



CNC - SERIE S4000

Manuale per lo Sviluppo della Logica di Macchina (PLC)



DIR. EMC 89/336
DIR. LVD 73/23 + 93/68

AGGIORNAMENTO

Ed.	Data agg.	Pagine aggiornate
00	16/07/2001	Emissione CMAPL401070I
01	Mar. 2002	Seconda Edizione CMAPL402031I Aggiornato Capitolo 2.4 della Parte II con le istruzioni di configurazione della scheda SerPLC.
02	Apr. 2002	Terza Edizione CMAPL402042I Aggiornate istruzioni Dual Drive Digitale nel Capitolo 2.5 della Parte II.
03	Ott. 2006	Quarta Edizione CMAPL406103I Introdotte nuove variabili interfaccia.

Nota: Le pagine senza contrassegni sono modificate, quelle con asterisco (*) devono essere annullate e quelle con il simbolo + sono nuove aggiunte

AGGIORNAMENTO (cont.)

Ed.	Data agg.	Pagine aggiornate

Nota: Le pagine senza contrassegni sono modificate, quelle con asterisco (*) devono essere annullate e quelle con il simbolo + sono nuove aggiunte

INTRODUZIONE

INTRODUZIONE

La nuova serie di CNC **S4000** è la sintesi tra le prestazioni dei Controlli **S3000**, molto apprezzati per la potenza e la facilità della programmazione e la ricchezza dell'interfaccia uomo-macchina, ed il sistema operativo Windows NT che fornisce all'utilizzatore maggiori strumenti grafici e multimediali e rende i CNC più flessibili ed aperti ad applicazioni standard di mercato. Infatti i CNC **S4000** hanno una sperimentata architettura basata su doppia CPU: una con processore PC Pentium II per la gestione dell'interfaccia operatore e dei sistemi esterni CAD/CAM ed una con processore Motorola per il controllo in tempo reale della macchina utensile.

SOMMARIO

Il manuale è stato suddiviso in tre parti indipendenti tra di loro:

Parte I Linguaggi di programmazione e procedure operative

In questa parte vengono descritte le singole istruzioni utilizzabili durante la programmazione, con esempi elementari, nonché le procedure di utilizzo e le softkey che consentono di operare in quest'ambiente.

Parte II Interfaccia di sistema

In questa parte sono descritti in dettaglio tutti i comandi interscambiati tra la LOGICA DI MACCHINA ed il C.N. con le relative caratteristiche e modalità di attuazione.

Parte III Esempi di programmazione

Questa parte contiene alcuni esempi reali di applicazioni realizzate con la programmazione PLC.

Il contenuto dei singoli capitoli che compongono le varie parti è il seguente:

Parte I

Cap. 1 Procedure preliminari

In questo capitolo sono elencate le procedure preliminari dei sistemi SELCA Serie S4000, operazioni elementari di avvio e arresto e descrizione dell'interfaccia utente.

Cap. 2 Procedure operative

Questo capitolo contiene la descrizione delle softkey attive nell'ambiente APPLICAZIONI e le procedure operative per eseguire le operazioni necessarie all'editing, compilazione, attivazione e debugging del programma.

Cap. 3 Organizzazione del programma

In questo capitolo viene descritta la struttura delle varie parti del programma nonché il formato delle variabili e delle costanti utilizzate all'interno del programma stesso.

Cap. 4 Dichiarative iniziali

Questo capitolo contiene la descrizione delle variabili, delle funzioni e dei parametri che devono precedere la stesura del programma vero e proprio, cioè gli input/output, le variabili interne, gli impulsi, i temporizzatori i contatori e le softkey definibili da logica.

Cap. 5 Operazioni e funzioni

In questo capitolo vengono descritte in dettaglio tutte le istruzioni e funzioni che possono essere utilizzate nella stesura del programma, con i relativi parametri, limiti, ecc. Le funzioni sono suddivise in logiche, aritmetico/matematiche, di conversione di formati delle variabili e funzioni per il trattamento delle variabili stringa.

Cap. 6 Istruzioni di controllo del flusso di programma

Questo capitolo descrive le funzioni che possono variare il flusso naturale del programma, cioè salti (condizionati o meno), loop e subroutine.

Cap. 7 Funzioni particolari

In questo capitolo, l'ultimo della I parte, sono state inserite le descrizioni di funzioni particolari che possono essere utilizzate dall'utente, cioè staticizzazione di segnali, scelte tra segnali diversi, messaggi per l'interfaccia operatore e comandi interscambiati tra logica e CN.

Parte II

- Cap. 1** Questo capitolo contiene la descrizione dei registri, delle variabili e dei segnali d'interfaccia tra il modulo standard del PLC ed il Controllo Numerico, con le caratteristiche di ognuno di essi in termini di formato, direzione e sincronismo, nonché l'interazione tra di loro. Detti registri, variabili e segnali sono stati raggruppati per argomento o funzione.
- Cap. 2** Questo capitolo svolge le stesse funzioni del capitolo precedente per i registri, variabili e segnali relativi ai moduli interni dedicati, cioè gestione mandrini, movimentazione assi indipendenti e gestione del cambio utensile.
- Cap. 3** In questo capitolo sono brevemente descritte le principali modifiche richieste per adattare un programma PLC sviluppato per i sistemi della Serie S1200 alla Serie S4000.
- Cap. 4** Questo capitolo contiene una tabella riassuntiva di tutti i registri, variabili e segnali descritti nei capitoli 1 e 2. Detta tabella è particolarmente utile come prontuario di consultazione delle caratteristiche peculiari di ognuno di essi, se si hanno già le informazioni descritte dettagliatamente nei due capitoli citati.

Parte III

La parte terza contiene un unico capitolo nel quale sono raggruppati esempi reali di programmazione PLC che possono essere utilizzati così come sono o adoperati come spunto per funzioni analoghe.





SIMBOLOGIA E TERMINOLOGIA



Tutti i nomi delle variabili e delle istruzioni predefinite sono scritti in grassetto corsivo con caratteri maiuscoli (es. **VARIAB**), mentre quelli scritti in grassetto corsivo con caratteri minuscoli sono riferiti a generici nomi di variabili o espressioni che devono essere assegnati dal programmatore (es. **operatore**).

Nella sintassi delle istruzioni tutto ciò che è compreso fra i simboli [e] è opzionale e può anche essere omesso.

Il simbolo | separa parametri alternativi tra loro, cioè deve essere introdotto uno solo dei termini divisi da questo simbolo (es. **A|B|C** deve essere introdotto o solo **A** o solo **B** o solo **C**).

I tasti della tastiera (esclusi quelli alfanumerici) vengono riportati come appaiono sulla tastiera del CN.


(es. , , , , ecc.).

Nota: Il tasto Return della tastiera del CN è posizionato in verticale (). Poichè però in questa forma esso provoca un allontanamento eccessivo delle righe di testo, si è preferito la rappresentazione , all'interno del testo del manuale.

Generalità

Con il termine "settare" (italianizzazione del termine inglese "set") si intende il forzamento a livello logico "1" o "vero" di una variabile di tipo booleano.

Con il termine "resettare" (italianizzazione del termine inglese "reset") si intende invece il forzamento a livello logico "0" o "falso" della variabile booleana.

 Con questo simbolo vengono messe in evidenza differenze di utilizzo di particolari funzioni o istruzioni tra i controlli della Serie S3000 e quelli precedenti S1200. Questo vale in particolare per quegli utilizzatori che hanno già installato e sviluppato programmi per il controllo S1200 , per evitare possibili errori o incongruenze.

INDICE

Parte I

1. PROCEDURE PRELIMINARI

1.1. ACCENSIONE CN	1-1
1.2. SPEGNIMENTO CN	1-1
1.3. ARRESTO DEL SISTEMA IN CASO DI ERRORI INTERNI	1-2
1.4. L'INTERFACCIA UTENTE	1-3

2. PROCEDURE OPERATIVE

2.1. PROGRAMMAZIONE PLC	2-2
2.2. EDIT PLC	2-4
2.3. COMPILA LOGICA	2-8
2.4. CARICA ATTIVA	2-8
2.5. DEBUG LOGICA	2-9
2.6. VISUALIZZAZIONE DINAMICA	2-9
2.7. TABELLE DI VISUALIZZAZIONE DINAMICA	2-10
2.8. FORZAMENTO ASSEGNAZIONE	2-11
2.9. ANALIZZATORE GRAFICO G SCOPE	2-11
2.9.1. Impostazione nomi variabili/espressioni da tracciare	2-12
2.9.2. Attivazione e arresto dell'acquisizione	2-12
2.9.3. Impostazione dei parametri di acquisizione	2-13
2.9.4. Altre impostazioni (ASPETTO)	2-13
2.9.5. Salvataggio e recupero delle tabelle analizzatore	2-14
2.9.6. La stampa dell'analizzatore	2-14
2.9.7. Analisi tracce acquisite	2-14
2.9.8. La barra dei menu dell'analizzatore grafico	2-15
2.10. VISUALIZZAZIONE, MODIFICA TABELLE DEL PLC	2-16

3. ORGANIZZAZIONE DEL PROGRAMMA

3.1. REGOLE GENERALI	3-1
3.2. STRUTTURA DEL PROGRAMMA	3-2
3.2.1. SEZIONE DICHIARAZIONE	3-3
3.2.2. SEZIONE DI INIZIALIZZAZIONE	3-3
3.2.3. SEZIONE DEL PROGRAMMA	3-3
Logica superveloce	3-3
Logica veloce	3-4
Logica lenta	3-4
Logica superlenta	3-4
Sincronizzazione	3-4

3.2.4. SEZIONE DELLE ROUTINE.....	3-4
3.3. FORMATO DELLE VARIABILI E DEI NUMERI.....	3-4
3.3.1. VARIABILI SINGOLE E VETTORIALI	3-5
3.3.2. VARIABILI STATICHE E DINAMICHE	3-6
3.3.3. COSTANTI.....	3-7
3.3.4. COSTANTI CONFIGURABILI PER LA LOGICA DI MACCHINA	3-7
3.3.5. DISPOSIZIONE DEI SINGOLI BIT ALL'INTERNO DELLE VARIABILI	3-7
3.3.6. ACCESSO AI SINGOLI BIT DI VARIABILI	3-8
Variabili singole.....	3-8
Variabili vettoriali.....	3-9
3.3.7. ACCESSO AI BIT DI VARIABILI ADIACENTI	3-10

4. DICHIARATIVE INIZIALI

4.1. DICHIARAZIONI DI INPUT/OUTPUT FISICI.....	4-2
4.1.1. DICHIARAZIONE DI INPUT/OUTPUT FISICI: MODULI DI I/O REMOTATI	4-4
4.2. DICHIARAZIONE VARIABILI INTERNE	4-5
4.3. DICHIARAZIONE DI STRINGHE.....	4-6
4.4. DICHIARAZIONI DI EQUIVALENZE	4-7
4.5. PULSE	4-9
4.6. TEMPORIZZATORI	4-9
4.7. CONTATORI.....	4-12
4.8. SOFTKEY DEFINIBILI DA LOGICA	4-13
4.9. SOFTKEY E MESSAGGI CON TESTO IN PIÙ LINGUE	4-15

5. OPERAZIONI E FUNZIONI

5.1. PROGRAMMAZIONE CON ELEMENTI LOGICI ELEMENTARI.....	5-1
5.2. OPERAZIONI ARITMETICHE	5-2
5.3. FUNZIONI MATEMATICHE IN FLOATING POINT	5-3
5.4. CONFRONTO O PARAGONE	5-3
5.5. ROTAZIONE	5-4
5.6. FUNZIONI DI TRASFORMAZIONE TRA FORMATI DIVERSI	5-5
ENC - Ricerca bit.....	5-5
DEC - Setta bit.....	5-5
HI - Estrae il byte alto di una word.....	5-5
LO - Estrae il byte basso di una word.....	5-5
EXT - Estensione di un byte in una word.....	5-6
BCD - Converte in codice BCD un numero binario.....	5-6
BIN - Converte un numero BCD in byte o word.....	5-6
IFP - Converte un byte o una word in formato floating point	5-6
FPI - Converte un formato floating point in byte o word	5-6
5.6.1. ESPRESSIONI COMPLESSE	5-6
5.7. OPERAZIONI e FUNZIONI SU STRINGHE	5-7
5.7.1. FUNZIONI NUMERICHE AD ARGOMENTO STRINGA	5-7
VAL - Trasforma un numero da formato ASCII a numerico	5-7
INSTR - Ricerca una stringa all'interno di un'altra.....	5-8
LEN - Lunghezza di una stringa.....	5-9
STRCMP - Confronto tra stringhe	5-9
5.7.2. FUNZIONI STRINGA AD ARGOMENTO NUMERICO.....	5-10
MKN\$ - Converte un numero in formato stringa.....	5-10
CHR\$ - Genera un carattere ASCII	5-11
STRNG\$ - Genera una stringa di caratteri uguali.....	5-11

5.7.3. FUNZIONI STRINGA AD ARGOMENTO STRINGA.....	5-12
MID\$ - Estrae una stringa da un'altra.....	5-12
LEFT\$ - Estrae una stringa a partire da sinistra.....	5-13
RIGHT\$ - Estrae una stringa a partire da destra.....	5-13
5.7.4. SOMMA DI STRINGHE	5-14

6. ISTRUZIONI DI CONTROLLO DEL FLUSSO DEL PROGRAMMA

6.1. SALTO INCONDIZIONATO	6-1
6.2. SALTO CONDIZIONATO	6-2
6.3. ESECUZIONE CONDIZIONATA.....	6-2
6.4. GO TO CALCOLATA	6-2
6.5. GO TO PONDERATA.....	6-3
6.6. LOOP	6-4
6.7. SUBROUTINE	6-5

7. FUNZIONI PARTICOLARI

7.1. FLIP FLOP.....	7-1
7.2. MULTIPLEXER	7-1
7.3. RICERCA SU TABELLA	7-2
7.4. MESSAGGI PER L'OPERATORE	7-3
7.5. COMANDO DI PROGRAMMI DA LOGICA DI MACCHINA	7-4
7.5.1. COMANDO DI PROGRAMMI DURANTE L'ESECUZIONE IN AUTOMATICO	7-5
7.5.2. COMANDO DI PROGRAMMI A PARTIRE DALLO STATO DI MANUALE	7-6
7.5.3. COMANDO DI PROGRAMMI DURANTE L' ESECUZIONE IN MODO SEMIAUTOMATICO	7-6
Comando di programmi da logica di macchina: unita' di misura	7-7
Comando di programmi da logica di macchina: funzioni non ammesse	7-7
Comando di programmi da logica di macchina: lancio in modo asincrono	7-7

Parte II

1. GESTIONE MOVIMENTI E SEGNALI D'INTERSCAMBIO

1.1. STATI DEL CN	1-1
1.2. TRASMISSIONE FUNZIONI AUSILIARIE SINCRONE E PREPARATORIE	1-2
1.2.1. ACQUISIZIONE INFORMAZIONI SINCRONE DA PLC A CN	1-3
1.2.2. SEGNALE TERMINE DI SOTTOPROGRAMMA COM	1-3
1.2.3. TRASMISSIONE PARAMETRI SUPPLEMENTARI I, J, K, Q	1-4
1.2.4. EMISSIONE DI FUNZIONI AUSILIARIE "AL VOLO"	1-4
Trasmissione funzioni ausiliarie: note sull' emissione della speed.....	1-5
1.3. COMANDI ASINCRONI DI START, ARRESTI, ALLARMI, CONSENSI EMERGENZA CN SU INTERRUZIONE COMUNICAZIONE CON SCHEDA PC	1-5
1.4. ORIGINI E CORRETTORI UTENSILI	1-8
1.4.1. CAMBIO UTENSILE MANUALE	1-8
1.4.2. CAMBIO UTENSILE MANUALE TIPO S1200.....	1-8
1.4.3. CAMBIO UTENSILE AUTOMATICO	1-8
1.5. COMANDI RELATIVI ALL'ASSERVIMENTO E MOVIMENTAZIONE ASSI	1-9

1.5.1. ABILITAZIONE E BLOCCAGGIO ASSI.....	1-9
1.5.2. ASSI SEMPRE ATTIVI O CON BLOCCAGGI (M10 - M11)	1-11
1.5.3. RILASCIO DEGLI ASSI (M45 - M46).....	1-11
1.5.4. DISABILITAZIONE DEI TRASDUTTORI.....	1-11
1.5.5. MOVIMENTO MANUALE IN JOG	1-12
1.5.6. MOVIMENTO IN MANUALE CON VOLANTINI.....	1-12
1.5.7. RICERCA ZERO ASSI.....	1-13
Ciclo di ricerca zero asse con microinterruttore	1-14
Ciclo di ricerca zero su zero elettrico del trasduttore (marker)	1-16
Ciclo di ricerca zero su righe ottiche codificate.....	1-17
1.5.8. MOVIMENTI IN MANUALE DURANTE LO STATO DI HOLD.....	1-18
1.5.9. MOVIMENTO IN MANUALE E RICERCA ZERO CONTEMPORANEI ALLA LAVORAZIONE.....	1-18
1.5.10. INFORMAZIONI SUGLI ASSI.....	1-18
1.5.11. COMPENSAZIONI DINAMICHE DI MISURA ASSI.....	1-20
1.5.12. OFFSET PER ASSI CONTROLLATI	1-20
Offset di origina aggiuntivo per gli assi controllati.....	1-21
1.6. GESTIONE DEL PROBE DI MISURA ON/OFF	1-21
1.7. FINE CORSA SOFTWARE ASSI CONTROLLATI	1-22
Gestione del probe di misura on/of: diagnostica	1-22
Fine corsa software assi controllati: disattivazione errore E93.....	1-22
1.7.1. FINE CORSA SOFTWARE SUPPLEMENTARI	1-22
1.8. GESTIONE DI ASSI DI TIPO PARTICOLARE.....	1-23
1.8.1. ASSI PARALLELI (GANTRY)	1-23
Ciclo di ricerca zero con micro per assi GANTRY	1-24
1.8.2. ASSI PROGRAMMABILI NON CONTROLLATI	1-24
1.8.3. ASSI MASTER SLAVE (OPZIONE CN «MS»).....	1-25
1.8.4. LETTURA INGRESSI E SCRITTURA USCITE ANALOGICHE: MODULI I/O REMOTATI.....	1-25
Ingressi analogici.....	1-25
Uscite analogiche	1-25
Ingressi per sonde di temperatura.....	1-26
1.9. LETTURA INGRESSI E SCRITTURA USCITE ANALOGICHE.....	1-26
1.10. SCAMBIO DATI TRA PLC E PART PROGRAM	1-27
1.11. FINESTRE PER LA VISUALIZZAZIONE SU VIDEO CN.....	1-28
Finestre per la visualizzazione su video CN: quadro video attivo.....	1-29
1.12. DATA E ORA DEL SISTEMA.....	1-29
1.13. SEGNALI PER LA COPIATURA E DIGITALIZZAZIONE DI SUPERFICI	1-29
Segnali per la copiatura e digitalizzazione: copia manuale attiva.....	1-31
1.13.1. REGISTRO DI STATO DEL TASTATORE DIGITALE DI COPIATURA.....	1-31
1.14. VARIABILI DI VERIFICA DEI TEMPI DI UTILIZZO DELLE RISORSE	1-32
1.15. SEGNALAZIONI DI ERRORE ACCESSIBILI DALLA LOGICA.....	1-32
Controllo degli indici per l' accesso a variabili e tabelle	1-33
1.16. LETTURA E MODIFICA DEI PARAMETRI DI CONFIGURAZIONE ASSI CN	1-33
1.17. GESTIONE DI PIU' GRUPPI D' ASSI (GDA) INTERPOLANTI SIMULTANEAMENTE.....	1-35
Informazione sul gruppo d' assi visualizzato.....	1-35
1.18. GESTIONE DEI DRIVE DIGITALI PER L' AVANZAMENTO ASSI ED IL MANDRINO	1-36

2. MODULI INTERNI DEDICATI

2.1. MODULO PER LA GESTIONE DEI MANDRINI	2-1
2.1.1. SEGNALI E REGISTRI PER LA ROTAZIONE MANDRINO	2-1
2.1.2. SEGNALI E REGISTRI PER LA SELEZIONE E CAMBIO GAMMA.....	2-2
2.1.3. SEGNALI E REGISTRI PER L'ORIENTAMENTO MANDRINO	2-3
Orientamento su posizione assoluta	2-3
Orientamento unidirezionale.....	2-4
2.1.4. SEGNALI E REGISTRI PER IL SINCRONISMO FRA MANDRINI	2-4
2.1.5. SEGNALI E REGISTRI COMUNI A TUTTE LE MODALITA' MANDRINO.....	2-5
2.1.6. MANDRINI CON O SENZA TRASDUTTORE	2-6
2.1.7. NOTE SUL CICLO FISSO G84	2-6
Riassunto segnali e registri interessati.....	2-7
2.2. MODULO DI MOVIMENTAZIONE ASSI INDIPENDENTI	2-8
Nuove variabili	2-10
2.3. MODULO PER LA GESTIONE DEL CAMBIO UTENSILE	2-11
2.3.1. DEFINIZIONI ELEMENTARI	2-12
2.3.2. MODALITA' POSSIBILI DI FUNZIONAMENTO CU	2-12
2.3.3. CONFIGURAZIONE DEI CU AUTOMATICI.....	2-13
Disposizione utensili	2-13
Geometria del magazzino.....	2-14
Tipo di gestione magazzino.....	2-14
2.3.4. DEFINIZIONE DELLE SEQUENZE	2-14
Cambi utensili asincroni.....	2-15
Cambi utensili sincroni.....	2-16
Cambi utensili semiasincroni.....	2-17
Implementazione nel programma PLC.....	2-19
Attivazione del modulo cambio utensile	2-19
Attuazione delle sequenze.....	2-19
Attivazione correttore lunghezza utensile	2-20
Decodifica della T programmata e selezione della sequenza di lavoro	2-20
2.3.5. INTERRUZIONE DELLA SEQUENZA.....	2-21
Gestione vita utensili integrata	2-21
Descrizione delle variabili PLC	2-22
2.3.6. DIFFERENZIAZIONE UTENSILI PER FAMIGLIE.....	2-22
2.3.7. DIFFERENZIAZIONE UTENSILI CON TAGLIE DIVERSE	2-22
2.3.8. DESCRIZIONE DELLE VARIABILI PLC.....	2-23
Nuove variabili di informazione.....	2-24
2.3.9. LA TABELLA UTENSILI	2-24
Scrittura dei campi della tabella utensili da parte del PLC	2-25
Lettura e scrittura dei correttori raggio e lunghezza	2-27
2.4. MODULO DI GESTIONE DELLA LINEA SERIALE PER IL PLC.....	2-27
2.4.1. SEGNALI E REGISTRI PER LA GESTIONE VIA PLC.....	2-28
2.4.2. DESCRIZIONE DEL FUNZIONAMENTO.....	2-29
Esempio programma PLC	2-30
2.5. DUAL DRIVE DIGITALE	2-32
2.5.1. CONFIGURAZIONE ASSI	2-32
2.5.2. CONFIGURAZIONE DRIVES.....	2-33
2.5.3. DESCRIZIONE DELL'ALGORITMO DI CONTROLLO.....	2-34
2.5.4. MODALITÀ DI TARATURA DEL SISTEMA	2-35
2.5.5. LOGICA DI ABILITAZIONE E DISABILITAZIONE	2-36
2.5.6. LOGICA DI EMERGENZA.....	2-36
Esempio PLC.....	2-37
Schema descrittivo Dual Drive Digitale	2-40

3. ADATTAMENTO DI UN PROGRAMMA PLC DA S1200 A S4000

4. RIASSUNTO DEI SEGNALI E REGISTRI PREDEFINITI

4.1. SIMBOLI E CONVENZIONI.....	4-1
4.2. MOVIMENTI E SEGNALI D'INTERSCAMBIO	4-3
Stati del CN.....	4-3
Colloquio sincrono con il CN.....	4-3
Trasmissione funzioni ausiliarie sincrone e preparatorie.....	4-3
Comandi asincroni di Start, Arresti, Allarmi, Consensi.....	4-4
Origini e correttori lunghezza utensili.....	4-4
Abilitazione e bloccaggio assi.....	4-4
Assi sempre attivi o con bloccaggio	4-5
Abbandono degli assi	4-5
Abbandono dei trasduttori.....	4-5
Movimento manuale JOG.....	4-5
Movimenti in manuale volantini.....	4-5
Ricerca zero assi	4-5
Movimento in manuale e ricerca zero durante la lavorazione	4-5
Informazioni sugli assi	4-5
Compensazioni dinamiche degli assi	4-6
Offset per assi controllati	4-6
Gestione del probe di misura on/off.....	4-6
Fine corsa software assi controllati	4-7
Gestione assi paralleli (Gantry)	4-7
Gestione assi programmabili non controllati.....	4-7
Lettura ingressi e scrittura uscite analogiche	4-7
Scambio dati tra PLC e Part program.....	4-7
Finestre per la visualizzazione su video CN	4-8
Data e ora del sistema.....	4-8
Segnale per la copiatura e digitalizzazione superfici	4-8
Variabili di verifica dei tempi di utilizzo delle risorse	4-10
Segnalazione di errore accessibili dalla logica	4-10
Costanti numeriche.....	4-10
Lettura e modifica dei parametri di configurazione assi	4-11
4.3. MODULI DEDICATI.....	4-12
Rotazione mandrino.....	4-12
Selezione del cambio gamma	4-12
Orientamento mandrino.....	4-12
Sincronismo tra mandrini.....	4-12
Comuni a tutti i modi di funzionamento	4-12
Ciclo fisso G84	4-13
Modulo di movimentazione assi indipendenti	4-13
Modulo per la gestione del cambio utensili.....	4-15
Tabella utensili.....	4-16

5. LIMITI

Parte III

1. ESEMPI DI PROGRAMMAZIONE PLC

BAS300F - Fresa base 3 assi e mandrino)	1-2
COMI3045 - Fresa 3 assi, mandrino	1-5
AXM11 - Bloccaggi selettivi per assi	1-10
AUXON - logica accensione ausiliari.....	1-11
GEVOL3 - Comando assi x, y, z con un solo volantino.....	1-12
SPIND1 - Rotazione mandrino	1-13
SPIND2 - Orientamento mandrino	1-15
SPIND3 - Cambio gamma.....	1-16
LUBMET - Lubrificazione assi a metri percorsi	1-17
LUBIN3 - Base lubrificazione a tempo.....	1-19
LUBMOV - Lubrificazione a tempo con assi in movimento	1-20
ZERIX - Ciclo di ricerca zero assi automatico.....	1-21
ESRNDU - Cambio utensile random con presa e posa in tempo mascherato.....	1-23
SCROLLIN - Gestione fino a 128 messaggi con scrolling su video	1-28
SHIFTZ - Esempio di compensazione caduta in Y in funzione di Z	1-29
AXBLOC1 - Bloccaggio assi con attesa a tempo	1-30
AXBLOC2 - Sbloccaggio assi con consenso esterno	1-31
ESSINC - Cambio utensile sincrono a rastrelliera	1-32
AXP2P - Movimentazione asse magazzino utensili da PLC	1-37
COMMUCM - Commutazione mandrino con asse C	1-39
NEWFILT - Filtro numerico	1-41
TABUTE1 - Riordino dei posti utensile nella tabella.....	1-42
TESTAR - Testa indexata mossa con motore mandrino	1-43

APPENDICI

APPENDICE A - TABELLA DEI CODICI ASCII.....	A-1
---	-----

APPENDICE B - TABELLA FUNZIONI AUSILIARIE.....	B-1
--	-----

APPENDICE C - MESSAGGI DIAGNOSTICI.....	C-1
---	-----

PARTE I

LINGUAGGIO DI PROGRAMMAZIONE E PROCEDURE OPERATIVE

1. PROCEDURE PRELIMINARI

1.1 ACCENSIONE CN

Dopo aver fornito l'alimentazione all'unità centrale ed al terminale grafico compaiono sul display CN i messaggi di avvio del sistema operativo WINDOWS e viene quindi automaticamente avviato il software CN S4000.

All'avvio del software Selca, compare una finestra con la visualizzazione del logo SELCA S4000 nella quale scorrono i messaggi di caricamento del software della scheda Motorola, ed a fine operazione la videata principale del CN.

Se la finestra con il logo non scompare automaticamente, significa che non è stato possibile stabilire il colloquio tra Unità videografica ed Unità centrale; in questi casi occorre spegnere il CN, verificare la correttezza dei collegamenti in fibra ottica e la presenza della tensione di alimentazione sull'Unità Centrale, quindi riaccendere.

1.2 SPEGNIMENTO CN

Il sistema operativo WINDOWS utilizzato per il CN necessita normalmente di una sequenza di spegnimento per evitare alla successiva accensione, procedure di ripristino di eventuali informazioni non salvate o salvate in modo incompleto.

Dalla videata principale del CN occorre premere:

[UTILITÀ] [SPEGNI CONTROLLO] e confermare premendo [Si],
quindi attendere il messaggio "Adesso è possibile spegnere il CN" e togliere l'alimentazione.

Sulle macchine dotate di dispositivo opzionale UPS (battery backup), la sequenza precedente è eseguita in modo automatico in caso di mancanza di tensione di alimentazione. Nei casi di normale spegnimento del CN occorrerà comunque premere:

[UTILITÀ] [SPEGNI CONTROLLO] e confermare premendo [Si],
quindi attendere il messaggio "Adesso è possibile spegnere il CN" e togliere l'alimentazione.

1. Procedure preliminari**1.3 ARRESTO DEL SISTEMA IN CASO DI ERRORI INTERNI**


Malgrado sia stata posta la massima cura nello sviluppo del software di base del CN, nelle versioni preliminari, potranno occasionalmente presentarsi delle situazioni di “blocco” del software dovuto ad errori interni. Nella maggior parte dei casi, questi ultimi, saranno intercettati in modo automatico da uno strumento di debug presente nel sistema operativo Windows NT e si manifesteranno con la comparsa di una finestra intitolata Dott. Watson. In questi casi è sufficiente togliere tensione al CN per ripristinare la condizione di errore.




1.4 L'INTERFACCIA UTENTE

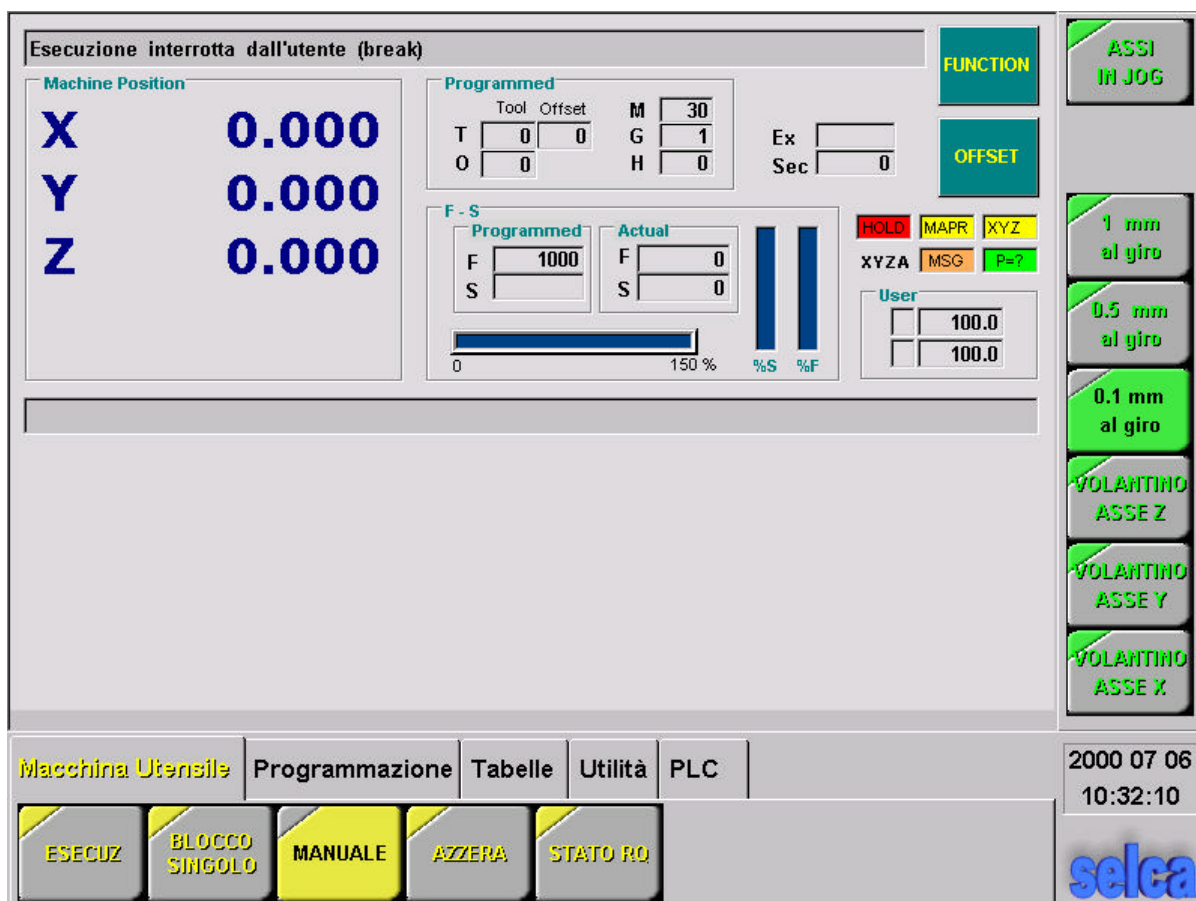
L'interfaccia utente dei sistemi S4000 è normalmente gestita tramite il consolidato sistema a softkey dei sistemi SELCA, arricchito con numerose finestre di dialogo, molto efficienti quando occorre introdurre dati o selezionare elementi appartenenti ad una lista predefinita.

Tutti i comandi sono attivabili "cliccando" (cioè premendo leggermente con le dita o con l'apposita "matita" in materiale plastico, sul video *touch screen* o con la pressione del tasto sinistro del *mouse* o *joystick*) direttamente sulle aree ove sono visualizzate le softkey, anziché con tasti funzione F1...F18, come per i CN della serie S3000.

Alcune softkey danno accesso ad un nuovo menu, il tasto  permette di tornare al menu precedente, cliccando sul logo SELCA che ruota nell'angolo in basso a destra, si ottiene la funzione

del tasto .

Il tasto , con funzione di ritorno al menu base, non è attualmente gestito.



La parte destra del video è dedicata alla visualizzazione dei menu di softkey gestiti dal PLC, sempre presenti nei vari ambienti CN.

La parte inferiore è invece dedicata alle softkey dei menu gestiti dal sistema, contestuali all'ambiente nel quale ci si trova.

1. Procedure preliminari

Una sostanziale modifica della nuova interfaccia, grazie al sistema operativo multi-tasking, è la presenza di cinque ambienti contemporanei per ognuno dei quali esistono una pagina video ed una struttura gerarchica indipendente di menu.

Come si può vedere nella precedente figura, gli ambienti disponibili sono:



Per passare da un ambiente all'altro occorre premere sul display o premere una volta il pulsante sinistro del mouse (cliccare) nella zona del video sulla quale appare il nome dell'ambiente stesso; queste aree verranno in seguito identificate semplicemente con il nome di "TAB".

Anche la zona centrale dello schermo (VIS_MC) è configurabile in modo indipendente per ogni ambiente attraverso l'utility VisMCEditor: un potente strumento che permette di creare pagine grafiche personalizzate posizionando e ridimensionando molto semplicemente una serie di oggetti predefiniti (caselle di visualizzazione testo e variabili di sistema, richiamo di immagini di sfondo tipo bitmap (*.bmp), pulsanti, indicatori a barra, indicatori a lancetta, visualizzazioni di potenziometri ed interruttori).

E' estremamente importante osservare che le attività all'interno di ogni ambiente proseguono indipendentemente dal fatto che sia attiva o meno la visualizzazione.

E' possibile, per esempio, lanciare in esecuzione un primo programma con la grafica attiva, all'interno dell'ambiente di [MACCHINA UTENSILE], quindi verificare un secondo programma in esecuzione grafica nell'ambiente [PROGRAMMAZIONE]: la grafica del programma in esecuzione proseguirà anche se non visualizzata.

Altresì è possibile attivare l'analizzatore grafico per visualizzare il comportamento degli asservimenti nell'ambiente [PLC] ed agire sulla taratura parametri nell'ambiente [TABELLE].

Di seguito viene riportata una breve descrizione delle principali funzioni accessibili all'interno di ogni ambiente:

Macchina Utensile

In quest'ambiente sono accessibili i menu di comandi per l'esecuzione di programmi, blocco singolo, movimenti manuali ed azzeramento origini.

Programmazione

Permette l'accesso all'edit dei programmi e all'esecuzione in grafica simulata.

Tabelle

Permette l'accesso alle tabelle di configurazione parametri macchina ed alle tabelle utente: origini, utensili, parametri programma, parametri in esecuzione grafica, tabelle PLC, enti geometrici ed enti geometri in esecuzione grafica. L'accesso indesiderato è bloccato da un tasto di abilitazione che deve essere attivato al menù [UTILITÀ]

Utilità

Permette l'accesso agli strumenti di utilità generale disponibili sul CN, quali ad esempio per test e diagnostica, cambio lingua, l'identificativo della versione del software, nonché il tasto funzione da utilizzare per l'arresto del sistema.

PLC

E' l'ambiente a disposizione dell'applicatore del CN per lo sviluppo e la messa a punto del software PLC, nonché dove sono disponibili gli strumenti per le tarature ed ottimizzazioni degli asservimenti.

La descrizione dettagliata degli ambienti di [MACCHINA UTENSILE] e [PROGRAMMAZIONE] esula dagli obbiettivi del presente manuale, occorre perciò fare riferimento alla documentazione specifica; di seguito saranno descritte le operatività relative agli ambienti specifici relativi all'applicazione CN.

2. PROCEDURE OPERATIVE

Le prestazioni di seguito descritte di attivazione logica e debug possono essere utilizzate solo se è stata fornita l'alimentazione +24 V alla piastra I/OMIX ed alle eventuali espansioni di I/O digitali (vedere *Manuale di Installazione*); ciò non è necessario per l'edit e la compilazione programmi.

L'ambiente di sviluppo del PLC, come pure la configurazione dei parametri macchina (APPLICAZIONE), non è normalmente accessibile all'utente con le softkey dei menù; per ottenere l'accesso a questi ambienti occorre premere la softkey ABILITA MANUTENZIONE nel menù [UTILITÀ]:

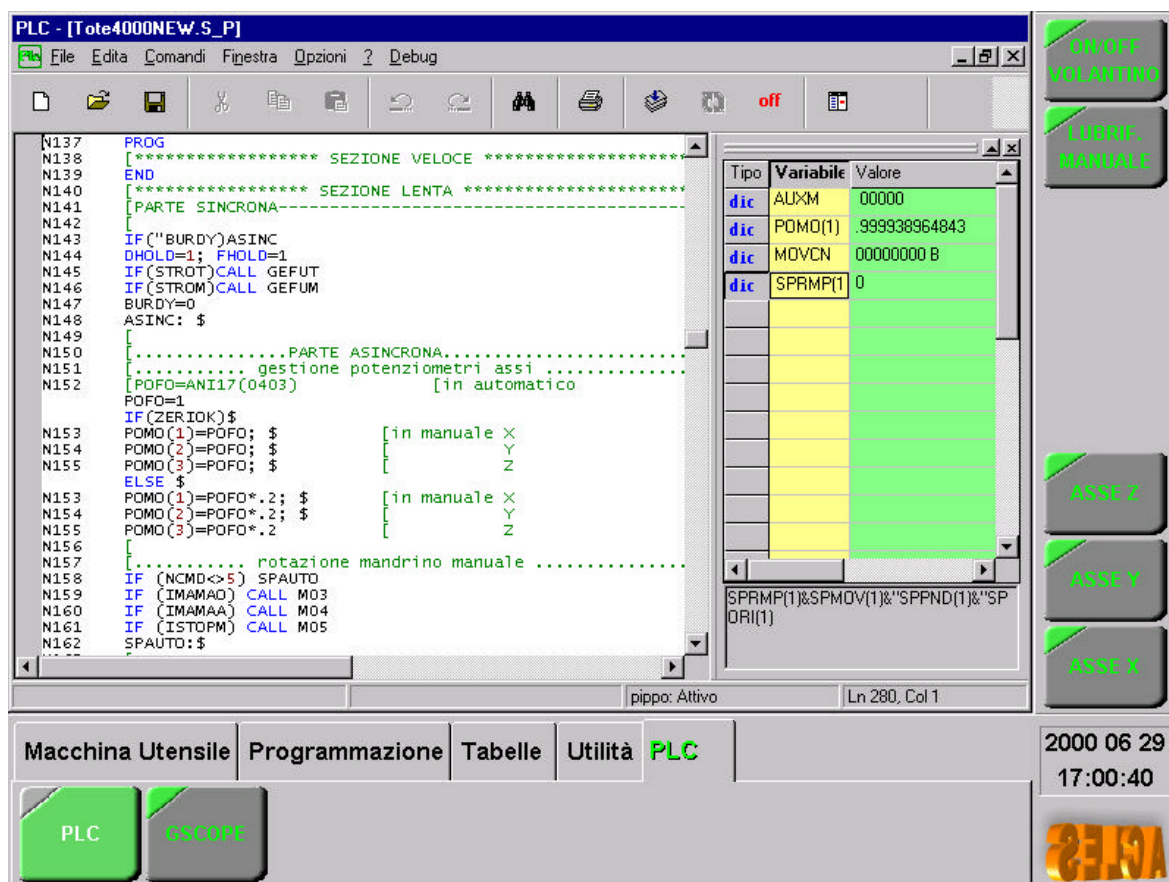


Gli ambienti rimarranno attivi fino allo spegnimento della macchina o a una successive pressione del tasto **ABILITA MANUTENZIONE**.

2. Procedure operative

2.1 PROGRAMMAZIONE PLC

In questo ambiente l'interfaccia utilizzata ha un'impostazione più vicina alle applicazioni per Windows, che in questi anni sono divenute famigliari agli applicatori ad a coloro che hanno dimestichezza nell'utilizzo dei personal computer.

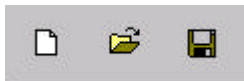


Nella parte superiore del video è presente la "barra dei menù" dove sono disponibili ai comandi di edit, di compilazione, di debug del PLC.

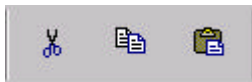
Nella regione sottostante si trova la "barra degli strumenti" dove i comandi di utilizzo più frequenti sono disponibili sotto forma di "icone".

Il significato delle "icone" è il seguente:

Comandi relativi all'editor:



- Crea nuovo programma,
- Apri programma esistente,
- Salva il programma in edit



- Taglia il testo selezionato, cioè copia la selezione in un'area di memoria temporanea (detta "clipboard o appunti") e quindi lo cancella da file in edit,
- Copia il testo selezionato negli appunti (**Ctrl** + **C**),
- Incolla il testo contenuto negli appunti a partire dal punto nel quale è posizionato il cursore (**Ctrl** + **V**).

Per selezionare una porzione di testo occorre posizionarsi sul punto iniziale della selezione, premere e mantenere premuto il tasto SHIFT, quindi estendere la selezione con i tasti frecce o trascinando tramite il mouse.



- Eliminazioni azioni effettuate (**Ctrl** + **Z**),
- Ripeti azioni annullate (**Ctrl** + **Y**)



- Cerca stringa (**Ctrl** + **F**),
- Stampa il programma (**Ctrl** + **P**)

Comandi di compilazione:



- Compila il programma (stessa funzione con tasto **F7**),
- Carica e attiva (oppure tasto **F8**),
- Attiva/Disattiva il PLC (l'icona on-off cambia in funzione dello stato del PLC)

Comandi di debug:



Richiama una tabella contenente una lista delle variabili PLC (utente e di sistema, suddivise per tipi).

2.2 EDIT PLC

Le procedure selezionabili in questo ambiente permettono di scrivere il programma del PLC direttamente a bordo macchina con tutte le istruzioni ed i comandi che sono riportati e spiegati in questo manuale.

Le funzioni messe a disposizione sui menu che compaiono in questo ambiente consentono l'inserzione e la modifica del testo, lo spostamento e cancellazione blocchi di testo, l'importazione del testo da altri programmi già scritti, la ricerca di parole, la ricerca e sostituzione di parole e la numerazione automatica delle linee.

Per iniziare a scrivere un nuovo programma occorre selezionare "FILE" dalla barra di menu e scegliere la voce nuovo.

Nuovo	Ctrl+N
Apri...	Ctrl+O
Chiudi	Ctrl+D
Salva	Ctrl+S
Salva con nome...	Ctrl+L
Stampa...	Ctrl+P
Anteprima di stampa	Ctrl+M
Imposta stampante...	Ctrl+T
1 D:\S4000\Plc\PROVA.S_P	
2 A:\plcscuf.S_P	

Ogni volta che viene creato un nuovo PLC è necessario salvarlo con un nome prima di poter effettuare le operazioni di compilazione e attivazione.

Per facilitare la lettura del programma tutte le istruzioni e parole chiave del linguaggio PLC sono utilizzate di colore blu, i commenti in colore verde ed il resto del testo in nero.

Per posizionarsi lungo il programma usando la tastiera, si possono utilizzare i seguenti tasti:



per spostarsi in alto di una riga



per spostarsi in basso di una riga



per posizionarsi a inizio riga



per posizionarsi a fine riga



per scorrere di una pagina in basso



per scorrere di una pagina in alto

Per spostare il cursore lungo una riga:



per spostarsi a destra di un carattere



per spostarsi a sinistra di un carattere



per posizionarsi sulla prima riga di programma



per posizionarsi sulla ultima riga di programma

Taglia	Ctrl+X
Copia	Ctrl+C
Incolla	Ctrl+V
Undo	Ctrl+Z
Redo	Ctrl+Z
Trova	Ctrl+F
Sostituisci	Ctrl+H
Seleziona tutto	Ctrl+A

Nel menu EDITA sono presenti le funzioni per copiare o spostare dalle parti di programmazione selezionate in precedenza.

Per selezionare con il mouse una parte di programma premere il tasto sinistro e senza rilasciarlo muovere il mouse fino a selezionare la parte desiderata e rilasciare il tasto, oppure con la tastiera si possono utilizzare il tasto contemporaneamente con le frecce.

TAGLIA:

copia la parte di programma selezionato negli "appunti" e la cancella dal programma.

COPIA:

copia la parte di programma selezionato negli appunti.

INCOLLA:

copia gli appunti nella posizione del cursore.

UNDO TYPING:

cancella l'ultimo carattere digitato, ripristina la situazione precedente.

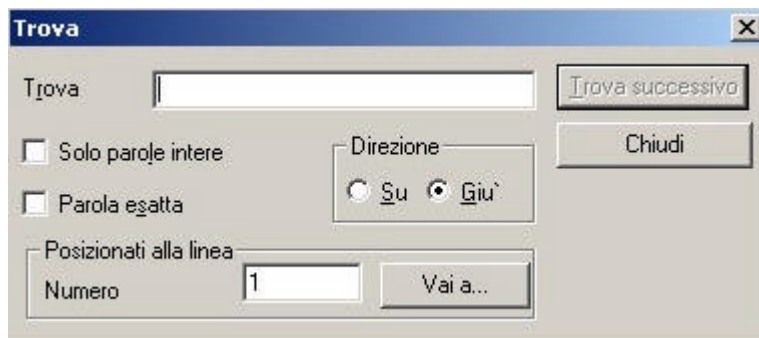
REDO:

ripristina la situazione prima dell'*UNDO TYPING*.

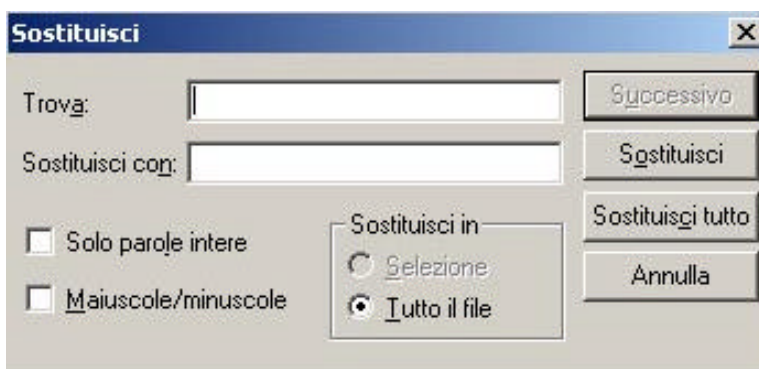
2. Procedure operative

TROVA:

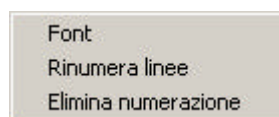
permette di cercare una parola o dei caratteri all'interno del programma, potendo scegliere la direzione di ricerca e se si vuole trovare una parola intera o una sola parte:

**SOSTITUISCI:**

con questa funzione si possono sostituire delle righe o parole con una nuova. Inserire nel campo "TROVA:" la parola che si vuole sostituire e nel campo "SOSTITUISCI CON:" la nuova parola. Con il tasto "SOSTITUISCI" l'editor chiede conferma ad ogni sostituzione da effettuare, con il tasto "SOSTITUISCI TUTTO" non chiede nessuna conferma.

**SELEZIONA TUTTO:**

serve per selezionare tutto il file. Lo si può usare nel caso in cui si voglia importare il file in un altro file.

**RINUMERA LINEE:**

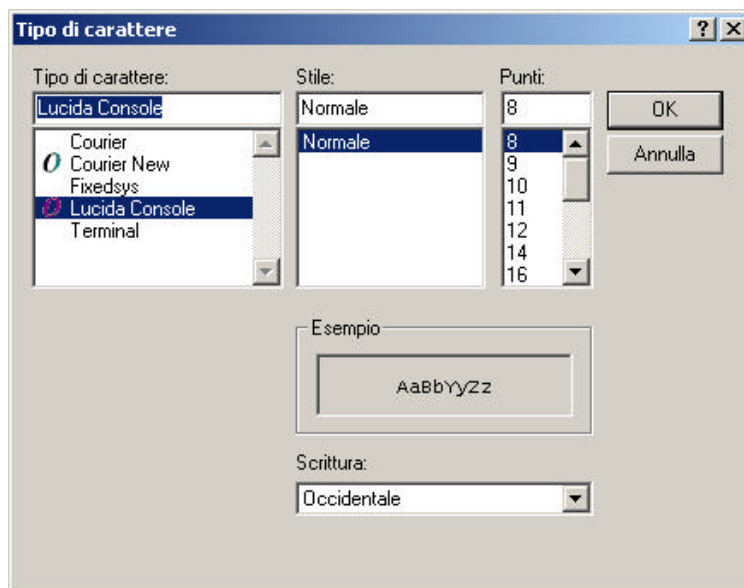
effettua una rinumerazione delle linee di programma o le inserisce nel caso in cui il programma ne fosse privo.

**ELIMINA
NUMERAZIONE:**

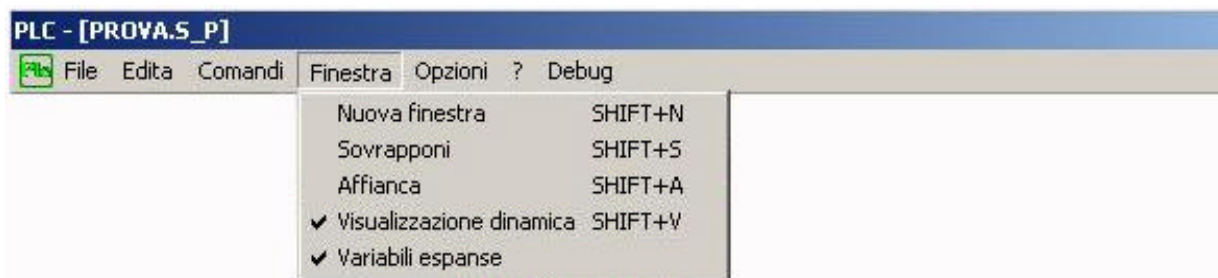
toglie la numerazione delle linee di programma.

FONT:

è possibile cambiare il tipo di carattere scegliendone uno presente nella list e la dimensione (punti).



L'editor permette di aprire più file contemporaneamente oppure di avere più finestre di edit relative allo stesso programma.



NUOVA FINESTRA: apre una nuova finestra relativa allo stesso programma PLC.

SOVRAPPONI: permette di sovrapporre le due finestre.


AFFIANCA: affianca le finestre aperte.

Per passare da una finestra all'altra è sufficiente cliccare sulla finestra voluta.

Per i menù comandi e debug vedi paragrafo relativo alla visualizzazione dinamica.


2.3 COMPILA LOGICA

E' questa la prima operazione da effettuare su di un programma, appena scritto o dopo una qualsiasi modifica, per tradurlo nel codice eseguibile da parte del calcolatore, nonché per verificarne la corretta sintassi.

Per **compilare** un programma PLC occorre posizionarsi sulla finestra contenente il testo sorgente, quindi utilizzare il comando di compilazione tramite l'apposita icona, il tasto , oppure la voce "Compila" dal menu "Comandi". In caso di errore durante la compilazione, compare una finestra a centro schermo con il messaggio relativo, quindi viene evidenziata con il colore rosso la riga contenente l'errore, correggere l'errore e ripetere l'operazione di compilazione.

2.4 CARICA E ATTIVA

La funzione **LOAD AND RUN** resetta la memoria delle variabili PLC e quindi attiva l'esecuzione dell'ultimo programma PLC compilato.

Per **caricare ed attivare** il programma (cioè trasferire il codice eseguibile alla scheda master) occorre premere l'apposita icona o scegliere la "Carica ed Attiva" dal menu "Comandi" oppure il tasto .

E' possibile disattivare volutamente la logica ripremendo la softkey stessa.

Il PLC può essere disattivato automaticamente nei seguenti casi:

- Errori di tipo hardware dovuti ad assorbimento di corrente troppo elevato su un OUTPUT, mancanza dell'alimentazione esterna 24 V su scheda principale o espansioni, ecc.
- Gravi errori software come CALL ed RTS sbilanciate, logica veloce o superveloce troppo lunga, errori sui calcoli in floating point (overflow, underflow, ecc.). In questi casi compare sul video un messaggio in chiaro con la causa che ha disattivato la logica.
- Cambiamento di alcuni parametri basilari nella configurazione della macchina Utensile (numero assi controllati, ecc.)

L'operazione di caricamento del PLC esegue automaticamente anche l'attivazione dello stesso; sono tuttavia presenti le icone ON e OFF che permettono rispettivamente di **attivare e disattivare** la logica.

2.5 DEBUG LOGICA

In questo ambiente sono disponibili tutte le funzioni di visualizzazione e di forzamento dei segnali e delle variabili utili alla diagnostica dell'impianto e alla verifica delle funzionalità del programma scritto.

E' inoltre possibile, a scopo di debug, memorizzare su tabelle l'insieme delle variabili da visualizzare o tracciare; in questo modo si conserva tutto l'insieme delle variabili per la verifica del funzionamento di particolari organi della macchina utensile in caso di anomalia, facilitando il compito in caso di riparazione o di messa in servizio della stessa.

2.6 VISUALIZZAZIONE DINAMICA

La finestra della **visualizzazione dinamica** è presente nella parte destra della videata base dell'ambiente PLC, per default, come riportato nell'immagine seguente; la contemporanea presenza di questa finestra accanto al sorgente PLC è estremamente utile in fase di messa a punto del programma.

Nel caso di operazioni di solo edit è possibile **rimuovere la finestra** cliccando sul simbolo "X" nell'angolo superiore destro della finestra.

Per riattivare la visualizzazione selezionare il menù “Finestra”, quindi “Visualizzazione dinamica”.


Nella parte inferiore della visualizzazione dinamica, viene riportata per esteso l'espressione della riga di visualizzazione selezionata (per cambiare selezione cliccare sulla tabella e scorrere con le frecce su e giù).






E' così possibile verificare immediatamente la relazione logica tra i segnali che compongono l'espressione (anche questa parte della finestra detta **"Finestra variabili espanse"** è disabilitabile con un comando dal menù "Finestra").

Tipo	Variable	Valore
in	AUXM	00000
dic	IMG1	1
in	BRKA	0
in	CICFI	00000
in	MOVCN	00000111 B
dic	SPRMP(1):	0


SPRMP(1)&SPMOV(1)&"SPND(1)&"SP
ORI(1)

2. Procedure operative

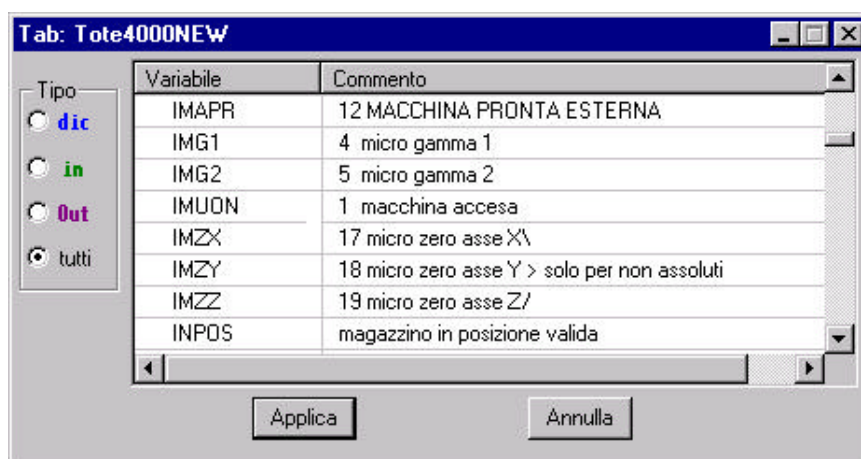
Per **inserire un simbolo** (nome/espressione) in visualizzazione è sufficiente cliccare su di una cella libera e digitare il nome della variabile; il tasto  permette di abbandonare l'inserimento.

Nel caso di debug di un programma è comodo poter selezionare una parte di testo dal file sorgente PLC (tipicamente l'espressione da verificare), premere  +  per copiarne il contenuto negli "appunti", quindi posizionarsi su una casella libera della visualizzazione dinamica e premere  +  per "incollarne" il contenuto, quindi confermare con .

Per **eliminare un simbolo** dalla visualizzazione occorre selezionare la riga voluta "cliccando sulla cella a sinistra del simbolo/espressione da eliminare (l'intera riga viene evidenziata), quindi premere <Canc> oppure selezionare la voce "Visualizzazione dinamica", "Elimina" dal Menu "Debug".

Per **modificare un simbolo** già visualizzato occorre selezionare la riga voluta "cliccando sulla cella a sinistra del simbolo/espressione da modificare (l'intera riga viene evidenziata), quindi effettuare la modifica e confermare con .

Altro strumento estremamente utile è la **finestra di riepilogo delle variabili**: essa viene richiamata dall'apposita icona sulla barra degli strumenti e si presenta nel modo seguente:



Si tratta di una lista delle variabili di sistema e di quelle dichiarate nel PLC, eventualmente divise per gruppi (in, out, ...), con a fianco il commento prelevato nel file sorgente; premendo il tasto [APPLICA], il simbolo selezionato viene aggiunto nella finestra di visualizzazione.

Altre funzioni disponibili nel menu "Debug" sono "**Ricerca assegnazione**", "**Espandi assegnazione**", "**Decimale/Binario**".


2.7 TABELLE DI VISUALIZZAZIONE DINAMICA

La voce "Debug" nella barra dei menu dà accesso ai comandi per il **salvataggio su file** ed il **richiamo** di tabelle dei simboli.

Eventuali simboli non disponibili nel programma PLC corrente, verranno visualizzati senza alcun valore a fianco, pur restando a far parte della tabella.

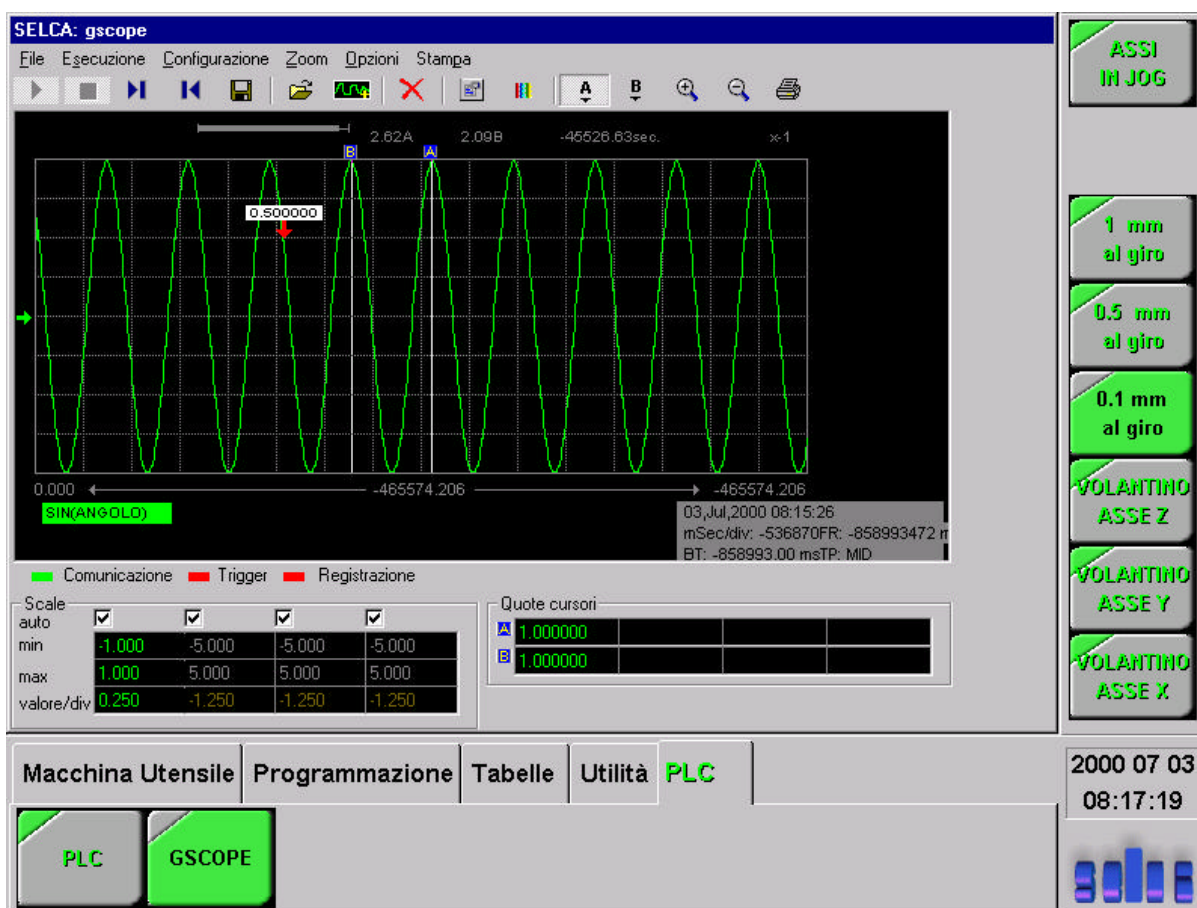
I simboli presenti in visualizzazione dinamica vengono mantenuti durante lo spegnimento e ripristinati alla successiva accensione del CN.

2.8 FORZAMENTO ASSEGNAZIONE

Per forzare il valore di una variabile è sufficiente inserirla in visualizzazione dinamica, posizionarsi sulla casella contenente la visualizzazione del valore, introdurre il nuovo dato e confermare con .

2.9 ANALIZZATORE GRAFICO GSCOPE

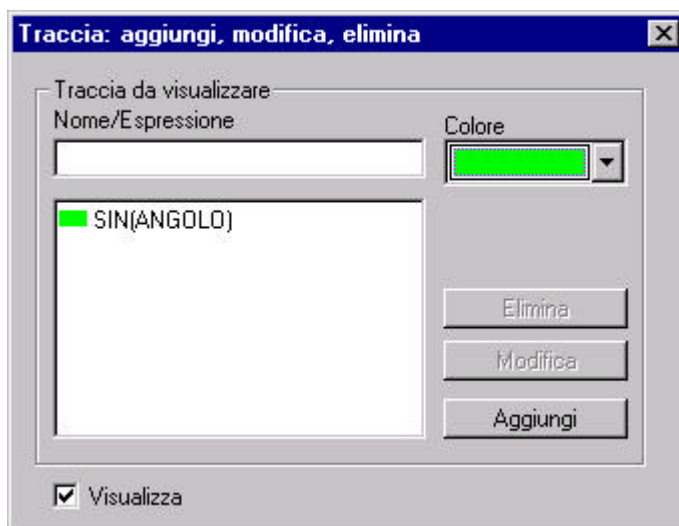
Per accedere al nuovo performante analizzatore grafico occorre premere la softkey [GSCOPE] all'interno dell'ambiente [PLC], compare la seguente videata:



L'interfaccia é dello stesso tipo usato nell'editor PLC e la visualizzazione dinamica.
Per tornare all'edit e la visualizzazione dinamica premere la softkey [PLC] (se l'analizzatore era attivo, rimarrà comunque in tale stato).

2.9.1 Impostazione nomi variabili /espressioni da tracciare


Cliccare sull'icona  compare la seguente finestra di dialogo:




Per **inserire** una nuova traccia occorre posizionarsi sul campo "Nome/Espressione", digitare quanto voluto, quindi cliccare su "Aggiungi"; la nuova espressione viene riportata nella finestra di riepilogo sottostante. E' possibile inserire un massimo di 4 tracce non bit e 16 tipo bit.


Per **modificare** una traccia occorre innanzitutto selezionarla dall'elenco cliccando sopra il nome, quindi cambiare l'espressione all'interno del campo "Nome/Espressione", il colore o la casella "Visualizza", quindi confermare cliccando su "Modifica".

Per **eliminare** una traccia occorre selezionarla dall'elenco cliccando sopra il nome, quindi confermare cliccando su "Elimina".

Per **eliminare tutte le tracce** dalla visualizzazione cliccare su:  dopo aver confermato l'azione, tutte le tracce impostate verranno eliminate.

Per ogni traccia (escluse le tracce logiche – di tipo bit) è possibile specificare il **limite minimo e massimo** e la casella di selezione **autoscala**, direttamente nell'area "Scale": è sufficiente introdurre i valori voluti e confermare con ; in quell'istante verrà calcolato il "**valore/divisione**" corrispondente ad un "quadretto" sulla scala verticale del grafico.

2.9.2 Attivazione e arresto dell'acquisizione

Dopo aver inserito le espressioni da tracciare, cliccare sull'icona  per **attivare l'acquisizione** dei dati.

Per **arrestare l'acquisizione** occorre, invece, cliccare sull'icona .

2.9.3 Impostazione dei parametri di acquisizione

Cliccare sull'icona , compare la seguente finestra:



Finestra di configurazione con titolo "Parametri di configurazione".

Trigger

nome/espressione

Posizione del trigger

☐ PRE ☒ MID ☐ END

☐ Trigger ON/OFF

Tempi

Base tempi (msec) 10

Durata acquisizione (sec) 5.12

☒ Punti interpolati

OK Esci


Per impostare una **equazione di trigger** è sufficiente digitarne l'espressione nel campo "Nome/Espressione", quindi specificare la posizione del trigger con **PRE, MID, END**.

Pur avendo impostato un'espressione di trigger, essa non avrà alcun effetto finché non verrà attivata la casella **Trigger ON/OFF**.

Nella stessa finestra è possibile impostare i parametri relativi a **Base tempi** e **Durata massima acquisizione**.

La casella **Punti interpolati** permette di scegliere se i punti acquisiti dovranno essere uniti da linee o meno, quando disegnati sul video.

2.9.4 Altre impostazioni (ASPETTO)

Cliccare sull'icona  per attivare la finestra di impostazione dei colori di sfondo e griglia e del tipo di griglia.



Finestra di impostazione con titolo "Aspetto: colore e griglia".

Colori

Sfondo

Griglia

Griglia



☐ 

☐ 

☒ 

Esci


2.9.5 Salvataggio e recupero delle tabelle analizzatore

Le icone seguenti permettono rispettivamente di memorizzare su file e recuperare le **tabelle delle tracce** da visualizzare  .

I simboli presenti nelle tracce in visualizzazione vengono mantenuti durante lo spegnimento e ripristinati alla successiva accensione del CN.

E' altresì possibile **salvare i dati acquisiti**, ma unicamente dalle apposite voci dal menu "File".

2.9.6 La stampa dell'analizzatore

L'icona  permette la stampa della finestra dell'analizzatore (sulla stampante di sistema, se configurata).

2.9.7 Analisi tracce acquisite






Quando la visualizzazione della traccia a video è stata completata (per arresto volontario o per l'arrivo del trigger), si passa in modalità di analisi traccia.

Nella **parte superiore della griglia** è sempre presente una barra che indica graficamente la relazione tra il "frame" corrispondente al tempo totale di acquisizione e l'intervallo relativo alla porzione di dati presenti all'interno della griglia. Questa visualizzazione sarà perciò funzione del fattore di zoom e della posizione dei cursori. Sono anche presenti le posizioni in tempo dei cursori A e B riferite al tempo 0 come primo valore acquisito ed il delta tempo tra i cursori.

Nella parte inferiore sono indicati rispettivamente i valori di tempo equivalente ai punti estremi della griglia ed il tempo corrispondente alla porzione di dati presente sulla griglia stessa (il tempo 0 è anche qui corrispondente al primo valore acquisito).



I cursori A e B


Nel **riquadro cursori** vengono visualizzati i valori numerici delle tracce in corrispondenza dei punti di intersezione con i due cursori orizzontali disponibili.

I due cursori A e B possono essere **abilitati e disabilitati** con le icone  ; quando essi sono attivati, usare i tasti  e  per spostarli; premendo contemporaneamente alle frecce anche il tasto , la **velocità di spostamento aumenta**.



Nel riquadro cursori vengono visualizzati **i valori numerici** delle tracce in corrispondenza dei punti di intersezione con i due cursori orizzontali disponibili.

Il cursore contestuale



Selezionare una traccia, cioè cliccare sul proprio nome (verrà visualizzato in reverse), quindi usare i tasti  e  per muovere il cursore contestuale: si tratta di una casella contenente il valore corrente che viene fatta scorrere direttamente lungo la traccia acquisita.

Premendo il tasto  contemporaneamente alle frecce, si ottiene il movimento veloce del cursore. Quando questo cursore è abilitato i cursori A e B non possono più essere mossi. Per disabilitare il cursore contestuale occorre nuovamente cliccare sul nome della variabile.

Lo zoom

Cliccare sulle icone   per **espandere** (fino ad arrivare a rappresentare ogni punto acquisito con un punto dello schermo) o **comprimere** (fino a far stare tutti i dati acquisiti all'interno della griglia) la traccia in corrispondenza della posizione del cursore principale A.

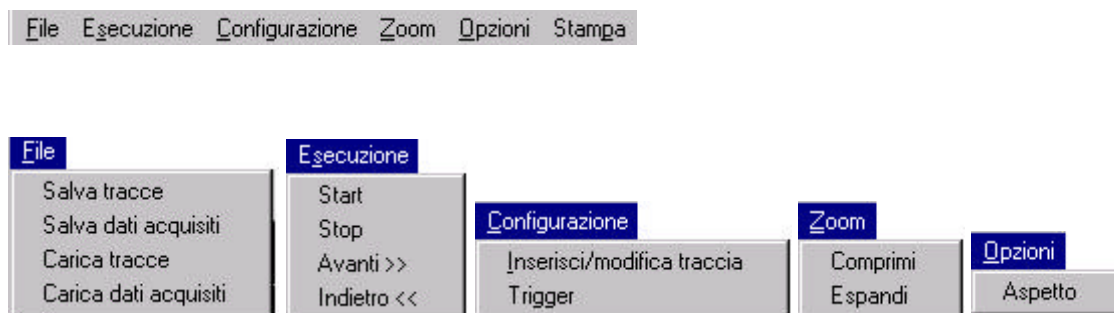
Lo spostamento verticale delle tracce

Per traslare verticalmente l'immagine della traccia, cliccare sul proprio nome (viene visualizzata in reverse), quindi usare i tasti  e .

Una freccia orizzontale sul lato sinistro della griglia sarà posizionata in corrispondente della posizione originaria di centro schermo.


2.9.8 La barra dei menu dell'analizzatore grafico

La maggiorparte dei comandi descritti è anche disponibile tramite le voci dei menu dall'omonima barra, cioè:



2.10 VISUALIZZAZIONE, MODIFICA TABELLE DEL PLC

Gli array di variabili (tabelle per l'utente) dichiarate all'interno del programma PLC sono visualizzabili e modificabili, purchè se ne conosca il nome anche a livello utente.

Premendo la softkey **CORRETT./ PARAMETRI** del menu **TABELLE**, compare tra le altre la softkey **TABELLA PLC**; dopo averla premuta occorre fornire il nome della tabella da modificare e confermare con .

Sul video viene presentata la lista degli elementi dell'array con a fianco il valore corrente acquisito al momento dell'inizio della visualizzazione della tabella; è quindi possibile cambiare i valori presentati che verranno trasferiti al PLC.

3. ORGANIZZAZIONE DEL PROGRAMMA

3.1. REGOLE GENERALI

Durante la stesura di un programma, occorre tener conto delle seguenti regole:

- Ogni programma PLC deve avere un nome per la sua identificazione composto da un massimo di 8 caratteri maiuscoli alfanumerici. Il primo carattere deve essere sempre alfabetico. Il nome del programma non deve contenere caratteri blank al suo interno.
- Non sono ammessi come nomi per i programmi: PRN, AUX, COM1, COM2, COM3, COM4, LPT1, LPT2, LPT3, LPT4, CON, NUL.
- Tutte le variabili ed i simboli utilizzati devono essere definiti con nomi mnemonici mediante stringhe alfanumeriche MAIUSCOLE di lunghezza massima pari a 6 caratteri.
- Tutti i simboli devono cominciare con una lettera e non possono contenere al loro interno i seguenti caratteri:

`^ ? ! \ # % & | ~ / () > [] < + - * @ = " ' , : . ;` *spazi bianchi (blank)*

poiché questi sono caratteri di controllo e operatori logici o aritmetici. Nel caso di blank, se ne sono stati inseriti all'interno delle istruzioni, essi vengono ignorati durante la compilazione.

- Non possono essere utilizzati come nome di variabili parole chiave dedicate alle istruzioni del linguaggio o nomi già assegnati a variabili predefinite.
- Anche se si sconsiglia l'uso di espressioni molto lunghe perché spesso poco leggibili, è possibile continuare l'editing dell'espressione stessa sulla riga successiva utilizzando il carattere "\$" in fondo alla riga interrotta.
- La massima lunghezza di una riga logica di programma (che può anche essere formata da più righe fisiche unite dal carattere \$) è di 500 caratteri, esclusi gli spazi (blank).
- E' possibile scrivere più equazioni su una stessa riga logica semplicemente separandole con il carattere ";" (punto e virgola).
- Le LABEL e i capostipiti dei simboli vengono riconosciuti perché seguiti dal terminatore ":".

3. Organizzazione del programma

- I commenti nel programma possono essere inseriti dovunque, sempre preceduti dal carattere "[". Se ne raccomanda un largo uso in quanto molto utili alla interpretazione del programma soprattutto in fase diagnostica visto che non consumano nè memoria nè tempo sul codice oggetto poiché ignorati in fase di compilazione.
- Per passare dai 6 caratteri massimi ammessi nella definizione delle variabili (default) a 9 oppure 12 occorre inserire, all'inizio del programma PLC le istruzioni:

CONST

_MXCHR=6 (oppure =9, =12)

(Il default è _MXCHR=6).

Occorre ricordare che l'utilizzo dei nomi di variabili lunghi comporta, oltre che naturalmente una maggiore occupazione di memoria da parte del listato sorgente, anche la generazione file eseguibile di dimensioni maggiori.

3.2. STRUTTURA DEL PROGRAMMA

Nella stesura del programma occorre descrivere, se necessarie, le SEZIONI nell'ordine in cui sono riportate nel seguente schema:

INP OUT	Sezione dichiarazione	P R O G R A M M A
INIT	Sezione inizializzazione (facoltativa)	
FAST	Sezione superveloce (facoltativa e solo se indispensabile) (se non necessaria omettere anche la parola chiave FAST)	
PROG END	Sezione veloce (facoltativa)	
..... END	Sezione lenta (logica ordinaria)	
..... END	Sezione superlenta (facoltativa)	
.....	Sezione routine (facoltativa)	

3.2.1. SEZIONE DICHIARAZIONE

Tutti i nomi delle variabili e dei segnali utilizzati nel corso del programma relativi tanto agli input/output fisici della macchina quanto alle variabili di elaborazione, ai timer, ai contatori, ecc. devono essere dichiarati all'inizio del programma nell'ordine prestabilito in questa sezione.

Accanto ai nomi, è utile inserire qualche breve commento in modo che il programma sia autonomamente documentato e interpretabile da tutti. Per esempio si può scrivere a fianco di input e output a quale morsetto e connettore sono riferiti.

I commenti a fianco dei simboli vengono anche riportati nella cross-reference dei segnali che viene stampata quando si esegue la Stampa della cross-reference.

La dichiarazione di ogni gruppo di variabili deve essere preceduta dalla corrispondente parola chiave (vedere il capitolo 4 *Dichiarative iniziali* più avanti).

3.2.2. SEZIONE DI INIZIALIZZAZIONE

La sezione successiva alla dichiarazione variabili è detta inizializzazione.

Questa sezione è facoltativa e contiene le parti di programma che devono essere eseguite solo una volta alla accensione del CN, per inizializzare o azzerare determinate variabili.

L'inizio della sezione inizializzazione è identificata dal simbolo **INIT**.

3.2.3. SEZIONE DEL PROGRAMMA

E' questa la sezione che contiene le istruzioni che il PLC ciclicamente scandisce. A sua volta può essere suddivisa in quattro ulteriori sezioni:

LOGICA SUPERVELOCE

LOGICA VELOCE

LOGICA LENTA

LOGICA SUPERLENTA

Logica superveloce

La sezione facoltativa, di *LOGICA SUPERVELOCE* comprende tutte le istruzioni scritte tra le parole chiave **FAST** e **PROG** che devono essere ripetute ciclicamente ad ogni "campionatura assi del CN" (vedere la documentazione relativa ai parametri di configurazione) per **gestire esclusivamente fenomeni estremamente veloci**. E' necessario ricordare infatti che, a parità di tipo di istruzioni, quelle poste in questa sezione occupano la CPU con un carico superiore di alcune decine di volte.

Se viene superato il tempo massimo a disposizione per questa sezione, compare il messaggio

Ciclo superveloce troppo lungo

Se non è necessaria alcuna logica superveloce, omettere anche la parola chiave **FAST**.

3. Organizzazione del programma

Logica veloce

La sezione di *LOGICA VELOCE* comprende tutte le istruzioni scritte tra la parole chiave **PROG** e la prima **END** che vengono ripetute ciclicamente ogni 10 msec.

Se viene superato il tempo massimo a disposizione per questa sezione, compare il messaggio

Ciclo veloce troppo lungo

Logica lenta

La sezione di *LOGICA LENTA* comprende tutte le istruzioni scritte tra la prima e la seconda istruzione **END**.

Queste parte di programma viene eseguita nei tempi lasciati liberi dalla sezione veloce all'interno del tempo dedicato al PLC dal sistema. Se tale tempo non è sufficiente, la sezione *LENTA* viene spezzata in più cicli.

Logica superlenta

La sezione di *LOGICA SUPERLENTA* comprende tutte le istruzioni scritte tra la seconda e la terza istruzione **END** che possono essere eseguite con bassa priorità e riguardanti fenomeni lenti (compensazioni termiche, gestione messaggi); anch'essa può venire spezzata in più cicli.

Sincronizzazione

Le sezioni di logica *SUPERVELOCE*, *VELOCE* e *LENTA* sono sincronizzate fra loro, cioè iniziano ad essere eseguite nell'ordine precedente dopo la sezione di **INIT**. La logica superlenta, invece, non è sincronizzata con le altre sezioni.

Gli INPUT vengono letti all'inizio della sezione superveloce, se presente; tutti gli OUTPUT vengono scritti alla fine della stessa.

Se la sezione superveloce è assente gli INPUT vengono letti all'inizio della sezione veloce e gli OUTPUT vengono scritti al termine della stessa.

3.2.4. SEZIONE DELLE ROUTINE

Qualsiasi routine che sia utilizzata esclusivamente in una certa sezione (veloce, ecc.) può essere scritta direttamente all'interno della stessa.

Considerato, tuttavia, che spesso una routine scritta per una sezione può essere valida anche per altre sezioni, è conveniente inserire le routine stesse al fondo del programma, cioè dopo la terza istruzione **END** (vedere il capitolo 6 *Istruzioni di controllo del flusso del programma*).

3.3. FORMATO DELLE VARIABILI E DEI NUMERI

Le variabili utilizzabili nel programma possono essere così classificate:

- BIT:** segnale logico elementare che può assumere solo i valori 1 e 0, ossia vero e falso
- BYTE:** variabile di 8 BIT che può contenere solo numeri interi compresi tra -128 e 127
- WORD:** variabile di 16 BIT che può contenere solo numeri interi compresi tra -32768 e 32767
- LONG:** variabile di 32 BIT capace di contenere numeri positivi e negativi compresi tra 1.2×10^{-38} e 3.4×10^{38} in formato floating point, con 7 cifre di mantissa
- DOUBLE:** variabile di 64 bit capace di contenere numeri positivi e negativi compresi tra 2.2×10^{-308} e 1.8×10^{307} in formato floating point a doppia precisione, con 15 cifre di mantissa
- STRINGA:** variabile di formato definibile contenente caratteri alfanumerici codificati in formato ASCII

I numeri *decimali* possono essere scritti nei formati:

± interi.decimali (es. -12.678) oppure

± interi.decimali e ±esponente con notazione scientifica (es. 12.3e-3).

Dovendo scrivere numeri in formato *esadecimale* si deve aggiungere in coda al numero la lettera **H** aggiungendo uno 0 prima della cifra più significativa se questa è maggiore di 9 (es. 0FA23H).

Si deve utilizzare la lettera **B** per i numeri binari (es. 01011101B).

Ad esempio il numero 35 in formato decimale corrisponde a 23H in esadecimale ed a 00100011B in binario; il numero 195 decimale corrisponde a 0C3H in esadecimale ed a 11000011B in binario.

Per le dichiarazioni del formato delle variabili, sia numeriche che di tipo stringa, vedere *Dichiarazioni di variabili interne* e *Dichiarazione di stringhe*, nel capitolo 4.

3. Organizzazione del programma

3.3.1. VARIABILI SINGOLE E VETTORIALI

Le variabili trattate nel sistema possono essere singole (dette anche **variabili interne**), nel qual caso il nome identifica in modo univoco il relativo contenuto, ovvero multiple (**vettoriali**), cioè contenere più elementi a fronte di uno stesso nome. In questo caso, il nome sarà composto da una parte alfanumerica, seguita da una parentesi che contiene un numero, detto **indice**, che identifica il singolo elemento. Queste variabili vengono definite **vettoriali** ed il loro formato è il seguente:

nome (indice)

Le variabili vettoriali possono contenere qualsiasi formato scelto tra quelli descritti precedentemente, esclusi i bit. E' ovvio che le variabili vettoriali conterranno tutti elementi omogenei, cioè tutti dello stesso tipo (BYTE, WORD, ecc.).

Il parametro **indice** può essere:

- un numero intero tra 1 e 32767

Esempio:

TABX(122) = 44.6565

Nell'elemento 122 sarà scritto il numero 44.6565

TABX(45)=TABX(77)+TABX(23)

L'elemento 45 conterrà la somma dei contenuti degli elementi 77 e 23

- il nome di una variabile di tipo BYTE che può assumere valori compresi tra 1 e 127, o di tipo WORD, con valori tra 1 e 32767

Esempio:

Si supponga di aver stabilito in precedenza che le variabili BTAB e WTAB siano rispettivamente di tipo BYTE e di tipo WORD

BTAB=18

Nella variabile BTAB si scrive il valore 18

TABUT(BTAB)=25

Si scrive il valore 25 nel 18° elemento di TABUT

...

WTAB=199

Nella variabile WTAB si scrive il valore 199

VALORE=TABCOY(WTAB)

Si scrive nella variabile singola VALORE il contenuto del 199° elemento di TABCOY

- una espressione con risultato nel formato BYTE o WORD con le stesse limitazioni numeriche del caso precedente

Esempio:

Si supponga che DATO1 e DATO2 siano variabili singole in formato BYTE e che ARRAY(x) sia una variabile vettoriale con un numero di elementi superiore a 11.

DATO1=4

DATO2=6

ARRAY(DATO1+DATO2+1)=66 *Si scrive 66 nell'11° elemento di ARRAY*

In generale, le variabili vettoriali occupano posizioni contigue della memoria, per cui è importante prestare particolare attenzione alla lunghezza e quantità dei dati trattati con queste variabili per evitare di invadere altre variabili (vedere più avanti). Infatti, se il valore dell'indice è maggiore del numero di elementi del VETTORE dichiarato, si andranno a invadere altre locazioni di memoria successive a quella a cui è riferito l'indice.

Sono da evitare scrupolosamente situazioni in cui l'indice assuma valore 0, negativo o al di fuori dei range stabiliti (es. 45000) in cui si vanno a sovrascrivere locazioni di memoria con esiti diversi da quelli attesi ed imprevedibili.

3.3.2. VARIABILI STATICHE E DINAMICHE

Le variabili utilizzate nel programma possono essere *ritentive* o *statiche*, cioè conservare il loro valore con il controllo spento, oppure *dinamiche*, a seconda del tipo di dichiarazione fatta (vedere *Dichiarazioni variabili interne* nel prossimo capitolo).

Le variabili numeriche non ritentive assumono valore 0 all'accensione, quelle stringa il valore ' ' (stringa vuota).

Il valore di conteggio dei COUNTER è conservato durante uno spegnimento, mentre non lo sono le variabili associate a TIMER, PULSE, SOFTKEY.

Tra le variabili interne predefinite del CN sono statiche quelle relative agli asservimenti degli assi controllati e degli assi indipendenti (assi sempre attivi, assi rilasciati, trasduttori disabilitati).

3.3.3. COSTANTI

Può essere utile nel programma fare riferimento in modo esplicito a delle costanti (numeriche o stringa); in questi casi le assegnazioni dei valori ai dati vanno fatte nella parte di inizializzazione del programma per evitare di ripetere inutilmente le istruzioni relative.

Esempio:

```
INIT  
SMAX=3500  
ALLM='MANDRINO FUORI SERVIZIO'
```

Il sistema mette a disposizione la seguente variabile mnemonica predefinita:

PI = 3.1415927 pigreco in formato DOUBLE

3.3.4. COSTANTI CONFIGURABILI PER LA LOGICA DI MACCHINA

Qualsiasi logica di macchina per poter essere utilizzata su macchine simili ma non perfettamente identiche, necessita di un certo numero di costanti configurabili al momento della installazione.

E' allora possibile parametrizzare, a livello di PLC, tempi di lubrificazione, rapporti di trasmissione di un cambio, la presenza di organi opzionali, l'intervallo di un timer, la quota assoluta di un asse in posizione di cambio utensile, ecc.

Sono per questo scopo definibili, in fase di configurazione, con i seguenti nomi

- 64 costanti di macchina comuni per tutto l'impianto con i nomi KMF(1), KMF(2), KMF(3),...,KMF(64) in formato floating point su 32 bit
- 64 costanti con i nomi KMW(1), KMW(2), KMW(3),...,KMW(64) in formato WORD.

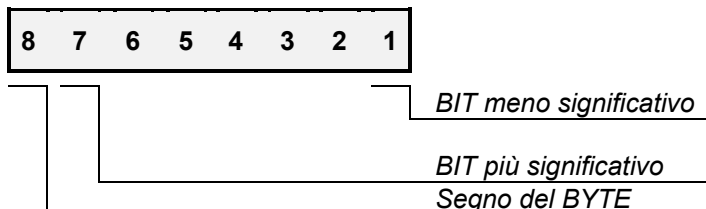
Nelle pagine di configurazione (setup), accanto ad ogni variabile è anche disponibile un campo commento nel quale introdurre la descrizione delle variabili.

3. Organizzazione del programma

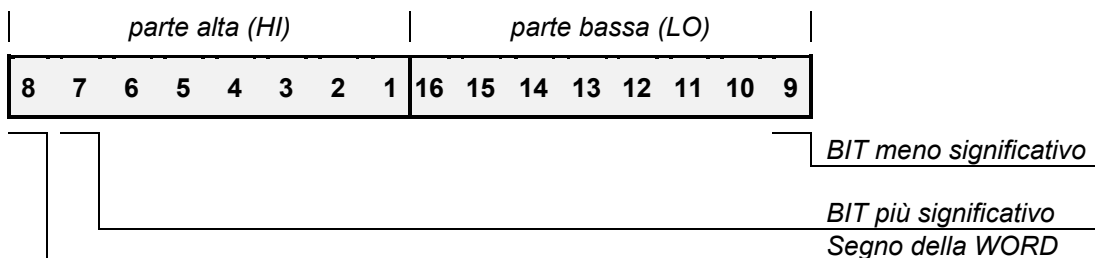
3.3.5. DISPOSIZIONE DEI SINGOLI BIT ALL'INTERNO DELLE VARIABILI

La disposizione dei singoli bit all'interno di BYTE, WORD è la seguente:

Formato BYTE



Formato WORD



Nota:

BYTE e WORD sono utilizzati dal PLC in formato binario con segno; cioè i numeri negativi sono rappresentati in complemento a 2.

Esempio:

BYTE 1 = 0 0 0 0 0 0 0 1 B
 -1 = 1 1 1 1 1 1 1 1 B

bit di segno

WORD 1 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 B
 -1 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 B

bit di segno

3.3.6. ACCESSO AI SINGOLI BIT DI VARIABILI

Variabili singole

Per accedere ai singoli bit di una variabile interna si tratta la variabile stessa come se fosse una matrice di 8 elementi, se di tipo BYTE, di 16 elementi se di tipo WORD, ecc. Si usa cioè una sintassi del tipo:

var(indice)

indice può essere una delle seguenti grandezze (negli esempi le variabili trattate si suppongono di tipo singolo e non vettoriale):

- un numero intero tra 1 e il numero massimo di bit previsto per quella variabile

Esempio:

BIT3=NUMBT(3) *con questa funzione, BIT3 avrà un valore 1 o 0 a seconda dello stato del bit 3 di NUMBT*

- il nome di una variabile già definita di tipo BYTE o WORD che può assumere valori compresi tra 1 ed il numero massimo di bit della variabile che deve trattare successivamente

Esempio:

INDEX=5
BYTE1(INDEX)=1 *si mette a 1 il bit 5 di BYTE1*

- una espressione con risultato nel formato BYTE o WORD con le stesse limitazioni numeriche del caso precedente

Esempio:

DAT01=8
DAT02=6
WORD1(DAT01+DAT02+1)=0 *Si pone a 0 il quattordicesimo bit della variabile WORD1 di formato WORD*

In ogni caso occorre tenere presente che, se il valore dell'indice è maggiore del formato della variabile, si andranno a sovrascrivere altre locazioni di memoria successive a quella a cui è riferito l'indice e presumibilmente occupate da altre variabili (vedere più avanti).

Sono da evitare scrupolosamente situazioni in cui l'indice assuma valore 0, negativo o al di fuori dei range stabiliti per la variabile trattata per cui si vanno a sovrascrivere locazioni di memoria con esiti diversi da quelli attesi.

3. Organizzazione del programma

Variabili vettoriali

Nel caso di variabili vettoriali, se si ha la necessità di leggere un bit di un elemento del vettore conviene prima copiare l'elemento voluto su una variabile di appoggio, quindi accedere al singolo BIT della variabile.

Esempio:

Supponendo che sia la variabile vettoriale CONF1(X) che la variabile TEMPOR siano di tipo WORD

TEMPOR=CONF1(2) *Si copia il contenuto del 2° elemento di CONF1 su TEMPOR*
 BIT12=TEMPOR(12) *Si legge il 12° BIT sulla variabile di appoggio TEMPOR e lo si pone nella variabile BIT12*

Se, invece, si desidera scrivere un singolo BIT di un elemento del vettore occorrerà dapprima scrivere il BIT su una variabile di appoggio e quindi sovrapporre quest'ultima all'elemento del vettore. Per maggiori dettagli sul trattamento dei bit, vedere il Capitolo 5 *Operazioni e funzioni* più avanti.

3.3.7. ACCESSO AI BIT DI VARIABILI ADIACENTI

Se il valore dell'indice è maggiore del formato della variabile a cui è riferito si andrà, come già detto in precedenza, ad accedere ai bit delle variabili successive nel modo seguente:

Esempi:

Supponendo che le variabili VAR1, ALARM e CONFIG siano di tipo BYTE e che VAR2 e FLAGS siano di tipo WORD, si ottiene quanto segue accedendo ai bit con le espressioni indicate sulla destra.

VAR1	8	7	6	5	4	3	2	1	VAR1 (3)
ALARM	8	7	6	5	4	3	2	1	VAR1(10)=ALARM(2)
CONFIG	8	7	6	5	4	3	2	1	

HI VAR2	8	7	6	5	4	3	2	1	VAR2(3)
LO VAR2	16	15	14	13	12	11	10	9	
HI FLAGS	8	7	6	5	4	3	2	1	VAR2(18) = FLAGS(2)
LO FLAGS	16	15	14	13	12	11	10	9	

4. DICHIARATIVE INIZIALI

Tutti i simboli che saranno utilizzati all'interno di un programma PLC devono essere dichiarati all'inizio del programma con una delle seguenti parole chiave descritte in dettaglio più avanti in questo capitolo.

INP	input fisici
TERM	terminale input
OUT	output fisici
TERM	terminale output
SRAM	variabili interne in memoria RAM ritentive allo spegnimento del CN
RAM	variabili interne in memoria RAM non ritentive allo spegnimento del CN
STR	stringhe
EQU	equivalenza o sinonimo
PULSE	derivate impulsive
FTIMER	temporizzatori veloci
STIMER	temporizzatori lenti
COUNT	contatori
LANG	lingue delle softkey
SOFTK	softkey

Le funzioni dichiarative elencate sopra non sono obbligatorie, ma, quando previste, devono rispettare questo ordine.

In generale, nei casi in cui per una stessa dichiarativa si possono avere più formati, tranne per input e output, occorre iniziare la descrizione dal formato maggiore.

Esempio:

```

SRAM,64      I nomi che seguono sono in formato DOUBLE
NOMEA
...
RAM,32       I nomi che seguono sono in formato LONG
NOMEF
...
SRAM,16     I nomi che seguono sono in formato WORD
NOMEL
...
SRAM,8      I nomi che seguono sono in formato BYTE
NOMEF
...

```

4. Dichiarative iniziali

4.1. DICHIARAZIONI DI INPUT/OUTPUT FISICI

Il programma dovrà iniziare sempre con le dichiarazioni dei nomi dei segnali collegati fisicamente alle piastre di INPUT/OUTPUT.

La corrispondenza fra variabili di INPUT/OUTPUT e posizione fisica sui connettori dipende dalla posizione in cui è dichiarato il nome variabile rispetto alla parola chiave di dichiarazione:

Per i morsetti di INPUT si deve usare la dichiarativa INP con il seguente formato:

```
INP[,attributo][,numero piastra]  
nome ingresso 1  
...  
nome ingresso n
```

mentre per i morsetti di OUTPUT:

```
OUT[,attributo][,numero piastra]  
nome uscita 1  
...  
nome uscita n
```

Dove :

[,attributo]	Definisce il tipo di input. Può essere: ,1 descrive 1 solo BIT (valore di default se omissso attributo) ,8 descrive un BYTE ,16 descrive una WORD
[,numero piastra]	indica quale è il POSTO PIASTRA della scheda base di I/OMIX che si sta descrivendo (vedere manuale di installazione sistema).

Nel caso questo parametro venga omissso, si assume 1 per default.

Dopo la dichiarazione del tipo di INPUT o di OUTPUT da descrivere si dovranno elencare tutti i nomi delle variabili di quel tipo.

Esempio :

```
INP  
NOMEA      nome dell'ingresso numero 1  
NOMEB      nome dell'ingresso numero 2  
NOMEC      nome dell'ingresso numero 3  
OUT  
NOMED      nome dell'uscita numero 1  
NOMEE      nome dell'uscita numero 2
```

Con il listato riportato sopra si esegue l'assegnazione di tre nomi ai primi tre input e di due nomi ai primi due output, tutti del tipo bit.

Le espansioni di I/O della piastra base mantengono lo stesso "numero piastra" della piastra I/OMIX a cui sono connesse.

Esempio:

Configurazione: - 1 scheda I/OMIX in posizione 1.
 - 2 schede espansione I/O digitali (I/OD)

In questo caso la dichiarazione degli INP dovrà essere:

```
INP[,attributo][,1]
NOME1             ingresso numero 1 su piastra base.
NOME2
...
NOME96            ingresso numero 96 (cioè ultimo) della seconda espansione
```

I numeri e la disposizione fisica degli I/O per ogni piastra sono descritti sul manuale di *Istruzioni di installazione*.

Nel caso invece esista un gruppo di 8 o di 16 segnali che debbano essere trattati come una unica grandezza, può essere utile definirla come BYTE o come WORD. In questo caso, per accedere ad un singolo segnale del gruppo valgono le stesse regole dell'accesso ai bit delle variabili (vedere *Accesso ai singoli bit di variabili* nel capitolo precedente).

Esempio:

```
INP,8
NOME
```

oppure:

```
INP,16
NOME
```

Generalizzando, è possibile avere la doppia dichiarazione per il trattamento misto tramite una sintassi del tipo:

gruppo:[nome1][,nome2][,....[,nomen]

Dove **gruppo** richiama il gruppo di segnali e **nome1....nomen** richiamano i singoli bit con **n** limitato dall'estensione della grandezza che si sta descrivendo: sino a 8 per un BYTE e sino a 16 per una WORD.

Tutti i termini che seguono il nome del gruppo di segnali (nome1 ... nomen) sono opzionali. Questo vuol dire che si può omettere qualsiasi elemento della lista.

Nel caso non vengano descritti alcuni dei nomi intermedi dei bit basta ometterne il nome ma deve essere mantenuta la virgola corrispondente.

Dopo l'ultimo nome non sono necessarie le virgole; il compilatore troncherà automaticamente la descrizione dei segnali a quel punto.

Esempio:

```
INP,8
INGR1:LIVOIL,IPLUBE,,,TERMAX,TERMAY
```

Non sempre i morsetti di input/output sono consecutivi a partire dal primo. In questo caso è necessario definire il numero dell'ultimo terminale inutilizzato con la funzione **TERM**, e quindi riprendere a dichiarare i restanti segnali. Il formato di detta funzione è il seguente:

TERM,numero

Il valore di **numero** può essere qualsiasi se si tratta di bit, ma multiplo di 8 o di 16 se dopo vengono dichiarati BYTE o WORD.

4. Dichiarative iniziali

Esempio:

INP
 IFCXP [ingresso 1
 TERM,5
 ISPOK [ingresso 6

Con il listato precedente non vengono utilizzati i morsetti da 2 a 5 compresi e si riparte dal morsetto 6 con il segnale ISPOK.

Se viene abilitato il parametro "diagnostica accesso I/O" nel setup di impianto, quando si tenta di accedere in lettura o scrittura da PLC o debug della logica, ad una risorsa non presente, compare il messaggio: "**E1994**: accesso a componente assente" tipicamente seguito dal numero della riga PLC dove si è verificata l'incongruenza.

La diagnostica verifica la congruenza tra l'indirizzamento e l'effettiva presenza delle risorse accessibili da PLC (cioè uscite ed ingressi digitali, uscite ed ingressi analogici, sonde termiche).

4.1.1. DICHIARAZIONE DI INPUT/OUTPUT FISICI: MODULI DI I/O REMOTATI

Per indirizzare gli I/O digitali sui moduli remotati occorre utilizzare la dichiarativa INP o OUT estesa, seguita dalla lista dei nomi attribuiti alle variabili.

Per i morsetti di INPUT si deve usare la dichiarativa INP con il seguente formato:

INP,attributo,numero piastra master,numero slave
nome ingresso 1

...

nome ingresso n

mentre per i morsetti di OUTPUT:

OUT,attributo,numero piastra master,numero slave
nome uscita 1

...

nome uscita n

Dove :

attributo

Definisce il tipo di input. Può essere:

,1 descrive 1 solo BIT (valore di default se omissso attributo)
 ,8 descrive un BYTE
 ,16 descrive una WORD

numero piastra master

indica quale è il POSTO PIASTRA della scheda con interfaccia master RIO analogamente al caso di I/O locali ove è relativo alla scheda I/OMIX. Nel caso di utilizzo della scheda master con RIO integrato, il numero piastra sarà 17.

numero slave

dichiara l'indirizzo impostato con i microinterruttori sul modulo remotato.

Esempio :

INP,1,17,60 *input formato bit, master17, slave 60*
 NOMEA *nome dell'ingresso numero 1*
 NOMEB *nome dell'ingresso numero 2*
 NOME C *nome dell'ingresso numero 3*
 OUT,1,17,60 *output formato bit, master17, slave 60*
 NOMED *nome dell'uscita numero 1*
 NOME E *nome dell'uscita numero 2*

4.2. DICHIARAZIONE VARIABILI INTERNE

Sono così definite quelle variabili o segnali di appoggio necessarie per calcoli o memorizzazioni interne non legate direttamente a segnali fisici.

A seconda che le variabili debbano o meno essere ritentive durante lo spegnimento del CN, cioè che a controllo spento conservino l'ultimo valore assegnato, si utilizzeranno rispettivamente due dichiarative differenti, il cui formato è quello che segue:

SRAM[,attributo] Variabili ritentive

Variabile interna 1

...

Variabile interna n

RAM [,attributo] Variabili non ritentive

Variabile interna 1

...

Variabile interna n

dove:

[,attributo] può assumere i valori seguenti per entrambe le dichiarative:

- ,1** per indicare una variabile di formato BIT (valore di default, se omesso)
- ,8** per indicare una variabile di formato BYTE
- ,16** per indicare una variabile di formato WORD
- ,32** per indicare una variabile di formato LONG
- ,64** per indicare una variabile di formato DOUBLE

Accanto ai tipi RAM,x e SRAM,x esiste la possibilità di gestire delle variabili, dette SSRAM, che non vengono resettate con le normali operazioni di reset del CN o con la ricompilazione del PLC.


Le SSRAM sono dimensionabili con le stesse taglie delle normali SRAM.

Esempio:

SSRAM,16

ORELAV [contatore ore di lavoro macchine

Lo spazio disponibile per le **SSRAM** è molto limitato (96 byte); l'area relativa a queste variabili viene resettata quando si compila un PLC con all'interno una dichiarazione di **SSRAM** diversa da quella precedente.

 Su S1200 le variabili non ritentive (**RAM [,attributo]**) erano implicitamente ritentive

A differenza degli INPUT/OUTPUT si possono anche descrivere vettori (cioè tabelle) di variabili interne, per tutti i formati tranne il BIT.

Si avrà quindi:

nome(numero):[nome1],[nome2],[.....],[nomen]

numero indica il numero di elementi del vettore. Opzionalmente, a destra del carattere ":" si possono specificare i nomi dei diversi elementi nello stesso formato del vettore.

Nel caso non vengano descritti alcuni dei nomi intermedi basta omettere il nome e mettere la virgola corrispondente.

4. Dichiarative iniziali

Dopo l'ultimo nome non sono necessarie le virgole; il compilatore troncherà automaticamente la descrizione dei segnali a quel punto.

Il numero di elementi di un vettore può arrivare a 32767 compatibilmente con la memoria disponibile.

Come si è già detto in precedenza, le variabili di tipo LONG e DOUBLE, essendo in formato floating point, saranno sempre utilizzate per effettuare calcoli matematici.

4.3. DICHIARAZIONE DI STRINGHE

Per stringa si intende una variabile che contiene caratteri alfanumerici, esclusi gli apici, codificati in formato ASCII. Generalmente questa dichiarativa sarà utilizzata per contenere messaggi.

La dichiarazione dei nomi delle stringhe si effettua dopo la parola chiave:

STR [,attributo]

Variabile stringa 1

...

Variabile stringa n

dove **[,attributo]** può assumere i seguenti valori:

,16 per stringhe con lunghezza max 14 caratteri
,32 per stringhe con lunghezza max 30 caratteri
,64 per stringhe con lunghezza max 62 caratteri
,128 per stringhe con lunghezza max 126 caratteri
,256 per stringhe con lunghezza max 254 caratteri

Se l'attributo viene omesso si assume come valore di default **,64**.

Se si tenta di scrivere una stringa più lunga del formato con cui è dichiarata, essa viene automaticamente troncata e compare un messaggio di errore sul video.

Anche per le stringhe è possibile utilizzare una dichiarativa esplicita o di tipo vettoriale:

- In forma esplicita elencando i singoli nomi attribuiti:

STR
 NOMEA
 NOMB

- Utilizzando una simbologia di tipo vettoriale con un solo nome ed il numero di elementi:

STR
 NOME(n)

Esempio:

....
 STR,64
 NOMEA
 MSG(12)
 ...
 PROG

....
 NOMEA='MESSAGGIO DIMOSTRATIVO'
 MSG(2)='INTERVENTO PROTEZIONE MANDRINO'

assegnazione del contenuto alla variabile stringa NOMEA
assegnazione del contenuto alla variabile stringa vettoriale MSG(2)

4.4. DICHIARAZIONI DI EQUIVALENZE

Con le equivalenze si possono assegnare nomi diversi a variabili già definite nelle dichiarative precedenti.

La funzione di equivalenza sarà del tipo:

EQU[*attributo*]
nome1:nome2
 ...
nomem:nomen

dove la dimensione della variabile introdotta è dichiarata dal valore di **attributo** e quindi può assumere tutti i valori permessi alle variabili interne (1, 8, 16, 32, 64).

Le assegnazioni che seguiranno devono essere del tipo

nomex:nomey

dove **nomex** è il nuovo simbolo da inserire e **nomey** una grandezza che deve già essere stata dichiarata.

Esempio:

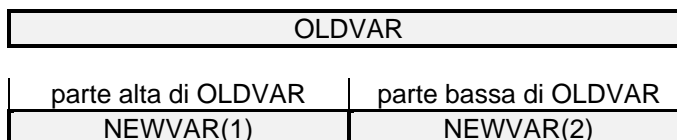
```
RAM,8
ARRAY(10)
...
EQU,8
NOMEX: ARRAY(3)   la nuova variabile NOMEX descrive il terzo BYTE di ARRAY definito prima con 10 elementi.
[
EQU,16
WORD: ARRAY(1)    la variabile WORD farà riferimento ai primi 2 BYTE dell'array
```

In aggiunta alla sintassi dell'esempio precedente è possibile dichiarare un nuovo operando di tipo vettoriale.

Esempio:

```
RAM,16
OLDVAR
EQU,8
NEWVAR(2):OLDVAR
```

Dove NEWVAR(2) è un vettore di due elementi di formato BYTE
 in cui NEWVAR(1) è equivalente alla parte alta di OLDVAR
 NEWVAR(2) è equivalente alla parte bassa di OLDVAR



E' possibile, tramite la dichiarativa **EQU**, assegnare equivalenze tra variabili stringa e vettori di byte (Questa prestazione è utile per avere disponibile un vettore contenente i caratteri ASCII di una stringa data).

4. Dichiarative iniziali

```

Es.
STR
BUFSTR          [variabile stringa
[
EQU,8
VETSTR(64):BUFSTR [associo un vettore di 64 byte alla stringa
[
PROG
BUFSTR='ABCD'
[
[VETSTR(1)=0      [questo byte e' sempre a 0
[VETSTR(2)=4      [il secondo byte contiene la lunghezza stringa
[VETSTR(3)=65     [codice ASCII della lettera A
[VETSTR(4)=66     [codice ASCII della lettera B
[VETSTR(5)=67     [codice ASCII della lettera C
[VETSTR(6)=68     [codice ASCII della lettera D
[VETSTR(7)=XX     [... altro

```

4.5. PULSE

La funzione pulse è la derivata del fronte di salita di un segnale. Ha lo scopo di creare un impulso che qualsiasi equazione della logica vede una sola volta. Essa viene attivata all'inizio di una scansione di logica lenta quando la variabile generatrice, o l'equazione generatrice, ha avuto una transizione dal livello logico "0" al livello "1" e disattivata alla fine della scansione completa di logica.

Nel caso di una pulse programmata nella logica veloce, è sufficiente che la variabile generatrice duri per il tempo minimo di cattura per attivare un impulso pari ad una scansione completa di tutta la logica: in questo caso si ottiene l'allungamento di un segnale che altrimenti sarebbe potuto sfuggire alla logica lenta.

Per la tecnica di sincronizzazione descritta, va considerato che il fronte di salita della pulse generalmente non coincide con il fronte del segnale generatore o del risultato dell'equazione generatrice, ma ritarda di un tempo che può arrivare ad essere pari una scansione completa del programma PLC.

Nota: Le pulse non sono ritentive per cui, all'accensione del CN, se sono associate ad un segnale già ad uno (es. un input), generano un segnale di derivata.

Nel programma l'equazione che definisce la PULSE viene scritta come quella di un segnale qualsiasi. Per il riconoscimento si devono definire i nomi dei segnali che si vogliono trattare come derivate, separatamente da quelli ordinari.

Si possono definire fino a 64 PULSE nella sezione dichiarativa, con una sintassi del tipo:

PULSE**nomea**

...

nomen**Esempio:**

PULSE

NOMEA

NOMEB

...

PROG

NOMEA=(NCMD=5) *il segnale nomea è un impulso di CN in stato manuale*

...

NOMEB=EMEA *il segnale nomeb è un impulso di CN in Emergenza*

4.6. TEMPORIZZATORI

Sono a disposizione dell'utente 32 timer veloci, con un tempo base di 10 msec (un ciclo), capaci di contare fino a 327.67 secondi e 64 timer lenti con un tempo base di 100 msec (10 cicli), capaci di contare fino a 3276.7 secondi (quasi un'ora).

I temporizzatori sono dichiarati come tali nella sezione delle dichiarative, ma la loro durata deve essere dichiarata all'interno del programma nel momento in cui sono utilizzati.

I timer devono essere definiti dopo la dichiarativa **FTIMER** se fast o **STIMER** se slow con una sintassi del tipo:

FTIMER (o **STIMER**)
ingresso, uscita, derivata, arresto, conta
 ...
ingresso, uscita, derivata, arresto, conta

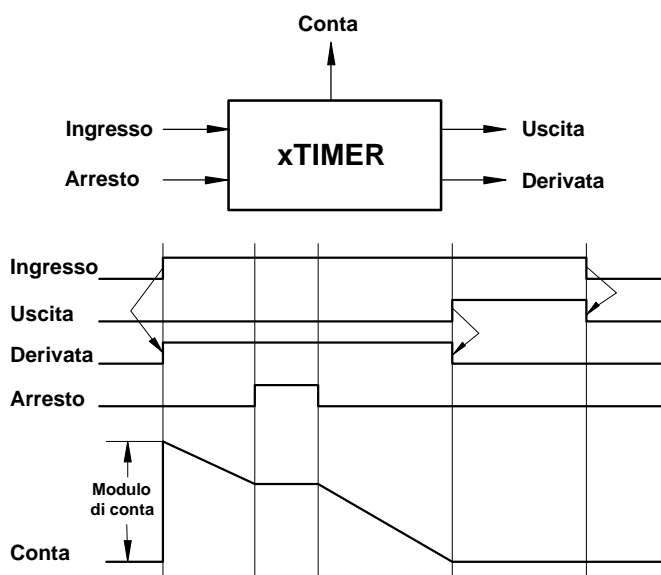
oppure

FTIMER (o **STIMER**)
ingresso, uscita
ingresso, uscita

Dove:

ingresso è il nome di un segnale che attiva il timer
uscita è il nome del segnale di tempo scaduto
derivata è il nome del segnale di tempo in corso
arresto è il nome del segnale di sospensione della conta
conta è il nome della WORD che contiene il valore corrente del conteggio.

Lo schema funzionale del temporizzatore è il seguente:



Nota: L'uscita del timer resta attiva (ad 1) fintanto che l'ingresso resta attivo.

4. Dichiarative iniziali

ingresso	Se = 1 il timer conta secondo la sua base tempi. Se = 0 l'uscita del timer viene azzerata, mentre il valore di conta è lasciato inalterato. Il timer viene ricaricato con il modulo di conta sulla transizione da 0 a 1 della variabile ingresso.
arresto	Con la transizione da 0 a 1 si congelano i valori di quel momento e inibisce il conteggio del timer. Con la transizione da 1 a 0 si riparte a contare da dove era rimasto sospeso.
uscita	Va ad 1 quando il tempo è scaduto. Ritorna a 0 quando l'ingresso va a 0.
derivata	Ad 1 durante l'intervallo di conta.

Tutti i simboli dichiarati per i timer possono essere sia scritti sia letti nel programma, ad eccezione delle uscite **uscita** e **derivata** che possono essere solo lette.

L'assegnazione del parametro di tempo, che non deve essere definito nella parte dichiarativa, dovrà essere eseguita nel testo del programma accanto al nome del segnale di ingresso del timer stesso laddove viene generato.

Questo offre la possibilità di modificarne il tempo nel corso del programma e quindi di utilizzare il timer con tempo costante o parametrico.

Per rendere identici i segnali del timer in qualsiasi punto del programma, si sincronizzano le loro transizioni sulla lettura dell'istruzione che definisce l'ingresso del timer stesso. Ciò implica la condizione che l'uscita del timer, come la sua derivata, siano aggiornate solo se il programma di logica di macchina legge l'istruzione di ingresso del timer.

La sintassi per attivare un **TIMER** all'interno del programma ,sarà del tipo

ingresso(modulo di conta) = espressione

dove **modulo di conta** può essere:

- un numero compreso tra 1 e 32767
- una variabile di tipo BYTE o WORD con un contenuto compreso nel range 1-32767
- un'espressione che restituisca un risultato nel formato BYTE o WORD con le stesse limitazioni numeriche del caso precedente.

Esempio 1:

```
FTIMER
T1I,T1U,T1D,T1A,T1W      dichiarazione timer 1
T2I,T2U,T2D,T2A,T2W      dichiarazione timer 2
...
PROG
T1I(25)=.....            utilizzazione del timer 1 con tempo fissato a 250 msec.
T2I(2*TIMBAS+10)=.....    utilizzazione del timer 2 con base tempo parametrica in funzione di TIMBAS
...
```

Esempio 2:

```
OUT
U1                          uscita oscillatore
STIMER
T2I,T2U,T2D,T2A,T2W      dichiarazione timer 2
...
PROG
T2I(10)= "T2U              timer 2 usato come oscillatore con base tempo 1 sec
U1=(T2W<5)                uscita ad onda quadra con periodo 1 sec
```

4.7. CONTATORI

Si possono definire fino a 48 contatori up/down a modulo programmabile entro 2 e 32767.

Anche i contatori, come i timer, devono essere definiti nella parte dichiarativa, ma il modulo, cioè la quantità da contare, deve essere definita all'interno del programma. La parte da inserire nella sezione delle dichiarative ha il seguente formato:

COUNT

azzeramento, avanti, indietro, riporto, conta

azzeramento, avanti, indietro, riporto, conta

Dove:

azzeramento è il nome del segnale che azzerava il contatore.

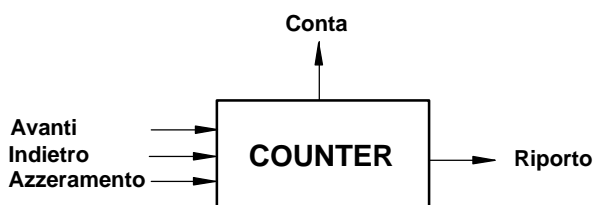
avanti è il nome del segnale che fa avanzare il contatore.

indietro è il nome del segnale che fa arretrare il contatore.

riporto è il nome del segnale generato dal contatore al passaggio per lo zero.

conta è il nome della WORD che contiene il valore cumulato.

Lo schema funzionale è il seguente:



Il funzionamento del contatore è il seguente:

azzeramento nella transizione da 0 a 1 viene caricato 0 sul valore di conta

avanti ad ogni fronte di salita il contatore viene incrementato

indietro ad ogni fronte di salita il contatore viene decrementato

riporto segnala il passaggio per lo zero del valore di conta ossia overflow o underflow sul modulo di conta a seconda se il contatore era UP o DOWN.

Le figure che seguono illustrano il comportamento di un contatore con modulo 10 con conta avanti e indietro.

Conta avanti

Azzeram.	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
Avanti	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	...
Conta	0	1	2	3	4	5	6	7	8	9	0	1	2	...								
Riporto	0	0	0	0	0	0	0	0	0	0	1	0	0	...								

4. Dichiarative iniziali

Conta indietro

Azzeram.	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
Indietro	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	...
Conta	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	...
Riporto	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	...

Come si vede, nella conta avanti, quando il contatore raggiunge il valore del modulo, si azzerava automaticamente, mentre nel caso della conta indietro, dopo aver raggiunto lo zero viene caricato il modulo di conta.

In entrambi i casi, il passaggio per lo zero viene segnalato dall'attivazione del segnale di riporto.

Tutti i simboli dichiarati per i contatori possono essere scritti e letti nel programma, salvo l'uscita che può solo essere letta.

L'assegnazione del parametro di conta, che non deve essere definito nella parte dichiarativa, dovrà essere effettuata nel programma accanto al nome del segnale di azzeramento del contatore stesso, laddove viene generato.

Questo offre la possibilità di modificarne il modulo nel corso del programma e quindi di utilizzare il contatore con modulo costante o parametrico.

Il caricamento del modulo di conta avviene sul rilascio del segnale di azzeramento.

Esempio:

```

INP
ICOMAI                ingresso conteggio magazzino indietro
COUNT
C1Z,C1A,C1I,C1R,C1C   dichiarazione counter 1
C2Z,C2A,C2I,C2R,C2C   dichiarazione counter 2
...
PROG
C1Z(50)=.....        utilizza il contatore 1 con modulo 50.
C1I=ICOMAI            l'input di conteggio magazzino indietro decrementa il contatore
...
C2Z(TEMPO/60)=.....    [utilizza il contatore 2 il cui modulo è definito in funzione della variabile TEMPO

```

4.8. SOFTKEY DEFINIBILI DA LOGICA

Il sistema mette a disposizione una serie di 8 tasti funzione disposti verticalmente sul lato destro del video, sempre accessibili con i tasti funzione da **F11** a **F18**, interamente gestibili e definibili dalla logica di macchina con una struttura configurabile da programma.

E' così possibile mantenere uniforme l'impostazione del colloquio uomo macchina estendendo la tecnica a menu anche alle funzioni solitamente realizzate con pulsanti, lampade, selettori e impostatori numerici che possono invece essere eliminati insieme a tutti i problemi di pannelli, serigrafia, cablaggio e consumo di ingressi e uscite del CN.

Un tasto softkey è trattato dal sistema come un pulsante luminoso con una targhetta.

Una volta dichiarati i nomi simbolici delle due variabili 'pulsante' e 'lampada' ed il testo della 'targhetta', sul video si avrà un nuovo tasto funzione con la targhetta desiderata in grado di trasmettere al PLC il segnale 'pulsante' e capace di illuminarsi quando il PLC attiva il segnale 'lampada'.

Le softkey possono essere 128 in totale e vanno definite in raggruppamenti composti da un numero massimo di otto elementi detti menu, identificati dalla dichiarativa **SOFTK**. Le softkey appartenenti allo stesso menu saranno contemporaneamente presenti sul video ed il passaggio ad un successivo menu o ad un sottomenu è ottenuto tramite un indice attribuito ad un tasto o tramite una apposita variabile da programma PLC (**SKMEN**).

Il tipo di segnale associato al parametro **pulsante** può essere di natura impulsiva cioè della durata di un'intera scansione di PLC (logica ultralenta) oppure può durare per tutto il tempo che la softkey associata resta premuta.

Alle softkey è anche possibile associare un messaggio ed una variabile numerica o stringa, al fine di stabilire con l'operatore un dialogo di richiesta e visualizzazione di dati.

La dichiarazione di ciascun menu di softkey *nella forma più semplice* si ottiene con una sequenza del tipo:

```
SOFTK,numero menu  
pulsante, lampada[,0|1], 'targhetta'  
...  
pulsante, lampada[,0|1], 'targhetta'
```

La selezione ,0 indica che il segnale **pulsante** è di tipo impulsivo cioè ha la durata di una scansione completa del programma di PLC. Questa è la condizione di default.

La selezione ,1 indica che il segnale **pulsante** è di tipo continuo cioè dura per tutto il tempo per cui viene mantenuto premuto il pulsante.

Nel caso si tratti della dichiarazione del primo menu, **numero menu** può essere omissso.


Sofkey associate a messaggi o variabili numeriche

Dovendo associare alla softkey un messaggio o una variabile numerica, la definizione diventa:

```
strobe, lampada, [pulsante,]'targhetta', 'messaggio', [FP:|STR:] variabile[,valore default]
```

In questo caso il segnale **pulsante** viene ricevuto dal PLC appena premuta la softkey, quindi compare sul video CN il **messaggio** seguito dal valore attuale della **variabile** associata e, solo se in seguito,

4. Dichiarative iniziali

l'operatore preme . Il segnale **strobe** viene trasmesso al PLC per segnalare l'introduzione di un nuovo dato o la conferma di quello visualizzato.

La **variabile** è implicitamente definita con il formato DOUBLE (**FP:**) a meno che non sia altrimenti specificato tramite l'identificatore **STR:** per il formato stringa.

Il **valore di default**, se definito, viene sempre proposto sulla riga di microeditor al posto del valore corrente della **variabile** quando si preme la softkey. Non è da intendersi come valore di inizializzazione della **variabile** all'accensione CN.

Softkey di richiamo menu

Quando una softkey deve richiamare un menu successivo o ritornare ad uno precedente la sintassi per il concatenamento diventa:

pulsante, lampada, 'targhetta', numero menu

In alternativa a questo metodo è possibile selezionare il menu di softkey da attivare direttamente scrivendone il numero nella variabile di PLC **SFKMEN**.





Questa variabile contiene sempre il numero del menu di softkey presente sul video, anche se il concatenamento è effettuato in modo automatico.

I rispettivi formati per le descrizioni dei testi sono di 18 caratteri su tre righe per la targhetta e di 20 per il messaggio su riga di microeditor. I testi di messaggio possono contenere tutti i caratteri ad esclusione dell'apice.

Esempio1:

SOFTK,1	<i>primo menu di softkey</i>
P1,L1, 1, 'JOG ASSE X+' ...	<i>la targhetta della prima softkey è JOG ASSE X+ ed il pulsante è di tipo continuo</i>
P7,L7,0,'RICERCA ZERO ASSI' P8,L8, 'VOLANTINO',2	<i>la targhetta della settima softkey è RICERCA ZERO ASSI ed il pulsante è impulsivo</i>
SOFTK,2	<i>la softkey, con targhetta VOLANTINO richiama il secondo menu di softkey</i>
P21,L21,1,'VOLANTINO ASSE X'	<i>secondo menu di softkey</i>
	<i>prima softkey del secondo menu ...</i>

4.9. SOFTKEY E MESSAGGI CON TESTO IN PIÙ LINGUE

I testi delle targhette e microedit associati alle softkey gestite dal PLC ( - ) possono essere definiti in modo "multilingue" per fare in modo che siano adeguate automaticamente alla lingua selezionata per i menu del CN ( - ).

Nella sezione dichiarativa del PLC occorre introdurre, prima della definizione **SOFTK**, l'istruzione:

LANG, cod_lingua_1[, cod_lingua_2][...][, cod_lingua_5]

dove i codici lingua possono essere:

- 1= Italiano
- 2= Francese
- 3= Tedesco
- 4= Inglese o Cinese
- 5= Spagnolo
- 6= Portoghese
- 7= Ceco

Nella dichiarazione dei menu, occorre specificare, per ogni softkey la targhetta ed il testo del microedit in tutte le lingue, rispettivamente, nell'ordine specificato nella dichiarativa **LANG** con la sintassi riportata nell'esempio seguente.

Esempio:

Messaggi in Italiano e Inglese:

LANG,1,4

...

SOFTK,1

P01,L01,'volantino X' 'handwheel X'

P02,L02,'tempo lubrif.' 'lubrif. time' , minuti= 'minuts= ', TIME

P03,L03,'parola chiave' 'password' , inserisci= 'insert= ', STR:CHIAVE, 'manutenzione' 'service'

...

Sulla variabile **SFKLNG** (word scritta da CN) è inoltre disponibile il codice della lingua attiva sul CN. Testando questa variabile è perciò possibile organizzare il programma PLC in modo di inizializzare le variabili stringa per la visualizzazione dei messaggi con testi in più lingue.

I codici di lingua sono gli stessi usati per la dichiarativa **LANG** e riportati sopra.

Esempio:

...

INIT

IF (SFKLNG<>1) INGL

[Inizializzazione testi messaggi in Italiano]

MSG(1)='STATO DI EMERGENZA'

MSG(2)='ANOMALIA AZIONAMENTO MANDRINO'

...

ENDMSG

[

INGL : \$

IF (SFKLNG<>4) ENDMSG

[Inizializzazione testi messaggi in Inglese]

MSG(1)='EMERGENCY STOP'

MSG(2)='FAULT ON SPINDLE DRIVE'

[

ENDMSG : \$

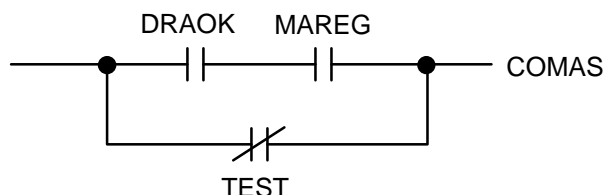
...

5. OPERAZIONI E FUNZIONI

5.1. PROGRAMMAZIONE CON ELEMENTI LOGICI ELEMENTARI

La prima rete logica che si incontra con qualsiasi PLC è una combinazione di contatti aperti o chiusi ossia di segnali veri o falsi che attivano una uscita.

Ad esempio, prendendo in esame lo schema elettrico seguente:



Volendo descrivere il funzionamento della rete logica realizzata con questo schema si può dire che l'uscita COMAS è attivata con DRAOK e con MAREG chiusi (o veri) o senza TEST (falso).

Nel linguaggio PLC S4000 si scriverà:

COMAS=DRAOK&MAREG~"TEST

utilizzando i seguenti operatori logici elementari:

&	operatore AND
~	operatore OR
 	operatore XOR
"	attributo negazione

Nell'impiego combinato dei diversi operatori logici all'interno di una equazione bisogna ricordare la priorità di esecuzione delle operazioni AND e XOR sull'operazione OR.

Nell'equazione $U=A\sim B\&C$ si intende fare l'OR di A con il risultato dell'AND tra B e C.

Se invece si desidera l'OR di A con B con il risultato in AND con C, si deve scrivere $U=(A\sim B)\&C$.

5. Operazioni e funzioni

L'introduzione delle parentesi cambia la priorità delle operazioni, come nella aritmetica convenzionale.

Le operazioni logiche analizzate applicate a segnali, ossia singoli bit, possono essere applicate indifferentemente anche a BYTE ed a WORD. In questi casi si ottengono gli stessi risultati bit a bit corrispondenti. E' evidente a questo punto che tutti gli operandi in gioco nella medesima equazione debbano essere omogenei.

Esempio:

```
RAM,16
CONFI(3)
TEMPOR
...
PROG
[esempio per negare il 4° bit del primo elemento di CONFI()]
TEMPOR=CONFI(1)
IF(TEMPOR(4)) TEMPOR(4)=0; CONFI(1)=CONFI(1) & TEMPOR; FINE
IF("TEMPOR(4)) TEMPOR(4)=1; CONFI(1)=CONFI(1) ~ TEMPOR
FINE:$
...
```

5.2. OPERAZIONI ARITMETICHE

Sono applicabili a BYTE, WORD, LONG, e DOUBLE.

Il formato tipico di un'espressione sarà:

risultato = operando operatore operando [...] [operatore operando]

I possibili operatori sono:

+	addizione
-	sottrazione
*	moltiplicazione
/	divisione
//	resto della divisione

Esempio:

$(10.5 // 7) = 3.5$

Naturalmente, eseguendo l'operazione tra BYTE o WORD si otterranno solo dei resti interi.

L'operazione // può anche essere utilizzata per ricavare la parte decimale di un numero in floating point dividendolo per 1.0

Esempio:

$$(4.123 // 1.0) = 0.123$$

La priorità tra gli operatori e l'uso delle parentesi seguono le regole classiche dell'aritmetica.

Attenzione al valore degli operandi: se il risultato della operazione è un numero maggiore della dimensione della variabili usate, si ottiene il suo complemento a 2.

Il risultato di una divisione per zero fornisce il valore massimo positivo contenuto nella dimensione scelta.

Tramite appositi codici di errore è possibile sapere se si sono verificati overflow, underflow o divisioni per zero nell'esecuzione del programma (vedere *Parte II - Lista riassuntiva dei segnali e registri predefiniti*).

5.3. FUNZIONI MATEMATICHE IN FLOATING POINT

Le funzioni seguenti possono essere utilizzate solo su variabili di tipo LONG o DOUBLE, sia singolarmente sia inserite in un algoritmo più complesso. Le funzioni trigonometriche lavorano in gradi.

SQR (<i>argomento</i>)	radice quadrata
INT (<i>argomento</i>)	intero troncato
NEI (<i>argomento</i>)	intero approssimato
SIN (<i>argomento</i>)	seno
COS (<i>argomento</i>)	coseno
TAN (<i>argomento</i>)	tangente
ATN (<i>argomento</i>)	arcotangente
LOG (<i>argomento</i>)	logaritmo
LGT (<i>argomento</i>)	logaritmo base 10
ACS (<i>argomento</i>)	arcocoseno
ASN (<i>argomento</i>)	arcoseno
NEG (<i>argomento</i>)	valore opposto o negato
SGN (<i>argomento</i>)	restituisce un valore nello stesso formato dell'operando uguale a 1 se il segno è positivo o -1 se il segno è negativo.
operando ^ operando	elevazione a potenza.
ABS (<i>argomento</i>)	fornisce il valore assoluto di una variabile BYTE, WORD, LONG, oppure DOUBLE.

Nota: Nel caso di elevazione al quadrato, è più efficiente, in termini di velocità di esecuzione, usare una sintassi del tipo **argomento*argomento** anziché **argomento^argomento**.

5.4. CONFRONTO O PARAGONE

E' spesso ricorrente la necessità di confrontare tra loro due variabili oppure una variabile con un valore costante e trattare poi il risultato, vero o falso, esattamente come il livello logico di un bit.

Sono previsti tutti i casi di confronto utilizzando i seguenti simboli:

=	per uguale
<>	per diverso
>	per maggiore
<	per minore
<=	per minore uguale
>=	per maggiore uguale

L'espressione di confronto deve essere racchiusa tra parentesi tonde e quindi può essere utilizzata come elemento logico (BIT) all'interno di una qualsiasi equazione.

Esempio:

$MAOR=(AUXM=3)\sim(AUXM=13)$ *MAOR è vera (a 1) quando AUXM=3 o AUXM=13*

Possono essere messe a confronto variabili e costanti di tipo BYTE, WORD, LONG e DOUBLE purché omogenee tra di loro.

Non è possibile mettere direttamente a confronto stringhe e variabili stringa; per tale scopo esiste l'apposita funzione **STRCMP()**, descritta in seguito.

5.5. ROTAZIONE

Ad una variabile, purché non dei tipi BIT, LONG o DOUBLE, si può applicare la funzione di rotazione mediante l'operando @ seguito dal numero di rotazioni da effettuare.

variabile @ + n effettua una ROTAZIONE a sinistra
variabile @ - n effettua una ROTAZIONE a destra

dove **n**, in formato BYTE o WORD, è il numero di rotazioni

Una rotazione a sinistra sposta tutti i bit nella direzione di quello più significativo, il quale, a sua volta, va ad occupare il meno significativo. Nella rotazione verso destra si verifica il contrario.

Esempio:

$STATP = STATP @+1$ *esegue una rotazione a sinistra di una posizione di tutti i bit.*

Prima della rotazione	1	0	0	0	1	0	0	0
Dopo la rotazione	0	0	0	1	0	0	0	1

5.6. FUNZIONI DI TRASFORMAZIONE TRA FORMATI DIVERSI

E' questo un set di funzioni che producono una variabile di **uscita** come **trasformazione** di una variabile data in input (**argomento**).

La sintassi, uguale per tutte le funzioni, è:

uscita = funzione (argomento)

dove **argomento** può essere anche un'espressione complessa.

ENC - Ricerca bit

Scandisce il valore dell'**argomento** a partire dal BIT meno significativo e fornisce in uscita un numero da 1 a 16 per le WORD, o da 1 a 8 per i BYTE che indica la posizione del primo bit trovato a 1.

Esempio:

ENC (10100000B) = 6 *poiché il sesto bit a partire da destra è il primo ad uno e i bit successivi vengono ignorati.*

DEC - Setta bit

Fornisce in uscita un numero nel formato BYTE o WORD con un bit a 1 nella posizione corrispondente al valore dell'**argomento**, purché detto valore non sia superiore a 16 o a 8 a seconda del formato WORD o BYTE.

Esempio:

DEC (7) = 01000000B *poiché il numero è 7 mette a 1 il settimo bit della parola di uscita.*

HI - Estrae il byte alto di una word

Converte in un BYTE gli otto bit più alti di una WORD (**argomento**).

Esempio:

BYT1=HI(WORD1) *isola la parte alta di WORD1*

LO - Estrae il byte basso di una word

Converte in un BYTE gli otto bit più bassi di una WORD (**argomento**).

Esempio:

BYT1=LO(WORD1) *isola la parte bassa di WORD1*

5. Operazioni e funzioni**EXT - Estensione di un byte in una word**

Estende un BYTE (**argomento**) in WORD con conservazione del segno. in altri termini, se il bit 8 (bit di segno) era 0 aggiunge 8 zeri a sinistra; se era 1 aggiunge otto 1 a sinistra.

Esempio:

WORD2=EXT(BYTE1)

BCD - Converte in codice BCD un numero binario

Trasforma un BYTE(**argomento**) in 2 cifre BCD oppure una WORD (**argomento**) in 4 cifre BCD.

Esempio:

BCD1=BCD(BYTE1) se BYTE1 era uguale a 00001100 (12 decimale), BCD1 sarà 0001 0010

BIN - Converte un numero BCD in byte o word

Converte in binario le 2 cifre BCD contenute in un BYTE (**argomento**) o le 4 contenute in una WORD (**argomento**). E' cioè la funzione opposta di **BCD**.

Esempio:

BYTE1=BIN(BCD1) se BCD1 era uguale a 0001 0010, BYTE1 sarà 00001100

IFP - Converte un byte o una word in formato floating point

Trasforma un numero dal formato BYTE o WORD (**argomento**) in floating point. Questa funzione è necessaria per poter eseguire operazioni matematiche ad essi riservate.

Esempio:

NUTF=IFP(DTOOL) conversione in floating della variabile DTOOL

FPI - Converte un formato floating point in byte o word

Trasforma un numero in FLOATING POINT (**argomento**) nel formato WORD o BYTE.

5.6.1. ESPRESSIONI COMPLESSE

Le funzioni di trasformazione tra i vari formati, descritte precedentemente, sono utilizzabili in combinazione con le funzioni aritmetiche e matematiche disponibili per comporre espressioni complesse.

Non tutte le funzioni sono però utilizzabili in espressioni complesse. In particolare, non sono utilizzabili:

- Funzioni con più di un argomento:
FF(...), (...), MUX(...), RIC(...)
- Funzioni ad argomento stringa:
VAL(...), LEN(...), INSTR(...), STRCMP(...)

Esempio 1:

RAM,8
ANGLE
RAM,32
RESULT

...

RESULT = SIN(IFP(ANGLE * 2 + 45)) *si trasforma in floating point il risultato dell'espressione ANGLE*2+45 e si calcola il seno relativo*

Esempio 2:

POWER = OFFSET + SIN(1 / FREQ * TIMBAS) + COS(ANGLE) *si esegue il seno del risultato dell'espressione 1/FREQ*TIMBAS, lo si somma al coseno di ANGLE e ad OFFSET per ottenere POWER.*

5.7. OPERAZIONI E FUNZIONI SU STRINGHE

In generale le stringhe possono contenere tutti i caratteri ad esclusione dell'apice.

5.7.1. FUNZIONI NUMERICHE AD ARGOMENTO STRINGA

Sono funzioni che restituiscono un valore numerico avendo come argomento una stringa:

VAL - Trasforma un numero da formato ASCII a numerico

Fornisce il valore numerico di una variabile stringa. Il suo formato è:

VAL(argomento)

dove **argomento** può essere:

- una variabile di tipo stringa
- una espressione che restituisca un risultato di tipo stringa.

5. Operazioni e funzioni

Il valore fornito dalla funzione può indifferentemente essere assegnato a variabili di tipo BYTE, WORD, LONG, DOUBLE purché rispetti i valori massimi ammessi.

La conversione si arresta al primo carattere non numerico.

Esempio:

```
RAM,32
NUMVAL
STR
NUMSTR
```

```
...
```

```
PROG
```

```
NUMSTR='123.56'
```

```
NUMVAL=VAL(NUMSTR)
```

NUMVAL contiene la conversione numerica di NUMSTR cioè NUMVAL=123.56

INSTR - Ricerca una stringa all'interno di un'altra

Ricerca una stringa all'interno di una stringa data a partire dalla posizione e per una lunghezza specificati e fornisce la posizione all'interno della stringa data in cui viene trovato il carattere iniziale della stringa da ricercare. Il suo formato è:

INSTR(argomento1, argomento2, argomento3, argomento4)

dove:

argomento1 è la stringa su cui effettuare la ricerca.

argomento2 è la stringa da ricercare

argomento3 è la posizione del carattere da cui iniziare la ricerca

argomento4 indica per quanti caratteri in avanti cercare **argomento2** a partire da **argomento3**.

argomento1 e **argomento2** possono essere:

- una sequenza di caratteri delimitata da apici
- una variabile di tipo stringa
- una espressione che restituisca un risultato di tipo stringa.

argomento3 e **argomento4** possono essere:

- un numero esplicito intero compresa tra 1 e 255
- una variabile di tipo BYTE con valore compreso tra 1 e 127 o WORD con valore compreso tra 1 e 255
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

Il valore fornito dalla funzione può indifferentemente essere assegnato a variabili di tipo BYTE, WORD.

In funzione dei valori assunti da **argomento1-4** o a fronte di condizioni rilevate, la funzione si comporta in modo diverso, come indicato sotto:

- Se la stringa non viene trovata viene restituito il valore 0
- se **argomento 2** è una stringa nulla viene restituito **argomento3**
- se **argomento1** è una stringa nulla viene restituito il valore 0
- se **argomento3** + **argomento4** è maggiore della lunghezza di **argomento1**, la ricerca viene effettuata da **argomento3** finché ci sono caratteri disponibili
- se **argomento4** è minore o uguale a 0 viene restituito 0.

Esempio:

Si supponga che VARIAB1 contenga 'ABCDEABCUPABCXY' e VARIAB2 contenga 'AB', con l'istruzione

```
POSIZ=INSTR(VARIAB1,VARIAB2,4,13)
```

si otterrà in POSIZ il numero 6

LEN - Lunghezza di una stringa

Fornisce la lunghezza della stringa **argomento**, cioè il numero di caratteri contenuti, compresi gli spazi. Il suo formato è:

LEN(argumento)

dove **argomento** può essere:

- una variabile di tipo stringa
- una espressione che restituisca un risultato di tipo stringa.

Il valore fornito dalla funzione può indifferentemente essere assegnato a variabili di tipo BYTE o WORD.

Esempio:

RAM,8

LUNST

STR

MSG1

...

PROG

MSG1='ALLARME NUMERO 3'

LUNST=LEN(MSG1)

LUNST contiene il numero di caratteri di MSG1 cioè LUNST=16

STRCMP - Confronto tra stringhe

Esegue l'operazione di confronto specificata con **operatore** fra **argomento1** e **argomento2** di tipo stringa e restituisce un valore vero o falso. **Argomento1** e **argomento2** possono avere anche formati diversi. Il suo formato è:

STRCMP(argumento1 operatore argomento2)

operatore può essere <, >, <=, >=, =, <>.

argomento1 e **argomento2** possono essere:

- una sequenza di caratteri delimitata da apici
- una variabile di tipo stringa
- una espressione che restituisca un risultato di tipo stringa.

Il valore restituito dalla funzione è di tipo BIT.

5. Operazioni e funzioni

Il valore del risultato rispecchia le seguenti condizioni:

argomento1 > argomento2 Se, analizzando le due stringhe, dal primo verso l'ultimo carattere, il codice ASCII del carattere incontrato su **argomento1** è maggiore di quello dello stesso carattere su **argomento2**.

Esempio:

STRCMP('COSE'>'COSA') *dà risultato vero*

argomento1 > argomento2 Se la condizione precedente non è verificata e la lunghezza di **argomento1** è maggiore della lunghezza di **argomento2**.

Esempio:

STRCMP('COSE'>'CO') *dà risultato vero*

argomento1 = argomento2 Se tutti i caratteri di **argomento1** sono uguali ai caratteri di **argomento2** (compresi i blank).

Esempio:

RAM,1

TEST

...

PROG

TEST = STRCMP ('AVARIA' = 'AVARIA') *restituisce TEST = 1*


TEST = STRCMP ('AVARIA' = 'AVARIA ') *restituisce TEST = 0*

5.7.2. FUNZIONI STRINGA AD ARGOMENTO NUMERICO

Si tratta di funzioni che restituiscono come risultato delle stringhe di caratteri che possono essere associati a variabile stringa con qualsiasi dimensione.

MKN\$ - Converte un numero in formato stringa

Converte il numero, specificato variabile **argomento** in qualsiasi formato numerico, tranne BIT, in una stringa di caratteri ASCII.

 Su S1200 questa conversione era svolta dalla funzione **ASC(argumento)** del PLC.

Questa funzione deve essere utilizzata ad esempio per visualizzare come messaggio il valore di una variabile numerica. Il valore fornito dalla funzione deve essere assegnato a variabili di tipo stringa. Il suo formato è:

MKN\$(argomento)

dove **argomento** può essere:

- un numero esplicito
- una variabile
- il risultato numerico di una espressione.

Se **argomento** è nel formato BYTE, il risultato della conversione sarà su 4 caratteri in cui il primo è il segno o un blank i tre successivi sono 0 o un numero.

Per esempio:

la conversione di un byte che contiene il valore 1 sarà ' 001'

la conversione di un byte che contiene il valore -11 sarà '-011'

Se **argomento** è nel formato WORD, il risultato della conversione sarà su 6 caratteri in cui il primo è il segno o un blank i cinque successivi sono 0 o un numero.

Per esempio:

la conversione di una word che contiene il valore 1 sarà ' 00001'

la conversione di una word che contiene il valore -11 sarà '-00011'

Esempio:

MSG4=MSG5+MKN\$(SSO) Se MSG5 contiene 'Numero utensile ' e SSO è un byte con valore 12, la funzione restituisce 'Numero utensile 012'

CHR\$ - Genera un carattere ASCII

Genera un carattere il cui codice ASCII è l' **argomento** della funzione (vedere Tabella codici ASCII al fondo del manuale). Il suo formato è:

CHR\$(argomento)

Dove **argomento** può essere:

- un numero esplicito intero di valore compreso tra 0 e 255
- una variabile di tipo BYTE o WORD con valore compreso tra 0 e 255
- una espressione che restituisca un valore nel formato BYTE o WORD compreso tra 0 e 255.

Il valore fornito dalla funzione deve essere assegnato a variabili di tipo STRINGA.

Esempio:

LETTERA=CHR\$(035) LETTERA conterrà il carattere #

STRNG\$ - Genera una stringa di caratteri uguali

Genera una stringa di caratteri tutti uguali il cui codice ASCII e la lunghezza sono specificati. Il suo formato è:

STRNG\$(argomento1, argomento2)

Dove:

argomento1 è il codice ASCII del carattere con cui generare la stringa.

argomento2 è il numero di caratteri da generare.

argomento1 e **argomento2** possono essere:

- un numero esplicito intero compreso tra 1 e 255
- una variabile di tipo BYTE o WORD con valore compreso tra 1 e 255
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

5. Operazioni e funzioni

Se **argomento2** è maggiore del formato della variabile da assegnare viene effettuato un troncamento.

Esempio:

STR	
MSG	
RAM,8	
NUMCAR	
CODCAR	
PROG	
NUMCAR=20	<i>lunghezza della stringa da generare</i>
CODCAR=42	<i>codice ASCII del carattere *</i>
MSG=STRNG\$(CODCAR,NUMCAR)	<i>genera una stringa di 20 asterischi</i>

5.7.3. FUNZIONI STRINGA AD ARGOMENTO STRINGA

MID\$ - Estrae una stringa da un'altra

Estrae un numero specificato di caratteri dalla stringa data a partire da una certa posizione. Il suo formato è:

MID\$(argomento1, argomento2, argomento3)

Dove:

argomento1 è la stringa da cui estrarre

argomento2 è la posizione del carattere da cui iniziare l'estrazione

argomento3 indica quanti caratteri estrarre.

argomento1 può essere:

- una variabile di tipo stringa.

argomento2 e **argomento3** possono essere:

- un numero esplicito intero compresa tra 1 e 254
- una variabile di tipo BYTE con valore compreso tra 1 e 127 o WORD con valore compreso tra 1 e 254
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

Il valore fornito dalla funzione deve essere assegnato a variabili di tipo STRINGA e rispecchia le seguenti condizioni:

- Se **argomento2** è maggiore della lunghezza di **argomento1** viene restituita una stringa vuota.
- Se **argomento3 + argomento2** è maggiore della lunghezza di **argomento1** l'estrazione avviene finché ci sono caratteri disponibili.
- Se la lunghezza di **argomento1** è 0 viene restituita una stringa vuota.

Esempio:

Si supponga che VARIAB1 contenga 'ABCDEFGHLMN'.

VARIAB2= MID\$(VARIAB1,4,5) *In VARIAB2 si otterrà la stringa di caratteri 'DEFGH'*

LEFT\$ - Estrae una stringa a partire da sinistra

Estrae un numero specificato di caratteri dalla stringa data a partire dall'inizio della stringa. Il suo formato è:

LEFT\$(argomento1, argomento2)

Dove:

argomento1 è la stringa da cui estrarre

argomento2 indica quanti caratteri estrarre.

argomento1 può essere:

- una variabile di tipo stringa

argomento2 può essere:

- un numero esplicito intero compresa tra 1 e la lunghezza della stringa
- una variabile di tipo BYTE o WORD con valore compreso tra 1 e la lunghezza della stringa
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

Il valore fornito dalla funzione deve essere assegnato a variabili di tipo STRINGA e rispecchia le seguenti condizioni:

Se **argomento2** è maggiore della lunghezza di **argomento1**, vengono estratti tutti i caratteri disponibili.

Se la lunghezza di **argomento1** è 0, viene restituita una stringa vuota.

RIGHT\$ - Estrae una stringa a partire da destra

Estrae un numero fissato di caratteri dalla stringa data a partire dalla fine della stringa. Il suo formato è:

RIGHT\$(argomento1, argomento2)

Dove:

argomento1 è la stringa da cui estrarre.

argomento2 indica quanti caratteri estrarre.

argomento1 può essere:

- una variabile di tipo stringa

argomento2 può essere:

- un numero esplicito intero compreso tra 1 e la lunghezza della stringa.
- una variabile di tipo BYTE o WORD con valore compreso tra 1 e la lunghezza della stringa.
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

Il valore fornito dalla funzione deve essere assegnato a variabili di tipo STRINGA e rispecchia le seguenti condizioni:

Se **argomento2** è maggiore della lunghezza di **argomento1**, vengono estratti tutti i caratteri disponibili.

Se la lunghezza di **argomento1** è 0, viene restituita una stringa vuota.

5.7.4. SOMMA DI STRINGHE

L'operazione di somma su variabili di tipo stringa è usata per comporre una nuova stringa che contiene l'accodamento di più argomenti di tipo stringa fino al raggiungimento della lunghezza della stringa destinazione.

La sintassi è:

nome=argomento1[+...][+argomentoN]

dove ***argomento1***, ***argomentoN*** possono essere:

- una sequenza di caratteri alfanumerici delimitata da apici
- una variabile di tipo stringa
- una espressione che restituisca un risultato di tipo stringa.

Esempio:

RAM,32

IPERC

STR

MSG(10)

PROG

MSG(10)='Corrente assorbita '

MSG(1)=MSG(10)+MKN\$(IPERC)+' Ampere'

6. ISTRUZIONI DI CONTROLLO DEL FLUSSO DEL PROGRAMMA

Una LABEL o etichetta è un nome che si deve applicare ad una riga di programma nel caso sia previsto di accedervi tramite una istruzione di salto o come chiamata ad una subroutine.

Le label sono riconoscibili in quanto seguite dal carattere ":" e quindi da una istruzione.

Si può modificare la sequenza di esecuzione della logica mediante istruzioni di:

- SALTO INCONDIZIONATO
- SALTO CONDIZIONATO
- ESECUZIONE CONDIZIONATA
- GO TO CALCOLATA
- GO TO PONDERATA
- LOOP
- SUBROUTINE

6.1. SALTO INCONDIZIONATO

Il formato è il seguente:

labelx *l'esecuzione salta sempre al punto labelx:*

...

labelx:...

Dove:

labelx è l'istruzione di salto

labelx: è la label a cui bisogna saltare.

Nota: Il salto incondizionato ha un formalismo (**labelx . . . labelx:**) simile per certi aspetti (**nome1:nome2**) a quello delle dichiarazioni di equivalenza (Vedere al Capitolo 4.4 *Dichiarazioni di equivalenza*). La differenza sostanziale consiste nel fatto che le dichiarazioni di equivalenza sono utilizzate solo nelle dichiarazioni iniziali, mentre i salti sono allocati nelle altre parti del programma.

6. Istruzioni di controllo del flusso del programma

6.2. SALTO CONDIZIONATO

Il formato è il seguente:

IF(equazione o segnale) labelx

...

labelx:

Se l'equazione od il segnale sono veri l'esecuzione del programma salta alla **labelx**, altrimenti continua alla riga successiva.

Esempio:

IF ("BURDY) ASINC

...

ASINC: ...

6.3. ESECUZIONE CONDIZIONATA

Il formato, nel caso minimo, è il seguente:

IF(condizione) equazione

L'equazione dopo la parentesi tonda viene eseguita solo se la condizione tra parentesi è vera.

In aggiunta è possibile usare una sintassi più complessa del tipo:

IF(condizione) equazione[;...][; equazione];ELSE equazione[;...][; equazione]

Le equazioni dopo la parentesi tonda vengono eseguite solo se la condizione tra parentesi è vera, altrimenti vengono eseguite le equazioni dopo **ELSE**; questa deve essere scritta sulla stessa riga logica della **IF** corrispondente.

Se lo spazio sulla riga non è sufficiente si può andare a capo con il simbolo **\$**; la limitazione è solo imposta dal numero max di caratteri (500 esclusi i blank).

Non è possibile nidificare più istruzioni **IF** sulla stessa riga logica.

Esempio:

IF (VEMA>=1) VEMA=.9999; LIMIT=1\$
;ELSE LIMIT=0

Esempio di uso di \$ anche se la lunghezza riga è limitata

6.4. GO TO CALCOLATA

Per poter gestire agevolmente più operazioni in sequenza, questa istruzione permette di saltare ad una delle label dichiarate in funzione di un valore numerico o del risultato di una espressione.

Il formato è il seguente:

GOTC(espressione)label1[,label2] [...][,label255]

dove **espressione** può essere:

- una variabile di tipo BYTE o WORD con valore compreso tra 1 e 255
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche del caso precedente.

Il numero max di LABEL ammesse è 255; se lo spazio sulla riga non è sufficiente si può andare a capo con il simbolo \$; la limitazione è solo imposta dal numero max di caratteri (500 esclusi i blank).

Esempio:

```
RAM,8
NLAB
PROG
NLAB = ...           Label corrente a cui saltare
GOTC(NLAB) L1,L2,L3,LFINE
LFINE
...
L1:...
...
LFINE
L2:...
...
LFINE
L3:...
...
LFINE:...
```

L'espressione viene elaborata ed il risultato determina il numero d'ordine della label a cui saltare.

Se il valore dell'espressione è uguale a 0 o la label non viene trovata, l'esecuzione prosegue sulla istruzione successiva.

6.5. GO TO PONDERATA

Permette di saltare ad una delle label dichiarate in funzione del bit settato su una variabile.

Il formato è il seguente:

GOTP(espressione)label1 [,label2] [...][,label16]

dove **espressione** può essere:

- una variabile di tipo BYTE o WORD con valore compreso tra 1 e 16
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

6. Istruzioni di controllo del flusso del programma

L'**espressione** viene elaborata e, ricercando la posizione del primo BIT ad 1, si determina il numero d'ordine della label a cui saltare:

BIT 1 prima label
BIT 2 seconda label
...
BIT 16 sedicesima label

Il numero di **label** possibile è 16.

Se in **espressione** compaiono più BIT a 1 viene comunque selezionato quello di peso minore.

Se **espressione** è uguale a zero viene eseguito il blocco successivo.

Esempio:

```
RAM,16  
SELECT  
PROG  
SELECT=0000000000000100 B  
GOTP(SELECT) FAS1,FAS2,FAS3      l'esecuzione passa alla LABEL FAS3
```

6.6. LOOP

Il formato è il seguente:

EXEC = espressione

...

ENDE

dove **espressione** può essere:

- un numero esplicito intero compreso tra 1 e 256
- una variabile di tipo BYTE o WORD con valore compreso tra 1 e 256
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

Le istruzioni comprese tra **EXEC** e **ENDE** vengono eseguite tante volte quanto definito dalla espressione.

Sono possibili sino a 4 loop **EXEC** annidati.

Esempio:

```
I=0  
EXEC=(2*XTAB)      azzera la tabella TAB  
I=I+1  
TAB(I)=0  
ENDE
```

6.7. SUBROUTINE

La chiamata ad una subroutine si ottiene con l'istruzione **CALL** seguita dal nome della label di identificazione.

L'ultima istruzione della subroutine deve essere **RTS** per il ritorno.

Se l'istruzione **CALL** è preceduta da dalla istruzione **IF(...)** nella medesima espressione si ottiene una chiamata a subroutine condizionata.

Esempio:

```
IF(STROM) CALL GEFUM
```

Se le routine vengono scritte all'interno di una sezione di logica (veloce, lenta, superlenta) esse possono essere richiamate unicamente da quella sezione.

Scrivendo, invece, le routine nell'apposita sezione ROUTINE a fine programma è possibile effettuare chiamate alle stesse routine da più sezioni del programma.

E' possibile l'annidamento delle chiamate a subroutine fino a 8 livelli.

7. FUNZIONI PARTICOLARI

In questo capitolo vengono descritte alcune funzioni che, non fanno parte dei gruppi descritti precedentemente, ma che vengono utilizzate per il trattamento di segnali di macchina (staticizzazione, selezione tra vari segnali) per la ricerca di dati all'interno di variabili vettoriali, per la gestione dell'interfaccia operatore e, infine, per la gestione di comandi generati nella logica di macchina e inviati al CN.

7.1. FLIP FLOP

Questa funzione può essere eseguita su un qualunque segnale con una sintassi del tipo:

segnale Out = *FF(equazione di set),(equazione di reset)*

La variabile di uscita assumerà i seguenti valori in funzione dei dati in ingresso.

Equazione di Set	Equazione di Reset	Segnale in uscita	Note
0	0	x	l'uscita non cambia
0	1	0	
1	0	1	
1	1	0	prevale il reset sul set

Esempio:

REME = FF(OLTREC ~ TERMIC),(EMEA)

7.2. MULTIPLEXER

Consente di assegnare ad una variabile il valore selezionato tra le variabili o costanti indicate, per mezzo di BIT di abilitazione.

La sintassi è del tipo:

varout = *MUX (sel1, sel2 [, sel3][, sel4] [, ...]),(var1, var2 [, var3] [, var4] [, ...])*

7. Funzioni particolari

dove:

sel1, sel2, sel3, sel4, ... sono le variabili di selezione nel formato BIT o espressioni che diano un risultato vero o falso.

var1, var2, var3, var4, .. sono BYTE, WORD, LONG o DOUBLE con lo stesso formato di **varout**.

Nella funzione vengono scandite le variabili di selezione all'interno della prima parentesi, e la prima che in quel momento assume valore 1 determina il numero d'ordine della quantità, contenuta nella seconda parentesi, che darà il valore a **varout**.

La funzione può operare su di un massimo di 16 variabili.

Se nessuna variabile di selezione è attiva, il valore di **varout** resta invariato.

Esempio:

MULTI1=MUX(SELEZ1,SELEZ2,SELEZ3),(VARIA1,VARIA2,VARIA3)

7.3. RICERCA SU TABELLA

Questa funzione restituisce la posizione di un valore ricercato in una tabella (vettore); se il valore ricercato non esiste, l'esecuzione del programma passa alla label specificata.

Il formato della funzione è il seguente :

posizione = RIC (tabella, primo indice, ultimo indice, valore da cercare) label

dove:

- **posizione** è la posizione della tabella in cui viene trovato il valore da cercare
- **tabella** è il nome della tabella contenente il valore da cercare
- **primo** e **ultimo indice** indicano l'intervallo di ricerca. Per considerare la tabella intera, **primo indice** sarà = 1 e **ultimo indice** = dimensione della tabella
- **valore da ricercare** è l'oggetto della ricerca
- **label** è la destinazione del salto dove prosegue il programma , se il valore da cercare non viene trovato.

posizione può essere una variabile in formato BYTE o WORD

tabella può essere un vettore di BYTE o WORD

primo indice, ultimo indice e valore da cercare possono essere:

- un numero esplicito intero compreso tra 1 e 32767.
- una variabile di tipo BYTE o WORD con valore compreso tra 1 e 32767.
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

Le tabelle (vettori) create nel PLC sono visualizzabili e modificabili a livello di utente finale con le modalità descritte nel Capitolo 2.6 *Visualizzazione e modifica delle tabelle del PLC*.

Esempio:

POMAG=RIC(TABUT,1,25,NEWTOL*2)ERRCU

cerca nuovo utensile in tabella TABUT

7.4. MESSAGGI PER L'OPERATORE

Sul video è previsto un quadro per la visualizzazione di 16 righe di messaggi, di 62 caratteri ciascuna, accessibile premendo la softkey **MESSAGGI DI LOGICA**.

La visualizzazione di un messaggio si ottiene con il comando **DISPL** seguito dal numero di riga e dal messaggio voluto.

Il messaggio rimane visualizzato fino a quando non viene emesso il comando di cancellazione **CLR** oppure una nuova visualizzazione sulla stessa riga.

Per richiamare il messaggio nel corso della logica si deve utilizzare il comando **DISPL** con la seguente sintassi:

DISPL, riga, messaggio

dove:

riga può essere:

- un numero esplicito intero compreso tra 1 e 16, oppure 0 per effettuare la visualizzazione in un'area riservata nella zona principale del video CN
- una variabile di tipo BYTE o WORD con valore compreso tra 0 e 16
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

messaggio può essere:

- una sequenza di caratteri delimitata da apici
- una variabile di tipo stringa
- una espressione che restituisca un risultato di tipo stringa.

Il **messaggio** può, quindi, essere ottenuto sommando più messaggi già definiti, testi in chiaro e stringhe ottenute per esempio con la funzione **MKN\$(...)** su variabili numeriche (o altre).

Esempio:

```
STR
MESDI
MESSAG
...
PROG
MESSAG = 'ARRESTO CICLO PER ANOMALIA ' + MESDI + MKN$(NUM)
DISPL,1,MESSAG
```

Nell'esempio si definisce il testo del messaggio con la prima espressione e lo si manda sulla riga 1 del video con la seconda.

7. Funzioni particolari

Se MESDI='AZIONAMENTO ASSE N.' e NUM=2 sulla riga 1 del video appare la frase:

ARRESTO CICLO PER ANOMALIA AZIONAMENTO ASSE N. 2

Se poi nel programma si cambia il testo ASCII contenuto in MESDI oppure si utilizza una struttura a vettore MESDI(n), in funzione delle variabili correnti si potrebbe con la stessa istruzione visualizzare:

ARRESTO CICLO PER ANOMALIA POMPA N. 1

ARRESTO CICLO PER ANOMALIA PALLET IN POSIZIONE 4

La cancellazione di una riga si ottiene con il comando:

CLR, riga

dove **riga** può essere:

- un numero esplicito intero compreso tra 0 e 16
- una variabile di tipo BYTE o WORD con valore compreso tra 0 e 16
- una espressione che restituisca un valore nel formato BYTE o WORD con le stesse limitazioni numeriche dei casi precedenti.

Spesso la definizione del testo di un messaggio come MESDI dell'esempio precedente è costante perciò è consigliabile definirlo nella sezione di inizializzazione; in alternativa si può utilizzare l'istruzione **DISPL** in modo immediato.

Esempio:

DISPL, 1 'ANOMALIA PRESSOSTATO LUBRIFICAZIONE'

Nota:

Nella terza parte di questo manuale (*Esempi di programmazione PLC*), è descritto un programma (*SCROLLIN - Gestione fino a 128 messaggi con scrolling su video*) che compatta automaticamente sulle 15 righe disponibili molti messaggi, presentando su video, con scorrimento di tutti i messaggi, quello richiesto in quel momento.

7.5. COMANDO DI PROGRAMMI DA LOGICA DI MACCHINA

Talvolta l'esecuzione di determinate funzioni ausiliarie quali, cambio utensile, cambio pezzo, implica sequenze molto complesse o movimentazione di assi e quindi risulta vantaggioso, se non indispensabile, ricorrere al lancio di un programma scritto nel linguaggio del CN.

La logica di macchina può attivare l'esecuzione dei comandi desiderati richiamando un programma CN presente nell'area di "EDIT COM" (vedere il *Manuale di Configurazione Sistema*) con una istruzione del tipo:

COM, 1, nome programma

dove:

nome programma può essere:

- una sequenza di caratteri delimitata tra apici
- una variabile di tipo stringa
- una espressione che restituisca un risultato di tipo stringa.

S1200

A differenza del sistema S1200 non è possibile lanciare direttamente un sottoprogramma contenuto all'interno di un programma ed individuato tramite una label; per contro i programmi lanciati con le funzioni **COM**, possono contenere qualunque blocco eseguibile di CN, compresi salti, cicli di misura e la geometria avanzata PROGET2.

S1200

Non esiste più lo sfalsamento di una posizione dei parametri **P1...P99** come su S1200, dove $P0 = P(1)$

I programmi richiamati con le funzioni **COM** possono utilizzare parametri specifici **P1...P99** indipendenti dagli omonimi del part program e direttamente accessibili dal PLC scrivendo sulle variabili del vettore **P** sugli elementi da **P(1)** a **P(99)**.

Sul lancio di una **COM** vengono automaticamente resettate le funzioni che alterano il sistema di coordinate (spostamenti origine, specularità, rototraslazioni, cicli fissi, ...).

I valori di FEED o SPEED si possono trasferire sui parametri **P()** per salvarli (esempio: $P1 = F$) e quindi ripristinarli con le istruzioni inverse ($F = P1$).

Particolare cura va posta nell'attivare l'istruzione **COM** per lanciare l'esecuzione di un dato programma una sola volta, altrimenti si incorre nell'errore di troppi sottoprogrammi annidati.

7.5.1. COMANDO DI PROGRAMMI DURANTE L'ESECUZIONE IN AUTOMATICO

Le istruzioni **COM** da eseguire durante l'esecuzione in automatico di un programma (sincrone) devono essere attivate esclusivamente in seguito a funzioni **T** o **M** di fine blocco (vedere *Parte II - Lista riassuntiva dei segnali e registri predefiniti*) in modo sincrono, cioè:

- prima che il segnale **BURDY** venga resettato
- in alternativa con **DHOLD** alto, se disattivato prima di resettare il segnale **BURDY**.

Vedere il paragrafo *Acquisizione informazioni sincrone da PLC a CN* nella *Parte II* di questo manuale dedicato all'interfaccia di sistema.

Il programma lanciato da una funzione ausiliaria può contenere funzioni che richiamano altri programmi (purché non richiamino se stesso) fino ad 8 livelli di annidamento.

Quando tutti i programmi lanciati con le **COM** sono terminati, cioè appena prima che l'esecuzione ritorni al blocco successivo a quello che ha richiamato la prima **COM**, viene settato dal CN lo strobe sincrone **STCOM** (valido solo con il segnale **BURDY** alto).

Questo strobe ha, dal punto di vista del sincronismo, le stesse caratteristiche di una funzione **M** di fine blocco o **T**, cioè, dà la possibilità di eseguire altre istruzioni COM sincrone con lo stesso metodo descritto sopra.

7.5.2. COMANDO DI PROGRAMMI A PARTIRE DALLO STATO DI MANUALE

E' possibile mandare in esecuzione un programma **COM** anche a partire dallo stato di manuale, cioè con NCMD = 5 (modo asincrono)

La sintassi è la stessa descritta ad inizio capitolo, ma naturalmente non vi sono vincoli di sincronismo con il segnale **BURDY**.

Lo strobe **STCOM** non è attivato al termine di questo tipo di **COM**.

All'interno di una **COM** "asincrona" è possibile inserire una funzione che richiami una **COM** sincrona con tutte le regole descritte nel paragrafo precedente.

Il lancio di questo tipo di **COM** deve naturalmente essere effettuato ad assi fermi. Per testare questa condizione si può costruire semplicemente un segnale di assi fermi con un'equazione del tipo:

bit ASSI FERMI = (("INTOL&MOVCN)=0)

Esempio:

Quella che segue è una chiamata del programma PALLETS da logica di macchina con una istruzione **COM** attivata in seguito alla funzione **M21** con **BURDY** alto, quindi in modo sincrono:

```
...
IF("BURDY) ASINC
DHOLD=1; FHOLD=1
IF(STROM) CALL GEFUM
BURDY=0
ASINC:$
...
END
...
GEFUM:$
IF (AUXM=21) COM, 1, 'PALLETS'; RTS
RTS
```

7.5.3. COMANDO DI PROGRAMMI DURANTE L'ESECUZIONE IN MODO SEMIAUTOMATICO

I sottoprogrammi di partprogram COM lanciati dal PLC sono eseguiti in modalità automatica (non c'è attesa di «via ciclo» tra un blocco e l'altro) anche se il CN è in esecuzione di un programma di lavorazione in semiautomatico.

La variabile **NCMD** resta comunque congruente con lo stato di esecuzione CN.

Le seguenti funzioni modali di programmazione pezzo permettono di variare tale comportamento:

- **G1011:** forza l'esecuzione dei sottoprogrammi COM in semiautomatico quando lo stato del CN è semiautomatico (da utilizzare in fase di verifica o messa in servizio).
- **G1010:** disabilita il funzionamento attivato con G1011 (ripristina la condizione di default).

COMANDO DI PROGRAMMI DA LOGICA DI MACCHINA: UNITÀ DI MISURA

I blocchi di movimento eseguiti all'interno dei sottoprogrammi COM lanciati dal PLC, sono sempre interpretati in millimetri anche se il CN è stato impostato per lavorare con il sistema di misura in pollici. Terminata l'esecuzione della COM, viene ripristinata la modalità attiva prima del lancio del sottoprogramma (pollici o millimetri).

COMANDO DI PROGRAMMI DA LOGICA DI MACCHINA: FUNZIONI NON AMMESSE

Il lancio di un sottoprogramma COM segnala errore "**E48**: mancanza di funzioni apertura/chiusura" qualora siano attive alcune funzioni che alterano il sistema di coordinate (**G846**, **G851**, **G68**, **G69**, cicli fissi, supercicli fissi, **G751**, **G16**, **G748**, **G749**).

Alcune altre funzioni (**G52**, **G51**, **G54**, **G55**, **G56**, **G57**, **G58**, **G59**, **G61**, **G76**) vengono invece disabilitate temporaneamente dal lancio della COM e ripristinate al suo termine.

COMANDO DI PROGRAMMI DA LOGICA DI MACCHINA: LANCIO IN MODO ASINCRONO

Il PLC può richiedere anche in modo asincrono rispetto al programma in esecuzione il lancio di un sottoprogramma COM.

Questa prestazione può, ad esempio, essere utilizzata per gestire una sequenza di cambio utensile in caso di vita utensile scaduta.

Nel momento in cui il PLC vuole lanciare la COM asincrona, deve settare il bit **RCOM**; in seguito a tale richiesta, il CN esaurisce i blocchi di programma precalcolati (max 256), quindi setta in modo sincrono (accompagnato dal segnale **BURDY**) lo strobe **STRCOM**.

Il PLC deve decodificare tale strobe e quindi eseguire l'istruzione COM, ...che in questo modo viene sincronizzata con il programma principale.

RCOM viene resettato immediatamente quando acquisito dal CN.

Nel caso di gruppi d'asse esistono i byte **RCOM_** e **STRCO_** dove ogni bit corrisponde ad un gruppo d'assi.

Nome	Dim	Direzione	Sincrono	Descrizione
RCOM	1	PLC \Rightarrow CN	no	Richiesta attivazione di una COM asincrona.
STRCOM	1	CN \Rightarrow PLC	si	Strobe di sincronizzazione per il lancio della COM richiesta con RCOM.
RCOM_	8	PLC \Rightarrow CN	no	Richieste di attivazione COM asincrone per i singoli gruppi d'assi (1..8).
STRCO_	8	CN \Rightarrow PLC	si	Strobe di sincronizzazione per il lancio della COM richieste con RCOM_ per i singoli gruppi d'assi (1..8).

PARTE II

INTERFACCIA DI SISTEMA

PREMESSA

Le informazioni contenute in questa sezione sono relative alle variabili ed ai segnali di interscambio tra la sezione PLC e la sezione CN dei controlli S4000. Questo vale per i moduli seguenti:

- Modulo standard, che si occupa della gestione dei movimenti e delle varie modalità di funzionamento e visualizzazione
- Moduli interni dedicati che sono:
 - Modulo per la gestione dei mandrini
 - Modulo per la movimentazione degli assi indipendenti
 - Modulo per la gestione del cambio utensile

La descrizione delle informazioni su menzionate è organizzata come segue.

A fronte di una determinata funzionalità, sia essa del modulo standard che dei moduli dedicati, vengono descritti i vari registri, variabili e segnali e la loro interazione. Al fondo della descrizione è riportata una tabella che riassume i segnali descritti con le loro peculiari caratteristiche (vedere più avanti). Dette tabelle vengono poi raggruppate nel Capitolo 4 di questa sezione per essere utilizzate come prontuario di consultazione durante lo sviluppo dell'applicazione.

Nella tabella, per ognuno dei registri, variabili o segnali, raggruppati per argomento, è specificato:

- il nome mnemonico
- il formato (nella colonna **Dim**)
 - 1 = bit
 - 8 = byte
 - 16 = word
 - 32 = floating point
 - 64 = double floating point
 - STR = stringa di caratteri
- i vincoli di sincronismo con il segnale **BURDY** (nella colonna **Scr**)
- la direzione dell'informazione: da PLC a CN, viceversa o in entrambe le direzioni (nella colonna **Direzione**).

Nota: La scrittura su variabili di sola lettura per il PLC, cioè con la direzione da CN a PLC e non viceversa, può avere conseguenze imprevedibili a priori.

- una breve **Descrizione** nella colonna relativa.

Le **unità di misura** utilizzate sono:

- | | |
|---|------------------------|
| - per quote, distanze, compensazioni di misura: | mm |
| - per dimensioni rotative | gradi |
| - per tempi | msec, sec o min |
| - per velocità: | mm/min |
| - per accelerazioni: | mm/(sec ²) |
| - per velocità mandrino | giri/min |
| - per tensioni | V |

Premessa

La **simbologia** usata è la seguente:

I caratteri **()** dopo il nome di un registro indicano che si tratta di un vettore di più elementi nel formato specificato (esempio **UTNUM()** indica che si tratta di un vettore, mentre **MOVCN** è un singolo registro).

Se, all'interno delle liste che seguono, compare il simbolo **(1..n)** si intende che il registro o il vettore devono essere interpretati analizzandone singolarmente gli elementi da (1 a n). Per discriminare un singolo registro di cui si descrivono i bit occorre tener conto che:

- Gli elementi di un vettore hanno dimensione superiore ad 1.
- Quando sono descritti i singoli bit di un registro, solitamente dette descrizioni sono precedute dalla descrizione del registro stesso, che sarà indicato senza parentesi.

Esempio:

Nome	Dim	Direzione	Scr	Descrizione
MOVCN	8	CN ⇨	PLC no	Richieste di abilitazione al movimento degli assi (1..8).
MOVCN(1)	1	CN ⇨	PLC no	(primo bit del byte) è la richiesta di abilitazione dell'asse 1
MOVCN(8)	1	CN ⇨	PLC no	(ottavo bit del byte) è la richiesta di abilitazione dell'asse 8
UTNUM()	16	CN ⇔	PLC no	Codice degli utensili in tabella (1 ... UTENRI), dove UTENRI rappresenta il numero delle righe della tabella utensili.
UTNUM(1)	16	CN ⇔	PLC no	(primo elemento del vettore word) è il codice dell'utensile presente nella riga 1 della tabella utensili.
UTNUM(8)	16	CN ⇔	PLC no	(ottavo elemento del vettore word) è il codice dell'utensile presente nella riga 8 della tabella utensili.

Nota: Le intestazioni delle colonne non sono riportate nelle tabelle del testo per non appesantirne la leggibilità. Si prega quindi di prendere nota in questa sede della posizione delle informazioni contenute nelle singole colonne.

1. GESTIONE MOVIMENTI E SEGNALI D'INTERSCAMBIO

1.1. STATI DEL CN

Il Controllo Numerico segnala al PLC il proprio stato e le relative variazioni per mezzo di due segnali **NCMD** per lo stato del funzionamento e **STBMD** come strobe di segnalazione del cambiamento di stato.

NCMD può assumere i seguenti valori:

- 1 rilevatore di quote
- 2 blocco singolo
- 3 esecuzione programma in semiautomatico
- 4 esecuzione programma in automatico
- 5 manuale
- 8 ritorno sul profilo
- 9 manuali attivi in stato di hold

Assegnando il valore 3 al registro **FNCMD** si forza il CN in stato di esecuzione programma in semiautomatico (**NCMD=3**). In condizioni normali il valore di **FNCMD** va mantenuto a zero e 3 è l'unico valore diverso da zero assegnabile.

Riassunto registri e segnali interessati

NCMD	8	CN \Rightarrow PLC	no	Codice di stato del funzionamento del CN: 1 = rilevatore di quote 2 = blocco singolo 3 = esecuzione programma in semiautomatico 4 = esecuzione programma in automatico 5 = manuale 8 = ritorno sul profilo 9 = manuali attivi in stato di hold.
STBMD	1	CN \Rightarrow PLC	no	Strobe impulsivo di segnalazione della variazione dello stato del CN; ha durata pari ad una scansione completa di logica lenta.
FNCMD	8	CN \Leftarrow PLC	no	Registro di forzamento CN in stato di esecuzione programma in semiautomatico.

1. Gestione movimenti e segnali d'interscambio

1.2. TRASMISSIONE FUNZIONI AUSILIARIE SINCRONE E PREPARATORIE

La presenza nei blocchi di programma di una funzione ausiliaria **M, S, T, H** eseguita singolarmente (in stato di blocco singolo) o all'interno di un programma (in stato di automatico o semiautomatico), viene segnalata al PLC tramite registri e segnali di colloquio *sincronizzati con i blocchi stessi*, che per brevità saranno detti sincroni.

Il segnale sincrono principale è **BURDY** (Buffer ReaDY); esso viene settato dal CN per segnalare al PLC la presenza di una nuova funzione ausiliaria.

Il codice della nuova funzione è contenuto nei registri **AUXM, SPEED, TOOL, AUXH**.

In aggiunta, per ottimizzare il colloquio, viene settato dal CN uno strobe (segnale), per individuare quale tipo di funzione è presente; si avrà quindi rispettivamente **STROM, STROS, STROT, STROH**.

⇒ Il programma PLC dovrà decodificare con questi segnali le nuove informazioni e, quindi, resettare immediatamente il segnale **BURDY** affinché il CN possa proseguire nelle sua attività. **BURDY** deve essere usato esclusivamente per la decodifica delle funzioni ausiliarie e non per arrestare l'avanzamento dei blocchi, per il quale esistono altri segnali dedicati.

Gli strobe sono segnali aggiornati dal CN unicamente nel momento in cui **BURDY** viene settato; non hanno perciò una durata fissa, non devono essere resettati dal PLC e sono significativi solamente quando il segnale **BURDY** è attivo.

La decodifica delle funzioni ausiliarie sarà solitamente gestita nella SEZIONE LENTA del PLC.

Poiché le funzioni ausiliarie possono essere emesse ad inizio e fine blocco del programma (vedere la tabella al fondo del manuale) occorre prestare particolare attenzione all'ordine con cui sono decodificati i vari strobe.

Le funzioni preparatorie **G** ed **F**, disponibili sui registri **AUXG** e **FEED**, non sono, invece, trasmesse con il segnale **BURDY** e sono, quindi, completamente asincrone rispetto all'esecuzione dei blocchi.

E' inoltre disponibile un ulteriore registro **CICFI** che contiene il codice del ciclo fisso in esecuzione.

Una qualunque funzione ausiliaria **M, H**, può essere predisposta in modo da agire selettivamente solo sugli assi specificati.

Il formato di programmazione da decodificare sarà per esempio M11XYZ; in questi casi viene comunicato sul registro **AXPGM** quali assi sono presenti nel blocco; nell'esempio precedente **AXPGM** sarà uguale a 00000111B. Non attivano questa prestazione gli assi per cui è richiesto un movimento nel blocco. (M11X100R comporta **AXPGM**= 00000000B).

Esempio di decodifica delle nuove informazioni e gestione del segnale BURDY:

...
PROG

...
END

[sezione lenta

IF("BURDY) ASINC

DHOLD=1; FHOLD=1

IF(STROT) CALL GEFUT

Se non è presente BURDY, salta alla parte asincrona

Arresto temporaneo

Gestione funzioni T

1. Gestione movimenti e segnali d'interscambio

IF(STROS) CALL GEFUS	<i>Gestione funzioni S</i>
IF(STROH) CALL GEFUH	<i>Gestione funzioni H</i>
IF(STROM) CALL GEFUM	<i>Gestione funzioni M</i>
IF(STCOM) ...	<i>Tutte le COM terminate</i>
BURDY=0	<i>Nuove funzioni acquisite</i>
ASINC:\$	<i>Operazioni relative al salto</i>
...	
DHOLD=...	<i>Conferma dell'arresto blocchi o rilascio</i>
FHOLD=...	
END	
...	
END	
[sezione routines	
GEFUM:\$	
IF(AUXM=3) ...; RTS	
IF(AUXM =11) M11	
...	
RTS	
M11:\$	
IF(AXPGM=0) SSA=00000111B; RTS; ELSE SSA=AXPGM; RTS	<i>Gestione M11</i>
[...	

1.2.1. ACQUISIZIONE INFORMAZIONI SINCRONE DA PLC A CN

Con il segnale **BURDY** a livello 1, settato da blocchi contenenti funzioni di fine movimento, sono inoltre acquisite tutte le informazioni *sincrone da PLC a CN* relative al comando di sottoprogrammi CN da logica, attivazione correttori utensili, ecc., indicate nella lista riassuntiva delle variabili predefinite (**INTOF**, istruzioni **COM** - vedere paragrafi relativi).

Le stesse informazioni vengono acquisite anche con il segnale **DHOLD** attivo quando esso viene settato prima di resettare **BURDY** in seguito ad una funzione **M** di fine blocco (o **M** di inizio blocco se programmata da sola).

1.2.2. SEGNALAZIONE TERMINE DI SOTTOPROGRAMMA COM

In modo sincrono viene segnalato dal CN il termine dei sottoprogrammi lanciati dal PLC (COM) tramite lo strobe **STCOM**. Questo segnale ha lo stesso comportamento degli strobe **STROM**, **STROH**, ma in aggiunta, quando settato, attiva l'acquisizione sincrona di ulteriori lanci di sottoprogrammi, come descritto nel paragrafo precedente.

E' importante ricordare che:

- nel caso di più sottoprogrammi annidati (un sottoprogramma contenente una funzione che a sua volta lancia un ulteriore sottoprogramma), **STCOM** viene emesso solo quando il sottoprogramma principale viene terminato
- nel caso di sottoprogrammi lanciati con il CN in stato manuale, **STCOM** non viene emesso

1. Gestione movimenti e segnali d'interscambio

1.2.3. TRASMISSIONE PARAMETRI SUPPLEMENTARI I, J, K, Q

I parametri **I, J, K, Q** programmati insieme alle funzioni ausiliarie M, H sono comunicati al PLC, ad inizio blocco, sull'array **AUXVAL** accompagnati dagli strobe **STRAUX** con i seguenti indici. Essi possono essere usati per esempio per definire sintassi del tipo M19 Q12.2.

AUXVAL(1) = parametro I	con lo strobe STRAUX(1)
AUXVAL(2) = parametro J	con lo strobe STRAUX(2)
AUXVAL(3) = parametro K	con lo strobe STRAUX(3)
AUXVAL(4) = parametro Q	con lo strobe STRAUX(4)

1.2.4. EMISSIONE DI FUNZIONI AUSILIARIE "AL VOLO"

Le funzioni ausiliarie di inizio blocco (vedere tabella in fondo a questo manuale) possono essere emesse "al volo" cioè durante un blocco di movimento in continuo senza alcuna decelerazione se programmate nel blocco di movimento stesso.

Esempio:

N1 X100F1000	
N2 X200F2000M7	<i>M7 viene emessa alla quota 200 con l'asse X alla velocità di 1000 mm/min.</i>
N3 X300	
N4 X400M9	<i>M9 viene emessa a X400 con l'asse X fermo</i>
N5 X450	

Riassunto registri e segnali interessati

BURDY	1	CN	↔	PLC	si	Segnalazione della presenza di nuove informazioni sincrone per la logica di macchina in arrivo dal CN.
AUXM	16	CN	⇒	PLC	si	Codice funzione ausiliaria M ultima programmata (M0-M9999).
STROM	1	CN	⇒	PLC	si	Strobe presenza funzione M.
TOOL	16	CN	⇒	PLC	si	Codice funzione T ultima programmata (T0-T32767).
STROT	1	CN	⇒	PLC	si	Strobe presenza funzione T.
AUXH	16	CN	⇒	PLC	si	Codice funzione ausiliaria H ultima programmata (H0-H9999).
STROH	1	CN	⇒	PLC	si	Strobe presenza funzione H.
SPEED	32	CN	⇒	PLC	si	Codice funzione ausiliaria S ultima programmata (S0-S99999).
STROS	1	CN	⇒	PLC	si	Strobe presenza funzione S.
STCOM	1	CN	⇒	PLC	si	Strobe di segnalazione del termine esecuzione su sottoprogrammi COM.
FEED	64	CN	⇒	PLC	no	Ultima Feed programmata.
AUXG	16	CN	⇒	PLC	no	Codice funzione ausiliaria G ultima programmata (G0-G9999).
CICFI	16	CN	⇒	PLC	no	Codice del ciclo fisso in esecuzione.
AXPGM	8	CN	⇒	PLC	si	Assi i cui nomi sono programmati nel blocco insieme alla funzione ausiliaria (es. M11XYZ genera AXPGM=00000111B).
AUXVAL()	64	CN	⇒	PLC	si	Array su cui vengono trasmessi i parametri I, J, K, Q programmati insieme alle funzioni ausiliarie M, H.
						AUXVAL(1) = parametro I
						AUXVAL(2) = parametro J
						AUXVAL(3) = parametro K
						AUXVAL(4) = parametro Q

STRAUX 8 CN \Rightarrow PLC si Strobe dei parametri I, J, K, Q.
STRAUX(1) = strobe di I
STRAUX(2) = strobe di J
STRAUX(3) = strobe di K
STRAUX(4) = strobe di Q

TRASMISSIONE FUNZIONI AUSILIARIE: NOTE SULL'EMISSIONE DELLA SPEED

Al termine dell'esecuzione simulata del programma (pressione della softkey **FINE RICERCA**), in seguito ad una sequenza di RIPRESA CICLO o RICERCA MEMORIZZATA, viene emesso automaticamente verso il PLC un blocco contenente l'ultima S incontrata in simulazione.


1.3. COMANDI ASINCRONI DI START, ARRESTI, ALLARMI, CONSENSI EMERGENZA CN SU INTERRUZIONE COMUNICAZIONE CON SCHEDA PC

Questo insieme di segnali permette al PLC di arrestare temporaneamente o definitivamente l'attività in corso sul CN *senza, peraltro, influenzare i mandrini, gli assi indipendenti comandati dal PLC ed il cambio utensile*.


Con questi segnali attivati sono inibite le transizioni di stato del CN (es. da manuale a Blocco Singolo).


FHOLD (Feed Hold): è il segnale che permette di sospendere temporaneamente i movimenti in corso facendo decelerare gli assi con l'accelerazione corrente. Quando viene rilasciato, l'esecuzione riprende senza ulteriori comandi.

DHOLD (Data Hold): settando questo segnale il PLC può temporaneamente bloccare l'avanzamento dei blocchi successivi di programma i quali presuppongono un arresto temporaneo dei movimenti con velocità finale uguale a zero, senza interrompere l'attività in corso. In altri termini, il programma procede nell'esecuzione della lavorazione fino a quando non incontra un blocco con movimento zero. Quando questo segnale viene rilasciato, il programma riprende l'esecuzione senza ulteriori comandi.
 E' molto importante ricordare che un profilo di interpolazione continua o una serie di movimenti senza interposizione di funzioni ausiliarie, è un blocco unico da questo punto di vista.


RHOLD (Richiesta di Hold): è il segnale che duplica il tasto rosso di  sulla tastiera del CN. Sospende temporaneamente l'eventuale movimento in automatico, con l'accelerazione corrente, ed arresta il programma in corso, mentre non agisce sui movimenti in manuale.


In risposta, quando gli assi si sono arrestati, viene attivato dal CN il segnale **HOLDA** (Hold Acquisito) per segnalare la presenza dello stato di HOLD.


Anche se viene rilasciata la richiesta di HOLD, l'esecuzione non riprende finché non si dà l'esplicito comando di VIA ciclo con il segnale **CYST** o premendo il tasto .


CYST (Cycle Start): è il segnale PLC che duplica il tasto verde di  sulla tastiera del CN per fornire un comando di START ciclo.

1. Gestione movimenti e segnali d'interscambio

SFKGRD (Tasto guardia): questa variabile è settata (in binario 11111111B, in esadecimale 0FFH) con la pressione del tasto “ guardia ”  (a sinistra del tasto barra spaziatrice) e viene resettata (00000000B, 00H) sul rilascio del tasto.

SFKCNS(1) Segnale di tipo impulsivo che registra la pressione del tasto verde di  sulla tastiera del CN.

SFKCNS(2) Segnale di tipo impulsivo che registra la pressione del tasto rosso di  sulla tastiera del CN.

SFKCNS(3) Segnale di tipo impulsivo che registra la pressione del tasto giallo di  sulla tastiera del CN.

CYON (Cycle On): è il segnale che fornisce il CN al PLC per informarlo che è in corso l'esecuzione di un blocco.

REME (Richiesta di emergenza): è il segnale per la richiesta di emergenza esterna da parte del PLC a cui il CN risponde settando il segnale **EMEA** (Emergenza Acquisita) per indicare la presenza dello stato di emergenza.
In questo stato vengono disabilitati istantaneamente gli assi controllati e forzati i riferimenti di velocità a 0 Volt. Ogni attività di programma o di movimento in corso, viene cancellata ed il CN passa in stato di rilevatore di quote (**NCMD=1**), visualizzando sul video il messaggio:

M.U. spenta causa emergenza.

Ogni ulteriore comando di esecuzione viene rifiutato.


Il segnale **EMEA** nasce anche in seguito ad allarmi interni al CN ed allarmi relativi ai trasduttori e agli asservimenti.


Per uscire da questo stato occorre rimuovere la causa che lo ha generato e dare un comando di **BREAK**.

RBRK (Richiesta di Break): è il segnale PLC che duplica il tasto giallo di  sulla tastiera CN.

Questo comando, settato dal PLC e resettato dal CN quando acquisito, cancella ogni attività CN in corso. Dopo aver fatto decelerare gli assi con l'accelerazione corrente, forza lo stato di Manuale (**NCMD=5**); non agisce sui movimenti in manuale.



A condizione di averne rimosso le richieste relative, **RBRK** cancella **EMEA** (stato di emergenza) e **HOLDA** (stato di HOLD).

BRKA (Break Acquisito): è il segnale di tipo impulsivo, con durata pari alla scansione completa della logica lenta, che trasmette un ordine di BREAK (reset) derivante dalla pressione del tasto  sulla tastiera o come risposta alla richiesta **RBRK**, affinché il PLC possa cancellare le proprie attività (per esempio arrestare il mandrino).

 **S1200** Mentre nel sistema S1200 un Break generava la funzione M30 (fine programma) e M30 generava un Break, questo non accade più nel sistema S4045.

1. Gestione movimenti e segnali d'interscambio
Riassunto registri e segnali interessati

DHOLD	1	CN ⇌ PLC	no	Arresto temporaneo dell'avanzamento del programma a partire dal primo blocco successivo che contiene un punto di arresto nel movimento in continuo (tipicamente una funzione ausiliaria), senza interruzione dell'attività in corso.
FHOLD	1	CN ⇌ PLC	no	Arresto temporaneo del movimento in esecuzione.
RHOLD	1	CN ⇌ PLC	no	Richiesta di HOLD esterna. Arresto temporaneo dei movimenti programmati e dei blocchi in esecuzione.
HOLDA	1	CN ⇌ PLC	no	Assi fermi in stato di Hold.
CYST	1	CN ⇌ PLC	no	Richiesta di START CICLO esterna.
SFKGRD	8	CN ⇌ PLC	no	Tasto guardia
SFKCNS	8	CN ⇌ PLC	no	Segnalazioni impulsive pressione tasti START CICLO (SFKCNS(1)), HOLD (SFKCNS(2)), BREAK (SFKCNS(3)).
CYON	1	CN ⇌ PLC	no	Blocco in esecuzione.
REME	1	CN ⇌ PLC	no	Richiesta di EMERGENZA esterna.
EMEA	1	CN ⇌ PLC	no	CN in stato di emergenza per allarmi interni o richiesta di emergenza esterna.
RBRK	1	CN ⇌ PLC	no	Richiesta di BREAK esterna. Interruzione del programma o blocco in corso. Uscita dallo stato di emergenza.
BRKA	1	CN ⇌ PLC	no	Ordine impulsivo di BREAK per il PLC.

SEGNALE	AZIONI SUL CN				
	Abort del programma	Arresto blocco successivo	Arresto movimento programmato	arresto movimento manuale	Arresto movimento manuale forzato
DHOLD		sì (su quello successivo che prevede un arresto movimento)			
FHOLD			sì	sì	
RHOLD o tasto 		sì	sì	no!	
RBRK o tasto 	sì	sì	sì		
REME	sì	sì	sì	sì	sì

Note relative alla visualizzazione dello stato dei segnali di arresto

- Per i segnali **FHOLD**, **DHOLD**, **HOLDA** esistono delle variabili di condizionamento da utilizzare nelle tabelle di configurazione del video per dare la possibilità di segnalare all'utente lo stato dei segnali descritti sopra in modo immediato (vedere la descrizione relativa nel *Manuale di Configurazione Sistema*).
- Nelle tabelle video fornite come default insieme al CN è implementata la seguente gestione:

FHOLD = 1 oppure **DHOLD** = 1 oppure **RDMOV** diverso **MOVCN** oppure **M6PGM** = 1 fanno lampeggiare la scritta in reverse **MAPR** (macchina pronta); sul quadro video di AREA MANUTENZIONE, oltre alla scritta citata, compaiono le scritte in reverse separate per ogni variabile.

HOLDA = 1 fa comparire la scritta in reverse **HOLD**.

EMEA = 1 fa comparire la softkey di **STATO R.Q.** nel menu principale ed elimina le altre di movimentazione.

1. Gestione movimenti e segnali d'interscambio

Nei casi di interruzione della comunicazione o di tempi troppo lunghi nel colloquio tra scheda PC e scheda MASTER il CN passa in stato di emergenza e sul video compare il messaggio: "**E32102**: M.U. spenta per interruzione della comunicazione con PC".

La segnalazione di questo allarme implica la segnalazione dello stato di emergenza (EMEA=1) con la conseguente disabilitazione degli assi e l'interruzione del programma.
Se non ci si trova in condizioni di guasto, l'allarme può essere cancellato con un comando di BREAK.

1.4. ORIGINI E CORRETTORI UTENSILI

Le azioni da intraprendere per attivare correttori utensili ed origini dipendono dalla scelta del tipo di cambio utensile fatta nella configurazione del CN. I dettagli sono riportati nel Capitolo 2.3. *Modulo per la gestione del cambio utensile*.

1.4.1. CAMBIO UTENSILE MANUALE

Non è necessaria alcuna gestione per richiamare i correttori utensili; essi vanno programmati con le funzioni **T**. Automaticamente è generata una "attesa di start" (con messaggio in chiaro per l'operatore); le origini sono attivate separatamente con le funzioni **O**.
Il codice **00** consente di passare in origine assoluta; **0-1** ripristina l'ultima origine presente prima di passare in origine assoluta.

La funzione **T0** annulla la correzione lunghezza attiva.

1.4.2. CAMBIO UTENSILE MANUALE TIPO S1200

I numeri da T0 a T9 selezionano una delle 10 diverse origini sul piano.

I numeri da T10 a T98 selezionano una delle 89 compensazioni della lunghezza utensile. Il numero T99 richiama, per tutti gli assi, le origini fisse ed assolute dei trasduttori. Esso serve per programmare spostamenti riferiti allo zero fisso della macchina e cioè indipendenti dallo zero pezzo.

Esempi

T1 *richiama l'origine 1 sul piano*
T23 *richiama la compensazione lunghezza utensile n. 23*

1.4.3. CAMBIO UTENSILE AUTOMATICO

Il codice **T** programmato viene unicamente passato al PLC sul registro **TOOL** con lo strobe **STROT**.

Il codice del correttore utensile è caricato nel registro **OFST** e quindi attivato dallo strobe sincrono **INTOF** (vedere il capitolo 2.3. *Modulo per la gestione del cambio utensile*).

Le origini sono attivate separatamente dalle funzioni **O**.

Il codice **00** consente di passare in origine assoluta; **0-1** ripristina l'ultima origine presente prima di passare in origine assoluta.

L'attivazione di **OFST=0** annulla la correzione lunghezza attiva.

In casi particolari, il PLC può attivare un'origine settando lo strobe sincrono **INORG** dopo aver caricato il codice dell'origine su **ORIG**.

Quando si ha la necessità di attivare l'origine assoluta, in alternativa a **00**, si può settare il segnale sincrono di bypass origini **BYORG**; si resta in questa situazione finché il bypass non viene resettato (in modo sincrono).

Il CN informa il PLC dello stato di origine assoluta presente con il segnale **ABSOR**.

Sia **INTOF** che **INORG** sono resettati dal CN quando acquisiti.

Essendo in origine assoluta è anche possibile attivare un correttore lunghezza, in questo caso, programmando O-1 si tornerà all'ultima origine attiva prima di O0, ma con il correttore attivato.

Riassunto registri e segnali interessati

OFST	16	CN	↔	PLC	si	Codice del correttore lunghezza da attivare.
INTOF	1	CN	↔	PLC	si	Strobe di segnalazione al CN per l'attivazione del correttore utensile selezionato.
ORIG	16	CN	↔	PLC	si	Codice dell'origine pezzo da attivare.
INORG	1	CN	↔	PLC	si	Strobe di segnalazione al CN per l'attivazione dell'origine pezzo selezionata.
BYORG	1	CN	↔	PLC	si	Annullamento temporaneo di origini e correttori utensili (origine assoluta).
ABSOR	1	CN	⇒	PLC	no	Segnalazione di origine assoluta attiva.

1.5. COMANDI RELATIVI ALL'ASSERVIMENTO E MOVIMENTAZIONE ASSI

La velocità di avanzamento durante l'esecuzione in modo automatico viene regolata da 0 al 200% in funzione del valore scritto sulla variabile **POFO** (tipicamente sarà uguale ad un ingresso analogico **ANI()** il cui range varia da 0 a 1).

Esempio:

POFO = ANI(1) regola tra 0 e 100%
 POFO = ANI(1)*2 regola tra 0 e 200%

1.5.1. ABILITAZIONE E BLOCCAGGIO ASSI

Dal CN viene fornito il registro **MOVCN** con la configurazione degli assi che devono essere abilitati al movimento, ad opera del PLC, in seguito a:

- un blocco programmato o particolari prestazioni geometriche (rototraslazioni, TCM)
- una richiesta di movimento in **JOG** o l'assegnazione di un volantino in modo manuale
- un movimento asse per ciclo di ricerca 0
- la richiesta da parte del PLC di mantenere l'asse sempre attivo.

La configurazione degli assi abilitati e sbloccati, cioè pronti a muovere, deve essere fornita in risposta sul registro **RDMOV**.

1. Gestione movimenti e segnali d'interscambio

Nel periodo in cui i registri **MOV CN** e **RDMOV** sono diversi, cioè in fase di bloccaggio/sbloccaggio assi, il CN resta in attesa di tale coincidenza prima di iniziare il movimento o passare al blocco successivo; non è quindi necessario operare attese con altri segnali.

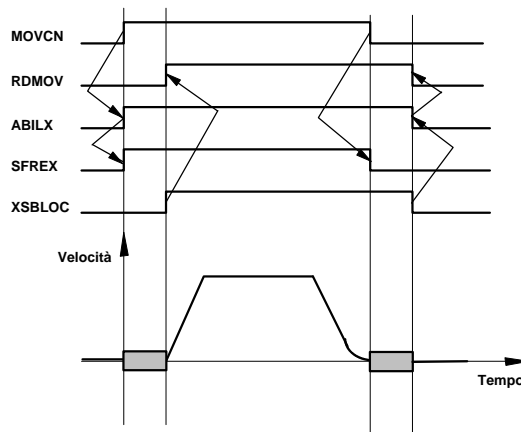
L'anello di posizione per ogni asse è chiuso quando sono presenti **MOV CN** o **RDMOV** relativi.

*Sono da evitare attivazioni di **RDMOV** che non corrispondano a richieste di **MOV CN**.*

Esempio:

```

INP
XSBLOC           asse X sbloccato
OUT
ABILX            abilitazione asse X
SFREX           comando di sfrenatura asse X
...
PROG
SFREX=MOV CN(1)
RDMOV(1)=XSBLOC
ABILX=MOV CN(1)~RDMOV(1)
..
    
```



Riassunto segnali e registri interessati

MOV CN	8	CN	⇒ PLC no	Richieste di abilitazione al movimento degli assi (1..8).
RDMOV	8	CN	⇐ PLC no	Assi abilitati al movimento; risposta al MOV CN (1..8).
POFO	64	CN	⇐ PLC no	Valore dell'override sulla Feed programmata (da 0 a 2 permette una regolazione tra 0 e 200 per cento).

1.5.2. ASSI SEMPRE ATTIVI O CON BLOCCAGGI (M10 - M11)

Tramite il registro asincrono **SSA** il PLC può richiedere al CN la configurazione degli assi che desidera siano comunque sempre abilitati e asserviti tramite l'anello di posizione.

In modalità manuale, il CN prende in considerazione e attua la configurazione richiesta in modo asincrono. In automatico, bisogna però evitare di alterare **SSA** durante programmi contenenti movimenti; conviene farlo in seguito a funzioni ausiliarie.

Sfruttando il registro **AXPGM** si può rendere la funzione selettiva solo per gli assi specificati (M11XYZ).

Riassunto segnali e registri interessati

SSA 8 CN ⇐ PLC no Assi che devono essere sempre attivi (1..8).


1.5.3. RILASCIO DEGLI ASSI (M45 - M46)

Se un asse, normalmente sotto controllo, deve essere movimentato da un sistema esterno, il PLC può richiedere al CN la configurazione degli assi che desidera siano rilasciati tramite il registro sincrono **DSERV**. Quando un asse è rilasciato viene disabilitato, viene trascurato se programmato e non ne viene gestito il riferimento.

Nel momento in cui viene ripreso sotto controllo, resettando **DSERV**, l'asse viene riasservito sulla posizione in cui si trova e abilitato o meno a seconda della configurazione presente nel registro **SSA**.

Il CN prende in considerazione e attua la configurazione richiesta in modo asincrono.

Sfruttando il registro **AXPGM** si può rendere la funzione selettiva solo per gli assi specificati (M45XYZ).

 Nel sistema S1200 questa funzionalità era implementata internamente, ma gestita rigidamente dalle funzioni (M45 e M46).

Riassunto segnali e registri interessati

DSERV 8 CN ⇐ PLC no Assi da rilasciare (1..8).

1.5.4. DISABILITAZIONE DEI TRASDUTTORI

Settando il bit corrispondente all'asse sul registro **DISRQ** è possibile disabilitare completamente la gestione del trasduttore nel caso in cui si debba scollegare fisicamente un trasduttore per la rimozione dell'organo meccanico a cui è connesso o per la commutazione tra più assi.

Questa operazione comporta un rilascio implicito dell'asse in questione.

Il CN prende in considerazione e attua la configurazione richiesta in modo asincrono.

1. Gestione movimenti e segnali d'interscambio

Riassunto segnali e registri interessati

DISRQ 8 CN ⇐ PLC no Assi con i trasduttori disabilitati (1..8).

1.5.5. MOVIMENTO MANUALE IN JOG

In stato di CN manuale (**NCMD=5**) è possibile comandare il movimento degli assi fornendo direzione e velocità, senza un obiettivo predefinito da raggiungere. Il movimento termina quando viene rilasciato il comando e l'asse si è fermato.

S1200 A differenza del sistema S1200 i JOG sono assolutamente necessari, anche durante la RICERCA MEMORIZZATA ed il RIPRISTINO CICLO, per movimentare gli assi (**NCMD=8**) nella fase di ritorno sul profilo; in questo stato non devono perciò essere disabilitati. (vedere *Manuale di Uso e Programmazione*).

La scelta del movimento in JOG assi si effettua settando il bit corrispondente all'asse sul registro di **MOVMA**; quindi, con i registri **JOGP** e **JOGM** si fornisce il comando di movimento e la direzione.

L'asse viene abilitato e preso sotto controllo di spazio, se non lo è, nel momento in cui viene fornito il **MOVMA** corrispondente.

La velocità viene regolata, singolarmente per ogni asse, tramite il registro **POMO(n)** relativo, con un valore compreso tra 0 e 1 (0 - 100% della velocità di rapido).

Riassunto segnali e registri interessati

MOVMA	8	CN	⇐ PLC no	Assi selezionati in movimento manuale (1..8).
JOGP	8	CN	⇐ PLC no	Comandi di jog positivo(1..8).
JOGM	8	CN	⇐ PLC no	Comandi di jog negativo (1..8).
POMO()	64	CN	⇐ PLC no	Velocità per i movimenti manuali e di ricerca zero di ogni singolo asse (1..8) (da 0 a 1 come percentuale della velocità di rapido).

1.5.6. MOVIMENTO IN MANUALE CON VOLANTINI

Gli assi possono essere movimentati, in stato manuale, anche con i volantini elettronici.

L'associazione volantino e asse da muovere, deve essere fatta per mezzo del programma PLC scrivendo il numero dell'asse da muovere nel registro **HWL(n)** corrispondente al volantino in questione.

Esempio:

HWL(1)=5 associa il volantino 1 all'asse 5

Il passo del volantino può essere selezionato scrivendo sulla variabile **STEP** il numero corrispondente al passo, scelto tra gli 8 valori dichiarati nella configurazione che si desidera attuare. Non deve quindi essere scritto il valore del passo in mm/giro.

Gli assi a cui è assegnato un volantino, in modo manuale, vengono automaticamente abilitati.

Il movimento manuale in JOG (selezionato con **MOVMA**) ha priorità sul comando impartito dal volantino, che in questo caso viene ignorato.

Riassunto segnali e registri interessati

HWL()	8	CN ⇐ PLC no	Uno per volantino (1..3) per indicare il numero dell'asse da comandare.
STEP	8	CN ⇐ PLC no	Selezione del passo volantino tra gli 8 valori definiti nei parametri di configurazione.

1.5.7. RICERCA ZERO ASSI

In stato di CN manuale (**NCMD=5**) è possibile eseguire il ciclo di ricerca zero assi, con o senza microinterruttore di zero, fornendo direzione e velocità.

La scelta del ciclo di ricerca zero su marker (encoder o righe ottiche) si effettua settando il bit corrispondente all'asse sul registro **MARK**.

Se la ricerca deve essere effettuata con la presenza del microinterruttore di zero occorrerà, in alternativa, settare il bit corrispondente all'asse sul registro **MICZE**.

In tutti i casi, il rifasamento elettrico del trasduttore di un certo asse sullo zero viene segnalato dallo stato del bit corrispondente all'asse sul registro **MIZEA**.

Nei dati di configurazione occorre specificare se è presente o meno il microinterruttore di zero coerentemente con la predisposizione della macchina utensile. Questa informazione è utilizzata dal CN per discriminare casi particolari come l'utilizzo di un resolver in rapporto meccanico 1:1 con l'organo su cui rilevare la quota, nel caso in cui il trasduttore è usato come assoluto e non necessita di alcuna gestione supplementare PLC.

Per i trasduttori assoluti, o usati come tali (vedere caso precedente) **MIZEA** è sempre presente a meno di errori sul sistema di misura.

E' altresì importante ricordare che i FINECORSO SOFTWARE sono attivi solo dopo che il trasduttore è stato azzerato sullo zero elettrico.

La priorità tra i registri di selezione del tipo di movimento asse in JOG (manuale, ricerche 0) è la seguente:

MICZE	- priorità più elevata
MARK	
MOVMA	- priorità minima

1. Gestione movimenti e segnali d'interscambio

Ciclo di ricerca zero asse con microinterruttore

fase A:

- Dopo aver settato il bit corrispondente all'asse sul registro **MICZE**, l'asse viene abilitato e preso sotto controllo di spazio (se non lo è già).
- Con i registri **JOGP** o **JOGM** si fornisce il comando di movimento che deve essere poi mantenuto fino alla fine del ciclo (cioè quando viene settato il registro **MIZEA**).
- La velocità viene, come per il manuale in JOG, regolata tramite il registro **POMO(n)**, relativo all'asse, con un valore compreso tra 0 e 1 (0 - 100% riferito alla velocità di rapido).
- Quando viene raggiunto il microinterruttore di zero (camma), segnalato sul registro **MIZER**, l'asse viene fatto decelerare fino a fermarsi.

fase B:

- Automaticamente viene invertita la direzione di movimento e ridotta la velocità ad 1/8 della velocità di andata.
- Dopo aver rilasciato il microinterruttore di zero, proseguendo in quella direzione, viene azzerato il trasduttore sul primo zero elettrico incontrato, e quindi sulla quota assoluta dell'asse viene caricato il valore di "*quota di 0 macchina*" definito nei dati di configurazione (vedere documentazione specifica).

fase C:

- Il ciclo prosegue, automaticamente, con il posizionamento dell'asse sulla quota specificata in fase di configurazione nel parametro "*Posizione di arresto ricerca 0*", con la stessa velocità con cui si è incontrato **MIZER**.
- Infine viene data la segnalazione di asse azzerato sul registro **MIZEA** con il bit relativo all'asse.

Se viene rilasciato il JOG durante il ciclo, l'asse viene comunque fermato con l'accelerazione corrente e si possono presentare i seguenti casi:

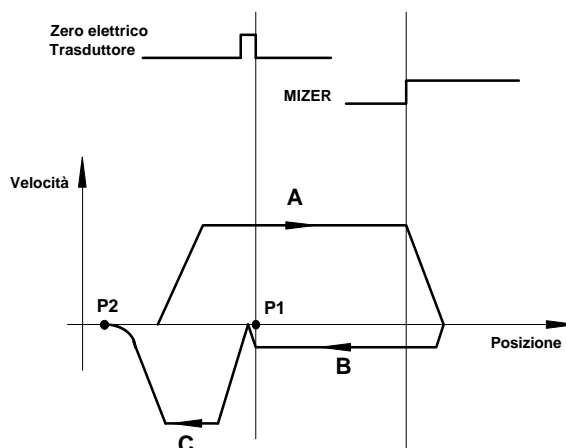
Il JOG viene rilasciato durante la fase A prima che venga impegnato MIZER :	Se il trasduttore era già stato azzerato, viene ripristinato il valore di MIZEA precedente.
Il JOG viene rilasciato dopo aver impegnato il MIZER ma prima di aver incontrato il segnale di zero elettrico:	MIZEA non viene ripristinato.
Il JOG viene rilasciato nella fase C, durante il posizionamento su " <i>Posizione di arresto ricerca 0</i> ":	Viene segnalato MIZEA in quanto il trasduttore è già stato rifasato elettricamente anche se l'asse non è posizionato.

Se il ciclo inizia con la camma già premuta, la sequenza inizia dalla fase B.

Il ciclo viene comunque sempre interrotto quando viene rilasciato il registro **MICZE**.

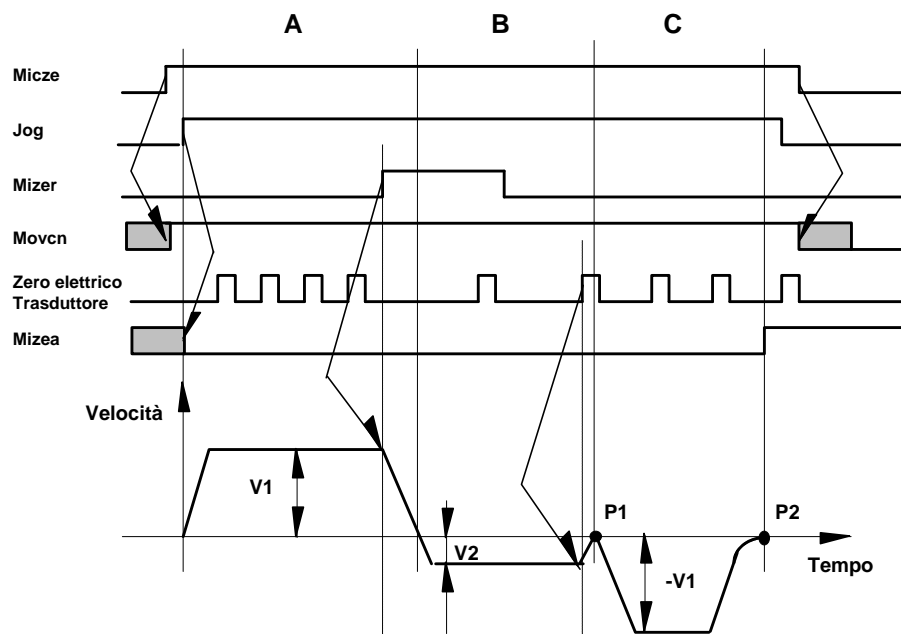
Volendo ripetere il ciclo di ricerca dopo averne terminato uno precedente, è sufficiente ripetere la sequenza di comandi descritta precedentemente. Il segnale **MIZEA** viene riазzerato e la sequenza inizia nuovamente.

Ciclo di ricerca zero su microinterruttore



P1 = punto in cui viene caricata la quota di zero macchina
P2 = posizione di arresto ricerca zero

Temporizzazione del ciclo di ricerca zero su microinterruttore



P1 = punto in cui viene caricata la quota di zero macchina
P2 = posizione di arresto ricerca zero
V1 = velocità di ricerca zero
V2 = velocità di disimpegno della camma (1/8 di V1)

1. Gestione movimenti e segnali d'interscambio

Ciclo di ricerca zero su zero elettrico del trasduttore (marker)

fase A:

- Dopo aver settato il bit corrispondente all'asse sul registro **MARK**, l'asse viene abilitato e preso sotto controllo di spazio (se non lo è già).
- Con i registri **JOGP** o **JOGM** si fornisce il comando di movimento che deve essere poi mantenuto fino alla fine del ciclo (cioè quando viene settato il registro **MIZEA**).
- La velocità viene, come per il manuale in JOG, regolata tramite il registro **POMO(n)**, relativo all'asse, con un valore compreso tra 0 e 1 (0 - 100% riferito alla velocità di rapido).
- Il trasduttore viene azzerato sul primo zero elettrico incontrato e fatto decelerare fino a fermarsi; quindi sulla quota assoluta dell'asse viene caricato il valore di "*quota di 0 macchina*" definito nei dati di configurazione (vedere documentazione specifica).

fase B:

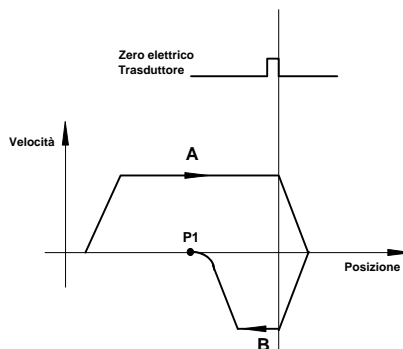
- Il ciclo prosegue, automaticamente, con il posizionamento dell'asse sulla quota specificata in fase di configurazione nel parametro "*Posizione di arresto ricerca 0*", con la stessa velocità con cui si è incontrato lo zero elettrico.
- Infine viene data la segnalazione di asse azzerato sul registro **MIZEA** con il bit relativo all'asse.

Se viene rilasciato il JOG durante il ciclo, l'asse viene comunque fermato con l'accelerazione corrente e si possono presentare i seguenti casi:

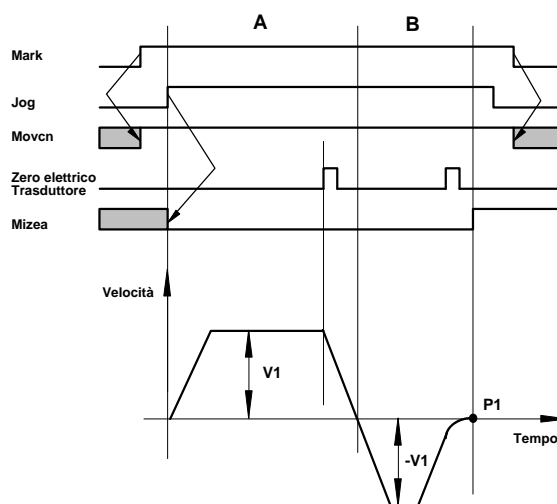
il JOG viene rilasciato prima di raggiungere lo zero elettrico del trasduttore:	MIZEA non viene ripristinato.
il JOG viene rilasciato nella fase B, durante il posizionamento su " <i>Posizione di arresto ricerca 0</i> ":	viene segnalato MIZEA in quanto il trasduttore è già stato rifasato elettricamente anche se l'asse non è posizionato.

Il ciclo viene comunque sempre interrotto quando viene rilasciato il registro **MARK**.

Volendo ripetere il ciclo di ricerca dopo averne terminato uno precedente, è sufficiente ripetere la sequenza di comandi descritta precedentemente. Il segnale **MIZEA** viene riазzerato e la sequenza inizia nuovamente.

Ciclo di ricerca zero su marker


P1= Posizione di arresto ricerca 0

Temporizzazione del ciclo di ricerca zero su marker


P1 = Posizione di ricerca zero
V1 = velocità di ricerca zero

Ciclo di ricerca zero su righe ottiche codificate

Per effettuare la ricerca di zero su righe ottiche codificate si deve utilizzare la sequenza di ricerca zero con microinterruttore, come precedentemente descritta.

Il microinterruttore di zero (**MIZER()**), che deve essere posizionato in prossimità dell'oltrecorsa, viene utilizzato per invertire la direzione di ricerca zero in modo automatico senza ulteriori azioni da parte del PLC.

Se durante la ricerca di zero viene percorso uno spazio maggiore del massimo previsto per il ciclo, si passa in stato di emergenza, **EMEA** = 1, e viene visualizzato un messaggio di errore. E' possibile cadere in questa situazione con dei parametri di configurazione non corretti.

1. Gestione movimenti e segnali d'interscambio

Riassunto segnali e registri interessati

MICZE	8	CN	⇐	PLC no	Assi selezionati in ricerca zero con microinterruttore (1..8).
MARK	8	CN	⇐	PLC no	Assi selezionati in ricerca zero senza microinterruttore (1..8).
MIZER	8	CN	⇐	PLC no	Microinterruttore di zero assi (1..8).
MIZEA	8	CN	⇒	PLC no	Assi riferiti allo zero elettrico del trasduttore (1..8).

1.5.8. MOVIMENTI IN MANUALE DURANTE LO STATO DI HOLD

Con l'esecuzione sospesa in seguito ad un comando di HOLD (segnale **HOLDA**=1), si ha la possibilità, senza dover interrompere il programma, di abilitare il movimento degli assi in JOG o volante, tramite softkey (che compaiono automaticamente nel menu di esecuzione).

In questo stato di MANUALI IN HOLD il registro **NCMD** vale 9; se si vuole utilizzare tale prestazione non bisogna quindi inibire i comandi di JOG.

Quando si desidera riprendere l'esecuzione del programma occorre, sempre tramite softkey, passare in stato di RITORNO SUL PROFILO (**NCMD** = 8), quindi riposizionare gli assi sul profilo in esecuzione, con i JOG (vengono abilitati automaticamente solo i comandi nella direzione di avvicinamento al pezzo).

1.5.9. MOVIMENTO IN MANUALE E RICERCA ZERO CONTEMPORANEI ALLA LAVORAZIONE

I cicli descritti, riguardanti i movimenti manuali e ricerca zero, possono essere effettuati anche durante l'esecuzione di un programma, contemporaneamente alla lavorazione, a condizione che venga settato il bit corrispondente all'asse del registro sincrono **FOMAN** per il forzamento in modo manuale.

Questo stato comporta un rilascio implicito dell'asse.

Il CN prende in considerazione e attua la configurazione richiesta in modo sincrono.

Riassunto segnali e registri interessati

FOMAN	8	CN	⇐	PLC si	Assi su cui forzare i comandi manuali contemporaneamente alla lavorazione (1..8).
--------------	---	----	---	--------	---

1.5.10. INFORMAZIONI SUGLI ASSI

Tramite un set di registri predefiniti è possibile leggere in qualunque istante alcune informazioni rilevanti relative ad ogni singolo asse CN a scopo di debug, taratura oppure, in casi sporadici, per l'implementazione di algoritmi di tipo particolare.

I registri sono stati suddivisi in tre aree nella tabella che segue in cui le descrizioni dei segnali e dei registri sono sufficientemente esaustivi da non richiedere ulteriori dettagli.

Riassunto segnali e registri interessati
Per l'asservimento degli assi

ERR()	64	CN	⇒	PLC	no	Errori di inseguimento assi (1..8).
VATT	64	CN	⇒	PLC	no	Velocità attuale teorica imposta sul percorso utensile.
TACH()	64	CN	⇒	PLC	no	Velocità assi (1..8) misurate dai trasduttori di posizione.
VFF()	64	CN	⇒	PLC	no	Velocità istantanea prescritta per gli assi (1..8).
AFF()	64	CN	⇒	PLC	no	Accelerazione istantanea prescritta per gli assi (1..8).
DAA()	64	CN	⇒	PLC	no	Riferimenti di tensione per assi controllati (1..8). Se l'asse è attivo e sotto il controllo del CN, il suo DAA può solamente essere letto. Il contenuto varia fra -1 e 1 in corrispondenza di tensioni in uscita fra -10 e +10 V.
POA()	64	CN	⇒	PLC	no	Posizioni riferite all'origine assoluta assi (1..8).
POO()	64	CN	⇒	PLC	no	Posizioni assi riferite all'origine corrente e correttore utensile attivo (1..8).
POATE()	64	CN	⇒	PLC	no	Quota teorica istantanea degli assi controllati lungo la traiettoria di interpolazione (1..8) riferita all'origine assoluta.
POOTE()	64	CN	⇒	PLC	no	Quota teorica istantanea degli assi controllati lungo la traiettoria di interpolazione (1..8) riferita all'origine attiva.
POORT()	64	CN	⇒	PLC	no	Quota teorica istantanea soggetta ad eventuali rototraslazioni del sistema di coordinate per gli assi controllati, lungo la traiettoria di interpolazione (1..8) riferita all'origine attiva.
PFCN()	64	CN	⇒	PLC	no	Quota finale programmata per gli assi (1..8).

Per lo stato degli assi

INTOL	8	CN	⇒	PLC	no	Assi (1..8) entro la "tolleranza di posizione asse" dichiarata nei dati di configurazione.
JOGIN	8	CN	⇒	PLC	no	Assi (1..8) in movimento in seguito a comando JOG (manuale o ricerca 0).
RAPI	1	CN	⇒	PLC	no	Blocco in esecuzione con velocità rapido.

Per il controllo dei sistemi di misura e dei volantini elettronici

MKSAX	8	CN	⇒	PLC	no	Segnalazione del marker (zero elettrico) degli assi nel caso di trasduttori encoder o righe ottiche (1..8). Viene settato dal CN quando si incontra il marker elettrico del trasduttore ed è resettato nella campionatura di sistema successiva; per questo motivo è visibile sull'analizzatore grafico solamente con base tempo uguale alla campionatura di sistema.
AIRGP()	64	CN	⇒	PLC	no	Livello del segnale nel caso di trasduttori analogici (INDUCTOSYN o RESOLVER) oppure numero degli impulsi persi nel caso di ENCODER con l'algoritmo "recupero step" presente per gli assi(1..8).
SPMANO()	64	CN	⇒	PLC	no	Spazio comandato da percorrere con i volantini (1..3) secondo il passo selezionato. Lo spazio accumulato viene resettato sui cambi di stato CN e dalle variazioni sugli assi (SSA, DSERV, ...)

1. Gestione movimenti e segnali d'interscambio

Informazioni sugli assi: entità dello spostamento origine (**G851**)

Sul vettore **OFHWL()** di 8 elementi vengono caricati, rispettivamente per ogni asse macchina, i valori in millimetri di spostamento dell'origine ottenuto tramite i volantini quando è attiva la funzione **G851**.

L'entità dello spostamento può quindi essere visualizzata sui quadri video CN utilizzando le variabili per la visualizzazione disponibili nel PLC.

Nome	Dim	Direzione	Descrizione
OFHWL()	64	CN=>PLC	Spostamenti (1..8) dell'origine pezzo tramite G851

Informazione sugli assi controllati: nuove variabili

Variabili per debug e taratura assi:

Nome	Dim	Direzione	Descrizione
AXRIF()	64	CN \Rightarrow PLC	Comando di velocità inviato agli assi (1..8) [mm/min].
OFSVA()	64	PLC \Rightarrow CN	Offset aggiuntivo di velocità per gli assi (1..8) [mm/min]. (Influenza anche AXRIF() - da usare solo per applicazioni particolari)
AFF()	64	CN \Rightarrow PLC	Comando di accelerazione impresso agli assi (1..8) [mm/sec ²]

1.5.11. COMPENSAZIONI DINAMICHE DI MISURA ASSI

Sui registri **SHIFT** il PLC ha la facoltà di scrivere un valore, direttamente in millimetri, per la compensazione dinamica delle quote, da applicare ai singoli assi per compensare fenomeni dovuti a deformazione sia termica che meccanica.

La compensazione agirà in due modi differenti a seconda che l'asse sia asservito o meno:

asse asservito: la quota visualizzata non varia, ma fisicamente l'asse si sposta della quantità indicata da **SHIFT**.

asse non asservito: l'asse non può evidentemente muoversi, quindi è la quota a variare della quantità impostata su **SHIFT**.

Riassunto segnali e registri interessati

SHIFT() 64 CN \Leftarrow PLC no Compensazione dinamica delle quote assi (1..8).

1.5.12. OFFSET PER ASSI CONTROLLATI

Per applicazioni particolari è possibile sommare un offset di tensione al riferimento analogico calcolato per gli assi controllati. Questa prestazione deve essere utilizzata con estrema cautela; valori non appropriati potrebbero causare anomalie sulla movimentazione degli assi CN.

Riassunto segnali e registri interessati

OFSDA() 64 CN \Leftarrow PLC no Offset da applicare al riferimento analogico sui canali degli assi controllati (1..8) nel range ± 1 per tensioni tra ± 10 Volt.

OFFSET DI ORIGINE AGGIUNTIVO PER GLI ASSI CONTROLLATI

Per applicazioni particolari può essere attivato un offset di posizione supplementare alle origini pezzo tramite il PLC. Lo spostamento origine rimane attivo anche dopo lo spegnimento del Controllo Numerico così da garantire la posizione in caso di trasduttori assoluti.

Il valore dell'offset, espresso in millimetri o in gradi, deve essere caricato sul vettore **PLORG()** di 8 elementi (rispettivamente uno per ogni singolo asse). Gli offset devono essere attivati con una funzione M di fine blocco che setti il bit **STORG_(1)** in modo sincrono con il segnale BURDY. Gli altri bit del byte **STORG_** sono riservati ad eventuali altri gruppi di assi.

Allo stesso modo impostando **STORG_** a 0 in modo sincrono vengono disattivati tutti gli offset aggiuntivi.

E' importante ricordare che l'attivazione e la disattivazione degli offset avviene solamente in seguito alla transizione del bit **STORG_(1)** da zero a uno o da uno a zero rispettivamente. Per esempio se il sistema si avvia con il bit a zero è attiva solo la salita a uno e viceversa. Allora, per coerenza con lo stato delle memorizzazioni interne del controllo assi, si suggerisce di creare un bit di appoggio in ram statica (SRAM) per memorizzare lo stato di **STORG_** a CN spento e reinizializzarlo in accensione.

Tipicamente questa prestazione viene utilizzata su macchine con testa rotativa con doppio mandrino contrapposto; gli offset aggiuntivi rappresentano le differenze di posizione tra il «naso» mandrino primario e «secondario».

In questo caso l'attivazione di **STORG_** viene fatta su una funzione ausiliaria M di fine blocco all'interno di un programma COM usato per lo scambio dei mandrini.

In origine assoluta gli offset origine sono disabilitati.

Nome	Dim	Direzione	Descrizione
STORG_	8	PLC ⇒ CN	Registro di attivazione degli offset origini aggiuntivi. STORG_(1) = 1 attiva gli offset (per tutti gli assi) STORG_(1) = 0 disattiva gli offset
PLORG()	8	PLC ⇒ CN	Registri contenenti gli offset di origine aggiuntivi

1.6. GESTIONE DEL PROBE DI MISURA ON/OFF

Se il sistema rileva il segnale di probe deflesso al di fuori del ciclo di misura segnala errore 210 e passa in stato di emergenza (urto probe on/off).

Il PLC può disabilitare la rilevazione di questo errore settando il bit 1 della variabile **CWDTF**.

Riassunto segnali e registri interessati

CWDTF 8 CN ⇐ PLC no Byte di controllo del Probe on/off:
Bit 1: disabilitazione errore 210 (urto)

GESTIONE DEL PROBE DI MISURA ON/OFF: DIAGNOSTICA

Tramite il registro **SWDTF** è possibile leggere lo stato del probe ON/OFF di misura (questo registro è da utilizzare principalmente per scopi diagnostici).

Nome	Dim	Direzione	Descrizione
SWDTF	8	PLC ⇒ CN SWDTF(2)	Stato del probe ON/OFF = 0 probe a riposo = 1 probe deflesso

1. Gestione movimenti e segnali d'interscambio

1.7. FINE CORSA SOFTWARE ASSI CONTROLLATI

Lo stato degli assi in fine corsa software viene segnalato sui registri **FICOP** e **FICOM** (fine corsa positivi e negativi). Il PLC ha la facoltà di disabilitare i *finecorsa software alzando i bit relativi agli assi sui registri DFCOP* (disabilitazione finecorsa positivi) e *DFCOM* (disabilitazione finecorsa negativi).

Riassunto segnali e registri interessati

FICOP	8	CN	⇒ PLC no	Assi (1..8) in fine corsa software positivo.
FICOM	8	CN	⇒ PLC no	Assi (1..8) in fine corsa software negativo.
DFCOP	8	CN	⇐ PLC no	Disabilitazione finecorsa software positivi assi (1..8).
DFCOM	8	CN	⇐ PLC no	Disabilitazione finecorsa software negativi assi (1..8).

GESTIONE DEL PROBE DI MISURA ON/OFF: DIAGNOSTICA

Tramite il registro **SWDTF** è possibile leggere lo stato del probe ON/OFF di misura (questo registro è da utilizzare principalmente per scopi diagnostici).

Nome	Dim	Direzione	Descrizione
SWDTF	8	PLC=>CN	Stato del probe ON/OFF
		SWDTF(2)	= 0 probe a riposo = 1 probe deflesso

FINE CORSA SOFTWARE ASSI CONTROLLATI: DISATTIVAZIONE ERRORE E93

Settando la variabile **CWFCS** è possibile disabilitare la rilevazione preventiva al movimento dei finecorsa software, e di conseguenza la segnalazione dell'errore «**E93: ASSI IN FINECORSA**»; la limitazione della corsa degli assi dovuta ai finecorsa software non viene comunque alterata.

Questa prestazione deve essere usata quando il PLC, per esigenze di impianto, agisce anche, ad assi in movimento, sulle variabili relative ai finecorsa software, per esempio disabilitando i finecorsa con **DFCOP**, **DFCOM** o cambiando la coppia di finecorsa attivi – variabile **FCA**).

Negli stati CN di esecuzione programma o blocco singolo, i settaggi di **CWFCS** devono essere effettuati in modo sincrono con il segnale BURDY.

Nome	Dim	Direzione	Descrizione
CWFCS	8	PLC ⇒ CN	Controllo errori finecorsa software
		CWFCS(1)	= 1 controllo E93 disabilitato = 0 controllo E93 abilitato (default).

1.7.1. FINE CORSA SOFTWARE SUPPLEMENTARI

Nei parametri di configurazione, singolarmente per ogni asse, è possibile introdurre una seconda coppia di fine corsa software supplementari da abilitare tramite PLC nei casi in cui si debbano modificare le dimensioni del campo operativo (per esempio nel caso di un cambio utensile “planare a rastrelliera” appoggiato sull’asse X).

Per attivare i fine corsa supplementari (secondari) occorre scrivere sull’array **FCA()** di 8 elementi (rispettivamente uno per ogni asse CN) quale coppia di fine corsa si intende abilitare.

Esempio :

Si consideri una configurazione con X, Y, Z in cui si debbano attivare i fine corsa secondari sull’asse Z; si scriverà:

FCA(3)=2 [Abilita coppia fine corsa secondari asse Z.

Per tornare ai fine corsa primari si potrà indifferentemente scrivere:

FCA(3)=1 *[Abilita coppia fine corsa primari asse Z.*

oppure:

FCA(3)=0 *[Disabilita gestione fine corsa supplementari asse Z.*

Se l'array **FCA** non viene utilizzato, sono attivi per default i fine corsa primari su tutti gli assi controllati.

Riassunto segnali e registri interessati

FCA() 8 CN ⇐ PLC no Array abilitazioni fine corsa secondari per gli assi CN (1..8)

1.8. GESTIONE DI ASSI DI TIPO PARTICOLARE

1.8.1. ASSI PARALLELI (GANTRY)

Gli assi Gantry, ossia controllati da due asservimenti paralleli comandati da un unico indirizzo di programmazione, sono completamente gestiti dal software di base del CN secondo i parametri di configurazione inseriti.

I parametri di configurazione relativi ad accelerazione e velocità debbono essere rigorosamente identici. L'asse definito MASTER è quello associato al nome utilizzato dall'utente, quello secondario è definito SLAVE.

L'interfaccia PLC con il CN tratta solamente l'asse MASTER ad eccezione dei segnali di riconoscimento del micro di zero. I comandi di movimento in manuale, (**JOG**), di regolazione velocità, (**POMO**), ricerca zero, (**MICZE**, **MARK**), sono richiesti solo sull'asse MASTER. I segnali di controllo e abilitazione servo, (**MOVCN**, **RDMOV**, **SSA**), sono anch'essi trattati solo per l'asse MASTER. Le segnalazioni di micro di zero raggiunto, (**MIZER**), devono invece, essere comunicate al CN per entrambi gli assi anche nel caso di provenienza dal medesimo ingresso. Nel normale funzionamento, i due assi devono essere agganciati in spazio con un offset di compensazione pari alla differenza dei trasduttori fissato nel parametro di configurazione **OFFSET NOMINALE GANTRY**.

Il comando di abilitazione di questo offset è il bit nella variabile **OFSGY** corrispondente al numero dell'asse SLAVE. Se **OFSGY()** è a zero, gli assi sono asserviti e movimentati mantenendo l'offset di posizione inizialmente rilevato durante l'accensione del CN. Quando gli assi non sono assoluti questo aggancio viene effettuato solo dopo il riconoscimento di entrambi gli zeri e prima di questo evento gli assi restano asserviti con l'offset iniziale. Durante le operazioni di messa in fase dei due asservimenti, quando non è ancora determinato il valore di offset nominale gantry, è preferibile mantenere disabilitato il bit di abilitazione nella variabile **OFSGY**.

1. Gestione movimenti e segnali d'interscambio

Ciclo di ricerca zero con micro per assi GANTRY

- Settare per l'asse MASTER il registro **MICZE**, quindi fornire il comando **JOG** della direzione richiesta, il valore di velocità su **POMO**, come per un asse comune.
- L'asse SLAVE insegue l'asse MASTER mantenendo l'offset inizialmente letto all'accensione fino a raggiungere entrambi i micro di zero (segnalati sui bit **MIZER** relativi ai due assi).
- La coppia di assi inverte la direzione a velocità ridotta a 1/8 fino a liberare entrambi i micro di zero.
- Il movimento prosegue fino alla lettura dei due marker di zero.
- Il CN trasmette al PLC i due bit su **MIZEA** relativi ai due assi e, se abilitato da **OFSGY**, applica l'offset gantry fissato nel parametro di configurazione **OFFSET NOMINALE GANTRY**.

Riassunto segnali e registri interessati

OFSGY 8 CN ⇐ PLC no Abilitazione offset nominale per assi gantry (1..8). Deve essere settato il bit corrispondente al numero asse SLAVE.

1.8.2. ASSI PROGRAMMABILI NON CONTROLLATI

Se in un blocco CN viene programmato il movimento di un asse il cui nome non rientra tra quelli definiti come assi controllati, la quota (così come programmata) viene passata al PLC sull'elemento relativo del vettore **AUXPF()** accompagnato dallo strobe sincro **STRPF**.

Per questi assi sarà perciò il PLC ad attuare il movimento, utilizzando se necessario il MODULO ASSI INDIPENDENTI.

Le posizioni da raggiungere sono passate sugli elementi di **AUXPF()** con la seguente disposizione:

AUXPF(1) = quota dell'asse A con lo strobe **STRPF(1)**
AUXPF(2) = quota dell'asse B con lo strobe **STRPF(2)**
AUXPF(3) = quota dell'asse C con lo strobe **STRPF(3)**
AUXPF(4) = quota dell'asse U con lo strobe **STRPF(4)**
AUXPF(5) = quota dell'asse V con lo strobe **STRPF(5)**
AUXPF(6) = quota dell'asse W con lo strobe **STRPF(6)**

Riassunto segnali e registri interessati

AUXPF() 64 CN ⇐ PLC si Quote programmate per gli assi che devono essere movimentati tramite il PLC (1..6).
STRPF 8 CN ⇐ PLC si Strobe di presenza nuove informazioni su **AUXPF()** (1..6).

1.8.3. ASSI MASTER SLAVE (OPZIONE CN «MS»)

Tramite la funzione **G15** (solo sistemi predisposti) è possibile «agganciare» due assi macchina (uno principale detto Master ed uno secondario detto Slave) in modo che tutti i comandi di movimento impartiti all'asse Master vengano parimenti eseguiti dallo Slave.

La sintassi è : **G15** asse_slave asse_master I...
(I rappresenta un fattore di scala tra i due movimenti).

La funzione **G14** cancella la **G15**.

Per informazioni più dettagliate vedere il *Notiziario Tecnico 1 del 1996*.

1.8.4. LETTURA INGRESSI E SCRITTURA USCITE ANALOGICHE: MODULI I/O REMOTATI

Per l'interfacciamento di ingressi, uscite analogiche, sonde di temperatura tramite moduli I/O remotati non sono richiesti parametri di configurazione nel CN.

La lettura di ingressi analogici fornisce al PLC un valore numerico in formato 64 bit variabile fra 0 e 1 come percentuale del valore di fondo scala.

Ingressi analogici

La sintassi è la seguente:

ANInumero piastra master (numero slave numero ingresso)

Dove :

numero piastra master indica quale è il POSTO PIASTRA della scheda con interfaccia master RIO analogamente al caso di I/O locali ove è relativo alla scheda I/OMIX. Nel caso di utilizzo della scheda master con RIO integrato, il numero piastra sarà 17.

numero slave dichiara l'indirizzo impostato con i microinterruttori sul modulo remotato.

numero ingresso dichiara l'ingresso utilizzato sul modulo.

Esempio:

AN17(6002) significa ingresso analogico n. 2 del modulo remotato SLAVE con indirizzo 60 collegato alla interfaccia MASTER RIO in posizione 17.

AN1(3) significa il canale 3 di ingresso analogico della prima scheda I/OMIX

Uscite analogiche

Le uscite analogiche scritte dal PLC con un valore numerico in formato 64 bit variabile fra -1 e 1 come percentuale del valore di fondo scala producono una tensione di uscita variabile fra -10V e +10V. Non sono richiesti parametri di configurazione nel CN.

Nel programma PLC l'accesso è ottenuto con una variabile VELO... così composta:

VELOnumero piastra master (numero slave numero uscita)

1. Gestione movimenti e segnali d'interscambio

Dove :

numero piastra master	indica quale è il POSTO PIASTRA della scheda con interfaccia master RIO analogamente al caso di I/O locali ove è relativo alla scheda I/OMIX.
numero slave	dichiara l'indirizzo impostato con i microinterruttori sul modulo remotato.
numero uscita	dichiara l'uscita utilizzata sul modulo.

Esempio:

VELO17(6002) significa uscita analogica n. 2 del modulo remotato SLAVE con indirizzo 60 collegato alla interfaccia MASTER RIO in posizione 17.

VELO(3) significa uscita analogica n. 3 della prima scheda I/OMIX.

Ingressi per sonde di temperatura

La lettura di ingressi per sonde di temperatura analogici fornisce al PLC un valore in gradi della temperatura rilevata dalle sonde termiche in formato 64 bit.

Non sono richiesti parametri di configurazione nel CN. Nel programma PLC l'accesso è ottenuto con una variabile TEMP... così composta:

TEMP*numero piastra master (numero slave numero ingresso)*

Dove :

numero piastra master	indica quale è il POSTO PIASTRA della scheda con interfaccia master RIO analogamente al caso di I/O locali ove è relativo alla scheda I/OMIX.
numero slave	dichiara l'indirizzo impostato con i microinterruttori sul modulo remotato.
numero ingresso	dichiara l'ingresso utilizzato sul modulo.

Esempio:

TEMP17(6002) significa ingresso sonda n. 2 del modulo remotato SLAVE con indirizzo 60 collegato alla interfaccia MASTER RIO in posizione 17.

1.9. LETTURA INGRESSI E SCRITTURA USCITE ANALOGICHE

Il PLC ha la facoltà di accedere direttamente ai canali fisici di ingresso e uscita analogici.

Ogni elemento dei registri seguenti ha come indice il numero del canale fisico e come desinenza del nome il numero della piastra a cui ci si riferisce *per le schede maggiori di uno*.

Esempio:

ANI2(3) significa il canale 3 di ingresso analogico della seconda scheda I/OMIX

ANI(2) significa il canale 2 di ingresso analogico della prima scheda I/OMIX

Riassunto segnali e registri interessati

ANIx()	64	CN \Rightarrow PLC no	Lettura ingressi analogici della scheda I/OMIX specificata e delle sue espansioni. Il valore letto varia fra 0 e 1 come percentuale del valore di fondo scala.
VELOx()	64	CN \Leftarrow PLC no	Scrittura delle uscite analogiche della scheda I/OMIX specificata e delle sue espansioni. Queste uscite possono sempre essere lette, ma scritte solo se non sono utilizzate dal CN per gli assi controllati o dai moduli interni per la gestione mandrini o assi indipendenti. Il contenuto può variare fra -1 e 1 come percentuale del valore di fondo scala (± 10 V).
TEMPx()	64	CN \Rightarrow PLC no	Valore in gradi della temperatura rilevata dalle sonde termiche (se presente l'interfaccia) relative alla scheda specificata.

1.10. SCAMBIO DATI TRA PLC E PART PROGRAM

Il PART PROGRAM ha la possibilità di scambiare dati con il PLC nei formati BIT, BYTE, WORD, LONG, tramite istruzioni:

OUT(formato) = argomento per inviare l'argomento al PLC
Pxx = INP(formato) per ricevere un valore dal PLC

dove:

formato può essere 1, 8, 16, 32 per identificare rispettivamente BIT, BYTE, WORD, LONG.

argomento può essere il risultato di un'espressione, un parametro **Pxx** o un numero in modo esplicito.

A seconda del formato e della direzione dell'informazione saranno interessate le variabili riportate nel riassunto che segue; nel caso di passaggio dati da part program a PLC è pure presente uno strobe che segnala un nuovo valore.

In alternativa, il PLC può accedere direttamente in lettura e scrittura ai parametri **Pxx** (da P1 a P99) del CN con le variabili del vettore **PCN()** (da **PCN(1)** a **PCN(99)**)

ATTENZIONE! La scrittura di parametri Pxx da PLC è sconsigliata perché può causare conflitti con programmi di lavorazione o particolari cicli del CN quindi va utilizzata con attenzione. Ad esempio i parametri **P90.....P99** sono utilizzati dal CN nei cicli di misura.

Per i part program lanciati in esecuzione con le istruzioni **COM**, invece, esiste nel PLC un set di parametri da **P(1)** a **P(99)** che corrispondono ai **Pxx** usati nel programma lanciato.

Questi hanno lo stesso nome, ma non hanno niente a che fare con i parametri **Pxx** dei part program eseguiti direttamente dall'operatore.

1. Gestione movimenti e segnali d'interscambio

Riassunto segnali e registri interessati

VPLFL	32	CN	⇒	PLC	si	Variabile in FLOATING da part program al PLC.
STVFL	1	CN	⇒	PLC	si	Strobe presenza variabile in FLOATING da part program al PLC.
VPLWO	16	CN	⇒	PLC	si	Variabile WORD da part program al PLC.
STVWO	1	CN	⇒	PLC	si	Strobe presenza variabile WORD da part program al PLC.
VPLBY	8	CN	⇒	PLC	si	Variabile BYTE da part program al PLC.
STVBY	1	CN	⇒	PLC	si	Strobe presenza variabile BYTE da part program al PLC.
VPLBI	1	CN	⇒	PLC	si	Variabile BIT da part program al PLC.
STVBI	1	CN	⇒	PLC	si	Strobe presenza variabile BIT da part program al PLC.
VLPFL	32	CN	⇐	PLC	si	Variabile in FLOATING inviata al part program dal PLC.
VLPWO	16	CN	⇐	PLC	si	Variabile WORD inviata al part program dal PLC.
VLPBY	8	CN	⇐	PLC	si	Variabile BYTE inviata al part program dal PLC.
VLPBI	1	CN	⇐	PLC	si	Variabile BIT inviata al part program dal PLC.
PCN()	32	CN	⇔	PLC	no	99 parametri in formato floating point condivisi, cioè leggibili e scrivibili sia da PLC che da part program a livello utente (1..99).
P()	32	CN	⇔	PLC	no	99 parametri in formato floating point condivisi, cioè leggibili e scrivibili sia da PLC che da sottoprogrammi lanciati dal PLC con le istruzioni COM (1..99).

1.11. FINESTRE PER LA VISUALIZZAZIONE SU VIDEO CN

Un set di variabili predefinite permette al PLC di visualizzare dati nell'area CN nelle posizioni e con i formati predefiniti dalla caratterizzazione delle tabelle video (vedere il *Manuale di Configurazione Sistema*).

Riassunto segnali e registri interessati

WINDOW()	64	CN	⇐	PLC	no	Registri per visualizzazione sul video del CN nelle aree prefissate (1..16) nei formati floating point long o double. La visualizzazione di queste aree è attivata tramite un condizionamento, nelle tabelle video di default fornite con il CN, dal corrispondente ASCW diverso da 0.
ASCW()	8	CN	⇐	PLC	no	Registri per visualizzazione di caratteri sul video del CN nelle aree prefissate (1..16). Occorre fornire il codice ASCII del carattere.
WNDINT()	16	CN	⇐	PLC	no	Registri per visualizzazione sul video del CN nelle aree prefissate (1..16) nel formato word.
WNDSTR()	str	CN	⇐	PLC	no	Registri stringa contenenti un max di 64 caratteri alfanumerici per la visualizzazione sul video del CN nelle aree prefissate (1..16).
GIRMI	64	CN	⇐	PLC	no	Registro per la visualizzazione del valore della funzione S nell'area prefissata sul video CN.

Si ricorda che, come descritto a riguardo delle softkey, il PLC può cambiare il menu corrente di softkey utilizzando la variabile **SKFMEN**.

Si ricorda altresì che il PLC ha a sua disposizione il codice della lingua attivata sul CN sulla variabile **SFKLNG** :

1=	Italiano
2=	Francese
3=	Tedesco
4=	Inglese
5=	Spagnolo
6=	Portoghese

E' inoltre disponibile appositamente per creare dei condizionamenti nelle tabelle di configurazione del video l' array **CNDVIS()** di 64 elementi in formato word (per l' utilizzo vedere il *Manuale di Configurazione Sistema*).

Riassunto segnali e registri interessati

SFKMEN	8	CN	⇔	PLC	no	Menu corrente di softkey del PLC.
SFKLNG	16	CN	⇒	PLC	no	Codice lingua attiva sul CN.
CNDVIS()	16	CN	⇐	PLC	no	Array di word da utilizzare per i condizionamenti all' interno delle tabelle video (1..64).

FINESTRE PER LA VISUALIZZAZIONE SU VIDEO CN: QUADRO VIDEO ATTIVO

La variabile **VISMC** (di sola lettura) contiene il numero del quadro video (**VIS_MC**) attivo. I quadri da **VIS_MC_A** a **VIS_MC_F** forniscono i codici da 10 a 15 rispettivamente.

Nome	Dim	Direzione	Descrizione
VISMC	16	CN ⇒ PLC	Numero del quadro video attivo

1.12. DATA E ORA DEL SISTEMA

La data e l'ora del sistema sono disponibili (in cifre e per sola lettura) su un vettore di 6 elementi nel formato WORD (i secondi hanno una tolleranza di ± 1).

Riassunto segnali e registri interessati

DATE(1)	16	CN	⇒	PLC	no	Anno (ultime due cifre)
DATE(2)	16	CN	⇒	PLC	no	Mese
DATE(3)	16	CN	⇒	PLC	no	Giorno del mese
DATE(4)	16	CN	⇒	PLC	no	Ora (0-24)
DATE(5)	16	CN	⇒	PLC	no	Minuti
DATE(6)	16	CN	⇒	PLC	no	Secondi

1.13. SEGNALI PER LA COPIATURA E DIGITALIZZAZIONE DI SUPERFICI

Per remotare su di una console esterna i comandi relativi alle funzionalità legate alla copiatura e digitalizzazione di superfici già presenti nei menu del CN, il PLC può agire sulle variabili descritte di seguito:

1. Gestione movimenti e segnali d'interscambio

Riassunto segnali e registri interessati

COPIA 8 CN ⇔ PLC no Primo byte per gestione remotata comandi copiatura

Il significato dei singoli bit è il seguente:

COPIA(1)	1	CN ⇔ PLC no	= 0	seleziona la modalità di digitalizzazione continua, in cui i punti vengono memorizzati in funzione dei parametri impostati da programma in copiatura manuale.
			= 1	seleziona la modalità di digitalizzazione ad impulsi, in cui i punti vengono memorizzati solo in seguito ad un impulso (transizione da 0 a 1) sul bit COPIA(2) in copiatura manuale.
COPIA(2)	1	CN ⇔ PLC no		Impulso di digitalizzazione; vedere COPIA(1) .
COPIA(3)	1	CN ⇔ PLC no		Segnalazione di ciclo di copiatura attivo. Quando resettato da PLC significa fine ciclo. E' importante terminare un ciclo con digitalizzazione azzerando questo bit (o con l'apposita softkey già implementata nel CN) altrimenti gli ultimi punti digitalizzati non vengono memorizzati.
COPIA(4)	1	CN ⇔ PLC no		Impulso di STEP +.
COPIA(5)	1	CN ⇔ PLC no		Impulso di STEP -.
COPIA(6)	1	CN ⇔ PLC no		Impulso di STEP ed inversione della passata.
COPIA(7)	1	CN ⇔ PLC no		Asservimento di copiatura attiva.
COPIA(8)	1			Non assegnato

COPIA2 8 CN ⇔ PLC no Secondo byte per gestione remotata comandi copiatura.

Il significato dei singoli bit è il seguente:

COPIA2(1)	1	CN ⇔ PLC no		Impulso di passaggio in stato manuale.
COPIA2(2)	1	CN ⇔ PLC no		0 = digitalizzazione punti disabilitata. 1 = digitalizzazione punti abilitata
COPIA2(3)	1	CN ⇔ PLC no		Impulso di acquisizione offset tastatore.
COPIA2(4)	1	CN ⇔ PLC no		1 = blocca 1° asse della terna attiva (di copiatura). 0 = sblocca
COPIA2(5)	1	CN ⇔ PLC no		1 = blocca 2° asse della terna attiva (di copiatura). 0 = sblocca
COPIA2(6)	1	CN ⇔ PLC no		1 = blocca 3° asse della terna attiva (di copiatura). 0 = sblocca
COPIA2(7)	1	CN ⇔ PLC no		Impulso di inversione passata.
COPIA2(8)	1	CN ⇔ PLC no		0 = autoapprendimento superficie disabilitato 1 = autoapprendimento della superficie attivo.

COPIA3 8 CN ⇔ PLC no Terzo byte per gestione remotata comandi copiatura.

Il significato dei singoli bit è il seguente:

COPIA3(1)	1	CN ⇔ PLC no		Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione negativa asse 3.
COPIA3(2)	1	CN ⇔ PLC no		Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione negativa asse 2.
COPIA3(3)	1	CN ⇔ PLC no		Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione negativa asse 1.
COPIA3(4)	1	CN ⇔ PLC no		Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione positiva asse 3.

1. Gestione movimenti e segnali d'interscambio

COPIA3(5)	1	CN ⇐ PLC no	Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione positiva asse 2.
COPIA3(6)	1	CN ⇐ PLC no	Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione positiva asse 1.
COPIA3(7)			Riserva
COPIA3(8)			Riserva
COPIA4	8	CN ⇔ PLC no	Quarto byte per gestione remotata comandi copiatura.

Il significato dei singoli bit è il seguente:

COPIA4(1)	1	CN ⇐ PLC no	Impulso di arresto tentativo di ripresa contatto.
COPIA4(2)			Riserva
COPIA4(3)			Riserva
COPIA4(4)			Riserva
COPIA4(5)			Riserva
COPIA4(6)			Riserva
COPIA4(7)			Riserva
COPIA4(8)			Riserva
POCOP	64	CN ⇐ PLC no	Regolazione guadagno in copiatura manuale. Il valore ammesso può essere compreso tra 0 e 1 e moltiplica il guadagno dell'asservimento in copiatura manuale da 1 a 5. Ciò permette di regolare la velocità di movimento degli assi a parità di deflessione del tastatore da un minimo (non 0) ad un massimo.

SEGNALI PER LA COPIATURA E DIGITALIZZAZIONE: COPIA MANUALE ATTIVA

Il CN setta il bit 8 del byte COPIA per segnalare l'esecuzione attiva di un ciclo di scansione in modalità manuale.

Nome	Dim	Direzione	Descrizione
COPIA	8	CN ⇒ PLC	Primo byte per gestione remotata comandi copiatura COPIA(8) scansione di copiatura manuale attiva

1.13.1 REGISTRO DI STATO DEL TASTATORE DIGITALE DI COPIATURA

Nel caso venga utilizzato per la copiatura e digitalizzazione un tastatore digitale, è disponibile il registro **PBSTS** in cui i singoli bit hanno il seguente significato:

PBSTS(1)	non usato
PBSTS(2)	non usato
PBSTS(3)	non usato
PBSTS(4)	non usato
PBSTS(5)	=1 se i segnali elettrici del tastatore sono corretti =0 in caso contrario
PBSTS(6)	=0 se il tastatore è connesso e non in sovradefflessione (oltrecorsa meccanico) =1 in caso contrario
PBSTS(7)	non usato
PBSTS(8)	non usato

1. Gestione movimenti e segnali d'interscambio

Con il tastatore montato vengono automaticamente generate dal sistema le segnalazioni di errore in caso di anomalia sui segnali del registro **PBSTS** con il conseguente passaggio allo stato di emergenza (**EMEA=1**).

Il tastatore viene considerato presente dal CN quando viene rilevata la configurazione di **PBSTS(5)=1** e **PBSTS(6)=0** e viene considerato assente con **PBSTS(5)=0** e **PBSTS(6)=1**.

Riassunto segnali e registri interessati

PBSTS 8 CN \Rightarrow PLC no Registro di stato tastatore digitale.

1.14. VARIABILI DI VERIFICA DEI TEMPI DI UTILIZZO DELLE RISORSE

Per valutare il rendimento del sistema durante le varie operazioni sono a disposizione le variabili elencate nel riassunto che segue:

Riassunto segnali e registri interessati

SMPTI	64	CN \Rightarrow PLC no	Intervallo di campionatura sistema (assi controllati) [msec]
OCCV	16	CN \Rightarrow PLC no	Tempo in microsecondi impiegato per la scansione della logica veloce.
OCCI	16	CN \Rightarrow PLC no	Tempo in microsecondi impiegato per la gestione degli assi controllati.
OCCT	16	CN \Rightarrow PLC no	Tempo in microsecondi impiegato dall'analizzatore grafico.
OCCP2P	16	CN \Rightarrow PLC no	Tempo in microsecondi impiegato per la gestione degli assi indipendenti.
CCL	16	CN \Rightarrow PLC no	Contatore dei cicli in cui viene spezzata la logica lenta.
CCUL	16	CN \Rightarrow PLC no	Contatore cicli in cui viene spezzata la logica superlenta.

1.15. SEGNALAZIONI DI ERRORE ACCESSIBILI DALLA LOGICA

Gli errori di sistema, oltre ad essere segnalati con un chiaro messaggio sul video, vengono comunicati con un codice numerico al PLC sulla variabile **ERSYS**; lo stesso codice viene poi scritto a seconda del tipo su una delle variabili elencate nel riassunto che segue.

La lista completa degli errori è riportata sul manuale *Uso e Programmazione*.

Riassunto segnali e registri interessati

ERSYS	16	CN	⇒	PLC no	Codice dell'errore di sistema rilevato su assi controllati, mandrini, assi indipendenti, errori runtime PLC, errori su modulo cambio utensile automatico, ecc.
ERAXS	16	CN	⇒	PLC no	Codice dell'errore rilevato su assi controllati (servo error, fuori tolleranza, errori su trasduttori, ecc.).
ERIOX	16	CN	⇒	PLC no	Codice dell'errore rilevato sulle schede di I/OMIX (mancanza alimentazioni encoder, errore su output digitali, ecc.).
ERINT	16	CN	⇒	PLC no	Codice dell'errore avvenuto durante i calcoli di interpolazione.
ERPLC	16	CN	⇒	PLC no	Codice dell'errore runtime rilevato durante l'esecuzione del programma PLC (divisione per 0, overflow, underflow, ecc.).
ERSPN	16	CN	⇒	PLC no	Codice dell'errore rilevato sui mandrini (trasduttori, ecc.).
ERP2P	16	CN	⇒	PLC no	Codice dell'errore rilevato su assi indipendenti (trasduttori, ecc.).
ERCU	16	CN	⇒	PLC no	Codice dell'errore rilevato durante il cambio utensile oppure tabella utensili non corretta, ecc.
ER2LN	16	CN	⇒	PLC no	Codice dell'errore rilevato per superamento del tempo di campionatura di sistema.
ERCPY	16	CN	⇒	PLC no	Codice dell'errore rilevato durante un ciclo di copiatura o relativo al tastatore.
FPERMK	8	CN	↔	PLC no	Maschera di disabilitazione rilevazioni errori sui calcoli floating point (divisione per zero, overflow).

CONTROLLO DEGLI INDICI PER L'ACCESSO A VARIABILI E TABELLE

Con lo scopo di diagnosticare se il valore degli indici utilizzati per accedere ai singoli bit di variabili semplici o agli elementi di un vettore rientri nei limiti delle dimensioni delle variabili stesse, è possibile aggiungere all'interno del programma PLC le istruzioni:

```

_ENIDX      = -1    per attivare la diagnostica
_ENIDX      = 0     per disattivarla (default)

```

Il controllo può venire attivato e disattivato più volte all'interno del programma PLC (solo in una sezione di programma alla volta).

L'esecuzione del programma PLC con attivi questi controlli risulta essere rallentata.

Nel caso sia rilevata una situazione di errore viene segnalato un messaggio in chiaro e viene disabilitato il PLC.

1.16. LETTURA E MODIFICA DEI PARAMETRI DI CONFIGURAZIONE ASSI CN

Il PLC, per poter implementare sofisticate tecniche di autotaratura degli asservimenti, ha la possibilità di leggere e modificare temporaneamente o definitivamente alcuni parametri di caratterizzazione degli assi CN controllati, normalmente definiti nei dati di configurazione.

Particolare attenzione va posta nell'uso di questa prestazione, in quanto dati non corretti comportano un funzionamento anomalo degli assi.

1. Gestione movimenti e segnali d'interscambio

Per accedere a questi parametri, occorre innanzitutto selezionare l'asse CN con cui si vuole operare fornendo sul registro **AXSTP** il numero d'ordine con cui è dichiarato nei dati di configurazione, quindi con il registro **HOWSTP** si selezionerà il parametro ed il tipo di accesso (lettura o scrittura).

Per effettuare l'operazione selezionata occorre attivare lo strobe **ACTSTP**; quando questa è conclusa, il CN risponde resettandolo.

Il valore da scrivere o leggere, per la grandezza selezionata, deve essere scritto o letto sul registro **VALSTP**.

Le modifiche ai parametri sono memorizzate permanentemente solo con l'operazione esplicita di SALVATAGGIO DATI PERMANENTE (**HOWSTP** = 0) a condizione che l'applicazione non sia protetta nella memoria flash.

Riassunto segnali e registri interessati

AXSTP	8	CN ⇐ PLC no	Numero dell'asse a cui sono riferiti i valori letti o scritti sui parametri di configurazione per mezzo del PLC.
VALSTP	64	CN ⇔ PLC no	Valore corrente letto o da scrivere nei parametri di configurazione sistema.
HOWSTP	16	CN ⇐ PLC no	Codice del parametro di configurazione a cui si vuole accedere per mezzo del PLC (finché non si scrive su disco, i parametri agiscono su una copia in memoria non statica); i nuovi valori vengono attuati unicamente quando viene incontrato un punto con velocità finale = 0:

Codice scrittura	Parametro	Codice lettura
0	Salvataggio dati permanente	
-1	velocità rapido	1
-2	accelerazione lavoro	2
-3	accelerazione di rapido	3
-4	gioco asse trasduttore	4
-5	K _v guadagno	5
-6	Compensazione dinamica	6
-7	Velocità ripresa gioco	7
-8	Tempo ripresa gioco	8
-9	Servo error massimo	9
-10	Velocità compensazione attrito	10
-11	Compensazione accelerazione	11
-12	Fine corsa negativo 1	12
-13	Fine corsa positivo 1	13
-14	Passo trasduttore	14
-15	Costante tempo integratore	15
-16	Guadagno integratore	16

ACTSTP	1	CN ⇔ PLC no	Segnale di attivazione operazione richiesta su HOWSTP . Viene resettato dal CN ad operazione conclusa.
INCH	1	CN ⇔ PLC no	Tipo del sistema di misura 0 = millimetri 1 = pollici Questa variabile viene settata dal CN in funzione dell'apposito parametro nella configurazione impianto. Il PLC può sovrascriverla alterando il sistema di misura ma la scelta non può venire salvata in modo permanente nei parametri di configurazione.

1.17. GESTIONE DI PIÙ GRUPPI D'ASSI (GDA) INTERPOLANTI SIMULTANEAMENTE

Previa la dichiarazione nei parametri di Setup del CNC, è possibile configurare fino ad 8 gruppi d'assi interpolanti, ciascuno in grado di eseguire un programma o parti di programma in modo completamente indipendente.

Sono di conseguenza anche state modificate le variabili PLC di colloquio con il programma di lavorazione.


Le regole utilizzate per generare i nuovi nomi di variabile sono le seguenti:

- per le variabili in formato bit è stato creato un byte in cui ogni bit corrisponde ad un gruppo d'assi;

Esempio: il bit BURDY viene esteso nel byte BURDY_
 indirizzare BURDY o BURDY_ (1) è la stessa cosa.
 Per i GDA superiori al primo occorre usare BURDY_ (..).

- per le variabili con altri formati sono stati creati degli array di 8 elementi (uno per ogni gruppo d'asse).

- il nome delle nuove variabili è ottenuto aggiungendo il carattere "_" (underscore) dopo il nome originario.

A livello di interfaccia utente il tasto posto al di sopra del  permette di passare (se configurati) dalla visualizzazione della videata di un gruppo d'assi al successivo.

Per la sincronizzazione ed il lancio di programmi su gruppi d'assi differenti sono state introdotte nuove istruzioni di part-program.

Per ulteriori dettagli fare riferimento al relativo *Notiziario Tecnico numero 4 del 1997*.

INFORMAZIONE SUL GRUPPO D'ASSI VISUALIZZATO

La variabile **GDAVIS** comunica a quale gruppo d'assi è riferita la visualizzazione della console.

Questa informazione serve per esempio in quanto è compito del PLC tenere aggiornata la visualizzazione dell'ultima M programmata per ogni gruppo d'asse, in funzione di quale GDA viene visualizzato sulla console dall'utente.

Nome	Dim	Direzione	Descrizione
GDAVIS	8	CN ⇒ PLC	Numero del gruppo d'assi al quale si riferisce la Visualizzazione

1.18. GESTIONE DEI DRIVE DIGITALI PER L'AVANZAMENTO ASSI ED IL MANDRINO

Con l'introduzione dell'interfacciamento dei drive digitali per assi e mandrini, molti dei segnali di colloquio tradizionalmente gestiti come input/output del PLC e degli azionamenti, sono entrati a far parte dei segnali registri di interfaccia.

La descrizione delle variabili PLC, e del loro funzionamento non è riportata sul presente manuale per motivi di corposità dell'argomento; si rimanda pertanto al *"Manuale di Installazione Schede di Regolazione DCMA"*.

2. MODULI INTERNI DEDICATI

Per semplificare la gestione di organi complessi usati frequentemente, è possibile sfruttare dei MODULI INTERNI nei quali, settando alcune variabili si ottengono gli effetti desiderati senza dover implementare complicati algoritmi. In questo modo si può ottenere un programma più leggibile, con tempi di sviluppo ridotti.

2.1. MODULO PER LA GESTIONE DEI MANDRINI

I mandrini utilizzabili possono essere un massimo di 4 con o senza trasduttore. Sono gestibili direttamente per mezzo di un set ridotto di registri predefiniti appartenenti al **MODULO INTERNO DI GESTIONE MANDRINI**.

Le funzioni svolte sono:

- rampe di accelerazione/decelerazione
- regolazione di velocità in base alla gamma inserita ed al valore del potenziometro
- orientamento, nel rispetto delle accelerazioni dichiarate, su posizione programmabile rispetto alla posizione di zero assoluta (anch'essa soggetta ad offset sui parametri di configurazione)
- pendolamento per cambio gamma
- sincronismo di più mandrini (slave) con un mandrino principale (master)
- limitazione per velocità maggiori di quella di gamma.

I registri a disposizione per la gestione sono tutti asincroni (cioè svincolati dai blocchi di programma e da **BURDY**).

Ogni registro deve essere utilizzato con l'indice relativo al mandrino a cui è riferito (nel caso di registri di n bit si andrà ad attivare un singolo bit all'interno del registro).

Tutti i parametri relativi ai vari mandrini (velocità nelle varie gamme, accelerazioni, tipi di trasduttori, soglie) devono essere inseriti nei dati di configurazione sistema (vedere documentazione relativa).

2.1.1. SEGNALI E REGISTRI PER LA ROTAZIONE MANDRINO

SPVEL() (Impostazione velocità di rotazione). Su questo registro deve essere scritta la velocità da raggiungere in giri/min. Essa viene automaticamente limitata se il valore impostato è maggiore del massimo ammesso per la gamma inserita.

2. Moduli interni dedicati

- SPSSO()** (Valore del potenziometro override). Scrivendo un valore compreso tra 0 e 2 su questo registro si regola la velocità tra lo 0 e il 200% della velocità impostata (sempre rispettando il massimo ammesso per la gamma inserita).
- SPDIR()** (Senso di rotazione mandrino). Dopo un comando di rotazione, se questo segnale è uguale a 0, il riferimento analogico di uscita sarà positivo; se uguale ad 1 il riferimento sarà negativo.
- SPROT** (Comando di rotazione). Agendo sui primi 4 bit (uno per mandrino) di questo registro di formato byte si fornisce il comando di start/stop rotazione mandrini con i parametri impostati.
- SPREG** (Velocità di regime raggiunta). I primi 4 bit di questo registro (uno per mandrino) sono attivati dal CN quando, dopo un comando di rotazione e avendo terminato la rampa di accelerazione teorica, la velocità effettiva del mandrino corrispondente ha raggiunto la percentuale specificata nei dati di configurazione rispetto alla velocità impostata. Se la velocità richiesta è minore della soglia minima di mandrino in moto dichiarata nei dati di configurazione, questo segnale è sempre a 1.
- SPMOT** (Mandrino in moto). I primi 4 bit di questo registro (uno per mandrino) sono attivati dal CN quando la velocità del mandrino corrispondente ha superato la soglia specificata nei dati di configurazione. Questa segnalazione è sempre aggiornata, anche se il mandrino non è comandato.
- SPRMP** (Mandrini in rampa). I primi 4 bit di questo registro (uno per mandrino) sono attivati dal CN in fase di accelerazione o decelerazione. Tipicamente sarà utilizzato per generare l'attesa su partenza e arresto mandrino.
- SPSGL** (Velocità effettiva in soglia). I primi 4 bit di questo registro (uno per mandrino) sono attivati dal CN quando la velocità effettiva del mandrino corrispondente, rilevata dal trasduttore, è nella soglia di tolleranza rispetto alla velocità prescritta. Quando il mandrino è fermo è sempre 0; per velocità inferiori alla soglia minima di mandrino in moto è sempre attivo.

2.1.2. SEGNALI E REGISTRI PER LA SELEZIONE E CAMBIO GAMMA

- SPGAM()** (Numero della gamma inserita). Scrivendo un valore da 1 a 4 su questo registro si attivano i parametri inseriti nei dati di configurazione e associati alla gamma corrispondente. Con **SPGAM(n)** = 0 si identifica la folle, cioè il riferimento viene comunque sempre forzato a 0 V, qualunque sia la velocità di rotazione impostata.
- SPPND** (Comando di pendolamento). Questo registro, sui primi 4 bit (uno per mandrino), fornisce la richiesta di attuare il pendolamento del riferimento mandrino corrispondente con i dati specificati nei parametri macchina.

Sui registri sottoelencati sono disponibili, per la sola lettura, i valori inseriti in configurazione relativi alle quattro gamme. L'utilizzo più comune risulta essere la determinazione della gamma fisica da inserire nel caso di un cambio automatico.

SPSMG1() Massime velocità in gamma 1 per i mandrini (1..4)

SPSMG2() Massime velocità in gamma 2 per i mandrini (1..4)

SPSMG3() Massime velocità in gamma 3 per i mandrini (1..4)

SPSMG4() Massime velocità in gamma 4 per i mandrini (1..4)

SPSMAX() Massime velocità assolute per i mandrini (1..4)

2.1.3. SEGNALI E REGISTRI PER L'ORIENTAMENTO MANDRINO

SPORI (Richieste di orientamento). Settando i primi 4 bit (uno per mandrino) di questo registro si fornisce la richiesta di orientamento alla posizione impostata su **SPPOS** secondo le modalità descritte di seguito. Nel caso il trasduttore non sia ancora stato riferito allo zero elettrico, viene automaticamente effettuato un ciclo di azzeramento.

SPTOL (Mandrino orientato in tolleranza di posizionamento). I primi 4 bit di questo registro (uno per mandrino) sono attivati dal CN in presenza di un comando di orientamento per il mandrino corrispondente, quando la posizione mandrino dista dalla posizione di orientamento meno della "tolleranza di posizionamento" dichiarata nei parametri di configurazione. Per evitare di considerare il mandrino orientato quando la tolleranza è raggiunta e poi oltrepassata, è bene non resettare immediatamente da PLC il comando di orientamento, ma mantenerlo finché il segnale **SPTOL** non si è stabilizzato.

SPPOS() (Posizioni di orientamento). Su questo registro, si deve fornire la posizione su cui deve essere orientato il mandrino. Occorre ricordare che il passo del trasduttore mandrino è 1 per definizione; i valori ammessi sono compresi tra 0 e 1.

Esempio: $SPPOS(1)=(NGRADI // 360)/360$

SPVEOR() (Limitazione velocità in orientamento). Il numero fornito su questo registro permette di limitare la velocità di rotazione durante l'orientamento. La velocità limite sarà data da:

$(1-SPVEOR) \times SPSMGx$. (SPVEOR = 0 non comporta quindi nessuna riduzione).

Orientamento su posizione assoluta

SPOAB (Selezione per orientamento su quota assoluta). Se viene settato questo bit (bit 1-4 della variabile, per mandrini 1-4) il valore caricato su **SPPOS()**, cioè la posizione su cui orientare, viene interpretato come valore assoluto da raggiungere (compresi i giri).

2. Moduli interni dedicati

Orientamento unidirezionale

Per attivare l'orientamento unidirezionale occorre settare il bit corrispondente al mandrino in questione su uno dei due registri di direzione **SPORP** o **SPORM**, caricare l'**SPPOS()** desiderato, quindi *solo successivamente* attivare l'orientamento settando **SPORI**.

SPORP Orientamento in direzione positiva

SPORM Orientamento in direzione positiva.

2.1.4. SEGNALI E REGISTRI PER IL SINCRONISMO FRA MANDRINI

SPSYN (Richieste di sincronismo mandrino con slave). Con i primi 4 bit (uno per mandrino) di questo registro si richiede di sincronizzare il mandrino **n** con quello dichiarato come master in **SPMAS(n)**.

La richiesta di sincronismo **SPSYN** può avvenire in qualsiasi istante; il mandrino slave (cioè quello secondario), partendo anche da fermo adatterà la sua velocità in funzione della velocità del master (principale) e del rapporto di velocità (**SPRTO**) quindi manterrà il sincronismo con l'offset **SPOFS** specificato. Quest'ultimo ha senso solo se il rapporto di velocità per il sincronismo è un numero intero.

Tutti i parametri relativi al mandrino slave da sincronizzare devono essere impostati con mandrino slave fermo.

Se ad un mandrino slave sincronizzato viene dato un ulteriore comando (**SPROT**, **SPORI**, ...), esso viene automaticamente sganciato dal sincronismo.

SPMAS() (Numeri mandrino master). Nel registro corrispondente al mandrino si imposta il numero del master con cui sincronizzarsi.

SPOFS() (Offset per sincronismo). Sono i registri in cui scrivere l'angolo di sfasamento tra mandrino master e slave (1=360 gradi) da mantenere durante il sincronismo. Ha senso solo se il rapporto di sincronismo **SPRTO** è un numero intero.

SPRTO() (Rapporti di velocità per sincronismo). Sono i registri in cui scrivere il rapporto tra la velocità del mandrino slave diviso quella del master da mantenere durante il sincronismo. (Vel. slave/Vel. master).

SPAGG (Mandrino slave sincronizzato con master). I primi 4 bit di questo registro (uno per mandrino) sono attivati dal CN quando, in seguito ad un comando di sincronismo, è presente la condizione di mandrino sincronizzato.

2.1.5. SEGNALI E REGISTRI COMUNI A TUTTE LE MODALITA' MANDRINO

Per alcuni dei comandi descritti precedentemente, relativi alle varie modalità di funzionamento, esistono delle priorità di esecuzione. Questo vale in particolare per i seguenti segnali, elencati in ordine di priorità:

1. **SPPND** (pendolamento) *priorità più alta*
2. **SPROT** (rotazione)
3. **SPORI** (orientamento)
4. **SPSYN** (sincronismo con slave) *priorità più bassa*

I registri e segnali comuni a tutti i modi di funzionamento sono i seguenti:

SPMOV (Abilitazione azionamento mandrino). E' il comando di abilitazione coppia mandrino per pilotare l'azionamento, fornito dal CN sui primi 4 bit (uno per mandrino) di questo registro; viene automaticamente mantenuto finché il mandrino non è fermo durante i cicli di rotazione o sincronismo e quando sono presenti i comandi di orientamento o pendolamento. Eventuali ulteriori protezioni o prolungamenti nel tempo dell'abilitazione mandrino devono essere implementati nel PLC.

Nota: La scrittura sul canale di riferimento analogico associato ad un mandrino è possibile solo in assenza di **SPMOV** e con il segnale **SPDIS** attivato.

SPDIS (Disabilitazione mandrino). E' il comando con cui il PLC richiede il rilascio del mandrino relativo (cioè il riferimento viene forzato a 0 V e il mandrino è istantaneamente disabilitato) sui primi 4 bit (uno per mandrino) di questo registro. Questo segnale può essere usato in caso di emergenza.

SPDRQ (Disabilitazione del trasduttore mandrino). Questo comando richiede che venga disabilitato il trasduttore del mandrino corrispondente ad uno dei primi 4 bit del registro. Da quel momento non viene più letta la quota, non vengono più rilevati gli errori trasduttore, viene perso l'azzeramento trasduttore (**SPMZA**).

SPTCH() (Velocità mandrino effettiva). Su ogni registro si legge direttamente in giri/min, la velocità del mandrino relativo, rilevata dal trasduttore. Nel caso non sia presente il trasduttore, questa variabile è la velocità calcolata per cui viene inviato il riferimento.

PASP() (Posizione angolare assoluta del mandrino). Dato che il passo del mandrino è 1 il trasduttore dovrà sempre essere in rapporto meccanico 1:1 con il mandrino relativo. La ciclicità di questo registro è +131071.9999, -131071.9999.

SPMZA() (Azzeramento trasduttore mandrino eseguito). Il bit corrispondente al mandrino interessato è attivo quando il relativo trasduttore è stato azzerato sullo zero elettrico. La ricerca zero è automaticamente effettuata sul primo orientamento o richiesta di sincronismo.

Nel caso in cui si voglia ripetere il ciclo di azzeramento del trasduttore è sufficiente resettare il bit corrispondente al mandrino desiderato su **SPMZA**.

SPMKS (Marker di zero). Questo segnale è settato dal fronte di salita del segnale di zero elettrico per i trasduttori dei mandrini costituiti da encoder. E' di tipo impulsivo con la durata pari ad una campionatura del sistema. Utilizzo tipico è la verifica del funzionamento del trasduttore.

2. Moduli interni dedicati

NUOVE VARIABILI

Variabili per debug e taratura:

Nome	Dim	Direzione	Descrizione
SPRIF()	64	CN \Rightarrow PLC	Comando di velocità inviato ai mandrini (1..4) [giri/min] può essere utilizzato per la verifica delle rampe di accelerazione/decelerazione paragonando SPRIF con SPTCH (velocità effettiva) per mandrini con trasduttore.

Aggiunta variabile **SPAGP** da utilizzare per diagnostica che assume i seguenti significati in funzione del tipo di trasduttore mandrino utilizzato:

Con RESOLVER rappresenta il livello del segnale analogico del trasduttore.

Con ENCODER rappresenta il numero di impulsi persi e recuperati (con il parametro **RECUPERO STEP ATTIVO**).

Nome	Dim	Direzione	Descrizione
SPAGP()	8	CN \Rightarrow PLC	Livello trasduttore o impulsi persi e recuperati per i mandrini (1..4).

2.1.6. MANDRINI CON O SENZA TRASDUTTORE

Se il mandrino non ha trasduttore, **SPTCH** è la velocità calcolata per cui viene inviato il riferimento, **SPSGL** sarà sempre a 1, mentre **SPREG**, **SPMOT** e **SPRMP** sono attivi ma rapportati al riferimento impostato (velocità prescritta) anziché alla velocità effettiva.

In questo caso non è evidentemente effettuabile la sincronizzazione con altri mandrini.

*Nel caso di mandrino con trasduttore, affinché i vari cicli funzionino correttamente è assolutamente necessario che il senso di conteggio positivo, rilevato dal trasduttore (**PASP**), corrisponda con l'invio di un riferimento analogico positivo.*

Un giro del trasduttore deve comunque e sempre corrispondere ad un giro del mandrino per il corretto funzionamento dei cicli di orientamento e, sui torni in special modo, per tutte quelle funzioni che necessitano di conoscere la velocità effettiva del mandrino.

2.1.7. NOTE SUL CICLO FISSO G84

Nel caso di ciclo fisso G84 con trasduttore presente occorre specificare sulla variabile **SPGDA** quale dei quattro mandrini possibili si deve sincronizzare con l'asse mandrino.

Se il ciclo fisso inizia ma non prosegue, occorre verificare che il trasduttore sia azzerato, cioè **SPMZA** sia settato (si azzerava automaticamente) e che la velocità reale abbia raggiunto il valore nominale (**SPREG**).

Eventuali **FHOLD**, **DHOLD** o **RHOLD** sono gestiti unicamente a fine del ciclo fisso corrente.

Dichiarando nei parametri di configurazione "numero mandrini = 0" il CN emette le funzioni M3 ed M4 automaticamente per invertire il senso di rotazione mandrino a inizio e fine foro (mandrino "in alternata").

Riassunto segnali e registri interessati

Rotazione mandrino

SPVEL()	64	CN	⇐	PLC	no	Velocità di rotazione mandrini (1..4).
SPSSO()	64	CN	⇐	PLC	no	Potenzimetro override mandrini (1..4).
SPDIR()	8	CN	⇐	PLC	no	Sensi di rotazione mandrini (1..4).
SPROT	8	CN	⇐	PLC	no	Comandi di rotazione mandrini (1..4).
SPREG	8	CN	⇒	PLC	no	Mandrini (1..4) a regime.
SPMOT	8	CN	⇒	PLC	no	Mandrini (1..4) in moto.
SPRMP	8	CN	⇒	PLC	no	Mandrini (1..4) in rampa.
SPSGL	8	CN	⇒	PLC	no	Velocità effettive mandrini (1..4) entro soglia.

Selezione del cambio gamma

SPGAM()	8	CN	⇐	PLC	no	Gamme mandrini inserite (0 = folle) (1..4).
SPPND	8	CN	⇐	PLC	no	Comandi di pendolamento per cambio gamma mandrini (1..4).
SPSMG1()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) per la gamma 1, impostate nella configurazione.
SPSMG2()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) per la gamma 2, impostate nella configurazione.
SPSMG3()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) per la gamma 3, impostate nella configurazione.
SPSMG4()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) per la gamma 4, impostate nella configurazione.
SPSMAX()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) impostate nella configurazione.

Orientamento mandrino

SPORI()	8	CN	⇐	PLC	no	Comandi di orientamento mandrini (1..4).
SPTOL	8	CN	⇒	PLC	no	Mandrini orientati all'interno della tolleranza di posizionamento (1..4).
SPPOS()	64	CN	⇐	PLC	no	Posizioni di orientamento mandrini (1..4).
SPVEOR()	64	CN	⇐	PLC	no	Riduzione di velocità (da 0 a 1) durante l'orientamento (1..4).
SPOAB	8	CN	⇐	PLC	no	Selezioni per orientamento su quota assoluta (1..4).
SPORP	8	CN	⇐	PLC	no	Selezione di orientamento unidirezionale positivo.
SPORM	8	CN	⇐	PLC	no	Selezione di orientamento unidirezionale negativo.

Sincronismo tra mandrini

SPSYN	8	CN	⇐	PLC	no	Comandi di sincronismo mandrini (1..4) con mandrini slave.
SPMAS()	8	CN	⇐	PLC	no	Numero dei mandrini master per sincronismo con slave (1..4).
SPOFS()	64	CN	⇐	PLC	no	Offset per sincronismo fra mandrini master e slave (1..4).
SPRTO()	64	CN	⇐	PLC	no	Rapporti di velocità per sincronismo fra mandrini master e slave (1..4).
SPAGG	8	CN	⇒	PLC	no	Mandrini slave (1..4) sincronizzati con master.

Comuni a tutti i modi di funzionamento

SPMOV	8	CN	⇒	PLC	no	Richieste di abilitazione azionamento mandrini (1..4).
SPDIS	8	CN	⇐	PLC	no	Disabilitazioni generali azionamento mandrini (1..4).
SPDRQ	8	CN	⇐	PLC	no	Disabilitazioni del trasduttore mandrini (1..4).
SPTCH()	64	CN	⇒	PLC	no	Velocità effettiva mandrini (1..4).
PASP()	64	CN	⇒	PLC	no	Posizioni angolari mandrini lette dai trasduttori (1..4).

2. Moduli interni dedicati

SPMZA	8	CN ⇔ PLC no	Mandrini (1..4) con trasduttore riferito allo zero elettrico. Può essere resettato per ripetere la ricerca zero.
SPMKS	8	CN ⇔ PLC no	Segnalazione del marker (zero elettrico) per gli encoder mandrini (1..4).

Ciclo fisso G84

SPGDA	8	CN ⇔ PLC no	Mandrino da utilizzare per il ciclo fisso G84 con trasduttore.
--------------	---	-------------	--

2.2. MODULO DI MOVIMENTAZIONE ASSI INDIPENDENTI

In tutti quei casi in cui è necessario posizionare un asse di servizio, cioè indipendente dagli assi interpolati del CN (cambi utensile, cambi pallets, assi di posizionamento), si dovrà utilizzare il **MODULO INTERNO DI MOVIMENTAZIONE ASSI INDIPENDENTI**. Esso consiste in un algoritmo di posizionamento di tipo punto-punto, interfacciabile con una logica minima nel programma di logica di macchina, per un massimo di 8 assi.

Lettura trasduttori e aggiornamento del riferimento vengono effettuati per questo tipo di asse ogni 10 msec o più secondo quanto specificato nei parametri di configurazione.

I parametri di impianto relativi alla caratterizzazione di questi moduli devono essere inseriti nei dati di configurazione come per ogni altro asse controllato della macchina; in aggiunta, parte di questi dati possono essere letti e riscritti temporaneamente tramite registri del PLC.

I registri a disposizione per la gestione sono tutti asincroni (cioè non vincolati ai blocchi di programma ed al **BURDY**) con funzionamento analogo a quello degli assi controllati.

Ogni registro deve essere utilizzato con l'indice relativo all'asse indipendente al quale è riferito.

Riassunto segnali e registri interessati

MOVP2P	8	CN ⇔ PLC no	Richieste di abilitazione al movimento per assi indipendenti (1..8).
RDMP2P	8	CN ⇔ PLC no	Assi indipendenti abilitati al movimento(1..8); risposte al MOVP2P .
SSAP2P	8	CN ⇔ PLC no	Assi indipendenti che devono essere sempre attivi (abilitati) (1..8).
DSVP2P	8	CN ⇔ PLC no	Assi indipendenti da rilasciare (1..8).
DRQP2P	8	CN ⇔ PLC no	Assi indipendenti di cui disabilitare i trasduttori (1..8).
MVMP2P	8	CN ⇔ PLC no	Assi indipendenti selezionati in modo manuale (1..8).
MRKP2P	8	CN ⇔ PLC no	Assi indipendenti selezionati in ricerca zero senza microinterruttore (1..8).
MCZP2P	8	CN ⇔ PLC no	Assi indipendenti selezionati in ricerca zero con microinterruttore (1..8).
MIZP2P	8	CN ⇔ PLC no	Microinterruttore di zero per assi indipendenti (1..8).
MZAP2P	8	CN ⇔ PLC no	Assi riferiti allo zero elettrico del trasduttore e posizionati alla quota di riposizionamento dopo ricerca 0 (1..8).
POTP2P()	64	CN ⇔ PLC no	Potenzimetri per la regolazione di velocità assi indipendenti (1..8). Da 0 a 1 come percentuale della velocità di movimento impostata, se in automatico o della velocità di rapido, se in manuale.
JGPP2P	8	CN ⇔ PLC no	Comandi di JOG positivo per assi indipendenti (1..8).
JGMP2P	8	CN ⇔ PLC no	Comandi di JOG negativo per assi indipendenti (1..8).

PFNP2P()	64	CN ⇌ PLC no	Posizioni programmate da raggiungere in automatico per assi indipendenti (1..8).
RUNP2P	8	CN ⇌ PLC no	Comandi di posizionamento in automatico alla quota programmata per assi indipendenti. Devono essere settati dal PLC per comandare il movimento alla posizione prefissata; vengono resettati dal CN quando l'asse, avendo terminato il movimento, entra nella soglia di posizionamento inserita nei dati di configurazione (1..8).
RHDP2P	8	CN ⇌ PLC no	Richieste di HOLD su assi indipendenti (1..8). Arresti temporanei del movimento in esecuzione; l'esecuzione riprende senza ulteriori comandi non appena vengono rilasciati.
HDAP2P	8	CN ⇌ PLC no	Richieste di HOLD su assi indipendenti (1..8). Arresti temporanei del movimento in esecuzione; l'esecuzione riprende senza ulteriori comandi non appena vengono rilasciati.
RBKP2P	8	CN ⇌ PLC no	Richieste di BREAK su movimento in automatico per assi indipendenti (1..8). RBKP2P vengono resettati dal CN quando acquisiti. Gli assi vengono fatti decelerare fino a fermarsi, quindi vengono resettati anche i RUNP2P eventuali. In stato di emergenza (EMAP2P) servono a cancellare lo stato di emergenza, sempre che la richiesta sia stata rimossa (REMP2P).
BKAP2P	8	CN ⇌ PLC no	Assi indipendenti fermi in seguito ad un comando di RBKP2P (1..8); possono essere resettati dal PLC, ma ciò non è vincolante.
REMP2P	8	CN ⇌ PLC no	Richieste di passaggio in stato di emergenza per assi indipendenti (1..8).
EMAP2P	8	CN ⇌ PLC no	Assi indipendenti in stato di emergenza. Nel passare in questo stato, gli assi vengono istantaneamente disabilitati senza alcuna decelerazione controllata (1..8).
POAP2P()	64	CN ⇌ PLC no	Posizioni assolute lette dai trasduttori per assi indipendenti (1..8).
TCHP2P()	64	CN ⇌ PLC no	Velocità effettive (rilevate dai trasduttori) per assi indipendenti (1..8).
SGLP2P	8	CN ⇌ PLC no	Assi indipendenti (1..8) entro tolleranza di posizionamento asse dichiarata nella configurazione.
MKSP2P	8	CN ⇌ PLC no	Segnalazione del marker (zero elettrico) degli assi indipendenti nel caso di trasduttori encoder o riga ottica (1..8).
FCPP2P	8	CN ⇌ PLC no	Assi indipendenti (1..8) la cui quota attuale risulta essere maggiore del finecorsa software positivo dichiarato nella configurazione.
FCMP2P	8	CN ⇌ PLC no	Assi indipendenti (1..8) la cui quota attuale risulta essere minore del finecorsa software negativo dichiarato nella configurazione.
VATP2P()	64	CN ⇌ PLC no	Velocità teoriche (calcolate) per assi indipendenti (1..8). Se nei dati di configurazione viene dichiarato "convertitore D/A assente" il riferimento in tensione non viene inviato sul canale di uscita ma la velocità su questo registro è sempre disponibile.
JINP2P	8	CN ⇌ PLC no	Assi (1..8) in movimento in seguito a movimento in JOGP2P .
DIRP2P	8	CN ⇌ PLC no	Direzioni del movimento comandato (rilevate dal segno del riferimento analogico) per assi indipendenti (1..8). Il valore 1 significa velocità negativa.

I registri seguenti sono inizializzati all'accensione del CN con i valori immessi nei dati di configurazione, dopodiché il PLC ha la facoltà di leggerli e modificarli, quando l'asse corrispondente è fermo.

2. Moduli interni dedicati

FEDP2P()	64	CN ⇔ PLC	no	Velocità di movimento per assi indipendenti (1..8).
RAPP2P()	64	CN ⇔ PLC	no	Velocità di rapido per assi indipendenti (1..8).
VLNP2P()	64	CN ⇔ PLC	no	Velocità nella "zona lenta" per assi indipendenti (1..8).
ZLNP2P()	64	CN ⇔ PLC	no	Distanze nella "zona lenta" per assi indipendenti (1..8).
DEXP2P()	64	CN ⇔ PLC	no	Distanze per decelerazioni esponenziali per assi indipendenti (1..8).
ACMP2P()	64	CN ⇔ PLC	no	Accelerazioni in movimento manuale per assi indipendenti (1..8).
ACCP2P()	64	CN ⇔ PLC	no	Accelerazioni in movimento automatico per assi indipendenti (1..8).
DECP2P()	64	CN ⇔ PLC	no	Decelerazione da velocità di movimento a velocità lenta per assi indipendenti (1..8).
DE2P2P()	64	CN ⇔ PLC	no	Decelerazione da velocità di lento a decelerazione esponenziale per assi indipendenti (1..8).
TOLP2P()	64	CN ⇔ PLC	no	Tolleranze di posizionamento per assi indipendenti (1..8).
OFSP2P()	64	CN ⇔ PLC	no	Offset trasduttore che viene applicato sulla quota letta per ottenere la quota assoluta POAP2P() (1..8).

NUOVE VARIABILI

Variabili per debug e taratura assi:

Nome	Dim	Direzione	Descrizione
SHIP2P()	64	PLC ⇒ CN	Shift di origine per assi indipendenti (1..8). Permette di definire una quota zero diversa dalla quota di zero assoluta. Le posizioni finali PFNP2P() sono sempre riferite alla POOP2P() .
POOP2P()	64	PLC ⇒ CN	Quota asse indipendente (1..8) che risente dello shift di origine SHIP2P() .

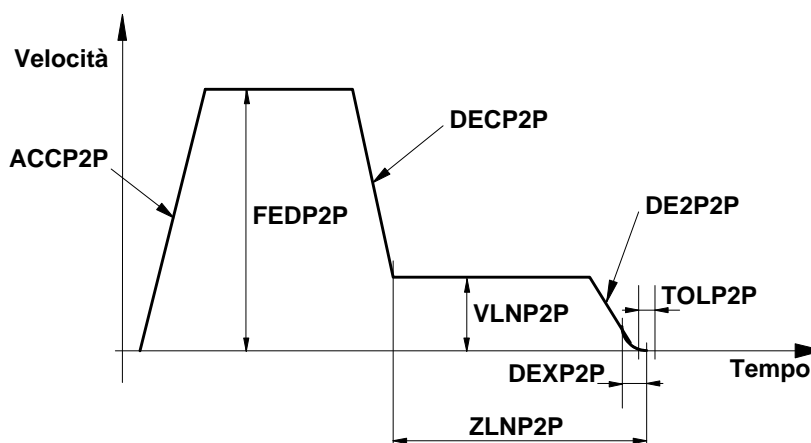
Note per l'utilizzo

Il diagramma di velocità per questi assi è riportato di seguito.

Nel caso si voglia eliminare il tratto a velocità lenta (**ZLNP2P**) è sufficiente impostarne il valore a 0 nei parametri di inizializzazione.

L'asservimento è del tipo punto-punto cioè il movimento di un asse è completamente svincolato dagli altri e la velocità impostata è funzione della distanza dal punto finale nel rispetto delle accelerazioni e velocità limite imposte nei parametri di configurazione; non esiste perciò un errore di inseguimento come per gli assi controllati.

Per la regolazione della velocità dell'asservimento occorre comparare la velocità reale **VATP2P** con la velocità effettiva **TCHP2P**.

diagramma di velocità per assi indipendenti


2.3. MODULO PER LA GESTIONE DEL CAMBIO UTENSILE

La gestione del cambio utensile (abbreviato in CU) è facilitata dalla presenza di un modulo integrato, interfacciabile con un numero ridotto di variabili.

La sua tipologia deve essere impostata nella configurazione del CN e tipicamente verrà attivato su comando del PLC, in seguito alla decodifica di una funzione **T** oppure **M6**.

Scopi principali del CU sono:

- Individuare quale **SEQUENZA** (carico, scarico, scambio, da magazzino oppure da terra) è necessaria per l'utensile richiesto analizzando la situazione del magazzino e del mandrino, la tabella utensili, la configurazione del tipo di cambio e le richieste di eventuale carico/scarico esplicitate. Ogni **SEQUENZA** viene poi identificata da un numero, per esempio sequenza 6=scambio utensile tra mandrino e magazzino.
- Gestire la tabella utensili e l'individuazione delle posizioni di prelievo e posa utensile.
- Gestire utensili con taglie differenti.
- Gestire utensili per famiglie di appartenenza.
- Agevolare l'esecuzione della sequenza tramite un opportuno sequenziatore integrato.

Non necessariamente l'applicazione deve gestire tutte le **SEQUENZE** possibili, ma solo quelle ritenute necessarie in funzione del tipo di macchina e della complessità che le si vuole dare.

Esse devono infatti essere definite nel programma PLC indicando per ognuna, accanto a dei codici interni indispensabili per l'aggiornamento dell'evoluzione della sequenza nella tabella utensili, tutti i codici delle **OPERAZIONI** (azioni elementari) per effettuare fisicamente il cambio (per esempio operazione 9001 = sblocco utensile, operazione 9021 = apertura pinza braccio di scambio).

2. Moduli interni dedicati

Nel momento dell'esecuzione della sequenza i codici relativi alle **OPERAZIONI** vengono comunicati al PLC nell'ordine definito. Quest'ultimo deve preoccuparsi prevalentemente della gestione degli organi meccanici per il cambio e la ricerca attuando le singole operazioni fisiche elementari, senza essere appesantito dalla gestione di tabelle utensili, taglie, famiglie o scelta delle sequenze di carico, scarico o scambio utensile.

Ciò significa che la gestione del sequenziatore CU deve essere del tutto simile a quella delle comuni funzioni ausiliarie **M**, **H** con la conseguente immediatezza.

2.3.1. DEFINIZIONI ELEMENTARI

OPERAZIONE: si intende il codice di una azione elementare che il sequenziatore del modulo CU comunica al PLC. Ogni azione elementare, in quanto tale, non deve avere vincoli di sequenzialità con le altre.

SEQUENZA: si intende la serie ordinata di OPERAZIONI che il modulo CU deve eseguire in funzione dello stato attuale del magazzino, della tabella utensili, ecc..,

PINZA: nel caso di cambio utensile con braccetto di scambio, si intende l'organo di estrazione dell'utensile dal mandrino.

STAZIONE INTERMEDIA: nel caso sia presente il braccetto di scambio, si intende l'organo di prelievo dell'utensile da magazzino in attesa di cambio.

Qualora sia necessario differenziare utensili per tipo e taglie diverse, occorre considerare anche:

TIPO UTENSILE: **utensile normale:** trattato in modo coerente con il tipo di CU selezionato (random, posto fisso)
utensile speciale: trattato sempre e solo come un utensile a posto fisso: verrà depositato nella stessa posizione di prelievo.

TAGLIA UTENSILE: sia gli utensili normali che quelli speciali possono avere le seguenti taglie:
utensile small: occupa sempre e solo un posto magazzino
utensile medium: occupano il numero di posti magazzino definito in
utensile large: configurazione
utensile extra :

2.3.2. MODALITÀ POSSIBILI DI FUNZIONAMENTO CU

La scelta principale nella configurazione riguarda la modalità di funzionamento del CU:

MANUALE TIPO S1200: Non è necessaria alcuna gestione PLC per l'attivazione della correzione utensile ed è generato automaticamente un arresto del programma in esecuzione (attesa di via) per tutte le **T** con valore da 10 a 98.

Le **T** da **T0** a **T9** sono origini pezzo. **T99** forza l'origine assoluta, ogni altra **T** esce da tale stato.

- MANUALE:** Non è necessaria alcuna gestione PLC per l'attivazione della correzione utensile ed è generato automaticamente un arresto del programma in esecuzione (attesa di via) per tutte le funzioni **T**.
- Le origini pezzo sono gestite separatamente con le funzioni **O**.
- Il codice **00**, cancellato da ogni altra **O**, consente il passaggio in origine assoluta.
- O-1** permette di riattivare l'origine presente prima di passare in origine assoluta.
- La funzione **T0** annulla la correzione lunghezza attiva.
- AUTOMATICO:** Il codice della funzione **T** viene passato al PLC ma non genera alcun arresto sul programma, né attiva alcuna correzione. Il programma PLC, salvo motivazioni particolari, dovrà attivare il modulo CU.
- Le origini pezzo sono gestite separatamente con le funzioni **O**.
- Il codice **00**, cancellato da ogni altra **O**, consente il passaggio in origine assoluta.
- O-1** permette di riattivare l'origine presente prima di passare in origine assoluta.
- L'attivazione di **OFST=0** annulla la correzione lunghezza attiva.
- Sull'origine **00** (che comunque elimina ogni ulteriore correzione) può anche essere attivato un correttore lunghezza. In questo caso riprogrammando **O-1** si ripristinerà l'ultima origine attiva precedente ad **00**, ma il correttore lunghezza rimarrà quest'ultimo attivato.

2.3.3. CONFIGURAZIONE DEI CU AUTOMATICI

Le scelte relative alla configurazione del magazzino, dei posti occupati per le varie taglie di utensili che devono essere impostate nella configurazione, sono riassunte di seguito:

Disposizione utensili

- posto fisso:** Ogni utensile è disposto nel magazzino nella posizione corrispondente al codice dell'utensile stesso. La sua posizione resta inalterata durante il funzionamento della macchina, cioè ogni utensile sarà sempre riposto nel magazzino nella posizione da cui è stato prelevato.
- random fisso:** A priori non ci sono vincoli tra il codice utensile ed il posto che occupa, ma ad ogni utensile è assegnata una precisa posizione nel magazzino all'interno della tabella utensili, la quale non verrà mai cambiata durante il funzionamento della macchina.
- random:** Ad ogni utensile non è predestinata una posizione specifica, ma viene depositato (e prelevato) in modo da ottimizzare la disposizione nel magazzino ed il tempo di cambio utensile.

2. Moduli interni dedicati

Geometria del magazzino

catena: Presuppone una disposizione dei posti utensile “consecutiva”, cioè in cui i vincoli dimensionali sono da considerarsi solo con l'utensile precedente ed il successivo.

planare: Presuppone una disposizione di utensili su una rastrelliera secondo un reticolo regolare XYZ allineato con gli assi.

Per questo tipo di cambio utensile non è prevista la gestione per taglie (tipicamente gli utensili sono introdotti nella rastrelliera dall'alto e quindi devono essere più piccoli del foro di alloggiamento, ovvero di dimensione **small**).

Tipo di gestione magazzino

sincrono: *La ricerca utensile non può avvenire in tempo mascherato*, cioè contemporaneamente alla lavorazione CN (il cambio utensile inizierà con il deposito del vecchio utensile, prima di ricercare il nuovo) in quanto non è prevista la stazione intermedia per il cambio.

asincrono: *La ricerca utensile può essere fatta in tempo mascherato* contemporaneamente alla lavorazione CN in quanto esiste un organo di scambio utensile tra mandrino e magazzino con una PINZA ed una STAZIONE INTERMEDIA.

semiasincrono: Nelle tipologie degli attuali cambi utensili automatici con disposizione degli utensili di tipo **RANDOM** si riscontra spesso la mancanza della **stazione intermedia**; la programmazione della funzione Txx genera unicamente una rotazione del magazzino senza alterare la situazione degli utensili.
In questi casi può essere utilizzata la configurazione del tipo di gestione magazzino **Semiasincrono**.

2.3.4. DEFINIZIONE DELLE SEQUENZE

Ogni SEQUENZA di CU deve essere definita all'interno del programma PLC con i codici obbligatori, identificati da numeri negativi, necessari per l'aggiornamento della situazione degli utensili in tabella, nella sequenza specificata nelle pagine seguenti.

In aggiunta si possono inserire tutti i codici delle OPERAZIONI ritenute necessarie dal PLC utilizzando numeri interi tra 1 e 32767.

Il significato dei codici interni delle OPERAZIONI predefinite è il seguente:

- **1** Richiesta della stazione per prelievo nuovo utensile da magazzino
- **4** Nuovo utensile prelevato da magazzino e inserito in mandrino
- **5** Nuovo utensile prelevato da magazzino e inserito in stazione intermedia
- **6** Attesa funzione di cambio utensile (**M6**)
- **10** Utensile vecchio estratto manualmente da mandrino e posato a terra
- **12** Utensile vecchio estratto da mandrino e inserito in pinza
- **13** Utensile vecchio estratto da mandrino e depositato in magazzino
- **16** Utensile nuovo prelevato da terra e inserito manualmente in mandrino
- **17** Utensile nuovo estratto da stazione intermedia e inserito in mandrino
- **23** Richiesta della stazione per posa utensile vecchio
- **27** Utensile vecchio estratto da pinza e depositato in magazzino

- **31** Utensile estratto da stazione intermedia e ridisposto in magazzino
- **34** Fine sequenza cambio utensile
- **0** Richiesta di analisi della situazione per intraprendere una nuova sequenza.

Non tutte le sequenze descritte devono necessariamente essere definite e quindi implementate. Quelle desiderate, scelte in funzione delle caratteristiche della macchina e della complessità che si vuole dare all'applicazione, devono essere dichiarate nella sezione **INIT** del PLC tramite l'istruzione:

DEF SEQC(U(num. sequenza) = codice predef., codice PLC, ... altri [,COM,1,'nome prog']

riportando nell'ordine previsto tutti i codici interni predefiniti ed indispensabili.

Gli errori di definizione nelle sequenze sono segnalati comunque con messaggi in chiaro sul video.

Oltre ai codici delle operazioni, nella istruzione **DEF SEQC(U(n)** è possibile specificare facoltativamente il nome di un sottoprogramma di CN (**COM,1,'nome prog'**) che verrà mandato automaticamente in esecuzione in concomitanza con l'operazione di attesa **M06** (-6) e del segnale PLC di **M06** programmata (**M6PGM = 1**), al fine di posizionare gli assi CN ed eventualmente eseguire la parte della sequenza di cambio utensile in tempo non mascherato.

Secondo il tipo di cambio utensile automatico configurato, le SEQUENZE possibili, sono riportate di seguito.

In ogni sequenza che presuppone il montaggio del nuovo utensile in mandrino occorrerà attivare il correttore lunghezza prima di iniziare la lavorazione (INTOF=1).

Cambi utensili asincroni

Sequenze di gestione utensili a terra (con **POSIZ. MAGAZ.** = 0 e **SELECU** = 0 o **SELECU** = 1):

Sequenza 1:	-6, -16, -34	presa ut. da terra e inserimento in mandrino(carico)
Sequenza 2:	-6, -10, -34	posa utensile da mandrino a terra (scarico)
Sequenza 3:	-6, -10, -16, -34	posa ut. mandrino a terra, presa utensile da terra e inserimento in mandrino (scambio)

Sequenze di scambio tra utensili a terra ed il magazzino (SELECU = 0):

Sequenza 4:	-1, -5, -6, -10, -17, -34	posa utensile mandrino a terra, presa utensile da magazzino e inserimento in mandrino
Sequenza 5:	-23, -6, -12, -16, -27, -34	posa utensile mandrino in magazzino, presa utensile da terra e inserimento in mandrino

Sequenze di gestione utensili da magazzino (SELECU = 0):

Sequenza 6:	-1,-5,-6,-12,-17,-23,-27,-34	posa utensile mandrino in magazzino, presa utensile da magazzino e inserimento in mandrino (scambio)
Sequenza 7:	-1, -5, -6, -17, -34	presa utensile da magazzino e inserimento in mandrino (carico)
Sequenza 8:	-23, -6, -12, -27, -34	posa utensile mandrino in magazzino (scarico)

2. Moduli interni dedicati

Altre sequenze (**SELECU** = 0):

Sequenza 11: -6, -34	utensile programmato uguale al precedente (cambio correttore).
Sequenza 19: -23, -31, -0	posa utensile da stazione intermedia in magazzino e analisi nuova situazione (due T consecutive).

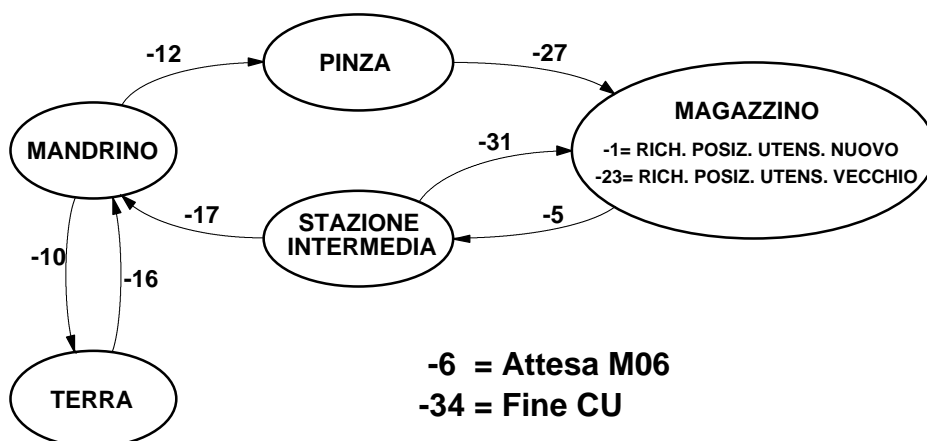
Sequenze per il carico e lo scarico utensili da terra a magazzino tramite il mandrino

(solo con **SELECU** = 2):

Sequenza 9: -23, -6, -16, -12, -27, -34	presa utensile da terra a mandrino, posa da mandrino in pinza, posa pinza in magazzino.
---	---

(solo con **SELECU** = 3):

Sequenza 10: -1, -5, -6, -17, -10, -34	presa utensile da magazzino in stazione intermedia, presa da stazione intermedia in mandrino, posa mandrino a terra.
--	--



Cambi utensili sincroni

Sequenze di gestione utensili a terra (con **POSIZ. MAGAZ.** = 0 e **SELECU** = 0 o **SELECU** = 1):

Sequenza 1: -6, -16, -34	presa utensile da terra e inserimento in mandrino
Sequenza 2: -6, -10, -34	posa utensile da mandrino a terra (scarico)
Sequenza 3: -6, -10, -16, -34	posa utensile mandrino a terra, presa utensile da terra e inserimento in mandrino (scambio)

Sequenze di scambio tra utensili a terra ed il magazzino (**SELECU** = 0):

Sequenza 4: -6, -10, -1, -4, -34	posa utensile mandrino a terra, presa utensile da magazzino e inserimento in mandrino
Sequenza 5: -6, -23, -13, -16, -34	posa utensile mandrino in magazzino, presa utensile da terra e inserimento in mandrino

Sequenze di gestione utensili da magazzino (SELECU=0):

Sequenza 6:	-6, -23, -13, -1, -4, -34	posa utensile mandrino in magazzino, presa utensile da magazzino e inserimento in mandrino (scambio)
Sequenza 7:	-6, -1, -4, -34	presa utensile da magazzino e inserimento in mandrino
Sequenza 8:	-6, -23, -13, -34	posa utensile mandrino in magazzino (scarico)

Altre sequenze (SELECU = 0):

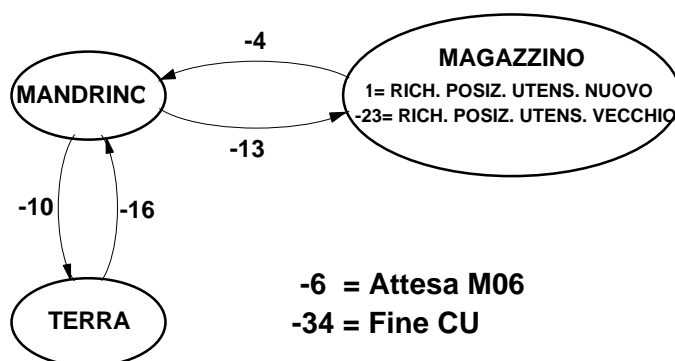
Sequenza11:	-6, -34	utensile programmato uguale al precedente (cambio correttore, effettuare INTOF=1 in modo sincrono).
-------------	---------	---

Sequenze per il carico e lo scarico utensili da terra a magazzino tramite il mandrino:
(SELECU = 2):

Sequenza 9:	-6, -16, -23, -13, -34	presa utensile da terra a mandrino, posa da mandrino in magazzino.
-------------	------------------------	--

(SELECU = 3):

Sequenza 10:	-6, -1, -4, -10, -34	presa utensile da magazzino in mandrino, posa mandrino a terra.
--------------	----------------------	---


Cambi utensili semiasincroni

Le caratteristiche di questa configurazione sono le seguenti:

- fisicamente non esiste una stazione intermedia, nella tabella utensili non hanno significato e non sono quindi gestite le celle di pinza e stazione intermedia.
- L'aggiornamento della tabella utensili risulta notevolmente semplificato: anche in fase di interruzione del ciclo di cambio, l'unico utensile che ha il segno "-" (meno) è quello in mandrino.
- Il cambio utensile semiasincrono ha come requisito che l'operazione di deposito del vecchio utensile avvenga sempre contemporanea alla presa del nuovo utensile (tramite un braccio di scambio a due pinze); ne consegue che il posto di presa e di posa devono essere coincidenti.
- Il caso di scambio tra utensili di taglie differenti viene differenziato dallo scambio tra utensili di taglie uguali per facilitare la codifica del PLC.

2. Moduli interni dedicati

Il significato dei codici interni delle OPERAZIONI predefinite è il seguente:

- 9 Scambio utensile tra mandrino e magazzino

Sequenze di gestione utensili a terra (manuali)

Sequenza 1: -6, -16, -34	carico da terra a mandrino
Sequenza 2: -6, -10, -34	scarico da mandrino a terra
Sequenza 3: -6, -10, -16, -34	scambio tra mandrino e terra

Sequenze di scambio tra utensili a terra ed il magazzino (miste) (SELECU=0)

Sequenza 4: -1, -6, -10, -4, -34	posa utensile mandrino a terra e presa da magazzino
Sequenza 5: -23, -6, -13, -16, -34	posa utensile mandrino in magazzino e presa da terra

Sequenze di gestione utensili da magazzino (automatico) (SELECU=0)

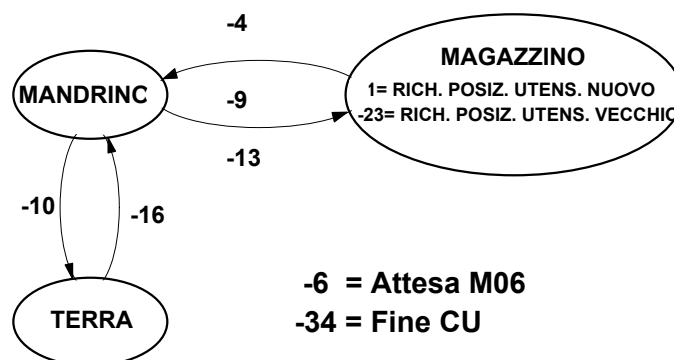
Sequenza 6: -1, -6, -9, -34	scambio utensile tra magazzino e mandrino (taglie uguali)
Sequenza 13: -23, -6, -13, -1, -4, -34	scambio utensile tra magazzino e mandrino (taglie differenti)
Sequenza 7: -1, -6, -4, -34	carico utensile da magazzino in mandrino
Sequenza 8: -23, -6, -13, -34	scarico utensile da mandrino a magazzino

Altre sequenze (SELECU=0)

Sequenza 11: -6, -34	utensile programmato uguale all'utensile in mandrino (solo cambio Correttore lunghezza)
----------------------	---

Con questo tipo di cambio utensili la sequenza 13 (scambio fra taglie differenti) può essere implementata almeno con un dei seguenti metodi:

- Doppio scambio: si posiziona innanzitutto il magazzino sul posto di deposito, che dovrà essere vuoto (operazione -23); all'atto dell'esecuzione della M6 si prosegue con un primo scambio tra l'utensile in magazzino ed il mandrino (dopo questa operazione il mandrino resta vuoto ed il braccio torna a riposo); il ciclo prosegue con un posizionamento del magazzino sul posto di prelievo del nuovo utensile; infine il ciclo termina con un ulteriore scambio tra magazzino e mandrino.
- Scambio singolo: la sequenza si comporta come un normale ciclo di scambio tra taglie uguali, ma quando si trovano entrambi gli utensili sulle pinze del braccio scambiatore (tipicamente braccio in basso) viene effettuata una rotazione del magazzino sul posto di deposito.



Implementazione nel programma PLC

esempio: CU ASINCRONO RANDOM CATENA

```
INIT
DEF SEQCU(6)=901,-1,902, 920,-5,-6, ...,COM,1,'SCAMBIO'           [scambio ut. con magazzino]
[901 = sblocca magazzino per rotazione
[-1 = codice predefinito: posiziona magazzino su posizione prelievo
[902 = blocca magazzino
[920 = braccio scambiatore in posizione di presa/posa
[-5 = codice predefinito: nuovo utensile prelevato da maggazz. e inserito in stazione intermedia
[-6 = codice predefinito: attesa M06
[COM,1,'SCAMBIO' = sottoprogramma da eseguire su operazione -6, quando M06 programmata (M06PGM=1)
PROG
...
```

Attivazione del modulo cambio utensile

Il PLC riceve in modo sincrono sul registro **TOOL**, con lo strobe **STROT**, il codice della nuova funzione **T** programmata, ma ciò non implica ancora alcuna attivazione del modulo di CU.

Per attivare il modulo CU si deve scrivere il codice dell'utensile desiderato nel registro **UTECU**, e settare il segnale **NEWCU**; questo viene resettato dal CU stesso non appena ha inizio la sequenza determinata per il cambio utensile richiesto a patto che il segnale **MAPRCU** sia a 1 (vedere più avanti).

Naturalmente la tabella utensili deve già essere stata compilata.

UTECU = 0 viene interpretato come richiesta di scarico dell'utensile da mandrino in magazzino, od a terra se non esiste posto vuoto, dell'utensile in mandrino.

Attuazione delle sequenze

Dopo essere stato attivato (**NEWCU** resettato), il modulo CU alza il segnale **CUATT** (cambio utensile attivo), quindi:

- comunica sul registro **NSEQCU** il numero della SEQUENZA intrapresa
- comunica sul registro **PPRECU** il numero della posizione magazzino di presa nuovo utensile
- comunica sul registro **PPOSCU** il numero della posizione magazzino di posa vecchio utensile
- prepara sul registro **OFST** il codice del correttore associato al nuovo utensile,
- infine passa al PLC, sul registro **OPERCU**, nella sequenza definita in **DEF SEQCU(n)**, i codici delle OPERAZIONI accompagnati dallo strobe **BRDYCU**.

Il PLC deve preoccuparsi di eseguire la singola operazione proposta senza vincoli di sequenzialità con le altre; gli unici vincoli da prevedere sono di natura meccanica e di sicurezza tra un organo e l'altro.

Il segnale di sincronismo del colloquio **BRDYCU** deve essere resettato dal PLC non appena acquisita la nuova operazione.

Se l'operazione richiesta presuppone una attesa per poter eseguire la fase successiva (qualunque sia), il PLC deve momentaneamente porre a zero il segnale **MAPRCU** (macchina pronta per il CU), che altrimenti deve stare a 1.

2. Moduli interni dedicati

Nei casi in cui l'operazione corrente sia una richiesta di stazione su cui prelevare/posare, il PLC dovrà preoccuparsi di posizionare il magazzino in funzione delle posizioni indicate su **PPRECU** e **PPOSCU**, sfruttando, se necessario, il MODULO DI POSIZIONAMENTO ASSI INDIPENDENTI.

Quando la sequenza, qualunque essa sia, arriva sulla operazione -6 (attesa **M06**), automaticamente si sospende e resta in tale situazione finché il PLC non attiva il segnale **M6PGM** (**M06** programmata).

Quando il modulo CU, essendo in fase di attesa **M06** (-6), rileva il segnale **M6PGM**, lancia l'eventuale sottoprogramma CN (**COM**) definito per la sequenza corrente, quindi resetta **M6PGM** e la sequenza CU prosegue con le fasi successive.

Il sottoprogramma CN così lanciato risolve molto bene la sequenza di operazioni da eseguire in modo sincrono (cioè quelle relative al cambio vero e proprio), nonché i posizionamenti assi CN.

Da notare che il segnale **M6PGM** attivo esegue in modo del tutto automatico una sospensione nell'avanzamento dei blocchi di programma, senza appesantire ulteriormente il programma PLC con complicati sincronismi. Si pensi al caso in cui la funzione **M6** viene eseguita prima che l'utensile specificato dalla funzione **T** sia disponibile dal magazzino in quanto la ricerca è in corso (CU random).

La sequenza CU corrente ha termine quando, essendo stato eseguita l'operazione di fine CU (-34), il PLC resetta il segnale di **CUATT**.

La richiesta di un nuovo cambio utensile **NEWCU** è acquisita, cioè dà luogo ad una nuova sequenza di cambio utensile, solo se:

- il CU non ha sequenze in corso;
- pur avendo una sequenza in corso, si sta attuando l'operazione di attesa **M6** (caso di due **T** consecutive senza **M6**).

In questo modo non è necessario che il PLC si preoccupi di creare complessi sincronismi.

Nel caso il CU consideri necessaria una sequenza non dichiarata con l'istruzione **DEF SEQCU(n)**, viene segnalato un messaggio in chiaro per l'operatore di **sequenza non prevista** e si passa in stato di cambio utensile in emergenza (segnale **EMACU** = 1); questo stato non condiziona implicitamente nessuna delle altre attività del CN.

Attivazione correttore lunghezza utensile

Per attualizzare la correzione utensile, proposta su **OFST**, il PLC deve attivare in modo sincrono con **BURDY**, lo strobe **INTOF** (che a sua volta viene resettato dal CN).

Se per qualche ragione si vuole attivare un correttore utensile diverso, è comunque possibile sovrascrivere **OFST** prima di settare **INTOF**.

Particolare attenzione va posta quando si gestiscono utensili suddivisi per famiglie (utensili alternativi): in questi casi non è detto che l'utensile da montare abbia come codice lo stesso della **T** programmata per cui sovrascrivendo **OFST** si potranno ottenere esiti indesiderati.

Decodifica della T programmata e selezione della sequenza di lavoro

Può essere necessario fornire una compatibilità con la sintassi dei sistemi della serie S1200 in cui le funzioni **T** da **T0** a **T9** rappresentano origini pezzo e non utensili e **T99** rappresenta il passaggio in coordinate assolute: in questi casi occorrerà decodificare la **T** programmata prima di attivare il modulo cambio utensile.

Prima di attivare il modulo CU si ha la facoltà di scegliere il modo di funzionamento scrivendo sul registro **SELECU** il codice desiderato; le selezioni effettuate con una sequenza CU in corso vengono ignorate.

- 0 = modo normale (default):**
l'utensile richiesto viene montato in mandrino prelevandolo da magazzino, se presente, o da terra.
- 1 = modo con magazzino escluso:**
l'utensile viene montato in mandrino da terra e posato a terra; il magazzino è considerato interdetto all'uso.
- 2 = modo con carico in magazzino dell'utensile programmato:**
l'utensile richiesto viene montato da terra in mandrino, quindi posato in magazzino.
- 3 = modo con scarico da magazzino dell'utensile programmato:**
l'utensile richiesto viene montato in mandrino prelevandolo dal magazzino se già non è presente ed immediatamente posato a terra.

2.3.5. INTERRUZIONE DELLA SEQUENZA

E' possibile interrompere volontariamente una sequenza di cambio utensile in due modi:

- interruzione istantanea per emergenza: si ottiene settando il segnale **REMCU** in questo caso:
 - il CU passa in stato di emergenza (**EMACU=1**)
 - La tabella utensili può non essere coerente con la situazione reale, è perciò necessario l'intervento dell'operatore per la verifica. Ogni ulteriore richiesta di cambio utensile viene ignorata.
- interruzione della sequenza con il segnale **RBKCU**: non viene segnalata **EMACU**.

Se il CN viene spento (mancata tensione) durante una sequenza di cambio utensile, alla successiva riaccensione compare un messaggio in chiaro e viene settato automaticamente **EMACU**.

Per uscire dallo stato di emergenza si deve rimuovere la richiesta **REMCU**, quindi attivare il segnale **RBKCU**. Esso viene a sua volta resettato automaticamente dal CU quando acquisito.

In ogni caso occorre prevedere le sicurezze nel PLC in modo che una qualsiasi sequenza di CU automatico non possa iniziare se non sono verificate le condizioni iniziali (CU a riposo).

Gestione vita utensili integrata

L'algoritmo di gestione vita utensili permette di controllare il tempo di lavorazione (VITA RESIDUA) dell'utensile montato in mandrino tramite un «contatore» che viene decrementato dal CNC ogni 10mS quando il PLC setta il flag di utensile in fase di asportazione **UTRUN**.

Quando la VITA RESIDUA diventa minore della soglia di VITA MINIMA l'utensile viene considerato scaduto.

Al successivo richiamo di tale utensile, esso potrà essere sostituito con uno alternativo (gestione famiglie utensili).

Nel caso non vi siano utensili alternativi (tipicamente con Cambio Utensile manuale) viene generato un messaggio di utensile non più disponibile.

Per informazioni più dettagliate vedere il *Notiziario Tecnico numero 1 del 1996*.

2. Moduli interni dedicati

DESCRIZIONE DELLE VARIABILI PLC

Nome	Dim	Direzione	Descrizione
UTRUN1	PLC \Rightarrow CN		Utensile in mandrino in fase di lavorazione: decrementa VITA RESIDUA
UTTIM	32	CN \Rightarrow PLC	Valore del contatore VITA RESIDUA dell'utensile in mandrino
UTSTS	8	CN \Rightarrow PLC	Registro di stato dell'utensile in mandrino: UTSTS (1) = vita scaduta UTSTS (2) = vita residua ≤ 0

2.3.6. DIFFERENZIAZIONE UTENSILI PER FAMIGLIE

La gestione per "famiglie" presuppone l'esistenza di una serie di utensili tecnologicamente equivalenti (es.: punta a forare diametro 10). A livello di programma esiste un solo utensile (**capostipite**) ed una serie di suoi sostituti (**figli**) che saranno montati al suo posto in caso di scadenza vita, rottura, interdizione all'uso, ecc. Se, per esempio, l'utensile T65 ha come capostipite l'utensile T23, significa che richiedendo T23 questo verrà montato finché possibile, poi sostituito da T65. Con questa gestione il PLC non conosce a priori né l'utensile, né il correttore da applicare.

La scelta tra gli utensili della famiglia viene fatta in funzione dei parametri di "vita scaduta" e "interdizione".

Ogni utensile è caratterizzato da:

- **una vita max**, in minuti e secondi che rappresenta il tempo max per cui può essere usato
- **una vita residua** che rappresenta la vita max meno il tempo di attività già trascorso
- **una vita minima** raggiunta la quale l'utensile viene considerato esaurito.

L'**interdizione**, cioè l'esclusione dell'utensile, ha priorità rispetto alla situazione della vita.

Al momento di scegliere un utensile della famiglia verranno scartati quelli interdetti e quelli con la vita scaduta.

2.3.7. DIFFERENZIAZIONE UTENSILI CON TAGLIE DIVERSE

Il modulo CU è in grado di gestire, in modo del tutto trasparente, senza implicare alcuna gestione PLC, utensili di taglie diverse (fino a 4) le cui dimensioni devono essere indicate nei dati di configurazione.

Durante il regolare funzionamento della macchina il modulo CU stesso dispone i depositi in magazzino rispettando i vincoli dimensionali fra gli utensili.

2.3.8. DESCRIZIONE DELLE VARIABILI PLC

Riassunto segnali e registri interessati

UTEUCU	16	CN	⇔	PLC	no	Numero utensile richiesto al modulo cambio utensile. UTEUCU = 0 è il codice particolare riservato alla sequenza di scarico dell'utensile da mandrino in magazzino (o a terra se non esiste posto).
NEWCU	1	CN	⇔	PLC	no	Comando di attivazione nuova sequenza per CU. Questo segnale viene settato dal PLC per attivare il modulo cambio utensile ed è resettato dal CU non appena acquisito.
NSEQCU	16	CN	⇒	PLC	no	Codice dell'ultima sequenza CU intrapresa.
BRDYCU	1	CN	⇔	PLC	no	Strobe di presenza nuovo codice su OPERCU . Viene settato dal CU e deve essere resettato dal PLC non appena è stato acquisita la nuova operazione.
MAPRCU	1	CN	⇒	PLC	no	Macchina pronta per il cambio utensile: se uguale a zero la sequenza viene sospesa finché non è rilasciato.
OPERCU	16	CN	⇒	PLC	no	Codice dell'operazione richiesta dal CU al PLC.
PPRECU	16	CN	⇒	PLC	no	Posizione da raggiungere per il prelievo del nuovo utensile.
PPOSCU	16	CN	⇒	PLC	no	Posizione da raggiungere per il deposito del vecchio utensile.
CUATT	1	CN	⇔	PLC	no	Segnale alzato dal CU ad inizio nuova sequenza, resettato dal PLC quando considera terminata la sequenza corrente.
M6PGM	1	CN	⇔	PLC	si	(M6 programmata) Deve essere alzato in modo sincrono con il burdy dal PLC, viene resettato dal CU quando, essendo arrivata l'operazione di attesa M06 , è stato lanciato l'eventuale sottoprogramma CN (COM). In mancanza di questo segnale, la sequenza si arresta sulla fase (-6). M6PGM attivo implica una sospensione automatica dell'esecuzione dei blocchi CN !
UTSPCU	16	CN	⇔	PLC	no	Numero utensile in mandrino.
UTSICU	16	CN	⇔	PLC	no	Numero dell'utensile in stazione intermedia.
UTPICU	16	CN	⇔	PLC	no	Numero dell'utensile in pinza.
EMACU	1	CN	⇒	PLC	no	Cambio utensile in stato di emergenza. E' settato quando la sequenza CU viene interrotta prima del termine in seguito a richiesta di emergenza CU. La presenza di questo segnale segnala che può non essere coerente la situazione degli utensili presente in tabella rispetto alla situazione reale; necessita quindi un intervento dell'operatore. Eventuali richieste di nuovi cambi utensili NEWCU vengono ignorate.
REMCU	1	CN	⇔	PLC	no	Richiesta di emergenza per CU. Questo comando interrompe la sequenza corrente di CU e l'operazione in corso, quindi pone il CU in stato di emergenza.
RBKCU	1	CN	⇔	PLC	no	Uscita dallo stato di emergenza CU EMACU e richiesta di interruzione su sequenza cambio utensile: deve essere settato dal PLC ed è resettato dal CU quando acquisito.
SELECU	8	CN	⇔	PLC	no	Selettore di modalità. Deve essere predisposto prima di attivare il modulo cambio utensile, esso è acquisito su inizio sequenza e non può essere modificato durante la stessa. 0 = modo CU normale 1 = modo CU con magazzino escluso 2 = modo con carico dell'utensile programmata in magazzino 3 = modo con scarico a terra dell'utensile programmato.

2. Moduli interni dedicati

ERCU 16 CN \Rightarrow PLC no Codice dell'errore riscontrato dal CU. Ad ogni operazione, vengono verificate le informazioni relative a magazzino, tabella utensili, configurazione. In caso di informazioni non congruenti o di situazioni non previste o non gestibili, il CU interrompe l'eventuale sequenza attiva e comunica l'errore riscontrato. Inoltre, nessuna sequenza di CU è attivabile se si è in condizione di errore.

NUOVE VARIABILI DI INFORMAZIONE

Il PLC può acquisire alcuni parametri di configurazione per poter implementare programmi più flessibili e generalizzati; le informazioni sono disponibili sulle seguenti variabili:

<i>Nome</i>	<i>Dim</i>	<i>Direzione</i>	<i>Descrizione</i>
CUATYP	16	CN \Rightarrow PLC	Tipo di cambio utensile selezionato 0 = manuale 1 = manuale S1200 2 = automatico
MAGGEO	16	CN \Rightarrow PLC	Geometria magazzino selezionata 0 = catena 1 = planare
MAGTYP	16	CN \Rightarrow PLC	Disposizione utensili in magazzino selezionata 0 = fisso 1 = random 2 = random fisso
MAGGST	16	CN \Rightarrow PLC	Gestione magazzino selezionata 0 = sincrono 1 = asincrono 2 = semiasincrono

2.3.9. LA TABELLA UTENSILI

All'interno della tabella utensili sono presenti tutte le informazioni relative agli utensili, organizzate per righe e disposte su più pagine:

PARAMETRI UTENSILI (pagina 1)

- codici degli utensili, correttori raggio e lunghezza, posizione in magazzino, stato di interdizione, tipo speciale e taglia
- codice degli utensili in mandrino, in pinza e stazione intermedia

PARAMETRI VITA UTENSILI (pagina 2)

- vita massima, vita minima, vita residua, utensile capostipite, vita utensile scaduta

PARAMETRI A DISPOSIZIONE DELL'APPLICAZIONE (pagina 3)

- word#1, word#2, float#2, float#3

I **Codici** validi per gli utensili sono tutti i numeri interi da 1 a 32767.

La **posizione in magazzino** va interpretata nel modo seguente:

- se è un numero compreso tra 1 ed il numero di posti massimo per il magazzino utensili, rappresenta il posto in cui dovrà essere prelevato l'utensile.
- se è uguale a 0 significa che l'utensile dovrà essere prelevato e quindi successivamente depositato manualmente da terra.
- se è un numero preceduto dal segno meno significa che l'utensile è stato prelevato e rappresenta il posto da cui l'utensile è stato prelevato (quest'informazione serve per esempio nel caso di random fisso).

Se il flag di **stato interdizione** è uguale a "si" l'utensile corrispondente non verrà mai montato e viene trattato come se non fosse presente in magazzino (un utensile può venire dichiarato interdetto in seguito ad una verifica negativa della sua integrità).

Naturalmente, se esiste un utensile non interdetto all'uso in cui il capostipite sia uguale all'utensile interdetto, questo verrà montato come utensile alternativo.

Gli utensili il cui flag di **vita scaduta** è uguale a "si" vengono trattati come utensili interdetti all'uso.

Capostipite, come già accennato indica per quale utensile è alternativo l'utensile.

Esempio:

- **T10** con vita scaduta, nessun capostipite
- **T11** con vita non scaduta, capostipite 10

Viene programmato **T10**: verrà montato in alternativa il primo utensile incontrato con vita non scaduta e capostipite 10 cioè l'utensile 11.

Le variabili **WORD#1** e **WORD#2** sono due word (RAM,16) a disposizione per poter contenere alcune informazioni aggiuntive relative all'utensile.

Allo stesso modo sono disponibili due variabili in formato floating point (**RAM,32**) di nome **FLOAT#1** e **FLOAT#2**.

Scrittura dei campi della tabella utensili da parte del PLC

(Solo per applicazioni particolari)

Normalmente la tabella utensili è completamente gestita dal modulo cambio utensile comunque, per applicazioni particolari, tutti i campi della tabella utensili sono accessibili da parte del PLC in lettura e scrittura.

La lettura può essere effettuata come per qualunque altra variabile PLC senza particolari precauzioni.

Occorre tenere conto che la scrittura su queste variabili comporta, per la struttura del sistema, una sequenza piuttosto lunga in quanto oltre alla tabella normalmente presente nella memoria di lavoro del sistema, occorre anche aggiornare la copia sulla memoria statica del sistema, operazione che ha tempi maggiori.

Nel PLC sono presenti gli array che rappresentano le colonne della tabella utensili, la cui dimensione è resa disponibile sulla variabile **UTENRI**, con i nomi riportati di seguito.

2. Moduli interni dedicati

Per poter accedere ai parametri di un certo utensile occorre ricercarne il codice con un'istruzione del tipo

RIC(UTNUM,1,UTENRI,TOOL) label

per determinare in quale "riga" della tabella si trova descritto l'utensile; con l'indice ottenuto si accederà quindi a tutti gli altri array.

Come accennato, essendo lenta l'operazione di scrittura dei campi della tabella è impensabile sospendere il programma PLC per attendere che sia terminata l'operazione. Esiste perciò una memoria di transito con capacità limitata (16 scritture) su cui le variabili relative ai campi vengono trasferite temporaneamente, per poi essere scritte definitivamente con il tempo necessario.

Per questo motivo esiste la variabile **UTEFRE** che rappresenta il numero di scritture ancora disponibili nella memoria temporanea; il PLC dovrà sempre verificare di avere ancora scritture disponibili prima di aggiornare i campi della tabella. Nel caso questa regola non venga rispettata, il PLC verrà disattivato con il conseguente messaggio su video.

Al PLC è reso anche disponibile un ulteriore array **MAGCUA()** che rappresenta l'immagine del magazzino utensili (**MAGCUA(1)** = posto 1 e così via); il numero di elementi dipende evidentemente dal quanti posti sono definiti nei parametri di configurazione (il PLC può leggere questo numero su **MAGNPO**).

Riassunto segnali e registri interessati

UTENRI	16	CN	⇒ PLC no	Numero di "righe" nella tabella utensili, cioè numero massimo di elementi dei vettori che rappresentano le colonne della tabella utensili.
UTNUM()	16	CN	⇔ PLC no	Codici degli utensili in tabella (1 .. UTENRI).
UTPOS()	16	CN	⇔ PLC no	Posti in magazzino degli utensili (1 .. UTENRI).
UTCAP()	16	CN	⇔ PLC no	Capostipiti degli utensili (1 .. UTENRI).
UTDIM()	8	CN	⇔ PLC no	Taglie degli utensili (1 .. UTENRI), dove: 0 = small 1 = medium 2 = large 3 = extra
UTSPC()	8	CN	⇔ PLC no	Utensili speciali (1 .. UTENRI), dove: 0 = utensile normale diverso da 0 = utensile speciale
UTPLKO()	8	CN	⇔ PLC no	Utensili interdetti (1 .. UTENRI), dove: 0 = utensili non interdetti diverso da 0 = utensili non interdetti
UTVTKO()	8	CN	⇔ PLC no	Vita scaduta (1 .. UTENRI), dove: 0 = vita non scaduta diverso da 0 = vita scaduta
UTVITA()	64	CN	⇔ PLC no	Vita MAX utensili (1 .. UTENRI) in centesimi di secondo.
UTVTRE()	64	CN	⇔ PLC no	Vita Residua utensili (1 .. UTENRI) in centesimi di secondo.
UTVTMI()	64	CN	⇔ PLC no	Vita Minima utensili (1 .. UTENRI) in centesimi di secondo.
UTWD1()	16	CN	⇔ PLC no	WORD#1 - variabili 1 nel formato word a disposizione dell'applicazione (1.. UTENRI).
UTWD2()	16	CN	⇔ PLC no	WORD#2 - variabili 2 nel formato word a disposizione dell'applicazione (1.. UTENRI).
UTFP1()	32	CN	⇔ PLC no	FLOAT#1 - variabili 1 nel formato floating point a disposizione dell'applicazione (1 .. UTENRI).
UTFP2()	32	CN	⇔ PLC no	FLOAT#2 - variabili 2 nel formato floating point a disposizione dell'applicazione (1 .. UTENRI).

UTEFRE	16	CN \Rightarrow PLC no	Numero delle scritture ancora disponibili nella memoria temporanea per l'aggiornamento dei campi della tabella utensili.
MAGNPO	16	CN \Rightarrow PLC no	Numero dei posti configurati nei parametri per il magazzino utensili.
MAGCUA()	16	CN \Rightarrow PLC no	Array che rappresenta l'immagine del magazzino utensili (0 .. MAGNPO).

LETTURA E SCRITTURA DEI CORRETTORI RAGGIO E LUNGHEZZA

Il PLC può accedere in lettura e scrittura sui campi relativi a correzione lunghezza e raggio della tabella utensile utilizzando le variabili riportate di seguito; la modalità di accesso è la stessa utilizzata per le altre variabili di accesso alla tabella utensili. Ogni elemento degli array corrisponde ad una riga della tabella utensile. Il numero di elementi ogni vettore dipende dalla dimensione della tabella utensili.

<i>Nome</i>	<i>Dim</i>	<i>Direzione</i>	<i>Descrizione</i>
CORR_Z()	32	CN \Leftrightarrow PLC	correzione lunghezza utensile su asse mandrino (o longitudinale per i torni)
CORR_R()	32	CN \Leftrightarrow PLC	correzione raggio utensile
CORR_X()	32	CN \Leftrightarrow PLC	correzione diametrale utensile (solo per i torni)

2.4. SERPLC – INTERFACCIA SERIALE PER IL PLC

Questo modulo opzionale ha lo scopo di realizzare un canale di comunicazione seriale full duplex per lo scambio di dati del programma PLC con semplici dispositivi di lettura/scrittura dati, strumenti di controllo e microterminali in genere.

Si possono gestire contemporaneamente al massimo due linee seriali con due schede distinte.

Ciascuna scheda può essere configurata RS232 o RS422.

Possono essere trasmessi e ricevuti caratteri ASCII o dati binari codificati in ASCII con formati fissi o variabili con o senza carattere terminatore.

La velocità è configurabile fra 110 e 38400 BAUD.

I bit di trasmissione possono essere configurati fra 7 o 8 più un bit di stop.

L' handshake può essere XON-XOFF oppure hardware (in RS232).

Il controllo di parità può essere pari, dispari o assente.

Per queste scelte è richiesto un insieme di parametri di configurazione inseriti nel programma PLC stesso.

2.4.1 SEGNALI E REGISTRI PER LA GESTIONE VIA PLC

I nomi delle variabili iniziano con RS1 o RS2 a seconda se si utilizza la prima o la seconda linea seriale, nelle descrizioni di seguito sostituire l'indice 1 con 2 nel caso di seconda scheda.

Nome	Dim	Direzione	Direzione	Descrizione
RS1ON	8	CN	← PLC	no -1 = richiesta apertura seriale 0 = risposta linea chiusa 1 = risposta linea aperta
RS1PAR	8	CN	← PLC	no Bitmap parametri protocollo hardware (default 0) bit 0: 0=dato a 7 bit, 1=dato a 8 bit bit 1: 0=nessuna parità, 1=con parità bit 2: 0=parità dispari, 1=parità pari bit 3: 0=xon/xoff disabilitato, 1=xon/xoff abilitato bit 4: 0=rts/cts disabilitato, 1=rts/cts abilitato bit 5: 0=originate, 1=answer 0=originate : il cn trasmette attivando rts e può essere rallentato dal ricevente quando quest'ultimo disattiva il cts del cn. 1=answer : il cn riceve i dati dal trasmittente e può rallentare il flusso dei dati disattivando il cts del trasmittente
RS1BRT	64	CN	← PLC	no Valore di baud rate (default 9600) 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
RS1XON	8	CN	← PLC	no Codice del carattere per XON(default 11h)
RS1XOFF	8	CN	← PLC	no Codice del carattere per XOFF(default 12h)
RS1PRR	16	CN	← PLC	no Tipo di protocollo in ricezione (default = 0) 0 = protocollo con terminatore > 0 = protocollo con lunghezza fissa definita in byte (max 128)
RS1SCR	8	CN	← PLC	no Codice del carattere terminatore nel caso di protocollo di ricezione=0 (default = 0Dh)
RS1PRW	16	CN	← PLC	no Tipo di protocollo in trasmissione (default = 0) 0 protocollo con terminatore > 0 protocollo con lunghezza in byte (max 128)
RS1SCW	8	CN	← PLC	no Codice del carattere terminatore nel caso di protocollo di trasmissione =0 (default = 0Dh)
RS1ERR	8	CN	⇒ PLC	no Codice per eventuale errore 0 nessun errore 1 errore di parità 2 errore di overrun (buffer di ricezione pieno) 3 numero slot errato 4 valore di baud rate errato 5 valore di sincronizzazione errato 6 lunghezza buffer troppo grande (> 128 caratteri) 7 non è pervenuto il terminatore entro 128 caratteri

RS1CIN	16	CN	↔	PLC	no	Contatore caratteri nel buffer di ricezione (max 128)
RS1COU	16	CN	↔	PLC	no	Contatore caratteri nel buffer di trasmissione (max 128)
RS1BIN()	8	CN	↔	PLC	no	Buffer di ricezione fino al massimo di 128 byte. Equivale alla variabile Stringa RS1SIN
RS1SIN	STR	CN	↔	PLC	no	Stringa di ricezione fino al massimo di 128 caratteri. Equivale al buffer di ricezione RS1BIN
RS1BOU()	8	CN	↔	PLC	no	Buffer di trasmissione fino al massimo di 128 byte. Equivale alla variabile Stringa RS1SOU
RS1SOU	STR	CN	↔	PLC	no	Stringa di trasmissione fino al massimo di 128 caratteri. Equivale al buffer di trasmissione RS1BOU

2.4.2 DESCRIZIONE DEL FUNZIONAMENTO

Il programma PLC per attivare la porta seriale deve prima impostare o modificare i parametri del protocollo hardware: RS1PAR, RS1BRT, RS1XON, RS1XOF, ed i parametri del protocollo software: RS1PRR, RS1SCR, RS1PRW; RS1SCW, quindi deve porre RS1ON=-1.

Il CN, in caso di configurazione corretta, pone RS1ON=1 ed è pronto a ricevere e trasmettere. In caso contrario pone RS1ON=0 ed il codice di errore in RS1ERR.

In caso di necessità, per cambiare la configurazione hardware occorre prima chiudere per poi riaprire la linea. La configurazione software può invece cambiare durante il funzionamento.

In ricezione e in trasmissione sono utilizzati 2 buffer indipendenti ciascuno di 128 byte o caratteri. Per semplicità di programmazione PLC, a seconda dell'utilizzo, ciascun buffer può essere indirizzato con nomi diversi, uno in formato "Byte" oppure una equivalente variabile tipo "Stringa" utile nel caso di messaggi.

Quindi per i dati ricevuti si utilizza il buffer RS1BIN (O RS1SIN) dovendo leggere sequenze di Byte oppure la stringa RS1SIN per prelevare una Stringa di caratteri.

Analogamente per i dati in uscita, RS1BOU (O RS1SOU) per le sequenze di Byte oppure la stringa RS1SOU per una Stringa di caratteri.

Nel caso di ricezione con protocollo a lunghezza fissa (RS1PRR>0), i caratteri ricevuti vengono posti nel buffer RS1BIN (O RS1SIN) e quando questi sono pervenuti tutti, nella variabile RS1CIN viene scritto il numero di caratteri previsti quindi RS1CIN cambia da 0 a RS1PRR. Il PLC preleva i caratteri in RS1BIN (o la stringa RS1SIN) e pone il contatore RS1CIN=0 per indicare che il buffer è pronto per una nuova ricezione.

Nel caso di ricezione con protocollo con terminatore (RS1PRR=0), vengono posti nel buffer RS1BIN (O RS1SIN) i caratteri pervenuti prima del carattere terminatore (predisposto in RS1SCR) e nella variabile RS1CIN viene scritto il numero di caratteri pervenuti escluso il terminatore. Il PLC, come nel caso precedente, preleva i caratteri in RS1BIN (O RS1SIN) e pone il contatore RS1CIN=0 per indicare che il buffer è pronto per una nuova ricezione.

In trasmissione, il PLC prima deve depositare i caratteri da trasmettere in RS1BOU e poi scrivere nella variabile RS1COU il numero di caratteri da trasmettere. Il CN quindi scarica sulla linea i dati del buffer e poi azzerà il contatore di caratteri RS1COU=0 per indicare che è pronto per una nuova trasmissione. Se la trasmissione riguarda un messaggio in formato stringa, occorre semplicemente copiare il messaggio nella variabile stringa RS1SOU. Il contatore di caratteri RS1COU viene automaticamente aggiornato con la lunghezza opportuna.

Nel caso di trasmissione con protocollo a lunghezza fissa (RS1PRW>0) l'informazione sul numero di caratteri scritta nella variabile RS1COU è ridondante perché coincidente con RS1PRW e serve solo come richiesta di invio. In caso di discordanza prevale RS1COU su RS1PRW.

2. Moduli interni dedicati

Nel caso di trasmissione con protocollo con terminatore (RS1PRW=0) questo viene inserito automaticamente dal CN e non entra nel conteggio dei caratteri.

In caso di errore la linea viene chiusa ponendo RS1ON=0 e segnalando il codice di errore in RS1ERR; il CN non genera nessun errore nel caso in cui i buffer non vengano svuotati: si limita ad aspettare, l'errore di overrun e' segnalato dalla piastra e significa che sono stati persi dei caratteri in ricezione. La gestione cn e' posta nella logica lenta del plc.

ESEMPIO PROGRAMMA PLC

```
[*****
[*   ESEMPIO per trasmettere / ricevere un messaggio      *
[*   _____                                          *
[*   DEMORS232.s_p      24-01-2002                      *
[*****
[L'esempio mostra l'invio di un messaggio formato da un testo
[ digitato da tastiera e seguito dalla quota dell'asse X.
[ In ricezione, visualizza il messaggio sul display, memorizza
[ i codici dei caratteri ricevuti e mostra come interpretare
[ un semplice comando codificato.

INP
OUT

RAM,16
NUMERO          [numero contenuto nel codice ricevuto

RAM,8
J
MEMO(128)       [128 byte di lavoro

STR,128
MSGIN           [memoria messaggio ricevuto
STRPOO         [stringa posizione asse
VALORE         [stringa cifre valore numerico

SOFTK,1
P01,L01,'  INVIO  MESSAGGIO','STRINGA =' ,STR:MSGOU

INIT
RS1ON=0         [chiude la linea per impostare i parametri
RS1PAR=00000010B [7bit,parità dispari,no XON/XOFF,RTS/CTS disabilitati
RS1BRT=9600     [9600 baude
RS1PRR=0        [protocollo con terminatore in RX
RS1SCR=0DH      [codice carattere terminatore in RX
RS1PRW=0        [protocollo con terminatore in TX
RS1SCW=0DH      [codice carattere terminatore in TX
RS1ON=-1        [richiesta di linea attiva
PROG
END
BURDY=0
CALL RXMSG      [visualizza in DISPL0 il messaggio ricevuto
IF(P01) L01=1   [prenota richiesta di trasmissione messaggio
IF(L01)CALL TXMSG [trasmette un messaggio digitato a tastiera
END
END
```

```
[
                                R O U T I N E S

[====ROUTINE DI TRASMISSIONE DEL MESSAGGIO CON LA POSIZIONE DELL'ASSE
X=====
TXMSG:$
IF(RS1ON<>1) RTS    [linea non aperta, attendi
IF(RS1COU>0) RTS    [buffer occupato, attendi
STRPOO=MKN$((NEI(POO(1)*1000))/1000)  [ stringa quota asse X
RS1SOU=MSGOU+' X '+STRPOO [componi stringa di uscita
L01=0                      [resetta richiesta di invio
RTS

[=====ROUTINE DI GESTIONE DEL MESSAGGIO RICEVUTO=====
RXMSG:$
IF(RS1ON<>1) RTS    [linea non aperta, attendi
IF(RS1CIN=0) RTS    [nessun carattere arrivato
CALL VISMSG         [SE IL MESSAGGIO DEVE ESSERE VISUALIZZATO
CALL STOREB         [SE INVECE I CODICI RICEVUTI DEVONO ESSERE MEMORIZZATI A
BYTE
CALL DECODE         [SE UN COMANDO TIPO "T1234" DEVE ESSERE DECODIFICATO
RS1CIN=0             [libera il buffer di ricezione
RTS

[=====ROUTINE PER LA VISUALIZZAZIONE DEL MESSAGGIO
RICEVUTO=====
VISMSG:DISPL,0,RS1SIN    [manda sul display
RTS

[=====ROUTINE PER MEMORIZZARE A BYTE I CODICI
RICEVUTI=====
STOREB:J=0
EXEC=RS1CIN            [carica contatore J
J=J+1
MEMO(J)=RS1BIN(J) [loop di trasferimento di J caratteri
ENDE
RTS

[=====ROUTINE PER DECODIFICARE UN MESSAGGIO TIPO "T1234"
DECODE:IF(RS1BIN(1)<>84) RTS    [controlla se T
VALORE =MID$(RS1SIN,2,4)      [prende 4 caratteri a partire dal secondo
NUMERO=VAL(VALORE)           [e li converte in un numero
RTS
```

2.5 DUAL DRIVE DIGITALE

Lo scopo dell'antigioco è di mantenere costante la coppia esercitata tra due motori che devono muoversi su una cremagliera.

Il sistema serve ad impedire che, al variare della velocità, vi possano essere dei battimenti tra i denti delle pulegge e quelli della cremagliera che causerebbero errori di velocità.

Il sistema utilizza le prescrizioni di coppia per ciascun motore.

Nella condizione di motori abilitati e riferimento uguale a zero queste assumono valori uguali ma di segno opposto.

Il valore assunto in questo caso è pari al doppio del valore di **OFFSET** impostato.

$$\text{OUTA} - \text{OUTB} = 2 \cdot \text{OFFSET}$$

Quando il riferimento è diverso da zero le due prescrizioni di coppia possono essere diverse per i due Drives a causa delle letture di velocità date dai trasduttori che dipendono dalla posizione del pignone sulla cremagliera.

In questo caso si avrà una differenza tra le prescrizioni di coppia:

$$\text{OUTERR} = \text{OUTA} - \text{OUTB} - 2 \cdot \text{OFFSET} \quad \neq 0$$

La differenza di coppia prescritta contenuta nella variabile OUTERR viene filtrata da una rete di tipo proporzionale-integrativo: PIDERR e utilizzata come offset di riferimento di velocità da sommare ai riferimenti dei Drives.

$$\begin{aligned} \text{VRIFa} &= \text{VRIFD} - \text{OFFRIF} \\ \text{VRIFb} &= \text{VRIFD} + \text{OFFRIF} \end{aligned}$$

2.5.1 CONFIGURAZIONE ASSI

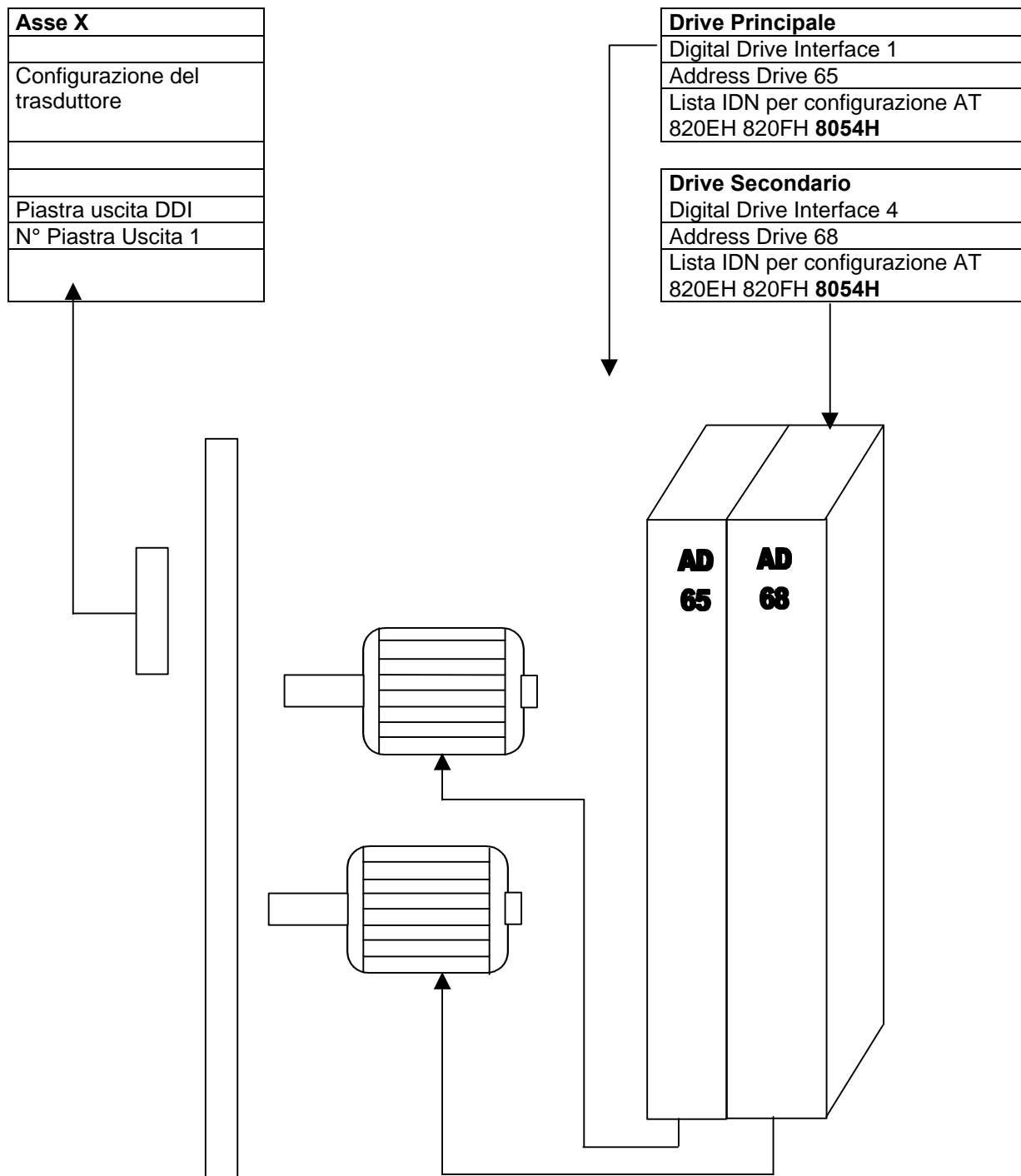
Si configurano i parametri dell'asse con antigioco con le consuete modalità, avendo cura di configurare gli ultimi tre parametri dell'ultima pagina di setup che vengono utilizzati dal CN per duplicare il riferimento di velocità del drive principale assegnandolo al drive secondario.

Piastra Uscita :	DDI
Numero Uscita Drive :	Numero del Drive principale
Piastra uscita secondaria :	DDI
Addr.slave D/A per RIO uscita secondaria :	0
N.uscita/Drive secondario :	Numero del Drive secondario

2.5.2 CONFIGURAZIONE DRIVES

Si configurano i Drive, principale e secondario che piloteranno i due motori con le modalità consuete.

Esempio: Volendo configurare l'asse X come asse con antigiooco si potrebbe utilizzare il drive numero uno come drive principale ed il drive numero quattro come drive secondario.



2. Moduli interni dedicati

2.5.3 DESCRIZIONE DELL'ALGORITMO DI CONTROLLO

Per poter essere implementato l'algoritmo di anti-gioco necessita dei seguenti parametri da configurare di tipo RAM,32:

PASSO	Avanzamento dell'asse per ogni giro motore [mm/g]
TNOM	Coppia nominale di un motore [Nm]

Per la taratura della rete proporzionale-integrativa sono necessari i seguenti parametri:

OFFSET	Offset di coppia per il twin Drive [Nm]
GP	Guadagno proporzionale compensatore twin Drive (PIDERR)
GI	Guadagno integrativo compensatore twin Drive (PIDERR)
FRI	Frequenza integrativa compensatore twin Drive (PIDERR)

Le variabili di lettura comando, feedback e di appoggio sono le seguenti:

OUTA	Lettura comando di coppia Drive primario
OUTB	Lettura comando di coppia Drive secondario
OUTERR	Ingresso compensatore twin Drive
OFFRIF	Uscita compensatore twin Drive [rpm]
TI	Costante di tempo rete integrativa
NWACC	Accumulatore aggiornato integratore

I comandi di coppia vengono letti nelle variabili **OUTA** e **OUTB** dai Drives mediante **IDN8054H** cui sono associate le variabili SRCATn(3) che divise per 2560 e moltiplicate per (TNOM/100) danno il valore effettivo di coppia prescritta.

NOTA: Si devono preferire le variabili SRCAT non scalate al posto DDAT già scalate per motivi di temporizzazione.

```
IF((DDRD(2)<3)~(DDRD(3)<3)~EMEA) NOTWIN
OUTA=SRCATM(3)/2560*TNOM/100      [M = N. drive asse principale]
OUTB=SRCATS(3)/2560*TNOM/100      [S = N. drive asse secondario]
```

La variabile di feedback in ingresso alla rete proporzionale-integrativa di feedback è OUTERR cui si sottrae l'OFFSET di coppia che devono avere i motori quando il riferimento di velocità è nullo.

$$\text{OUTERR} = \text{OUTA} - \text{OUTB} - 2 * \text{OFFSET}$$

La rete proporzionale-integrativa è implementata dall'algoritmo che segue:

$$\begin{aligned} \text{NWACC} &= (\text{OUTERR} - \text{NWACC}) * \text{TI} + \text{NWACC} \\ \text{OFFRIF} &= (\text{NWACC} * \text{GI} + \text{OUTERR}) * \text{GP} \end{aligned}$$

La variabile, definita come costante di tempo dell'integratore, viene calcolata in logica lenta in funzione della frequenza integrativa e del tempo di campionamento:

$$\text{TI} = (2 * \text{PI} * \text{SMPTI} / 1000 * \text{FRI}) \quad [\text{Aggiornamento TI twin drive}]$$

Dove **SMPTI** è il tempo di campionamento del CN in [ms].

La variabile **OFFRIF** è l'uscita della rete proporzionale-integrativa e va sommata ai riferimenti di velocità con segni opposti.

$$\text{OFSVA}(\mathbf{M}) = -\text{OFFRIF}$$

$$\text{OFSVA2}(\mathbf{M}) = \text{OFFRIF}$$

La variabile PLC OFSVA rappresenta l'offset di velocità da sommare alla prescrizione del Drive ed è espressa in [mm/min].

La variabile OFSVA2 rappresenta l'offset di velocità da sommare alla prescrizione del Drive secondario.

2.5.4 MODALITÀ DI TARATURA DEL SISTEMA

Prima di procedere alle operazioni di taratura è consigliabile creare un set di variabili da visualizzare con l'analizzatore grafico:

DDATM(1) Velocità Drive master [rpm]

DDATS(1) Velocità Drive slave [rpm]

DDATM(3) Percentuale di coppia nominale Drive master

DDATS(3) Percentuale di coppia nominale Drive slave

Per visualizzare le velocità e le coppie dei due Drives è necessario configurare preventivamente i test points TPA e TPB con gli indirizzi 0028H (Velocità) e 0050H (coppia).

Taratura degli assi

I due Drive: master e slave devono essere tarati separatamente con l'antigioco disabilitato.

Taratura della rete antigioco

Si procede innanzitutto con lo stabilire un offset di coppia da fornire ai due Drives e che può essere fissato intorno al 20% della coppia nominale dei motori.

Si verificano i valori di default di **GI**, **GP** ed **FRI** rispettivamente: guadagno integrativo, guadagno proporzionale e frequenza integrativa della rete PI antigioco:

$$\mathbf{GP} = 0.01 \quad \mathbf{GI} = 100 \quad \mathbf{FRI} = 5$$

Si abilitano i due Drives Master e Slave.

Tenendo gli assi fermi in coppia si aumenta il valore del guadagno proporzionale **GP** fino all'innesco delle oscillazioni alla frequenza di risonanza del sistema visibili sui segnali di percentuale di coppia nominale.

Il valore di **GP** da utilizzare è il **50%** circa di quello trovato al punto precedente.

Fissato il valore di **GP** si passa alla taratura della frequenza integrativa **FRI**: Si aumenta il valore di questa fino all'innesco delle oscillazioni alla frequenza di risonanza del sistema.

Il valore di **FRI** da utilizzare è il **50%** circa di quello trovato al punto precedente.

Con i valori così ottenuti si può procedere nell'esecuzione di un semplice programma di avanti indietro con pause, in modo da verificare che i Drive si fermino con i riferimenti di coppia opposti.

Osservando i profili di velocità, coppia ed errore di inseguimento si potranno ritoccare ulteriormente i parametri di cui sopra.

La taratura del guadagno *Kv* e della compensazione dinamica *Kc* va fatta con l'antigioco abilitato.

2.5.5 LOGICA DI ABILITAZIONE E DISABILITAZIONE

In normale funzionamento i due Drives devono essere abilitati e disabilitati contemporaneamente.

Si devono abilitare separatamente i due Drives in fase di taratura prevedendo una variabile di appoggio gestita da da softkey (vedere esempio di PLC allegato).

2.5.6 LOGICA DI EMERGENZA

Nel caso di allarmi del Drive o di richieste di emergenza da parte del controllo numerico non si può garantire un funzionamento coerente delle rampe di decelerazione dei due Drives (Si prenda ad esempio il caso di un Drive in allarme mentre l'altro non lo è).

Per questo motivo è opportuno togliere l'abilitazione software **DDEN** ad entrambi i Drives in modo da permettere all'asse di fermarsi per inerzia.

Eventuali gestioni di rampe di decelerazione rapide in emergenza devono essere valutate caso per caso a seconda delle possibilità della macchina.

ESEMPIO DI PLC

```

N1      [-----]
N2      [          ESEMPIO ROUTINE PER COMPENSAZIONE GIOCO          ]
N3      [-----]
N4      [          GIOCOE      981124          ]
N5      [-----]
N6      [  Asse 2-Y: asse twin Drive interpolato                    ]
N7      [  DDrive 2 : twin Drive primario                          ]
N8      [  DDrive 3 : twin Drive secondario                        ]
N9      [-----]
N10     [-----]
N11     RAM,32
N12     [
N13     [          Parametri da configurare _____ ]
N14     PASSO  [ Avanzamento dell'asse per ogni giro motore [mm/g] ]
N15     TNOM   [ Coppia nominale di un motore [Nm] ]
N16     [-----]
N17     [
N18     [          Parametri da tarare _____ ]
N19     OFFSET [ Offset di coppia per il twin Drive [Nm] ]
N20     GP      [ Guadagno proporzionale compensatore twin Drive ]
N21     GI      [ Guadagno integrativo compensatore twin Drive ]
N22     FRI     [ Frequenza integrativa compensatore twin Drive ]
N23     [-----]
N24     [
N25     [          Variabili Comando e Feedback _____ ]
N26     OUTA    [ Lettura comando di coppia Drive primario ]
N27     OUTB    [ Lettura comando di coppia Drive secondario ]
N28     OUTERR  [ Ingresso compensatore twin Drive ]
N29     OFFRIF  [ Uscita compensatore twin Drive ]
N30     [-----]
N31     [
N32     [          Variabili di appoggio _____ ]
N33     TI      [ Costante di tempo rete integrativa ]
N34     NWACC   [ Accumulatore aggiornato integratore ]
N35     ACC     [ Accumulatore non aggiornato integratore ]
N36     [-----]
N37     [
N38     RAM,8
N39     PCSTEP  [ Contatore sequenziatore P.C. ]
N40     [
N41     RAM,1
N42     PCON    [ Flag P.C. attiva ]
N43     PCOK    [ Flag P.C. terminata correttamente ]
N44     PCKO    [ Flag P.C. terminata in modo errato ]
N45     [
N46     SOFTK,1
N47     P11,L11,1,'JOG -'
N48     P12,L12,1,'JOG +'
N49     P13,L13,0,'ENABLE  DRIVE  SLAVE '
N50     P14,L14,0,'ENABLE  DRIVE  MASTER '
N51     P15,L15,0,''
N52     P16,L16,0,''
N53     P17,L17,0,'RESET    ALLARMI '
N54     P18,L18,0,'EMERGENZA'
N55     [
N56     [----- SEZIONE INIZIALIZZAZIONE -----]
N57     INIT
N58     [
N59     [----- Inizializzazione Parametri comp. GIOCO -----]

```

2. Moduli interni dedicati

```

N60      [
N61      PASSO= ...           [ Vedi u.car avanzamento giro motore
N62      TNOM= ...           [ Vedi coppia nominale motore
N63      OFFSET= ...         [ 20% circa della coppia nominale
N64      GP=.01
N65      GI=100
N66      FRI=5
N67      [
N68      [----- Inizializzazione variabili generiche <-----
N69      .....
N70      [
N71      [----- SEZIONE SUPERVELOCE -----
N72      FAST
N73      [
N74      [----- COMPENSATORE TWIN DRIVE <-
N75      IF((DDRD(2)<3)~(DDRD(3)<3)~EMEA) NOTWIN
N76      OUTA=SRCAT2(3)/2560*TNOM/100    [preferibile fare una costante con
N77      OUTB=SRCAT3(3)/2560*TNOM/100    [ TNOM/100/2560 per velocizzare
N78      OUTERR=OUTA-OUTB-2*OFFSET
N79      ACC=NWACC
N80      NWACC=(OUTERR-ACC)*TI+ACC
N81      OFFRIF=(NWACC*GI+OUTERR)*GP
N82      TWIEND
N83      [
N84      [----- RESET COMPENSATORE IN EMEA O DISAB <-
N85      NOTWIN:
N86      ACC=0
N87      NWACC=0
N88      OFFRIF=0
N89      DDMD3(1)=0
N90      DDMD3(2)=0
N91      TWIEND:
N92      [
N93      [----- COMANDI AI DRIVES <-----
N94      OFSVA(2)=-OFFRIF
N95      OFSVA2(2)= OFFRIF
N96      [
N97      [
N98      [----- SEZIONE VELOCE -----
N99      PROG
N100     END
N101     [
N102     [----- SEZIONE LENTA -----
N103     IF("BURDY)ASINC
N104     BURDY=0
N105     ASINC:
N106     TI=(2*PI*SMPTI/1000*(FRI/(GI+1)) [Aggiornamento TI twin Drive
N107     [
N108     [----- GESTIONE POTENZIOMETRI <-----
N109     ...
N110     [----- GESTIONE JOG ←-----
N111     [
N112     ...
N113     ...
N114     ...
N115     ...
N116     ...
N117     ...
N118     [----- Abilitazioni SW assi twin Drive <-----
N119     IF(P13)L13="L13
N120     IF(P14)L14="L14
N121     [

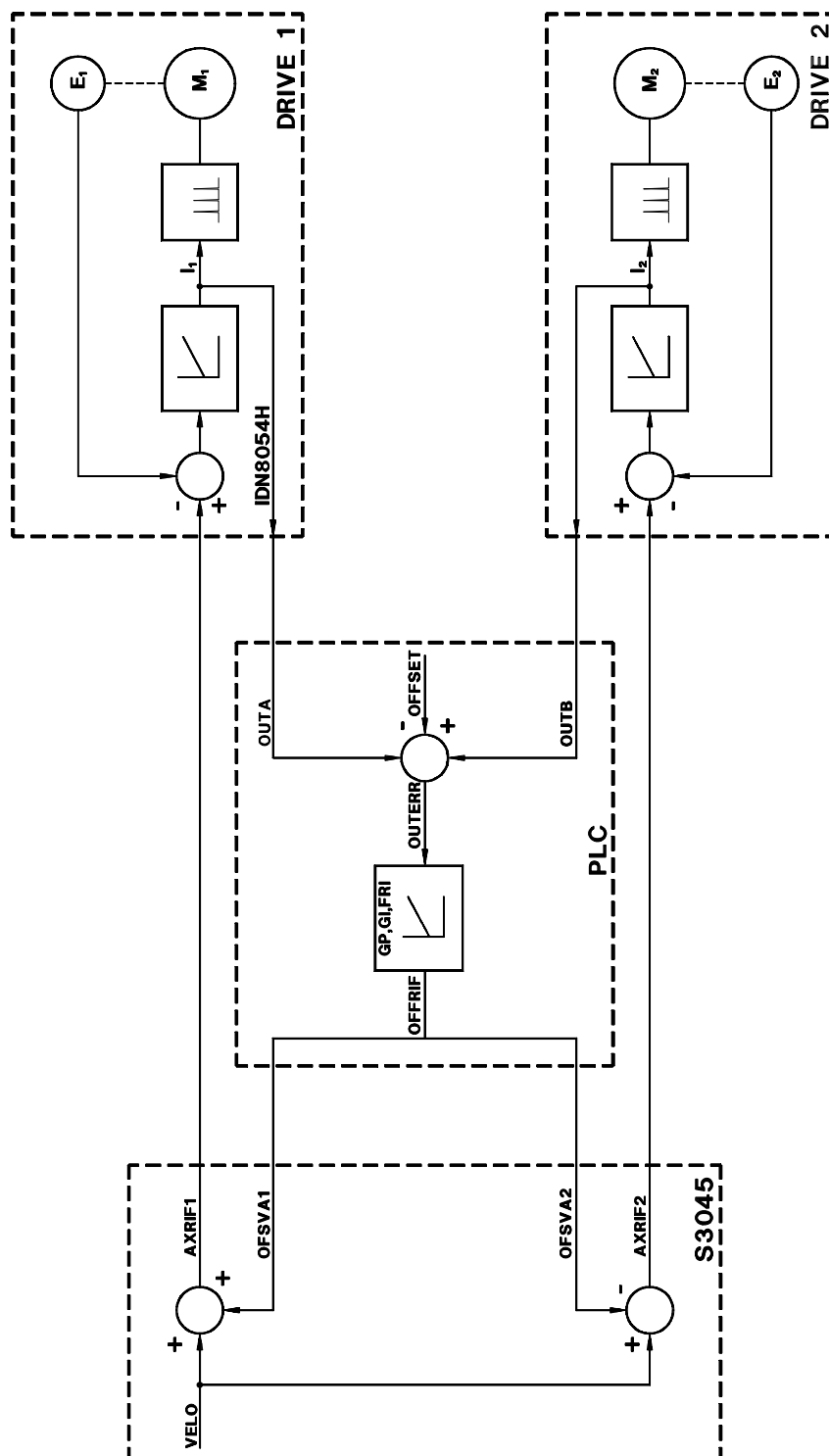
```

```

N122      DDEN(2)=( (DDRD(2)>1)&"DDC1D(2)~DDEN(2)&(DDRD(2)>2)$
           &(DDRD(3)>2)) [& ...SICUREZZE
N123      DDEN(3)=( (DDRD(3)>1)&"DDC1D(3)~DDEN(3)&(DDRD(3)>2)$
           &(DDRD(2)>2)) [& ...SICUREZZE
N124      [
N125      [----- Consensi di coppia twin Drive <-----
N126      [
N127      DDON(2)=DDEN(2)&MOVCN(2)&(DDRD(2)>1)&(DDRD(3)>1)
N128      DDON(3)=DDEN(3)&MOVCN(2)&(DDRD(3)>1)&(DDRD(2)>1)
N129      RDMOV(2)=DDON(2)&(DDRD(2)=3)~DDON(3)&(DDRD(3)=3)
N130      [
N131      [
N132      [----- GESTIONE EMERGENZE E GENERALI <-----
N133      REME=FF(P18~(DDC1D<>0)),(EMEA)
N134      L18=EMEA
N135      IF (EMEA)L14=0;L13=0
N136      [
N137      END[
N138      [..... FINE FILE .....

```

SCHEMA DESCRITTIVO DUAL DRIVE DIGITALE



3. ADATTAMENTO DI UN PROGRAMMA PLC DA S1200 A S4000

Quelle riportate nelle pagine che seguono sono le principali modifiche per ottenere la compatibilità dei programmi PLC scritti per il sistema S1200 con il linguaggio del sistema S4000, senza sfruttare le nuove potenzialità di linguaggio ed i MODULI INTERNI PER LA GESTIONE MANDRINO, ASSI INDIPENDENTI, CAMBI UTENSILI.

3. Adattamento di un programma PLC da S1200 a S4000

PROGRAMMA GENERICO SU S1200

```

INP
IMAPR          [Macchina pronta]
OUT
ABX            [abilitazione asse X]

[***** DICHIARAZ. VARIABILI *****]
RAM,32        [variabili]
LEPOTE         [lettura potenziometro]
POSX           [lettura della quota assoluta dell'asse X]
COMPX         [compensazione termica asse X]
VELX           [Convertitore asse X]
VEMA           [Velocità mandrino]
RAM,8
NUMUT          [Var. numerica per istruzione ASC()]

INIT
PROG

[***** GESTIONE POTENZIOMETRI *****]
POTER =1       [gestione potenziometri da PLC]
LEPOTE=LAD(POMA) [lettura ingresso potenziometro]
               [manuale e conversione]

formato
POMO=SDA(LEPOTE) [scrittura del valore per il CN e]
               [conversione formato]
POFO=SDA(LEPOTE)

[*** LETTURA QUOTE ASSI E SHIFT ORIGINI *****]
POSX=LRQ(POA(1)) [lettura quota asse X]
SHIFT(1)=SRQ(COMPX) [compens. asse X]

[ **** DECODIFICA FUNZIONI *****]
[Sintassi dell'istruzione COM, 1, 'LABEL']
IF(AUXM=6) COM, 1, 'L1'
RTS

[***** GESTIONE ABILITAZIONI *****]
ABX=MOVE(1)      [abilitazione asse X]

```

PROGRAMMA GENERICO SU S4000

```

INP
IMAPR          [Macchina pronta]
OUT
ABX            [abilitazione asse X]

[***** DICHIARAZ. VARIABILI *****]
[Sostituire RAM con SRAM in quanto le prime non sono più autoritative]
[allo spegnimento]
SRAM,32      [variabili]
LEPOTE         [lettura potenziometro]
POSX           [lettura della quota assoluta dell'asse X]
COMPX         [compensazione termica asse X]
VELX           [Convertitore asse X]
VEMA           [Velocità mandrino]
SRAM,8
NUMUT          [Var. numerica per istruzione MKN$()]
SRAM,1
[Le softkey di selezione dei passi dei volantini elettronici sono state eliminate, ma il PLC può selezionare uno dei passi predefiniti nei dati di configurazione con la variabile STEP]
SOFTK
P01,L01, 'passo .1 mm'
P02,L02, 'passo .5 mm'
P03,L03, 'passo 5 mm'
P04,L04, 'passo 10 mm'
P05,L05, 'ricerca zeri assi'
INIT
PROG

[***** GESTIONE POTENZIOMETRI *****]
[La gestione è sempre ad opera del PLC la variabile POTER è stata eliminata.]
[Occorre eliminare le funzioni LAD() e SDA(): le variabili relative a ingressi]
[Le uscite analogiche sono già in floating point.]
[Le variabili POFE, POMA, POSP sono state sostituite da ANI(1), ANI(2), ANI(3).]
[Per il modo manuale è presente un potenziometro per ogni asse]
LEPOTE=ANI(1)
POMO(1)=LEPOTE
POMO(2)=LEPOTE
POMO(3)=LEPOTE
POFO=LEPOTE

[***** LETTURA QUOTE ASSI E SHIFT ORIGINI *****]
[Occorre eliminare le funzioni LRQ() e SRQ(): le variabili relative a ingressi]
[Le uscite analogiche sono già in floating point.]
POSX= POA(1)    [lettura quota asse X]
SHIFT(1)=COMPX [compens. asse X]

[**** DECODIFICA FUNZIONI *****]
[Cambiare la sintassi dell'istruzione COM, 1, nomeprogramma]
IF(AUXM=6) COM, 1, 'CAMBUT'; RTS
[Su SSA si deve scrivere la configurazione assi in M11]
IF(AUXM=11) SSA=11111111B; RTS [Assi sempre attivi]
IF(AUXM=10) SSA=00000000B; RTS [Assi bloccati]

[***** GESTIONE ABILITAZIONI *****]
[Sostituire MOVE con MOVCN e fornire la cofiguraz. assi abilitati su]
[RDMOV]
ABX=MOVCN(1)    [abilitazione asse X]
RDMOV=MOVCN     [assi abilitati]

```


3. Adattamento di un programma PLC da S1200 a S4000
[*** GESTIONE MANDRINO *****]**

[Interamente implementata da PLC]

[*** GESTIONE CAMBIO UTENSILE *****]**

[Interamente implementata da PLC]

[*** ACQUISIZIONE DEL BREAK *****]**

[Sul Break viene emessa M30]

IF(AUXM=30) CALL M30

[*** GESTIONE MACCHINA PRONTA *****]**

MAPR=IMAPR

[arresto assi e programma]

[** VISUALIZZAZIONE MESSAGGI *****]**

[Sintassi dell'istruzione DISPL, riga (variabile)]

DISPL,1(MSG1) [visualizzazione MSG1]

[Conversione da numero a stringa]

MSG1=ASC(NUMUT)

[** SCRITTURA USCITE ANALOGICHE **]**

OEDA(1)=1

[abilitazione scrittura DAA X]

DAA(1)=SDA(VELX)

[Convertitore asse X]

DASP=SDA(VEMA)

[Velocità mandrino]

[** MOVIMENTI MANUALI IN JOG *****]**

[In manuale solo jog]

[** RICERCA ZERI ASSI *****]**

[Gestione non remotabile da parte del CN]

IF(NCMD=6) ...

[** GESTIONE PASSI MANOPOLE *****]**

[Gestione non remotabile da parte del CN]

END

[*** GESTIONE MANDRINO *****]**

[Non trasformabile con semplice sostituzioni, vedere il paragrafo relativo.]

[*** GESTIONE CAMBIO UTENSILE *****]**

[Non trasformabile con semplice sostituzioni, vedere il paragrafo relativo.]

[*** ACQUISIZIONE DEL BREAK *****]**

[Sul Break non viene emessa M30.]

[Viene settato il segnale impulsivo BRKA: occorre chiamare la routine di Break]

IF(AUXM=30) CALL M30

[CALL alla routine di M30]

IF(BRKA) CALL M30

[CALL alla routine di BREAK]

[*** GESTIONE MACCHINA PRONTA *****]**

[Il MAPR è stato sdoppiato nei suoi due significati]

DHOLD="IMAPR

[arresto avanzamento blocchi]

FHOLD="IMAPR

[arresto movimento assi]

[** VISUALIZZAZIONE MESSAGGI *****]**

[Cambiare la sintassi dell'istruzione DISPL, riga, variabile]

DISPL,1,MSG1

[visualizzazione MSG]

[sostituire la funzione ASC() con MKN\$()]

MSG1=MKN\$(NUMUT)

[** SCRITTURA USCITE ANALOGICHE **]**

[Eliminare gli OEDA() e le funzioni di conversione formato]

DAA(1)=VELX

[Convertitore asse X]

DASP=VEMA

[Velocità mandrino]

[** MOVIMENTI MANUALI IN JOG *****]**

[Per selezionare il movimento JOG in manuale occorre settare il registro]

[MOVMA]

MOVMA = JOGP ~ JOGM

[** RICERCA ZERI ASSI *****]**

[Non esiste più lo stato CN di RICERCA 0 (NCMD=6) in alternativa occorre scrivere la configurazione degli assi per cui ricercare lo zero sulle variabili MARK (ricerca senza micro) o MICZE (ricerca con micro).]

[Per esempio si può creare una softkey da PLC (P05,L05)]

L05=FF(P05),((NCMD<>5)~(MIZEA=7)) [lampada softk]

IF(L05) MICZE= 11111111B; ELSE MICZE=0 [con micro oppure]

IF(L05) MARK= 11111111B; ELSE MARK=0 [su marker]

[** GESTIONE PASSI MANOPOLE *****]**

[La scelta dei passi deve essere gestita dal PLC per poter utilizzare una eventuale console remotata.]

IF(P01) L01=1; L02=0; L03=0; L04=0

IF(P02) L02=1; L01=0; L03=0; L04=0

IF(P03) L03=1; L02=0; L01=0; L04=0

IF(P04) L04=1; L02=0; L03=0; L01=0

IF(L01) STEP=1 [selezione primo passo predefinito]

IF(L02) STEP=2 [selezione secondo passo predefinito]

IF(L03) STEP=3 [selezione terzo passo predefinito]

IF(L04) STEP=4 [selezione quarto passo predefinito]

END

4. RIASSUNTO DEI SEGNALI E REGISTRI PREDEFINITI

4.1. SIMBOLI E CONVENZIONI

In questo capitolo sono riassunti tutti i registri e le variabili predefinite che il Controllo Numerico (CN) scambia con il programma di Logica di Macchina (PLC) e viceversa.

Per ognuno di essi, raggruppati per argomento, è specificato:

- il nome mnemonico
- il formato (nella colonna **Dim**)
 - 1 = bit
 - 8 = byte
 - 16 = word
 - 32 = floating point
 - 64 = double floating point
 - STR = stringa di caratteri
- i vincoli di sincronismo con il segnale **BURDY** (nella colonna **Scr**)
- la direzione dell'informazione: da CN a PLC, viceversa o in entrambe le direzioni (nella colonna **Direzione**)
- una breve **Descrizione** nella colonna relativa

Per avere informazioni più dettagliate, quando necessarie, occorre consultare i paragrafi specifici in questa parte del presente manuale.

Le **unità di misura** utilizzate sono:

- | | |
|---|------------------------|
| - per quote, distanze, compensazioni di misura: | mm |
| - per dimensioni rotative | gradi |
| - per tempi | msec, sec o min |
| - per velocità: | mm/min |
| - per accelerazioni: | mm/(sec ²) |
| - per velocità mandrino | giri/min |
| - per tensioni | V |

4. Riassunto dei segnali e registri predefiniti

La **simbologia** usata è la seguente:

I caratteri **()** dopo il nome di un registro indicano che si tratta di un vettore di più elementi nel formato specificato (esempio **UTNUM()** indica che si tratta di un vettore, mentre **MOV CN** è un singolo registro).

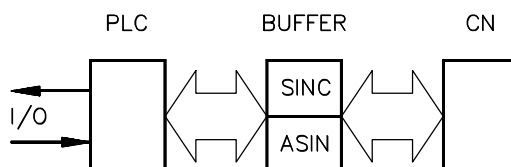
Se, all'interno delle liste che seguono, compare il simbolo **(1..n)** si intende che il registro o il vettore devono essere interpretati analizzandone singolarmente gli elementi da (1 a n). Per discriminare un singolo registro di cui si descrivono i bit occorre tener conto che:

- Gli elementi di un vettore hanno dimensione superiore ad 1.
- Quando sono descritti i singoli bit di un registro, solitamente dette descrizioni sono precedute dalla descrizione del registro stesso, che sarà indicato senza parentesi.

Esempio:

Nome	Dim	Direzione	Scr	Descrizione
MOV CN	8	CN ⇨ PLC	no	Richieste di abilitazione al movimento degli assi (1..8).
MOV CN(1)	1	CN ⇨ PLC	no	(primo bit del byte) è la richiesta di abilitazione dell'asse 1
MOV CN(8)	1	CN ⇨ PLC	no	(ottavo bit del byte) è la richiesta di abilitazione dell'asse 8
UTNUM()	16	CN ⇔ PLC	no	Codice degli utensili in tabella (1 ... UTENRI), dove UTENRI rappresenta il numero delle righe della tabella utensili.
UTNUM(1)	16	CN ⇔ PLC	no	(primo elemento del vettore word) è il codice dell'utensile presente nella riga 1 della tabella utensili.
UTNUM(8)	16	CN ⇔ PLC	no	(ottavo elemento del vettore word) è il codice dell'utensile presente nella riga 8 della tabella utensili.

Nota: Le intestazioni delle colonne non sono riportate nelle tabelle del testo per non appesantirne la leggibilità. Si prega quindi di prendere nota in questa sede della posizione delle informazioni contenute nelle singole colonne.



4.2. MOVIMENTI E SEGNALI D'INTERSCAMBIO

Stati del CN

NCMD	8	CN \Rightarrow PLC	no	Codice di stato del funzionamento del CN: 1 = rilevatore di quote 2 = blocco singolo 3 = esecuzione programma in semiautomatico 4 = esecuzione programma in automatico 5 = manuale 8 = ritorno sul profilo 9 = manuali attivi in stato di hold
STBMD	1	CN \Rightarrow PLC	no	Strobe impulsivo di segnalazione della variazione dello stato del CN; ha durata pari ad una scansione completa di logica lenta.
FNCMD	8	CN \Leftarrow PLC	no	Registro di forzamento CN in stato di esecuzione programma in semiautomatico.

Colloquio sincrono con il CN

BURDY	1	CN \Leftrightarrow PLC	si	Segnalazione della presenza di nuove informazioni sincrone per la logica di macchina in arrivo dal CN. Viene settato dal CN e <i>deve assolutamente essere resettato dal PLC non appena acquisite le informazioni.</i>
--------------	---	--------------------------	----	--

Trasmissione funzioni ausiliarie sincrone e preparatorie

AUXM	16	CN \Rightarrow PLC	si	Codice funzione ausiliaria M ultima programmata (M0-M9999).
STROM	1	CN \Rightarrow PLC	si	Strobe presenza funzione M.
TOOL	16	CN \Rightarrow PLC	si	Codice funzione T ultima programmata (T0-T32767).
STROT	1	CN \Rightarrow PLC	si	Strobe presenza funzione T.
AUXH	16	CN \Rightarrow PLC	si	Codice funzione ausiliaria H ultima programmata (H0-H9999).
STROH	1	CN \Rightarrow PLC	si	Strobe presenza funzione H.
SPEED	64	CN \Rightarrow PLC	si	Codice funzione ausiliaria S ultima programmata (S0-S99999).
STROS	1	CN \Rightarrow PLC	si	Strobe presenza funzione S.
STCOM	1	CN \Rightarrow PLC	si	Strobe di segnalazione del termine esecuzione su sottoprogrammi COM.
FEED	64	CN \Rightarrow PLC	no	Ultima Feed programmata.
AUXG	16	CN \Rightarrow PLC	no	Codice funzione ausiliaria G ultima programmata (G0-G9999).
CICFI	16	CN \Rightarrow PLC	no	Codice del ciclo fisso in esecuzione.
AXPGM	8	CN \Rightarrow PLC	si	Assi i cui nomi sono programmati nel blocco insieme alla funzione ausiliaria (es. M11XYZ genera AXPGM=00000111B).
AUXVAL()	64	CN \Rightarrow PLC	si	Array su cui vengono trasmessi i parametri I, J, K, Q programmati insieme alle funzioni ausiliarie M, H. AUXVAL(1) = parametro I AUXVAL(2) = parametro J AUXVAL(3) = parametro K AUXVAL(4) = parametro Q
STRAUX	8	CN \Rightarrow PLC	si	Strobe dei parametri I, J, K, Q. STRAUX(1) = strobe di I STRAUX(2) = strobe di J STRAUX(3) = strobe di K STRAUX(4) = strobe di Q

4. Riassunto dei segnali e registri predefiniti

RCOM	1	CN	⇐	PLC	Richiesta attivazione di una COM asincrona.
STRCOM	1	CN	⇒	PLC	Strobe di sincronizzazione per il lancio della COM richiesta con RCOM.
RCOM_	8	CN	⇐	PLC	Richieste di attivazione COM asincrona per i singoli gruppi d'assi (1..8).
STRCO_	8	CN	⇒	PLC	Strobe di sincronizzazione per il lancio della COM richieste con RCOM_ per i singoli gruppi d'assi (1..8).

Comandi asincroni di Start, Arresti, Allarmi, Consensi

DHOLD	1	CN	⇐	PLC no	Arresto temporaneo dell'avanzamento del programma a partire dal primo blocco successivo che contiene un punto di arresto nel movimento in continuo (tipicamente una funzione ausiliaria), senza interruzione dell'attività in corso.
FHOLD	1	CN	⇐	PLC no	Arresto temporaneo del movimento in esecuzione.
RHOLD	1	CN	⇐	PLC no	Richiesta di HOLD esterna. Arresto temporaneo dei movimenti programmati e dei blocchi in esecuzione.
HOLDA	1	CN	⇒	PLC no	Assi fermi in stato di Hold.
CYST	1	CN	⇐	PLC no	Richiesta di START CICLO esterna.
SFKGRD	8	CN	⇒	PLC no	Tasto guardia
SFKCNS	8	CN	⇒	PLC no	Segnalazioni impulsive pressione tasti START CICLO (SFKCNS(1)), HOLD (SFKCNS(2)), BREAK (SFKCNS(3)).
CYON	1	CN	⇒	PLC no	Blocco in esecuzione.
REME	1	CN	⇐	PLC no	Richiesta di EMERGENZA esterna.
EMEA	1	CN	⇒	PLC no	CN in stato di emergenza per allarmi interni o richiesta di emergenza esterna.
RBRK	1	CN	⇐	PLC no	Richiesta di BREAK esterna. Interruzione del programma o blocco in corso. Uscita dallo stato di emergenza.
BRKA	1	CN	⇒	PLC no	Ordine impulsivo di BREAK per il PLC.

Origini e correttori lunghezza utensili

OFST	16	CN	⇐	PLC si	Codice del correttore lunghezza da attivare.
INTOF	1	CN	⇐	PLC si	Strobe di segnalazione al CN per l'attivazione del correttore selezionato.
ORIG	16	CN	⇐	PLC si	Codice dell'origine pezzo da attivare.
INORG	1	CN	⇐	PLC si	Strobe di segnalazione al CN per l'attivazione dell'origine pezzo selezionata.
BYORG	1	CN	⇐	PLC si	Annullamento temporaneo di origini e correttori (origine assoluta).
ABSOR	1	CN	⇒	PLC no	Segnalazione di origine assoluta attiva.
STORG_	8	CN	⇐	PLC	Registro di attivazione degli offset origini aggiuntivi. STORG_(1) = 1 attiva gli offset (per tutti gli assi) STORG_(1) = 0 disattiva gli offset
PLORG()	8	CN	⇐	PLC	Registri contenenti gli offset di origine aggiuntivi

Abilitazione e bloccaggio assi

MOVCN	8	CN	⇒	PLC no	Richieste di abilitazione al movimento degli assi (1..8).
RDMOV	8	CN	⇐	PLC no	Assi abilitati al movimento; risposta al MOVCN (1..8).
POFO	64	CN	⇐	PLC no	Valore dell'override sulla Feed programmata (da 0 a 200 permette una regolazione tra 0 e 200 per cento – oltre 100% solo per prova programma).

4. Riassunto dei segnali e registri predefiniti
Assi sempre attivi o con bloccaggio

SSA 8 CN ⇐ PLC no Assi che devono essere sempre attivi (1..8).

Abbandono degli assi

DSERV 8 CN ⇐ PLC no Assi da abbandonare (1..8).

Abbandono dei trasduttori

DISRQ 8 CN ⇐ PLC no Assi con i trasduttori abbandonati (1..8).

Movimento manuale JOG

MOVMA 8 CN ⇐ PLC no Assi selezionati in movimento manuale (1..8).

JOGP 8 CN ⇐ PLC no Comandi di jog positivo (1..8).

JOGM 8 CN ⇐ PLC no Comandi di jog negativo (1..8).

POMO() 64 CN ⇐ PLC no Velocità per i movimenti manuali e di ricerca zero di ogni singolo asse (1..8) (da 0 a 1 come percentuale della velocità di rapido).

Movimenti in manuale volantini

HWL() 8 CN ⇐ PLC no Uno per volantino (1..3) per indicare il numero dell'asse da comandare.

STEP 8 CN ⇐ PLC no Selezione del passo volantino tra gli 8 valori definiti nei parametri di configurazione.

Ricerca zero assi

MICZE 8 CN ⇐ PLC no Assi selezionati in ricerca zero con microinterruttore (1..8).

MARK 8 CN ⇐ PLC no Assi selezionati in ricerca zero senza microinterruttore (1..8).

MIZER 8 CN ⇐ PLC no Microinterruttore di zero assi (1..8).

MIZEA 8 CN ⇒ PLC no Assi riferiti allo zero elettrico del trasduttore (1..8).

Movimento in manuale e ricerca zero durante la lavorazione

FOMAN 8 CN ⇐ PLC si Assi su cui forzare i comandi manuali contemporaneamente alla lavorazione (1..8).

Informazioni sugli assi
Per l'asservimento degli assi

ERR() 64 CN ⇒ PLC no Errori di inseguimento assi (1..8).

VATT 64 CN ⇒ PLC no Velocità attuale teorica imposta sul percorso utensile.

TACH() 64 CN ⇒ PLC no Velocità assi (1..8) misurate dai trasduttori di posizione.

VFF() 64 CN ⇒ PLC no Velocità istantanea prescritta per gli assi (1..8).

AFF() 64 CN ⇒ PLC no Accelerazione istantanea prescritta per gli assi (1..8).

DAA() 64 CN ⇒ PLC no Riferimenti di tensione per assi controllati (1..8). Se l'asse è attivo e sotto il controllo del CN, il suo DAA può solamente essere letto. Il contenuto varia fra -1 e 1 in corrispondenza di tensioni in uscita fra -10 e +10 V.

POA() 64 CN ⇒ PLC no Posizioni riferite all'origine assoluta assi (1..8).

POO() 64 CN ⇒ PLC no Posizioni assi riferite all'origine corrente e correttore attivo (1..8).

4. Riassunto dei segnali e registri predefiniti

POATE()	64	CN	⇒	PLC no	Quota teorica istantanea degli assi controllati lungo la traiettoria di interpolazione (1..8) riferita all'origine assoluta.
POOTE()	64	CN	⇒	PLC no	Quota teorica istantanea degli assi controllati lungo la traiettoria di interpolazione (1..8) riferita all'origine attiva.
POORT()	64	CN	⇒	PLC no	Quota teorica istantanea soggetta ad eventuali rototraslazioni del sistema di coordinate per gli assi controllati, lungo la traiettoria di interpolazione (1..8) riferita all'origine attiva.
PFCN()	64	CN	⇒	PLC no	Quota finale programmata per gli assi (1..8).
AXRIF()	64	CN	⇒	PLC	Comando di velocità inviato agli assi (1..8) [mm/min]
OFSVA()	64	CN	⇐	PLC	Offset aggiuntivo di velocità per gli assi (1..8) [mm/min]. (Influenza anche AXRIF() - da usare solo per applicazioni particolari)
AFF()	64	CN	⇒	PLC	Comando di accelerazione impresso agli assi (1..8) [mm/sec ²]
OFHWL()	64	CN	⇒	PLC	Spostamenti (1..8) dell'origine con G851 (in mm).
GDAVIS	8	CN	⇒	PLC	Numero del gruppo d'assi al quale si riferisce la visualizzazione.

Per lo stato degli assi

INTOL	8	CN	⇒	PLC no	Assi (1..8) entro la "tolleranza di posizione asse" dichiarata nei dati di configurazione.
JOGIN	8	CN	⇒	PLC no	Assi (1..8) in movimento in seguito a comando JOG (manuale o ricerca 0).
RAPI	1	CN	⇒	PLC no	Blocco in esecuzione con velocità rapido.

Per il controllo dei sistemi di misura e dei volantini elettronici

MKSAX	8	CN	⇒	PLC no	Segnalazione del marker (zero elettrico) degli assi nel caso di trasduttori encoder o righe ottiche (1..8). Viene settato dal CN quando si incontra il marker elettrico del trasduttore, ed è resettato nella campionatura di sistema successiva; per questo motivo è visibile sull'analizzatore grafico solamente con base tempo uguale alla campionatura di sistema.
AIRGP()	64	CN	⇒	PLC no	Livello del segnale nel caso di trasduttori analogici (INDUCTOSYN o RESOLVER) oppure numero degli impulsi persi nel caso di ENCODER con l'algoritmo "recupero step" presente per gli assi(1..8).
SPMANO()	64	CN	⇒	PLC no	Spazio comandato da percorrere con i volantini (1..3) secondo il passo selezionato. Lo spazio accumulato viene resettato sui cambi di stato CN e dalle variazioni sugli assi (SSA, DSERV ...)

Compensazioni dinamiche degli assi

SHIFT()	64	CN	⇐	PLC no	Compensazione dinamica delle quote assi (1..8).
----------------	----	----	---	--------	---

Offset per assi controllati

OFSDA()	64	CN	⇐	PLC no	Offset da applicare al riferimento analogico sui canali degli assi controllati (1..8) nel range ±1 per tensioni tra ±10 Volt.
----------------	----	----	---	--------	---

Gestione del probe di misura on/off

CWDTF	8	CN	⇐	PLC no	Byte di controllo del Probe on/off: Bit 1: disabilitazione errore 210 (urto)
SWDTF	8	CN	⇒	PLC	Stato del probe ON/OFF di misura. SWDTF(2) = 0 probe a riposo = 1 probe deflesso

4. Riassunto dei segnali e registri predefiniti
Fine corsa software assi controllati

FICOP	8	CN	⇒	PLC	no	Assi (1..8) in fine corsa software positivo.
FICOM	8	CN	⇒	PLC	no	Assi (1..8) in fine corsa software negativo.
DFCOP	8	CN	⇐	PLC	no	Disabilitazione finecorsa software positivi assi (1..8).
DFCOM	8	CN	⇐	PLC	no	Disabilitazione finecorsa software negativi assi (1..8).
FCA()	8	CN	⇐	PLC	no	Array abilitazioni fine corsa secondari.
CWFCS	8	CN	⇐	PLC		Controllo errori finecorsa software.
						CWFCS(1) = 1 segnalazione errore E93 disabilitata
						CWFCS (1) = 0 segnalazione errore E93 abilitata

Gestione assi paralleli (Gantry)

OFSGY	8	CN	⇐	PLC	no	Abilitazione offset nominale per assi gantry (1..8). Deve essere settato il bit corrispondente al numero asse SLAVE.
--------------	---	----	---	-----	----	--

Gestione assi programmabili non controllati

AUXPF()	64	CN	⇒	PLC	si	Quote programmate per gli assi che devono essere movimentati tramite il PLC (1..6).
STRPF	8	CN	⇒	PLC	si	Strobe di presenza nuove informazioni su AUXPF() (1..6).

Lettura ingressi e scrittura uscite analogiche

ANIx()	64	CN	⇒	PLC	no	Lettura ingressi analogici della scheda I/O MIX specificata e delle sue espansioni. Il valore letto varia fra 0 e 1 come percentuale del valore di fondo scala.
VELOx()	64	CN	⇐	PLC	no	Scrittura delle uscite analogiche della scheda I/O MIX specificata e delle sue espansioni. Queste uscite possono sempre essere lette, ma scritte solo se non sono utilizzate dal CN per gli assi controllati o dai moduli interni per la gestione mandrini o assi indipendenti. Il contenuto può variare fra -1 e 1 come percentuale del valore di fondo scala (±10 V).
TEMPx()	64	CN	⇒	PLC	no	Valore in gradi della temperatura rilevata dalle sonde termiche (se presente l'interfaccia) relative alla scheda specificata.

Scambio dati tra PLC e Part program

VPLFL	32	CN	⇒	PLC	si	Variabile in FLOATING da part program al PLC.
STVFL	1	CN	⇒	PLC	si	Strobe presenza variabile in FLOATING da part program al PLC.
VPLWO	16	CN	⇒	PLC	si	Variabile WORD da part program al PLC.
STVWO	1	CN	⇒	PLC	si	Strobe presenza variabile WORD da part program al PLC.
VPLBY	8	CN	⇒	PLC	si	Variabile BYTE da part program al PLC.
STVBY	1	CN	⇒	PLC	si	Strobe presenza variabile BYTE da part program al PLC.
VPLBI	1	CN	⇒	PLC	si	Variabile BIT da part program al PLC.
STVBI	1	CN	⇒	PLC	si	Strobe presenza variabile BIT da part program al PLC.
VLPFL	64	CN	⇐	PLC	si	Variabile in FLOATING inviata al part program dal PLC.
VLPWO	16	CN	⇐	PLC	si	Variabile WORD inviata al part program dal PLC.
VLPBY	8	CN	⇐	PLC	si	Variabile BYTE inviata al part program dal PLC.
VLPBI	1	CN	⇐	PLC	si	Variabile BIT inviata al part program dal PLC.
PCN()	32	CN	⇔	PLC	no	99 parametri in formato floating point condivisi, cioè leggibili e scrivibili sia da PLC che da part program a livello utente (1..99).
P()	32	CN	⇔	PLC		99 parametri in formato floating point condivisi, cioè leggibili e scrivibili sia da PLC che da sottoprogrammi lanciati dal PLC con le istruzioni COM (1..200).

4. Riassunto dei segnali e registri predefiniti

Finestre per la visualizzazione su video CN

WINDOW()	64	CN	↔	PLC	no	Registri per visualizzazione sul video del CN nelle aree prefissate (1..128) nei formati floating point long o double. La visualizzazione di queste aree è attivata tramite un condizionamento, nelle tabelle video di default fornite con il CN, dal corrispondente ASCW diverso da 0.
ASCW()	8	CN	↔	PLC	no	Registri per visualizzazione di caratteri sul video del CN nelle aree prefissate (1..128). Occorre fornire il codice ASCII del carattere.
WNDINT()	16	CN	↔	PLC	no	Registri per visualizzazione sul video del CN nelle aree prefissate (1..128) nel formato word.
WNDSTR()	str	CN	↔	PLC	no	Registri stringa contenenti un max di 64 caratteri alfanumerici per la visualizzazione sul video del CN nelle aree prefissate (1..128).
GIRMI	64	CN	↔	PLC	no	Registro per la visualizzazione del valore della funzione S nell'area prefissata sul video CN.
SFKMEN	8	CN	↔	PLC	no	Menu corrente di softkey del PLC.
SFKLNG	16	CN	⇒	PLC	no	Codice lingua attiva sul CN.
CNDVIS()	16	CN	↔	PLC	no	Array di word da utilizzare per i condizionamenti all' interno delle tabelle video (1..64).

Data e ora del sistema

DATE(1)	16	CN	⇒	PLC	no	Anno (ultime due cifre)
DATE(2)	16	CN	⇒	PLC	no	Mese
DATE(3)	16	CN	⇒	PLC	no	Giorno del mese
DATE(4)	16	CN	⇒	PLC	no	Ora (0-24)
DATE(5)	16	CN	⇒	PLC	no	Minuti
DATE(6)	16	CN	⇒	PLC	no	Secondi

Segnali per la copiatura e digitalizzazione superfici

COPIA	8	CN	↔	PLC	no	Primo byte per gestione remotata comandi copiatura
COPIA(1)	1	CN	↔	PLC	no	= 0 seleziona la modalità di digitalizzazione continua, cioè i punti vengono memorizzati in funzione dei parametri impostati da programma in copiatura manuale. = 1 seleziona la modalità di digitalizzazione ad impulsi, cioè i punti vengono memorizzati solo in seguito ad un impulso (transizione da 0 a 1) sul bit COPIA(2) in copiatura manuale.
COPIA(2)	1	CN	↔	PLC	no	Impulso di digitalizzazione; vedere COPIA(1) .
COPIA(3)	1	CN	↔	PLC	no	Segnalazione di ciclo di copiatura attivo. Quando resettato da PLC significa fine ciclo. E' importante terminare un ciclo con digitalizzazione azzerando questo bit (o con l'apposita softkey già implementata nel CN) altrimenti gli ultimi punti digitalizzati non vengono memorizzati.
COPIA(4)	1	CN	↔	PLC	no	Impulso di STEP +.
COPIA(5)	1	CN	↔	PLC	no	Impulso di STEP -.
COPIA(6)	1	CN	↔	PLC	no	Impulso di STEP ed inversione della passata.
COPIA(7)	1	CN	⇒	PLC	no	Asservimento di copiatura attiva.
COPIA(8)	1					Non assegnato

4. Riassunto dei segnali e registri predefiniti

COPIA2	8	CN ⇔ PLC	no	Secondo byte per gestione remotata comandi copiatura.
COPIA2(1)	1	CN ⇔ PLC	no	Impulso di passaggio in stato manuale.
COPIA2(2)	1	CN ⇔ PLC	no	0 = digitalizzazione punti disabilitata. 1 = digitalizzazione punti abilitata
COPIA2(3)	1	CN ⇔ PLC	no	Impulso di acquisizione offset tastatore.
COPIA2(4)	1	CN ⇔ PLC	no	1 = blocca 1° asse della terna attiva (di copiatura). 0 = sblocca
COPIA2(5)	1	CN ⇔ PLC	no	1 = blocca 2° asse della terna attiva (di copiatura). 0 = sblocca
COPIA2(6)	1	CN ⇔ PLC	no	1 = blocca 3° asse della terna attiva (di copiatura). 0 = sblocca
COPIA2(7)	1	CN ⇔ PLC	no	Impulso di inversione passata.
COPIA2(8)	1	CN ⇔ PLC	no	0 = autoapprendimento superficie disabilitato 1 = autoapprendimento della superficie attivo.
COPIA3	8	CN ⇔ PLC	no	Terzo byte per gestione remotata comandi copiatura.
COPIA3(1)	1	CN ⇔ PLC	no	Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione negativa asse 3.
COPIA3(2)	1	CN ⇔ PLC	no	Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione negativa asse 2.
COPIA3(3)	1	CN ⇔ PLC	no	Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione negativa asse 1.
COPIA3(4)	1	CN ⇔ PLC	no	Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione positiva asse 3.
COPIA3(5)	1	CN ⇔ PLC	no	Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione positiva asse 2.
COPIA3(6)	1	CN ⇔ PLC	no	Direzione di ripresa copiatura dopo perdita di contatto con il pezzo verso direzione positiva asse 1.
COPIA3(7)				Riserva
COPIA3(8)				Riserva
COPIA4	8	CN ⇔ PLC	no	Quarto byte per gestione remotata comandi copiatura.
COPIA4(1)	1	CN ⇔ PLC	no	Impulso di arresto tentativo di ripresa contatto.
COPIA4(2)				Riserva
COPIA4(3)				Riserva
COPIA4(4)				Riserva
COPIA4(5)				Riserva
COPIA4(6)				Riserva
COPIA4(7)				Riserva
COPIA4(8)				Riserva
PBSTS	8	CN ⇔ PLC	no	Registro di stato tastatore digitale.
POCOP	64	CN ⇔ PLC	no	Regolazione guadagno in copiatura manuale. Il valore ammesso può essere compreso tra 0 e 1 e moltiplica il guadagno dell'asservimento in copiatura manuale da 1 a 5. Ciò permette di regolare la velocità di movimento degli assi a parità di deflessione del tastatore da un minimo (non 0) ad un massimo.
COPIA	8	CN ⇔ PLC		Primo byte per gestione remotata dei comandi di copiatura COPIA(8) = 1 Segnalazione dell'esecuzione di un ciclo di copiatura in modalità Manuale

4. Riassunto dei segnali e registri predefiniti

Variabili di verifica dei tempi di utilizzo delle risorse

SMPTI	64	CN	⇒	PLC	no	Intervallo di campionatura sistema (assi controllati) in msec.
OCCV	16	CN	⇒	PLC	no	Tempo in microsecondi impiegato per la scansione della logica veloce.
OCCI	16	CN	⇒	PLC	no	Tempo in microsecondi impiegato per la gestione degli assi controllati.
OCCT	16	CN	⇒	PLC	no	Tempo in microsecondi impiegato dall'analizzatore grafico.
OCCP2P	16	CN	⇒	PLC	no	Tempo in microsecondi impiegato per la gestione degli assi indipendenti.
CCL	16	CN	⇒	PLC	no	Contatore dei cicli in cui viene spezzata la logica lenta.
CCUL	16	CN	⇒	PLC	no	Contatore cicli in cui viene spezzata la logica superlenta.

Segnalazione di errore accessibili dalla logica

ERSYS	16	CN	⇒	PLC	no	Codice dell'errore di sistema rilevato su assi controllati, mandrini, assi indipendenti, errori runtime PLC, errori su modulo cambio utensile automatico, ecc.
ERAXS	16	CN	⇒	PLC	no	Codice dell'errore rilevato su assi controllati (servo error, fuori tolleranza, errori su trasduttori, ecc.).
ERIOX	16	CN	⇒	PLC	no	Codice dell'errore rilevato sulle schede di I/O MIX (mancanza alimentazioni encoder, errore su output digitali, ecc.).
ERINT	16	CN	⇒	PLC	no	Codice dell'errore avvenuto durante i calcoli di interpolazione.
ERPLC	16	CN	⇒	PLC	no	Codice dell'errore runtime rilevato durante l'esecuzione del programma PLC (divisione per 0, overflow, underflow, ecc.).
ERSPN	16	CN	⇒	PLC	no	Codice dell'errore rilevato sui mandrini (trasduttori, ecc.).
ERP2P	16	CN	⇒	PLC	no	Codice dell'errore rilevato su assi indipendenti (trasduttori, ecc.).
ERCU	16	CN	⇒	PLC	no	Codice dell'errore rilevato durante il cambio utensile oppure tabella utensili non corretta, ecc.
ER2LN	16	CN	⇒	PLC	no	Codice dell'errore rilevato per superamento del tempo di campionatura di sistema.
ERCPY	16	CN	⇒	PLC	no	Codice dell'errore rilevato durante un ciclo di copiatura o relativo al tastatore.
FPERMK	8	CN	↔	PLC	no	Maschera di disabilitazione rilevazioni errori sui calcoli floating point (divisione per zero, overflow).

Costanti numeriche

PI	64	CN	⇒	PLC	no	Valore di Pi greco (3.14.15927)
-----------	----	----	---	-----	----	---------------------------------

Lettura e modifica dei parametri di configurazione assi

AXSTP	8	CN	↔	PLC	no	Numero dell'asse a cui sono riferiti i valori letti o scritti sui parametri di configurazione per mezzo del PLC.
VALSTP	64	CN	↔	PLC	no	Valore corrente letto o da scrivere nei parametri di configurazione sistema.
HOWSTP	16	CN	↔	PLC	no	Codice del parametro di configurazione a cui si vuole accedere per mezzo del PLC (finchè non si scrive su disco, i parametri agiscono su una copia in memoria non statica); i nuovi valori vengono attuati unicamente quando viene incontrato un punto con velocità finale = 0:

4. Riassunto dei segnali e registri predefiniti

				Codice scrittura	Parametro	Codice lettura
				0	Salvataggio dati permanente	
				-1	velocità rapido	1
				-2	accelerazione lavoro	2
				-3	accelerazione di rapido	3
				-4	gioco asse trasduttore	4
				-5	K _v guadagno	5
				-6	Compensazione dinamica	6
				-7	Velocità ripresa gioco	7
				-8	Tempo ripresa gioco	8
				-9	Servo error massimo	9
				-10	Velocità compensazione attrito	10
				-11	Compensazione accelerazione	11
				-12	Fine corsa negativo 1	12
				-13	Fine corsa positivo 1	13
				-14	Passo trasduttore	14
				-15	Costante tempo integratore	15
				-16	Guadagno integratore	16
ACTSTP	1	CN ⇌ PLC no			Segnale di attivazione operazione richiesta su HOWSTP . Viene resettato dal CN ad operazione conclusa.	
INCH	1	CN ⇌ PLC no			Tipo del sistema di misura 0 = millimetri 1 = pollici Questa variabile viene settata dal CN in funzione dell' apposito parametro nella configurazione impianto. Il PLC può sovrascriverla alterando il sistema di misura ma la scelta non può venire salvata in modo permanente nei parametri di configurazione.	
Varie						
_ENIDX	8	CN ⇌ PLC			attiva/disattiva la diagnostica di controllo della validità degli indici per l'accesso alle variabili singole ed ai vettori. _ENIDX = -1 diagnostica attivata _ENIDX = 0 diagnostica disattivata (default)	

4.3. MODULI DEDICATI
Rotazione mandrino

SPVEL()	64	CN ⇌ PLC no	Velocità di rotazione mandrini (1..4).
SPSSO()	64	CN ⇌ PLC no	Potenzimetro override mandrini (1..4).
SPDIR()	8	CN ⇌ PLC no	Sensi di rotazione mandrini (1..4).
SPROT	8	CN ⇌ PLC no	Comandi di rotazione mandrini (1..4).
SPREG	8	CN ⇌ PLC no	Mandrini (1..4) a regime.
SPMOT	8	CN ⇌ PLC no	Mandrini (1..4) in moto.
SPRMP	8	CN ⇌ PLC no	Mandrini (1..4) in rampa.
SPSGL	8	CN ⇌ PLC no	Velocità effettiva mandrini (1..4) entro soglia.

4. Riassunto dei segnali e registri predefiniti

Selezione del cambio gamma

SPGAM()	8	CN	⇐	PLC	no	Gamme mandrini inserite (0 = folle) (1..4).
SPPND	8	CN	⇐	PLC	no	Comandi di pendolamento per cambio gamma mandrini (1..4).
SPSMG1()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) per la gamma 1, impostate nella configurazione.
SPSMG2()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) per la gamma 2, impostate nella configurazione.
SPSMG3()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) per la gamma 3, impostate nella configurazione.
SPSMG4()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) per la gamma 4, impostate nella configurazione.
SPSMAX()	64	CN	⇒	PLC	no	Velocità massime mandrini (1..4) impostate nella configurazione.

Orientamento mandrino

SPORI()	8	CN	⇐	PLC	no	Comandi di orientamento mandrini (1..4).
SPTOL	8	CN	⇒	PLC	no	Mandrini orientati all'interno della tolleranza di posizionamento (1..4).
SPPOS()	64	CN	⇐	PLC	no	Posizioni di orientamento mandrini (1..4).
SPVEOR()	64	CN	⇐	PLC	no	Riduzione di velocità (da 0 a 1) durante l'orientamento (1..4).
SPOAB	8	CN	⇐	PLC	no	Selezioni per orientamento su quota assoluta (1..4).
SPORP	8	CN	⇐	PLC	no	Selezione di orientamento unidirezionale positivo.
SPORM	8	CN	⇐	PLC	no	Selezione di orientamento unidirezionale negativo.

Sincronismo tra mandrini

SPSYN	8	CN	⇐	PLC	no	Comandi di sincronismo mandrini (1..4) con mandrini slave.
SPMAS()	8	CN	⇐	PLC	no	Numero dei mandrini master per sincronismo con slave (1..4).
SPOFS()	64	CN	⇐	PLC	no	Offset per sincronismo fra mandrini master e slave (1..4).
SPRTO()	64	CN	⇐	PLC	no	Rapporti di velocità per sincronismo fra mandrini master e slave (1..4).
SPAGG	8	CN	⇒	PLC	no	Mandrini slave (1..4) sincronizzati con master.

Comuni a tutti i modi di funzionamento

SPMOV	8	CN	⇒	PLC	no	Richieste di abilitazione azionamento mandrini (1..4).
SPDIS	8	CN	⇐	PLC	no	Disabilitazioni generali azionamento mandrini (1..4).
SPDRQ	8	CN	⇐	PLC	no	Disabilitazioni del trasduttore mandrini (1..4).
SPTCH()	64	CN	⇐	PLC	no	Velocità effettiva mandrini (1..4).
PASP()	64	CN	⇒	PLC	no	Posizioni angolari mandrini lette dai trasduttori (1..4).
SPMZA	8	CN	⇐	PLC	no	Mandrini (1..4) con trasduttore riferito allo zero elettrico. Può essere resettato per ripetere la ricerca zero.
SPMKS	8	CN	⇒	PLC	no	Segnalazione del marker (zero elettrico) per gli encoder mandrini (1..4).
SPAGP()	8	CN	⇒	PLC		Livello trasduttore o impulsi persi e recuperati per i mandrini (1..4).
SPRIF()	64	CN	⇒	PLC		Comando di velocità inviato ai mandrini (1..4) [giri/min] può essere utilizzato per la verifica delle rampe di accelerazione/decelerazione paragonando SPRIF ed SPTCH (velocità effettiva) per mandrini con trasduttore.

Ciclo fisso G84

SPGDA	8	CN	⇐	PLC	no	Mandrino da utilizzare per il ciclo fisso G84 con trasduttore.
--------------	---	----	---	-----	----	--

Modulo di movimentazione assi indipendenti

MOV2P	8	CN ⇔ PLC	no	Richieste di abilitazione al movimento per assi indipendenti (1..8).
RDMP2P	8	CN ⇐ PLC	no	Assi indipendenti abilitati al movimento(1..8); risposte al MOV2P .
SSAP2P	8	CN ⇐ PLC	no	Assi indipendenti che devono essere sempre attivi (abilitati) (1..8).
DSVP2P	8	CN ⇐ PLC	no	Assi indipendenti da rilasciare (1..8).
DRQP2P	8	CN ⇐ PLC	no	Assi indipendenti di cui disabilitare i trasduttori (1..8).
MVMP2P	8	CN ⇐ PLC	no	Assi indipendenti selezionati in modo manuale (1..8).
MRKP2P	8	CN ⇐ PLC	no	Assi indipendenti selezionati in ricerca zero senza microinterruttore (1..8).
MCZP2P	8	CN ⇐ PLC	no	Assi indipendenti selezionati in ricerca zero con microinterruttore (1..8).
MIZP2P	8	CN ⇐ PLC	no	Microinterruttore di zero per assi indipendenti (1..8).
MZAP2P	8	CN ⇔ PLC	no	Assi riferiti allo zero elettrico del trasduttore e posizionati alla quota di riposizionamento dopo ricerca 0 (1..8).
POTP2P()	64	CN ⇐ PLC	no	Potenzimetri per la regolazione di velocità assi indipendenti (1..8). Da 0 a 1 come percentuale della velocità di movimento impostata, se in automatico o della velocità di rapido, se in manuale.
JGPP2P	8	CN ⇐ PLC	no	Comandi di JOG positivo per assi indipendenti (1..8).
JGMP2P	8	CN ⇐ PLC	no	Comandi di JOG negativo per assi indipendenti (1..8).
PFNP2P()	64	CN ⇐ PLC	no	Posizioni programmate da raggiungere in automatico per assi indipendenti (1..8).
RUNP2P	8	CN ⇔ PLC	no	Comandi di posizionamento in automatico alla quota programmata per assi indipendenti. Devono essere settati dal PLC per comandare il movimento alla posizione prefissata; vengono resettati dal CN quando l'asse, avendo terminato il movimento, entra nella soglia di posizionamento inserita nei dati di configurazione (1..8).
RHDP2P	8	CN ⇐ PLC	no	Richieste di HOLD su assi indipendenti (1..8). Arresti temporanei del movimento in esecuzione; l'esecuzione riprende senza ulteriori comandi non appena vengono rilasciati.
HDAP2P	8	CN ⇔ PLC	no	Richieste di HOLD su assi indipendenti (1..8). Arresti temporanei del movimento in esecuzione; l'esecuzione riprende senza ulteriori comandi non appena vengono rilasciati.
RBKP2P	8	CN ⇐ PLC	no	Richieste di BREAK su movimento in automatico per assi indipendenti (1..8). RBKP2P vengono resettati dal CN quando acquisiti. Gli assi vengono fatti decelerare fino a fermarsi, quindi vengono resettati anche i RUNP2P eventuali. In stato di emergenza (EMAP2P) servono a cancellare lo stato di emergenza, sempre che la richiesta sia stata rimossa (REMP2P).
BKAP2P	8	CN ⇔ PLC	no	Assi indipendenti fermi in seguito ad un comando di RBKP2P (1..8); possono essere resettati dal PLC, ma ciò non è vincolante.
REMP2P	8	CN ⇐ PLC	no	Richieste di passaggio in stato di emergenza per assi indipendenti (1..8).
EMAP2P	8	CN ⇔ PLC	no	Assi indipendenti in stato di emergenza. Nel passare in questo stato, gli assi vengono istantaneamente disabilitati senza alcuna decelerazione controllata (1..8).
POAP2P()	64	CN ⇔ PLC	no	Posizioni assolute lette dai trasduttori per assi indipendenti (1..8).
TCHP2P()	64	CN ⇔ PLC	no	Velocità effettive (rilevate dai trasduttori) per assi indipendenti (1..8).

4. Riassunto dei segnali e registri predefiniti

SGLP2P	8	CN ⇔ PLC no	Assi indipendenti (1..8) entro tolleranza di posizionamento asse dichiarata nella configurazione.
MKSP2P	8	CN ⇔ PLC no	Segnalazione del marker (zero elettrico) degli assi indipendenti nel caso di trasduttori encoder o riga ottica (1..8).
FCPP2P	8	CN ⇔ PLC no	Assi indipendenti (1..8) la cui quota attuale risulta essere maggiore del finecorsa software positivo dichiarato nella configurazione.
FCMP2P	8	CN ⇔ PLC no	Assi indipendenti (1..8) la cui quota attuale risulta essere minore del finecorsa software negativo dichiarato nella configurazione.
VATP2P()	64	CN ⇔ PLC no	Velocità teoriche (calcolate) per assi indipendenti (1..8). Se nei dati di configurazione viene dichiarato "convertitore D/A assente" il riferimento in tensione non viene inviato sul canale di uscita ma la velocità su questo registro è sempre disponibile.
JINP2P	8	CN ⇔ PLC no	Assi (1..8) in movimento in seguito a movimento in JOGP2P .
DIRP2P	8	CN ⇔ PLC no	Direzioni del movimento comandato (rilevate dal segno del riferimento analogico) per assi indipendenti (1..8). Il valore 1 significa velocità negativa.
FEDP2P()	64	CN ⇔ PLC no	Velocità di movimento per assi indipendenti (1..8).
RAPP2P()	64	CN ⇔ PLC no	Velocità di rapido per assi indipendenti (1..8).
VLNP2P()	64	CN ⇔ PLC no	Velocità nella "zona lenta" per assi indipendenti (1..8).
ZLNP2P()	64	CN ⇔ PLC no	Distanze nella "zona lenta" per assi indipendenti (1..8).
DEXP2P()	64	CN ⇔ PLC no	Distanze per decelerazioni esponenziali per assi indipendenti (1..8).
ACMP2P()	64	CN ⇔ PLC no	Accelerazioni in movimento manuale per assi indipendenti (1..8).
ACCP2P()	64	CN ⇔ PLC no	Accelerazioni in movimento automatico per assi indipendenti (1..8).
DECP2P()	64	CN ⇔ PLC no	Decelerazione da velocità di movimento a velocità lenta per assi indipendenti (1..8).
DE2P2P()	64	CN ⇔ PLC no	Decelerazione da velocità di lento a decelerazione esponenziale per assi indipendenti (1..8).
TOLP2P()	64	CN ⇔ PLC no	Tolleranze di posizionamento per assi indipendenti (1..8).
OFSP2P()	64	CN ⇔ PLC no	Offset trasduttore che viene applicato sulla quota letta per ottenere la quota assoluta POAP2P() (1..8).
SHIP2P()	64	CN ⇔ PLC	Shift di origine per assi indipendenti (1..8). Permette di definire una quota zero diversa dalla quota di zero assoluta. Le posizioni finali PFNP2P() sono sempre riferite alla POOP2P() .
POOP2P()	64	CN ⇔ PLC	Quota asse indipendente (1..8) che risente dello shift di origine SHIP2P() .

Modulo per la gestione del cambio utensili

UTECU	16	CN ⇔ PLC no	Numero utensile richiesto al modulo cambio utensile. UTECU = 0 è il codice particolare riservato alla sequenza di scarico dell'utensile da mandrino in magazzino (o a terra se non esiste posto).
NEWCU	1	CN ⇔ PLC no	Comando di attivazione nuova sequenza per CU. Questo segnale viene settato dal PLC per attivare il modulo cambio utensile ed è resettato dal CU non appena acquisito.
NSEQCU	16	CN ⇔ PLC no	Codice dell'ultima sequenza CU intrapresa.
BRDYCU	1	CN ⇔ PLC no	Strobe di presenza nuovo codice su OPERCU . Viene settato dal CU e deve essere resettato dal PLC non appena è stato acquisita la nuova operazione.
MAPRCU	1	CN ⇔ PLC no	Macchina pronta per il cambio utensile: se uguale a zero la sequenza viene sospesa finché non è rilasciato.

4. Riassunto dei segnali e registri predefiniti

OPERCU	16	CN	⇒	PLC	no	Codice dell'operazione richiesta dal CU al PLC.
PPRECU	16	CN	⇒	PLC	no	Posizione da raggiungere per il prelievo del nuovo utensile.
PPOSCU	16	CN	⇒	PLC	no	Posizione da raggiungere per il deposito del vecchio utensile.
CUATT	1	CN	↔	PLC	no	Segnale alzato dal CU ad inizio nuova sequenza, resettato dal PLC quando considera terminata la sequenza corrente.
M6PGM	1	CN	↔	PLC	si	(M6 programmata) Deve essere alzato in modo sincrono con il burdy dal PLC, viene resettato dal CU quando, essendo arrivata l'operazione di attesa M06 , è stato lanciato l'eventuale sottoprogramma CN (COM). In mancanza di questo segnale, la sequenza si arresta sulla fase (-6). M6PGM attivo implica una sospensione automatica dell'esecuzione dei blocchi CN !
UTSPCU	16	CN	↔	PLC	no	Numero utensile in mandrino.
UTSICU	16	CN	↔	PLC	no	Numero dell'utensile in stazione intermedia.
UTPICU	16	CN	↔	PLC	no	Numero dell'utensile in pinza.
EMACU	1	CN	⇒	PLC	no	Cambio utensile in stato di emergenza. E' settato quando la sequenza CU viene interrotta prima del termine in seguito a richiesta di emergenza CU. La presenza di questo segnale segnala che può non essere coerente la situazione degli utensili presente in tabella rispetto alla situazione reale; necessita quindi un intervento dell'operatore. Eventuali richieste di nuovi cambi utensili NEWCU vengono ignorate.
REMCU	1	CN	↔	PLC	no	Richiesta di emergenza per CU. Questo comando interrompe la sequenza corrente di CU e l'operazione in corso, quindi pone il CU in stato di emergenza.
RBKCU	1	CN	↔	PLC	no	Uscita dallo stato di emergenza CU EMACU e richiesta di interruzione su sequenza cambio utensile: deve essere settato dal PLC ed è resettato dal CU quando acquisito.
SELECU	8	CN	↔	PLC	no	Selettore di modalità. Deve essere predisposto prima di attivare il modulo cambio utensile, esso è acquisito su inizio sequenza e non può essere modificato durante la stessa. 0 = modo CU normale 1 = modo CU con magazzino escluso 2 = modo con carico dell'utensile programmata in magazzino 3 = modo con scarico a terra dell'utensile programmato.
ERCU	16	CN	⇒	PLC	no	Codice dell'errore riscontrato dal CU. Ad ogni operazione, vengono verificate le informazioni relative a magazzino, tabella utensili, configurazione. In caso di informazioni non congruenti o di situazioni non previste o non gestibili, il CU interrompe l'eventuale sequenza attiva e comunica l'errore riscontrato. Inoltre, nessuna sequenza di CU è attivabile se si è in condizione di errore.

Tabella utensili

UTENRI	16	CN	⇒	PLC	no	Numero di "righe" nella tabella utensili, cioè numero massimo di elementi dei vettori che rappresentano le colonne della tabella utensili.
UTNUM()	16	CN	↔	PLC	no	Codici degli utensili in tabella (1 .. UTENRI).
UTPOS()	16	CN	↔	PLC	no	Posti in magazzino degli utensili (1 .. UTENRI).
UTCAP()	16	CN	↔	PLC	no	Capostipiti degli utensili (1 .. UTENRI).

4. Riassunto dei segnali e registri predefiniti

UTDIM()	8	CN ⇔ PLC	no	Taglie degli utensili (1 .. UTENRI), dove: 0 = small 1 = medium 2 = large 3 = extra
UTSPC()	8	CN ⇔ PLC	no	Utensili speciali (1 .. UTENRI), dove: 0 = utensile normale diverso da 0 = utensile speciale
UTPLKO()	8	CN ⇔ PLC	no	Utensili interdetti (1 .. UTENRI), dove: 0 = utensili non interdetti diverso da 0 = utensili non interdetti
UTVTKO()	8	CN ⇔ PLC	no	Vita scaduta (1 .. UTENRI), dove: 0 = vita non scaduta diverso da 0 = vita scaduta
UTVITA()	64	CN ⇔ PLC	no	Vita MAX utensili (1 .. UTENRI) in centesimi di secondo.
UTVTRE()	64	CN ⇔ PLC	no	Vita Residua utensili (1 .. UTENRI) in centesimi di secondo.
UTVTMI()	64	CN ⇔ PLC	no	Vita Minima utensili (1 .. UTENRI) in centesimi di secondo.
UTWD1()	16	CN ⇔ PLC	no	WORD#1 - variabili 1 nel formato word a disposizione dell'applicazione (1.. UTENRI).
UTWD2()	16	CN ⇔ PLC	no	WORD#2 - variabili 2 nel formato word a disposizione dell'applicazione (1.. UTENRI).
UTFP1()	32	CN ⇔ PLC	no	FLOAT#1 - variabili 1 nel formato floating point a disposizione dell'applicazione (1 .. UTENRI).
UTFP2()	32	CN ⇔ PLC	no	FLOAT#2 - variabili 2 nel formato floating point a disposizione dell'applicazione (1 .. UTENRI).
UTEFRE	16	CN ⇒ PLC	no	Numero delle scritture ancora disponibili nella memoria temporanea per l'aggiornamento dei campi della tabella utensili.
MAGNPO	16	CN ⇒ PLC	no	Numero dei posti configurati nei parametri per il magazzino utensili.
MAGCUA()	16	CN ⇒ PLC		Array che rappresenta l'immagine del magazzino utensili (0 .. MAGNPO).
UTRUN	1	CN ⇐ PLC		Utensile in mandrino in fase di lavorazione: decrementa VITA RESIDUA
UTTIM	32	CN ⇒ PLC		Valore del contatore VITA RESIDUA dell'utensile in mandrino.
UTSTS	8	CN ⇒ PLC		Registro di stato dell'utensile in mandrino: UTSTS (1) = vita scaduta UTSTS (2) = vita residua <= 0
CUATYP	16	CN ⇒ PLC		Tipo di cambio utensile selezionato 0 = manuale 1 = manuale S1200 2 = automatico
MAGGEO	16	CN ⇒ PLC		Geometria magazzino selezionata 0 = catena 1 = planare
MAGTYP	16	CN ⇒ PLC		Disposizione utensili in magazzino selezionata 0 = fisso 1 = random 2 = random fisso
MAGGST	16	CN ⇒ PLC		Gestione magazzino selezionata 0 = sincrono 1 = asincrono 2 = semiasincrono

5. LIMITI

I dati riportati riassumono i limiti del compilatore da tener presente durante la stesura di un programma:

Max lunghezza istruzioni di programma (riga logica)	500 caratteri
Max lunghezza riga di programma (riga fisica)	62 caratteri (+ 8 numeri)
Max numero linee concatenabili con \$	24 righe fisiche
Max area memoria per variabili retentive	circa 3 KByte
Max area memoria per variabili non retentive	circa 50 KByte
Numero max di temporizzatori fast	32
Numero max di temporizzatori slow	64
Numero max di counters	48
Numero max di pulse	64
Numero max di EXEC annidate	4
Numero max di multiplexer	16
Numero max di rami per GOTC	255
Numero max di rami per GOTP	16
Max lunghezza per riga microeditor softkey	20
Max numero softkey complessivo (tutti i menu)	128
Max numero positivo rappresentabile nel formato byte	127
Max numero negativo rappresentabile nel formato byte	-128
Max numero positivo rappresentabile nel formato word	32767
Max numero negativo rappresentabile nel formato word	-32768
Max numero rappresentabile nel formato long	3.4×10^{38}
Min numero rappresentabile nel formato long	1.2×10^{-38}
Max numero rappresentabile nel formato double	1.8×10^{307}
Min numero rappresentabile nel formato double	2.2×10^{-308}

PARTE III

ESEMPI DI PROGRAMMAZIONE

1. ESEMPI DI PROGRAMMAZIONE PLC

Nelle pagine che seguono sono riportati diversi esempi reali di programmazione PLC, da cui è possibile prendere spunto per lo sviluppo di nuove applicazioni.

Gli esempi sono auto-documentati al loro interno per cui non necessitano di spiegazioni particolari. E' ovvio che le persone che hanno necessità di interpretarli devono avere una conoscenza di programmazione PLC o, almeno, aver letto approfonditamente le prime due parti del presente manuale.

Gli esempi sono suddivisi in moduli, ognuno dei quali compie una determinata funzione descritta nel titolo del programma stesso. Il titolo riporta anche il nome del file che può essere richiesto alla SELCA in caso di necessità.

1. Esempi di programmazione

BAS300F - Fresa base 3 assi e mandrino)

```

N1      [*****]
N2      [*      MACCHINA FRESA BASE 3 ASSI E MANDRINO CON S3045      ]
N3      [*      ]
N4      [*      BAS300F 941008      ]
N5      [*****]
N6      [Nota: le righe di edit sono lunghe 62 caratt. + 8 numeraz.
N7      [
N8      [***** SEZIONE DICHIARATIVA *****]
N9      [
N10     [      input fisici
N11     INP
N12     IMAPR      [ 1 macchina pronta
N13     IHOLD      [ 2 hold esterno
N14     ISTART     [ 3 via esterno
N15     IMZX       [ 4 micro zero asse X\
N16     IMZY       [ 5 micro zero asse Y > solo per non assoluti
N17     IMZZ       [ 6 micro zero asse Z/
N18     TERM,23 [ salta gli input fino a 23
N19     IREME      [24 emergenza esterna
N20     [
N21     [      output fisici
N22     OUT
N23     UMOVE1     [ 1 abilitazione asse 1
N24     UMOVE2     [ 2 abilitazione asse 2
N25     UMOVE3     [ 3 abilitazione asse 3
N26     TERM,4
N27     UMAN       [ 5 abilitazione mandrino
N28     UREF       [ 6 refrigerante
N29     ALARM      [ 7 CN in emergenza
N30     LAHOLD     [ 8 lampada hold assi
N31     LACYON     [ 9 lampada ciclo in esecuzione
N32     [
N33     [      variabili interne
N34     RAM,1
N35     ROTMA      [memorizzazione comando rotaz. mandrino
N36     CICL       [predisposizione assi ciclici
N37     [
N38     [      stringhe per messaggi
N39     STR
N40     MSG1
N41     [
N42     [      menu di softk gestiti da PLC
N43     SOFTK,1
N44     P1,L1,1,'JOG ASSE      X+'
N45     P2,L2,1,'JOG ASSE      X-'
N46     P3,L3,1,'JOG ASSE      Y+'
N47     P4,L4,1,'JOG ASSE      Y-'
N48     P5,L5,1,'JOG ASSE      Z+'
N49     P6,L6,1,'JOG ASSE      Z-'
N50     P7,L7,'RICERCA ZERO    ASSI'
N51     P8,L8,'VOLANTINO',2
N52     [
N53     SOFTK,2
N54     P21,L21,'VOLANTINO ASSE X '
N55     P22,L22,'VOLANTINO ASSE Y '
N56     P23,L23,'VOLANTINO ASSE Z '
N57     P24,L24,' 0.5 mm  al giro'
N58     P25,L25,' 1 mm   al giro'
N59     P26,L26,' 5 mm   al giro'
N60     P27,L27,' 10 mm  al giro'
N61     P28,L28,' ASSI    IN JOG',1
N62     [
N63     [***** SEZIONE INIZIALIZZAZIONE *****]
N64     INIT
N65     [test di KMW(1): se 1 abilito ricerca zeri
N66     IF(KMW(1)=1) CICL=1; ELSE CICL=0
N67     SPGAM(1)=1
N68     [inizializzazione messaggi

```

[dichiaro gamma 1 mandr.


```

N69 MSG1='Eseguire ricerca zero assi ' [messaggio zeri da fare
N70 L24=1 [passo volantini default
N71 SSA=00000111B [assi XYZ sbloccati
N72 [
N73 PROG
N74 [***** SEZIONE VELOCE *****
N75 END
N76 [***** SEZIONE LENTA *****
N77 [PARTE SINCRONA con i blocchi di programma _____
N78 [ ..... decodifica delle funzioni ausiliarie .....
N79 IF("BURDY)ASINC
N80 DHOLD=1; FHOLD=1
N81 IF(STROM) CALL GEFUM
N82 BURDY=0
N83 ASINC:$
N84 [
N85 [_____ PARTE ASINCRONA _____
N86 [ ..... lettura potenziometri .....
N87 POFO=ANI(1) [avanzamento in automatico
N88 POMO(1)=ANI(2) [avanzamento in manuale
N89 POMO(2)=ANI(2)
N90 POMO(3)=ANI(2)
N91 [
N92 [.....mandrino.....
N93 SPSSO(1)=ANI(3) [legge spindle override
N94 SPVEL(1)=SPEED [velocita' mandrino
N95 SPROT(1)=ROTMA&"HOLDA [comando rotazione e HOLD
N96 UMAN=SPMOV(1) [abilitazione azionamento
N97 [
N98 [..... gestione assi.....
N99 UMOVE1=MOVCN(1) [abilitazione azion. X
N100 UMOVE2=MOVCN(2) [abilitazione azion. Y
N101 UMOVE3=MOVCN(3) [abilitazione azion. Z
N102 RDMOV=MOVCN [assi abilitati per il CN
N103 [
N104 [.....jog.....
N105 [Attenzione a non inibire i JOG con NCMD=8 e NCMD=9
N106 [necessari per il riposizionamento sul profilo e manuali
N107 [durante lo stato di HOLD.
N108 [Gestione softkey: in manuale JOG+ e JOG-
N109 [ in ricerca 0 solo JOG+
N110 L1=P1
N111 L2=P2&"L7
N112 L3=P3
N113 L4=P4&"L7
N114 L5=P5
N115 L6=P6&"L7
N116 [
N117 JOGP(1)=L1 [assegnazione JOG
N118 JOGM(1)=L2
N119 JOGP(2)=L3
N120 JOGM(2)=L4
N121 JOGP(3)=L5
N122 JOGM(3)=L6
N123 MOVMA=JOGP~JOGM [selezione modo manuale in JOG
N124 [
N125 [ ..... volante .....
N126 [softkey per selezione asse da muovere con volante
N127 IF(P21) L21="L21;L22=0;L23=0
N128 IF(P22) L22="L22;L23=0;L21=0
N129 IF(P23) L23="L23;L21=0;L22=0
N130 IF(L21) HWL(1)=1;L8=1
N131 IF(L22) HWL(1)=2;L8=1
N132 IF(L23) HWL(1)=3;L8=1
N133 IF("L21&"L22&"L23) HWL(1)=0;L8=0
N134 [softkey per assegnazione passo
N135 IF(P24) L24=1;L25=0;L26=0;L27=0
N136 IF(P25) L24=0;L25=1;L26=0;L27=0
N137 IF(P26) L24=0;L25=0;L26=1;L27=0
N138 IF(P27) L24=0;L25=0;L26=0;L27=1
N139 IF(L24) STEP=1

```

1. Esempi di programmazione

```

N140 IF(L25) STEP=2
N141 IF(L26) STEP=3
N142 IF(L27) STEP=4
N143 [
N144 [.....ricerca zeri.....
N145 IF(P7) L7="L7 [softkey abilitazione ricerca 0
N146 IF((SFKMEN<>1)~(NCMD<>5)~IREME~BRKA) L7=0
N147 [selezione ricerca zero con micro presente
N148 MICZE(1)=L7
N149 MICZE(2)=L7
N150 MICZE(3)=L7
N151 MIZER(1)=IMZX
N152 MIZER(2)=IMZY
N153 MIZER(3)=IMZZ
N154 [
N155 [.....generali.....
N156 FHOLD="IMAPR ~ SPRMP(1)&"RAPI ~ $
N157 (((NCMD<>5)&(MIZEA<>7))&CICL) [arresto avanz. assi
N158 DHOLD="IMAPR [arresto avanzamento blocchi
N159 RHOLD=IHOLD [richiesta di hold esterna
N160 REME=IREME [emergenza esterna
N161 CYST=ISTART [richiesta di via esterna
N162 [
N163 ALARM=EMEA [CN in stato di emergenza
N164 [
N165 IF(BRKA~EMEA) CALL RESET [reset funzioni PLC da CN
N166 [
N167 END
N168 [***** SEZIONE SUPERLENTA *****
N169 [..... visualizzazioni e segnalazioni .....
N170 IF((MIZEA<>7)&CICL) DISPL,0,MSG1; ELSE CLR,0 [messaggio zeri
N171 [
N172 LAHOLD=HOLDA [lampada di hold
N173 LACYON=CYON [lampada di esecuzione in corso
N174 [
N175 WNDINT(2)=AUXH [visualizzazione codice H
N176 GIRMI=INT(ABS(SPTCH(1))) [visual. speed effettiva
N177 END
N178 [
N179 [***** SEZIONE ROUTINES *****
N180 [
N181 [ ..... decodifica funzioni M .....
N182 GEFUM:$
N183 WNDINT(1)=AUXM [visualizzazione M
N184 IF (AUXM=3) ROTMA=1; SPDIR(1)=0; RTS [mandrino orario
N185 IF (AUXM=4) ROTMA=1; SPDIR(1)=1; RTS [mandrino anti
N186 IF (AUXM=5) ROTMA=0; RTS [stop mandrino
N187 IF (AUXM=7) UREF=1; RTS [refrigerante
N188 IF (AUXM=9) UREF=0; RTS [stop refriger.
N189 IF (AUXM=10) SSA=0; RTS [bloccaggio assi
N190 IF (AUXM=11) SSA=00000111B; RTS [sbloccaggio assi XYZ
N191 IF (AUXM=13) ROTMA=1; SPDIR(1)=0; UREF=1; RTS [man.+ ref.
N192 IF (AUXM=14) ROTMA=1; SPDIR(1)=1; UREF=1; RTS [man.+ ref.
N193 IF (AUXM=30) CALL RESET; RTS [reset CN e PLC
N194 RTS
N195 [
N196 [..... routine di reset .....
N197 RESET:$
N198 ROTMA=0 [arresto mandrino
N199 UREF=0 [arresto refrigerante
N200 SFKMEN=1 [ripristino menu principale softkey
N201 WNDINT(1)=30 [visualizzazione M30
N202 RTS
N203 [..... fine programma .....

```

COMI3045 - Fresa 3 assi, mandrino

```

N1  [*****]
N2  [*      MACCHINA 3 ASSI CON BLOCCAGGI      ]
N3  [*      MANDRINO CON ORIENTAMENTO E 2 GAMME      ]
N4  [*      CICLO RICERCA ZERI AUTOMATICO (Z e XY)      ]
N5  [*      LOGICA DI ACCENSIONE AUSILIARI      ]
N6  [*      LUBRIFICAZIONE A TEMPO CON ASSI IN MOVIMENTO      ]
N7  [*      _____      ]
N8  [*      COMI3045:          941008      ]
N9  [*****]
N10 [Nota: Nell' editor le righe sono di 62 caratt. + 8 numeraz.]
N11 [
N12 [***** SEZIONE DICHIARATIVA *****]
N13 [
N14 [      input fisici
N15 [
N16 INP
N17 IMUON          [1 macchina accesa
N18 ISTART          [2 via esterno
N19 IHOLD          [3 Hold esterno
N20 IMG1          [4 micro gamma 1
N21 IMG2          [5 micro gamma 2
N22 IMAMAO          [6 mandrino manuale orario
N23 IMAMAA          [7 mandrino manuale orario
N24 ISTOPM          [8 stop mandrino manuale
N25 IGIROK          [9 giri mandrino raggiunti
N26 IDRAOK          [10 drives assi OK
N27 IDRMOK          [11 drives mandrino OK
N28 ILIVOL          [12 livello olio
N29 ILIVRE          [13 livello refrigerante
N30 ITERMI          [14 termici OK
N31 IOLTRC          [15 oltrecorsa assi OK
N32 IFICUT          [16 pulsante fine cambio utensile
N33 IMZX          [17 micro zero asse X\
N34 IMZY          [18 micro zero asse Y > solo per non assoluti
N35 IMZZ          [19 micro zero asse Z/
N36 [
N37 [      output fisici
N38 OUT
N39 UMOVE1          [1 abilitazione asse 1
N40 UMOVE2          [2 abilitazione asse 2
N41 UMOVE3          [3 abilitazione asse 3
N42 TERM,4          [salto all' uscita 5
N43 UMAN          [5 abilitazione mandrino
N44 USFREX          [6 sbloccaggio asse X
N45 USFREY          [7 sbloccaggio asse Y
N46 USFREZ          [8 sbloccaggio asse Z
N47 UREF          [9 refrigerante
N48 CNOK          [10 CN ok per ausiliari
N49 LAHOLD          [11 lampada Hold
N50 LACYON          [12 lampada Ciclo in corso
N51 OKVG1          [13 comando gamma 1
N52 OKVG2          [14 comando gamma 2
N53 UKLUBA          [15 e.v. lubrif. assi
N54 ULAM06          [16 lampada M06
N55 [
N56 [      variabili interne
N57 RAM,8
N58 MOVCPN          [copia dei MOVCPN vecchi per derivata
N59 NM          [numero messaggio
N60 NR          [numero riga per messaggio
N61 NMAX          [numero massimo di messaggi
N62 SG          [byte per flag messaggi 1 - 8
N63 SG2          [byte per flag messaggi 9 -16
N64 SG3          [byte per flag messaggi 17 -24
N65 SG4          [byte per flag messaggi 25 -32
N66 [
N67 RAM,1
N68 RICOX          [ricerca zero asse X in corso

```

1. Esempi di programmazione

```

N69  RIC0Y          [ricerca zero asse Y in corso
N70  RIC0Z          [ricerca zero asse Z in corso
N71  ZERIOK         [zeri assi effettuati
N72  SJOG           [stato di JOG abilitati
N73  RORMA          [memoria M3
N74  RANMA          [memoria M4
N75  RM41           [forzamento gamma 1
N76  RM42           [forzamento gamma 2
N77  GAM1           [memoria richiesta gamma 1
N78  GAM2           [memoria richiesta gamma 2
N79  CAUT           [cambio utensile attivo
N80  G84            [ciclo fisso maschiatura attivo
N81  [
N82  STR
N83  MSG(32)        [tabella testo di 32 messaggi
N84  [
N85  STIMER
N86  TIM06,TUM06,TDM06,TAM06,TWM06          [lampeggio lamp. CU
N87  TIM19,TUM19,TDM19,TAM19,TWM19          [mandrino M19
N88  TIMUON,TUMUON,TDMUON,TAMUON,TWMUON     [accensione ausiliari
N89  TISBX,TUSBX,TDSBX,TASBX,TWSBX          [sbloccaggio asse X
N90  TISBY,TUSBY,TDSBY,TASBY,TWSBY          [sbloccaggio asse X
N91  TISBZ,TUSBZ,TDSBZ,TASBZ,TWSBZ          [sbloccaggio asse X
N92  TIBLX,TUBLX,TDBLX,TABLX,TWBLX          [bloccaggio asse X
N93  TIBLY,TUBLY,TDBLY,TABLY,TWBLY          [bloccaggio asse Y
N94  TIBLZ,TUBLZ,TDBLZ,TABLZ,TWBLZ          [bloccaggio asse Z
N95  TLUBI,TLUBU,TLUBD,TLUBA,TLUBW          [lubrificazione assi
N96  [
N97  SOFTK,1
N98  P1,L1,1,'JOG ASSE      X+'
N99  P2,L2,1,'JOG ASSE      X-'
N100 P3,L3,1,'JOG ASSE      Y+'
N101 P4,L4,1,'JOG ASSE      Y-'
N102 P5,L5,1,'JOG ASSE      Z+'
N103 P6,L6,1,'JOG ASSE      Z-'
N104 P7,L7,' RICERCA ZERO ASSI'
N105 P8,L8,'VOLANTINO',2
N106 SOFTK,2
N107 P21,L21,'VOLANTINO ASSE X '
N108 P22,L22,'VOLANTINO ASSE Y '
N109 P23,L23,'VOLANTINO ASSE Z '
N110 P24,L24,' 0.4 mm  al giro'
N111 P25,L25,' 1 mm   al giro'
N112 P26,L26,' 5 mm   al giro'
N113 P27,L27,'
N114 P28,L28,' ASSI      IN JOG',1
N115 [
N116 INIT
N117 [***** SEZIONE INIZIALIZZAZIONE *****
N118 L24=1 [passo volantini default
N119 [
N120 NMAX=32 [numero massimo messaggi ammessi
N121 MSG(1)= 'AUSILIARI DISINSERITI'
N122 MSG(2)= 'EFFETTUARE RICERCA ZERO ASSI'
N123 MSG(3)= '- per iniziare il ciclo automatico premere JOG Z+'
N124 MSG(4)= 'MANDRINO NON A REGIME'
N125 MSG(5)= 'CAMBIO GAMMA IN CORSO'
N126 MSG(6)= 'ANOMALIA AZIONAMENTO ASSI'
N127 MSG(7)= 'ANOMALIA AZIONAMENTO MANDRINO'
N128 MSG(8)= 'LIVELLO OLIO INSUFFICIENTE'
N129 MSG(9)= 'LIVELLO REFRIGERANTE INSUFFICIENTE'
N130 MSG(10)= 'TERMICI SCATTATI'
N131 MSG(11)= 'ASSI IN OLTRECORSA'
N132 MSG(12)= 'SOSTITUIRE UTENSILE MANUALMENTE'
N133 MSG(13)= 'ATTESA BLOCCAGGIO / SBLOCCAGGIO ASSI'
N134 [...]
N135 MSG(32)= 'MESSAGGIO32'
N136 [
N137 PROG
N138 [***** SEZIONE VELOCE *****
N139 END

```

```

N140 [***** SEZIONE LENTA *****]
N141 [PARTE SINCRONA-----]
N142 [
N143 IF("BURDY)ASINC
N144 DHOLD=1; FHOLD=1
N145 IF(STROT)CALL GEFUT
N146 IF(STROM)CALL GEFUM
N147 BURDY=0
N148 ASINC: $
N149 [
N150 [.....PARTE ASINCRONA.....]
N151 [..... gestione potenziometri assi .....]
N152 POFO=ANI(1) [in automatico]
N153 POMO(1)=ANI(2) [in manuale X]
N154 POMO(2)=ANI(2) [ Y]
N155 POMO(3)=ANI(2) [ Z]
N156 [
N157 [..... rotazione mandrino manuale .....]
N158 IF (NCMD<=5) SPAUTO
N159 IF (IMAMAO) CALL M03
N160 IF (IMAMAA) CALL M04
N161 IF (ISTOPM) CALL M05
N162 SPAUTO:$
N163 [
N164 G84=(CICFI=84) [ciclo fisso G84 in corso]
N165 [spindle speed override e velocità (in manuale max di gamma)
N166 [in automatico 70% - 130%
N167 [in maschiatura 100%
N168 [in manuale 0% - 100%
N169 IF (NCMD=5) SPVEL(1)=SPSMAX; SPSSO(1)=ANI(3); NOVEMA
N170 SPVEL(1)=SPEED
N171 IF(G84) SPSSO(1)=1; $
N172 ELSE SPSSO(1)=0.7 + ANI(3)*0.6
N173 NOVEMA:$
N174 [
N175 [comando rotazione e HOLD
N176 SPROT(1)=(RORMA~RANMA)&"HOLDA [comando di rotazione
N177 SPDIR(1)=RORMA&"RANMA [direzione di rotazione
N178 UMAN=SPMOV(1)&IMUON [abilitazione azionamento
N179 [
N180 [..... orientamento mandrino .....]
N181 TIM19(10)=SPTOL(1)&SPORI(1) [timer fine orientam. M19
N182 IF(TUM19) SPORI(1)=0 [verifico toll. x 1 sec.
N183 [
N184 [..... CAMBIO GAMMA .....]
N185 [Nota: SPPND viene settato anche se il mandrino non è sotto
N186 [ soglia di giri min (SPMOT) per cambio gamma al volo.
N187 GAM1=RM41~(SPEED<=SPSMG1(1))&"RM42 [richiesta gamma 1
N188 GAM2=RM42~(SPEED>SPSMG1(1))&"RM41 [richiesta gamma 2
N189 OKVG1=GAM1&"IMG1&IMUON&"SPMOT(1) [comando e.v. gamma 1
N190 OKVG2=GAM2&"IMG2&IMUON&"SPMOT(1) [comando e.v. gamma 2
N191 SPPND(1)=(GAM1&"IMG1)~(GAM2&"IMG2)&IMUON [comando pendolam.
N192 IF(IMG1) SPGAM(1)=1 [seleziona gamma 1
N193 IF(IMG2) SPGAM(1)=2 [seleziona gamma 2
N194 [
N195 [..... gestione assi.....]
N196 TISBX(3)=MOVCN(1) [timer sbloccaggio
N197 TISBY(3)=MOVCN(2)
N198 TISBZ(3)=MOVCN(3)
N199 TIBLX(5)=("MOVCN(1)&MOVCNP(1))~TDBLX [timer bloccaggio
N200 TIBLY(5)=("MOVCN(2)&MOVCNP(2))~TDBLY
N201 TIBLZ(5)=("MOVCN(3)&MOVCNP(3))~TDBLZ
N202 [
N203 UMOVE1=(MOVCN(1)~TDBLX)&IMUON [abilitazione assi
N204 UMOVE2=(MOVCN(2)~TDBLY)&IMUON
N205 UMOVE3=(MOVCN(3)~TDBLZ)&IMUON
N206 USFREX=MOVCN(1)&IMUON [sbloccaggio
N207 USFREY=MOVCN(2)&IMUON
N208 USFREQ=MOVCN(3)&IMUON
N209 RDMOV(1)=(MOVCN(1)&"TDSBX)~TDBLX [risposte al CN
N210 RDMOV(2)=(MOVCN(2)&"TDSBY)~TDBLY

```

1. Esempi di programmazione

```

N211 RDMOV(3)=(MOVCN(3)&"TDSBZ)~TDBLZ
N212 MOVCNP=MOVCN [per derivata MOVCN
N213 [
N214 [.....jog.....
N215 [nota: i JOG devono essere abilitati con NCMD=5, 8, 9
N216 SJOG=((NCMD=5)&"L7)~(NCMD=8)~(NCMD=9) [jog + e - abilit.
N217 L1=JOGP(1)
N218 L2=JOGM(1)
N219 L3=JOGP(2)
N220 L4=JOGM(2)
N221 L5=JOGP(3)
N222 L6=JOGM(3)
N223 [la ricerca zeri X Y Z avviene in direzione positiva
N224 JOGP(1)=P1&SJOG~RIC0X
N225 JOGM(1)=P2&SJOG
N226 JOGP(2)=P3&SJOG~RIC0Y
N227 JOGM(2)=P4&SJOG
N228 JOGP(3)=P5&SJOG~RIC0Z
N229 JOGM(3)=P6&SJOG
N230 MOVMA=JOGP~JOGM [Selezione movimenti manuali JOG
N231 [
N232 [.....VOLANTINO.....
N233 [assegnazione asse da muovere
N234 IF(P21) L21="L21;L22=0;L23=0
N235 IF(P22) L22="L22;L23=0;L21=0
N236 IF(P23) L23="L23;L21=0;L22=0
N237 IF(L21) HWL(1)=1;L8=1
N238 IF(L22) HWL(1)=2;L8=1
N239 IF(L23) HWL(1)=3;L8=1
N240 IF("L21&"L22&"L23) HWL(1)=0;L8=0
N241 [selezione passo
N242 IF(P24) L24=1;L25=0;L26=0
N243 IF(P25) L24=0;L25=1;L26=0
N244 IF(P26) L24=0;L25=0;L26=1
N245 IF(L24) STEP=1
N246 IF(L25) STEP=2
N247 IF(L26) STEP=3
N248 [
N249 [.....ricerca zeri.....
N250 [la ricerca zeri inizia con la pressione della softkey F17
N251 [termina con un break o quando tutti gli zeri sono eseguiti.
N252 [Il ciclo inizia muovendo l' asse Z quindi X, Y in contemporanea
N253 ZERIOK=MIZEA(1)&MIZEA(2)&MIZEA(3)
N254 L7=FF(P7&"L7),(ZERIOK~(NCMD<>5)~BRKA~P7&L7) [ricerca zeri
N255 RIC0X=FF(L7&MIZEA(3)),("L7~MIZEA(1)) [memoria ciclo su X
N256 RIC0Y=FF(L7&MIZEA(3)),("L7~MIZEA(2)) [memoria ciclo su Y
N257 RIC0Z=FF(L7&(P6~P5)),("L7~MIZEA(3)) [memoria ciclo su Z
N258 [
N259 [selezione ricerca di zero con micro su softkey F7 menu 1
N260 MICZE(1)=L7
N261 MICZE(2)=L7
N262 MICZE(3)=L7
N263 [assegnazione micro di zero fisici
N264 MIZER(1)=IMZX
N265 MIZER(2)=IMZY
N266 MIZER(3)=IMZZ
N267 [..... cambio utensile manuale .....
N268 ULAM06=CAUT&(TWM06>5)&"SPMOT(1)
N269 IF(IFICUT) CAUT=0
N270 TIM06(10)="TUM06
N271 [
N272 [.....lubrificazione .....
N273 [All' accensione (IMUON) da' una pompata
N274 [il tempo scorre solo con assi in movimento.
N275 TLUBI(6000)="TLUBU&IMUON&ILIVOL [oscillatore 10 minuti
N276 TLUBA=((MOVCN&"INTOL)=0) [pausa ad assi fermi o disabilitati
N277 UKLUBA=(TLUBW>5950)&"TLUBA&IMUON&ILIVOL [pompa per 5 secondi
N278 [
N279 [.....generali.....
N280 [Nota: ILIVRE e ILIVOL
N281 [ non intervengono all' interno di una maschiatura

```

1. Esempi di programmazione

```

N282 Fhold= ("ILIVRE~"ILIVOL)&("G84~RAPI)) ~"ITERMI ~"IDRAOK ~ $
N283 "IDRMOK ~CAUT ~(SPRMP(1)~"IGIROK&SPROT(1))&"RAPI~SPORI(1)~ $
N284 SPPND(1) ~((NCMD<>5)&"ZERIOK) [inibiz. avanzamento assi
N285 DHOLD=FHOLD [inibizione avanzamento blocchi
N286 [
N287 RHOLD=FF(IHOLD&("G84~RAPI)),(HOLDA) [richiesta di hold
N288 LAHOLD=HOLDA [lampada di hold
N289 CYST=ISTART [richiesta di via
N290 LACYON=CYON [lampada di esecuzione in corso
N291 [
N292 [.....accensione ausiliari.....
N293 TIMUON(5)=IMUON [derivata all' accensione
N294 RBRK=TDMUON [BREAK all' accensione
N295 CNOK="EMEA~"TUMUON [uscita CN pronto
N296 REME=FF("IMUON~"IOLTRC),(EMEA) [richiesta di emergenza
N297 [
N298 [.....break.....
N299 IF(BRKA~EMEA) CALL RESET
N300 [
N301 IF(STBMD) SFKMEN=1 [su cambio stato CN ripristino menu SOFTK 1
N302 END
N303 [***** SEZIONE SUPERLENTA *****
N304 [.....visualizzazioni.....
N305 [
N306 WNDINT(2)=AUXH [visualizz. H
N307 GIRMI=INT(ABS(SPTCH(1))) [visual. speed effettiva
N308 [
N309 [..... preparazione messaggi .....
N310 SG(1)="IMUON
N311 SG(2)="ZERIOK&IMUON
N312 SG(3)=SG(2)&L7&"L5
N313 SG(4)=(SPRMP(1)~"IGIROK)&SPROT(1)
N314 SG(5)=SPPND(1)
N315 SG(6)="IDRAOK
N316 SG(7)="IDRMOK
N317 SG(8)="ILIVOL
N318 SG(9)="ILIVRE
N319 SG(10)="ITERMI
N320 SG(11)="IOLTRC
N321 SG(12)=CAUT
N322 SG(13)=(MOVCN<>RDMOV)
N323 CALL SCROLL [richiamo visualizzaz. messaggi
N324 END
N325 [***** SEZIONE ROUTINES *****
N326 [..... funzioni T .....
N327 GEFUT:$
N328 CALL M05; CAUT=1 [cambio utensile manuale
N329 RTS
N330 [..... funzioni M .....
N331 GEFUM:$
N332 WNDINT(1)=AUXM [visualizzazione codice funzione
N333 IF (AUXM=3) M03
N334 IF (AUXM=4) M04
N335 IF (AUXM=5) M05
N336 IF (AUXM=7) UREF=1; RTS [refrigerante
N337 IF (AUXM=9) UREF=0; RTS [stop refriger.
N338 IF (AUXM=10) SSA=0; RTS
N339 IF (AUXM=11) CALL M11
N340 IF (AUXM=13) CALL M03; UREF=1; RTS [M3 + ref.
N341 IF (AUXM=14) CALL M04; UREF=1; RTS [M4 + ref.
N342 IF (AUXM=19) CALL M05; SPPOS(1)=0; SPORI(1)=1; RTS [orientam.
N343 IF (AUXM=30) CALL M05; CALL RESET; RTS [reset da CN
N344 IF (AUXM=40) CALL M05; RM41=0; RM42=0; RTS [gamma auto
N345 IF (AUXM=41) CALL M05; RM41=1; RM42=0; RTS [gamma 1
N346 IF (AUXM=42) CALL M05; RM42=1; RM41=0; RTS [gamma 2
N347 RTS
N348 [
N349 M03: RORMA=1; RANMA=0; RTS [mandrino orario
N350 M04: RORMA=0; RANMA=1; RTS [mandrino antiorario
N351 M05: RORMA=0; RANMA=0; RTS [stop mandrino
N352 M11: IF(AXPGM=0) SSA=00000111B; RTS; $

```

1. Esempi di programmazione

```

N353 ELSE SSA=AXPGM&00000111B; RTS [assi sbloccati
N354 [..... reset comandi .....
N355 RESET:$
N356 RORMA=0; RANMA=0 [reset rotazione mandrino
N357 SPORI=0 [reset orientamento mandrino
N358 UREF=0 [reset refrigerante
N359 CAUT=0 [reset cambio utensile in corso
N360 WNDINT(1)=30 [aggiorn. visualizzazione funzione M
N361 RTS
N362 [..... GESTIONE MESSAGGI .....
N363 SCROLL:$
N364 NM=1; NR=1
N365 LOOVIS:IF(NM>NMAX) CLRSCR
N366 IF(NR>16) RTS
N367 IF(SG(NM)) DISPL,NR,MSG(NM); NR=NR+1
N368 NM=NM+1; LOOVIS
N369 CLRSCR:IF(NR>16) RTS
N370 CLR,(NR); NR=NR+1; CLRSCR
N371 [..... fine programma .....

```

AXM11 - Bloccaggi selettivi per assi

```

N1 [*****]
N2 [* FUNZIONE M11 SELETTIVA PER ASSI SPECIFICATI ]
N3 [* ]
N4 [* AXM11 941008 ]
N5 [*****]
N6 [
N7 [ASSI X, Y, Z bloccati o sbloccati (M10 o M11)
N8 [ASSE 4° sempre con bloccaggio
N9 [***** SEZIONE DICHIARATIVA *****]
N10 [
N11 INP
N12 OUT
N13 UMOVE1 [abilitazione asse 1
N14 UMOVE2 [abilitazione asse 2
N15 UMOVE3 [abilitazione asse 3
N16 [
N17 INIT
N18 SSA=00000111B [assi X, Y, Z sempre attivi e sbloccati
N19 [
N20 PROG
N21 END
N22 [***** SEZIONE LENTA *****]
N23 [PARTE SINCRONA con i blocchi di programma ]
N24 [ ..... decodifica delle funzioni ausiliarie .....]
N25 IF("BURDY)ASINC
N26 DHOLD=1; FHOLD=1
N27 IF(STROM) CALL GEFUM
N28 BURDY=0
N29 ASINC:$
N30 [
N31 [----- PARTE ASINCRONA -----]
N32 [..... gestione assi.....]
N33 UMOVE1=MOVCN(1) [abilitazione azion. X
N34 UMOVE2=MOVCN(2) [abilitazione azion. Y
N35 UMOVE3=MOVCN(3) [abilitazione azion. Z
N36 RDMOV=MOVCN [assi abilitati per il CN
N37 [..... generali .....]
N38 FHOLD=0 [~ .. [arresto avanzamento assi
N39 DHOLD=0 [~ .. [arresto avanzamento blocchi
N40 END
N41 END
N42 [***** SEZIONE ROUTINES *****]
N43 [ ..... decodifica funzioni M .....]
N44 GEFUM:$
N45 WNDINT(1)=AUXM [visualizzazione M

```



```

N46 IF (AUXM=11) M11 [sblocca assi (selettivo)
N47 IF (AUXM=10) M10 [blocca assi
N48 RTS
N49 [
N50 M10: SSA=0; RTS
N51 M11: IF(AXPGM=0) SSA=00000111B; RTS; $
N52 ELSE SSA=AXPGM&00000111B; RTS
N53 [
N54 [..... fine programma .....

```

AUXON - logica accensione ausiliari

```

N1 [*****]
N2 [*]
N3 [* LOGICA DI ACCENSIONE AUSILIARI]
N4 [*]
N5 [* AUXON 941008]
N6 [*****]
N7 [Si prevede una uscita di CNOK la quale comanda un rele
N8 [posto in serie alla catena che permette l' accensione
N9 [degli ausiliari.
N10 [Il CN non vede come input il pulsante di inserzione
N11 [ausiliari, ma un input che significa ausiliari inseriti.
N12 [
N13 INP
N14 IMUON [macchina accesa
N15 IDRAOK [drive assi ok
N16 [
N17 OUT
N18 UMOVE1 [abilitazione asse 1
N19 UMOVE2 [abilitazione asse 2
N20 UMOVE3 [abilitazione asse 3
N21 CNOK [CN ok per ausiliari
N22 [
N23 STR
N24 MSG1 [messaggio ausiliari disinseriti
N25 [
N26 STIMER
N27 TIMUON,TUMUON,TDMUON,TAMUON,TWMUON [accensione ausiliari
N28 [
N29 INIT
N30 SSA=00000111B [assi XYZ sempre abilitati
N31 MSG1='AUSILIARI DISINSERITI'
N32 [
N33 PROG
N34 END
N35 [***** sezione lenta *****]
N36 UMOVE1=MOVCN(1) [abilitazione asse X
N37 UMOVE2=MOVCN(2) [abilitazione asse Y
N38 UMOVE3=MOVCN(3) [abilitazione asse Z
N39 RDMOV=MOVCN [risposta assi abilitati
N40 [
N41 POFO=ANI(1) [override avanzamento assi
N42 BURDY=0 [... acquisizione funzioni da CN
N43 [
N44 [.....accensione ausiliari.....]
N45 TIMUON(5)=IMUON [derivata all' accensione
N46 RBRK=TDMUON [BREAK all' accensione
N47 CNOK="EMEA~"TUMUON [uscita CN pronto
N48 REME=FF("IMUON~"IDRAOK),(EMEA)[richiesta di emergenza
N49 [
N50 IF("IMUON) DISPL,0, MSG1; ELSE CLR,0 [messaggio a video
N51 [
N52 [..... fine programma .....

```

1. Esempi di programmazione

GEVOL3 - Comando assi x, y, z con un solo volantino

```

N1      [*****]
N2      [*]
N3      [*          ESEMPIO COMMUTAZIONE VOLANTINO ]
N4      [*          GEVOL3 941008                 ]
N5      [*]
N6      [*****]
N7      [Se si ha a disposizione un solo volantino che deve essere
N8      [commutato per muovere i tre assi della macchina occorre
N9      [prevedere un selettore esterno oppure crearlo con le soft key
N10     [come in questo esempio.
N11     [Ciascuno dei tre volantini puo' essere assegnato ad
N12     [un qualunque asse controllato.
N13     [
N14     SOFTK
N15     P21,L21,'VOLANTINO ASSE X '
N16     P22,L22,'VOLANTINO ASSE Y '
N17     P23,L23,'VOLANTINO ASSE Z '
N18     P24,L24,' 0.5 mm  al giro'
N19     P25,L25,' 1 mm   al giro'
N20     P26,L26,' 5 mm   al giro'
N21     P27,L27,' 10 mm  al giro'
N22     [
N23     INIT
N24     L25=1          [default all' accensione (le lampade delle softk
N25     [              [non sono retentive allo spegnimento)
N26     [
N27     PROG
N28     [softkey per selezione l' asse da muovere con volantino
N29     IF(P21) L21="L21;L22=0;L23=0          [softk asse X
N30     IF(P22) L22="L22;L23=0;L21=0          [softk asse Y
N31     IF(P23) L23="L23;L21=0;L22=0          [softk asse Z
N32     IF(L21) HWL(1)=1                      [assegno asse X volantino 1
N33     IF(L22) HWL(1)=2                      [assegno asse Y volantino 1
N34     IF(L23) HWL(1)=3                      [assegno asse Z volantino 1
N35     IF("L21&"L22&"L23) HWL(1)=0          [nessun asse assegnato
N36     [
N37     [softkey per la selezione del passo (come configurato)
N38     IF(P24) L24=1;L25=0;L26=0;L27=0      [softk passo 0.5
N39     IF(P25) L24=0;L25=1;L26=0;L27=0      [softk passo 1
N40     IF(P26) L24=0;L25=0;L26=1;L27=0      [softk passo 5
N41     IF(P27) L24=0;L25=0;L26=0;L27=1      [softk passo 10
N42     IF(L24) STEP=1                      [assegno passo numero 1
N43     IF(L25) STEP=2                      [assegno passo numero 2
N44     IF(L26) STEP=3                      [assegno passo numero 3
N45     IF(L27) STEP=4                      [assegno passo numero 4
N46     [
N47     BURDY=0                             [...acquisizione funz. da CN
N48     RDMOV=MOVCN                         [risposta assi abilitati
N49     END
N50     [..... fine programma .....

```

SPIND1 - Rotazione mandrino

```

N1      [*****]
N2      [
N3      [      ESEMPIO GESTIONE ROTAZIONE MANDRINO
N4      [      CON TRASDUTTORE O SENZA
N5      [      SPIND1 941008
N6      [
N7      [*****]
N8      [
N9      [Gestione della rotazione mandrino in automatico M3,M4,M13,M14
N10     [e con comandi manuali, attesa regime mandrino per assi,
N11     [hold su mandrino, emergenza su mandrino non in rotazione.
N12     [Nell' attesa regime mandrino sono considerati sia
N13     [il segnale del CN (senza trasduttore determinato sulla rampa),
N14     [sia l' effettivo segnale dall' azionamento.
N15     INP
N16     IMAMAO           [comando rotazione manuale mandrino orario
N17     IMAMAA           [comando rotazione manuale mandrino antiorario
N18     ISTOPM           [comando di arresto mandrino
N19     IGIROK           [segnalazione mandrino a regime
N20     [
N21     OUT
N22     TERM,4
N23     ABM              [abilitazione azionamento mandrino
N24     [
N25     RAM,1
N26     ROTMA           [comando di rotazione
N27     G84             [memorizzazione ciclo fisso G84
N28     [
N29     STIMER
N30     TRMI,TRMU       [timer per verifica mandrino fermo x emerg.
N31     [
N32     INIT
N33     SPGAM(1)=1      [gamma 1 (unica)
N34     [
N35     PROG
N36     END
N37     IF("BURDY) ASINC
N38     FHOLD=1; DHOLD=1
N39     IF(STROM) CALL GEFUM
N40     BURDY=0
N41     ASINC:$
N42     [
N43     [***** gestione mandrino *****]
N44     [
N45     [— comandi in manuale
N46     IF (NCMD<>5) NOMANU
N47     IF (IMAMAO) CALL M03
N48     IF (IMAMAA) CALL M04
N49     IF (ISTOPM) CALL M05
N50     NOMANU:$
N51     [
N52     [—
N53     [Se in automatico la velocità è uguale a S altrimenti
N54     [forzo la velocità max (SPSMAX).
N55     [Potenziometro 3 da 70% a 130% di SPEED in automatico
N56     [ al 100% in maschiatura
N57     [ da 0% a 100% di max velocità in manuale
N58     G84=(CICFI=84) [maschiatura in corso
N59     IF (NCMD=5) SPVEL(1)=SPSMAX(1); SPSSO(1)=ANI(3); NOVEMA
N60     SPVEL(1)=SPEED
N61     IF(G84) SPSSO(1)=1; $
N62     ELSE SPSSO(1)=0.7 + ANI(3)*0.6
N63     NOVEMA:$
N64     [
N65     SPROT(1)=ROTMA&"HOLDA           [comando di rotazione e HOLD
N66     ABM=SPMOV(1)[&...             [abilitazione + consensi
N67     [
N68     IF(BRKA-EMEA) CALL RESET       [break o emergenza

```

1. Esempi di programmazione

```

N69      [
N70      [con mandrino con trasduttore emergenza se non in moto
N71      TRMI(5)=SPROT(1)&"SPMOT(1)"&"SPREG(1)           [se att. 0.5 sec -> REME
N72      IF(TRMU) DISPL,0,'MANDRINO NON IN ROTAZIONE'     [messaggio a video
N73      IF(BRKA) CLR,0           [cancello msg
N74      [
N75      [— generali —————
N76      [Attenzione al fatto che non è garantito che da
N77      [quando viene dato il comando di rotazione sia immediatamente
N78      [disponibile il segnale SPRMP(1) di mandrino in rampa.
N79      [arresto avanzamento assi
N80      FHOLD = (SPRMP(1)~"IGIROK&SPROT(1))&"RAPI [~... &("G84~RAPI)
N81      DHOLD = FHOLD [~
N82      REME = FF(TRMU),(EMEA) [emergenza per mandrino fermo
N83      END
N84      [ ..... sezione super lenta .....
N85      GIRMI=INT(ABS(SPTCH))           [visualizzazione S
N86      END
N87      [
N88      [— ROUTINES —————
N89      GEFUM: $
N90      WNDINT(1)=AUXM           [visualizzazione M
N91      IF(AUXM=3) M03
N92      IF(AUXM=4) M04
N93      IF(AUXM=5) M05
N94      RTS
N95      M03: SPDIR(1)=0; ROTMA=1; RTS
N96      M04: SPDIR(1)=1; ROTMA=1; RTS
N97      M05: ROTMA=0; RTS
N98      [
N99      RESET: $
N100     ROTMA=0           [arresto mandrino
N101     WNDINT(1)=30      [visualizzazione M30
N102     RTS
N103     [..... fine programma .....

```

SPIND2 - Orientamento mandrino

```

N1      [*****]
N2      [
N3      [          ESEMPIO GESTIONE ORIENTAMENTO MANDRINO
N4      [          SPIND2  941008
N5      [
N6      [*****]
N7      [
N8      [Gestione dell' orientamento mandrino in automatico
N9      [anche su angolo programmabile con funzione H.
N10     INP
N11     [
N12     OUT
N13     TERM,4
N14     ABM                                [abilitazione azionamento mandrino
N15     [
N16     STIMER
N17     TM19I,TM19U,TM19D,TM19A,TM19C [timer per verifica tolleranza
N18     [
N19     INIT
N20     SPGAM(1)=1                        [gamma 1 (unica)
N21     [
N22     PROG
N23     END
N24     IF("BURDY) ASINC
N25     FHOLD=1; DHOLD=1
N26     IF(STROH) CALL GEFUH
N27     IF(STROM) CALL GEFUM
N28     BURDY=0
N29     ASINC:$
N30     [
N31     [*** gestione mandrino *****]
N32     IF(BRKA-EMEA) CALL RESET          [break o emergenza
N33     [
N34     TM19I(20)=SPTOL(1)&SPORI(1)        [verifico toll. per 2 sec.
N35     IF(TM19U) SPORI(1)=0              [reset comando di orientam.
N36     [
N37     ABM=SPMOV(1)[&...                 [abilitazione + consensi
N38     [
N39     [— generali —————]
N40     DHOLD = SPORI(1)                  [attesa su blocchi seguenti
N41     FHOLD = DHOLD                    [attesa su movimento
N42     END
N43     [ ..... sezione super lenta .....
N44     END
N45     [
N46     [— ROUTINES —————]
N47     GEFUH: SPPOS(1)=(IFP(AUXH)/360)//1.0; RTS
N48     [      nota SPPOS deve essere un valore tra 0 e 1
N49     [      e rappresenta l' angolo riferito al passo (che e' 1)
N50     [
N51     GEFUM: $
N52     WNDINT(1)=AUXM                    [visualizzazione M
N53     IF(AUXM=19) M19
N54     RTS
N55     M19:SPROT(1)=0
N56     [Se si desidera l' orientamento unidirezionale settare
N57     [SPORP(1) o SPORM(1) prima di SPORI(1)!
N58     SPORI(1)=1
N59     RTS
N60     RESET:$
N61     SPORI=0
N62     WNDINT(1)=30                      [visualizzazione M30
N63     RTS
N64     [..... fine programma .....

```

1. Esempi di programmazione

SPIND3 - Cambio gamma

```

N1      [*****]
N2      [ ]
N3      [      ESEMPIO MANDRINO CON CAMBIO A DUE GAMME      ]
N4      [      SPIND3  941008      ]
N5      [ ]
N6      [*****]
N7      [ ]
N8      [Gestione del cambio gamma mandrino]
N9      INP
N10     IMG1                [micro gamma 1]
N11     IMG2                [micro gamma 2]
N12     ISGLMI              [giri mandrino sopra soglia minima]
N13     [ ]
N14     OUT
N15     TERM,4
N16     ABM                [abilitazione azionamento mandrino]
N17     KVG1               [comando elettrovalvola gamma 1]
N18     KVG2               [comando elettrovalvola gamma 2]
N19     [ ]
N20     RAM,1
N21     GAM1               [gamma 1 selezionata]
N22     GAM2               [gamma 2 selezionata]
N23     MM41               [forzamento gamma 1]
N24     MM42               [forzamento gamma 2]
N25     [ ]
N26     PROG
N27     END
N28     IF("BURDY) ASINC
N29     FHOLD=1; DHOLD=1
N30     IF(STROM) CALL GEFUM
N31     BURDY=0
N32     ASINC:$
N33     [ ]
N34     [*** gestione mandrino *****]
N35     [ ]
N36     GAM1=MM41~(SPEED<SPSMG1(1))&"MM42                [selezione gamma 1]
N37     GAM2=MM42~(SPEED>=SPSMG1(1))&"MM41                [selezione gamma 2]
N38     [ ]
N39     [attivo le e.v. solo quando sono sotto la soglia di giri min.]
N40     KVG1=GAM1&"IMG1&"ISGLMI&"SPMOT(1)                [comando e.v. gamma 1]
N41     KVG2=GAM2&"IMG2&"ISGLMI&"SPMOT(1)                [comando e.v. gamma 2]
N42     [ ]
N43     IF(IMG1) SPGAM=1                [attiva gamma 1]
N44     IF(IMG2) SPGAM=2                [attiva gamma 2]
N45     [attenzione: SPGAM=0 non permette il pendolamento.]
N46     [ ]
N47     SPPND(1)=(GAM1&"IMG1)~(GAM2&"IMG2)                [pendolamento]
N48     [Nota: CAMBIO GAMMA "AL VOLO"
N49     [      SPPND ha priorità sugli altri comandi;
N50     [      se viene chiesto un cambio gamma con mandrino
N51     [      in moto, prima di attivare il pendolamento
N52     [      viene forzata automaticamente una decelerazione fino
N53     [      alla soglia di mandrino fermo.
N54     [ ]
N55     ABM=SPMOV(1)&...                [abilitazione + consensi]
N56     [ ]
N57     [— generali —]
N58     DHOLD = SPPND(1)                [attesa su blocchi seguenti]
N59     FHOLD = DHOLD                [attesa su movimento successivo]
N60     END
N61     [ ..... sezione super lenta .....]
N62     IF(SPPND(1)) DISPL,0,'CAMBIO GAMMA IN CORSO'; ELSE CLR,0
N63     END
N64     [ ]
N65     [— ROUTINES —]
N66     GEFUM: $
N67     IF (AUXM=40) MM41=0; MM42=0; RTS
N68     IF (AUXM=41) MM41=1; MM42=0; RTS

```

```

N69 IF (AUXM=42) MM41=0; MM42=1; RTS
N70 RTS
N71 [..... fine programma .....

```

LUBMET - Lubrificazione assi a metri percorsi

```

N1 [***** ]
N2 [*          LUBRIFICAZIONE a spazio percorso ]
N3 [*          ]
N4 [*          LUBMET 941008 ]
N5 [***** ]
N6 [
N7 INP
N8 IMUON          [ausiliari inseriti
N9 ILIVOL          [livello olio
N10 [
N11 OUT
N12 ABILX          [abilitazione asse X
N13 ABILY          [abilitazione asse Y
N14 ABILZ          [abilitazione asse Z
N15 UKLUBA        [e.v. lubrif. assi
N16 [
N17 RAM,32
N18 CORSAX          [spazio percorso da X
N19 CORSAY          [spazio percorso da Y
N20 CORSAZ          [spazio percorso da Z
N21 POAOLX          [posizione assoluta X vecchia
N22 POAOLY          [posizione assoluta Y vecchia
N23 POAOLZ          [posizione assoluta Z vecchia
N24 ML              [intervallo di spazio max per lubrificazione
N25 [
N26 STR
N27 MSG1            [messaggio livello insufficiente
N28 MSG2            [messaggio ausiliari non inseriti
N29 [
N30 STIMER
N31 TLUBI,TLUBU,TLUBD,TLUBA,TLUBW [lubrificazione
N32 [
N33 SOFTK,1
N34 P1,L1,1,'LUBRIFICA'
N35 [
N36 INIT
N37 ML=15000        [spazio percorso il quale inizia la lubrif.
N38 MSG1='LIVELLO OLIO INSUFFICIENTE'
N39 MSG2='AUSILIARI DISINSERITI'
N40 [
N41 PROG
N42 END
N43 [
N44 [.....lubrificazione .....
N45 [quando almeno un asse ha percorso ML metri lubrifico
N46 TLUBI(50)=(CORSAX>ML)~(CORSAY>ML)~(CORSAZ>ML)~TLUBD
N47 [gli spazi percorsi sono incrementati
N48 [solo con assi abilitati e fuori tolleranza
N49 [cioe' in movimento; non ho problemi se asse oscilla
N50 [oppure se si e' in errore 8,asse
N51 IF("INTOL(1)&MOVCN(1)) CORSAX=CORSAX+ABS(POA(1)-POAOLX)
N52 IF("INTOL(2)&MOVCN(2)) CORSAY=CORSAY+ABS(POA(2)-POAOLY)
N53 IF("INTOL(3)&MOVCN(3)) CORSAZ=CORSAZ+ABS(POA(3)-POAOLZ)
N54 POAOLX=POA(1)    [aggiorno posizioni vecchie
N55 POAOLY=POA(2)
N56 POAOLZ=POA(3)
N57 [ad ogni lubrificazione resetto le corse fatte
N58 IF(TLUBU) CORSAX=0; CORSAY=0; CORSAZ=0
N59 [con "IMUON carico il max su CORSA in modo da lubrificare
N60 [subito all' accensione

```

1. Esempi di programmazione

```

N61 [la stessa cosa su mancanza olio
N62 IF("IMUON~"ILIVOL) CORSAX=ML; CORSAY=ML; CORSAZ=ML
N63 [
N64 [pompa per 5 secondi o su softkey P1
N65 [
N66 UKLUBA=(TLUBD~P1)&ILIVOL&IMUON [pompa lubrificazione
N67 L1 = UKLUBA [lampada lubrificazione
N68 [
N69 [.....generali.....
N70 ABILX=MOVCN(1) [abilitazioni assi
N71 ABILY=MOVCN(2)
N72 ABILZ=MOVCN(3)
N73 RDMOV=MOVCN [risposta assi abilitati
N74 BURDY=0 [acquisizione funzioni CN
N75 POFO=ANI(1) [override avanzamento
N76 FHOLD="ILIVOL [inibizione avanzamento assi
N77 DHOLD=HOLD [inibizione avanzamento blocchi
N78 REME=FF("IMUON),(EMEA) [emergenza su MU spenta
N79 [
N80 END
N81 IF("ILIVOL) DISPL,1, MSG1; ELSE CLR,1 [messaggio livello min.
N82 IF("IMUON) DISPL,2, MSG2; ELSE CLR,2 [messaggio ausiliari
N83 END
N84 [..... fine programma .....

```


LUBIN3 - Base lubrificazione a tempo

```

N1      [***** ]
N2      [* ]
N3      [* LUBRIFICAZIONE INTERMITTENTE ]
N4      [* LUBIN3 941010 ]
N5      [* ]
N6      [***** ]
N7      [===== esempio di principio 1 =====]
N8      [
N9      [La UPOMPA e' attiva per 5 secondi ogni 10 minuti
N10     [
N11     OUT
N12     UPOMPA [comando della pompa
N13     [
N14     STIMER
N15     TLI,TLU,TLD,TLA,TLW [timer del ciclo
N16     [
N17     PROG
N18     TLI(6000)="TLU [oscillatore di 600 secondi
N19     UPOMPA=(TLW>5950) [attiva per i primi 5 sec.
N20     END
N21     [..... fine programma 1 .....
N22     [
N23     [
N24     [===== esempio di principio 2 =====]
N25     [Per ottenere TEMPI LUNGHI da 1 ora a " 2 anni "
N26     [si deve combinare un timer con un contatore, esempio
N27     [attiva la UPOMPA per 5 secondi ogni 60 minuti.
N28     [
N29     OUT
N30     UPOMPA [comando della pompa
N31     [
N32     STIMER
N33     TLI,TLU,TLD,TLA,TLW [timer del clock
N34     COUNT
N35     CLZ,CLA,CLI,CLC,CLW [contasecondi
N36     [
N37     INIT
N38     CLZ(3600)=1 [preset contatore a 3600 sec
N39     CLZ(3600)=0
N40     [
N41     PROG
N42     TLI(10)="TLU [oscillatore da 1 sec
N43     CLA=TLU [conteggio
N44     POMPA=(CLW<5) [attiva la pompa per 5 sec
N45     END
N46     [..... fine programma 2 .....

```

1. Esempi di programmazione

LUBMOV - Lubrificazione a tempo con assi in movimento

```

N1      [*****]
N2      [*      LUBRIFICAZIONE a tempo con ASSI in movimento      ]
N3      [*      _____      ]
N4      [*      LUBMOV 941010      ]
N5      [*****]
N6      [
N7      INP
N8      IMUON                      [accensione ausiliari
N9      ILIVOL                      [livello olio
N10     [
N11     OUT
N12     ABILX                      [abilitazione asse X
N13     ABILY                      [abilitazione asse Y
N14     ABILZ                      [abilitazione asse Z
N15     UKLUBA                    [e.v. lubrif. assi
N16     [
N17     STR
N18     MSG1                      [messaggio livello insufficiente
N19     MSG2                      [messaggio ausiliari non inseriti
N20     [
N21     STIMER
N22     TLUBI,TLUBU,TLUBD,TLUBA,TLUBW          [lubrificazione
N23     [
N24     SOFTK,1
N25     P1,L1,1,'LUBRIFICA MANUALE'
N26     [
N27     INIT
N28     MSG1='LIVELLO OLIO INSUFFICIENTE'
N29     MSG2='AUSILIARI NON INSERITI'
N30     [
N31     PROG
N32     END
N33     [
N34     [.....lubrificazione .....
N35     [All' accensione (IMUON) il tempo si resetta
N36     [in modo da dare una pompata al primo movimento.
N37     [Il tempo scorre solo con assi in movimento.
N38     TLUBI(6000)="TLUBU&IMUON&ILIVOL          [oscillatore 10 minuti
N39     [pausa ad assi fermi o disabilitati
N40     TLUBA=((MOVCN&"INTOL)=0)
N41     [pompa per 5 secondi o su softkey P1
N42     UKLUBA=((TLUBW>5950)&"TLUBA~P1)&IMUON&ILIVOL
N43     L1 = UKLUBA          [lampada lubrificazione
N44     [
N45     [.....generali.....
N46     ABILX=MOVCN(1)          [abilitazioni assi
N47     ABILY=MOVCN(2)
N48     ABILZ=MOVCN(3)
N49     RDMOV=MOVCN            [risposta assi abilitati
N50     BURDY=0                [... acquisizione funzioni CN
N51     POFO=ANI(1)            [potenziometro override avanzamenti
N52     [
N53     [se il livello olio non e' sufficiente il programma si
N54     [ferma sul primo blocco in rapido o sulla prima funzione aux.
N55     FHOLD="ILIVOL          [inibizione avanzamento assi
N56     DHOLD="FHOLD           [inibizione avanzamento blocchi
N57     REME=FF("IMUON),(EMEA) [emergenza con MU spenta
N58     [
N59     END
N60     [..... sezione superlenta .....
N61     IF("ILIVOL) DISPL,1, MSG1; ELSE CLR,1          [messaggio livello min.
N62     IF("IMUON) DISPL,2, MSG2; ELSE CLR,2          [messaggio ausiliari
N63     END
N64     [..... fine programma .....

```

ZERIAX - Ciclo di ricerca zero assi automatico

```

N1 [*****]
N2 [*      ESEMPIO DI CICLO RICERCA ZERO AUTOMATICO SU XYZ      ]
N3 [*      _____      ]
N4 [*      ZERIAX  941008      ]
N5 [*****]
N6 [
N7 [Ciclo di ricerca zero automatico su assi
N8 [con trasduttori non assoluti:
N9 [Dapprima Z cerca zero in direzione +, quindi a zero fatto
N10[X e Y ricercano lo zero contemporaneamente verso +.
N11[
N12[***** SEZIONE DICHIARATIVA *****]
N13[
N14[      input fisici
N15[      INP
N16[      IMZX      [micro zero asse X
N17[      IMZY      [micro zero asse Y
N18[      IMZZ      [micro zero asse Z
N19[
N20[      output fisici
N21[      OUT
N22[      UMOVE1      [abilitazione asse X
N23[      UMOVE2      [abilitazione asse Y
N24[      UMOVE3      [abilitazione asse Z
N25[
N26[      variabili interne
N27[      RAM,1
N28[      RICOX      [ricerca zero X in corso
N29[      RICOY      [ricerca zero Y in corso
N30[      RICOZ      [ricerca zero Z in corso
N31[      ZERIOK      [zeri assi effettuati
N32[
N33[      stringhe per messaggi
N34[      STR
N35[      MSG1      [messaggio zero assi non effettuato
N36[      MSG2      [messaggio premere JOG Z+ per inizio ciclo
N37[
N38[      SOFTK,1
N39[      P1,L1,1,'JOG ASSE      X+'
N40[      P2,L2,1,'JOG ASSE      X-'
N41[      P3,L3,1,'JOG ASSE      Y+'
N42[      P4,L4,1,'JOG ASSE      Y-'
N43[      P5,L5,1,'JOG ASSE      Z+'
N44[      P6,L6,1,'JOG ASSE      Z-'
N45[      P7,L7, ' RICERCA ZERO  ASSI'
N46[
N47[***** SEZIONE INIZIALIZZAZIONE *****]
N48[      INIT
N49[      [inizializzazione messaggi
N50[      MSG1='Eseguire RICERCA ZERO ASSI' [messaggio zeri
N51[      MSG2='premere JOG Z+ per iniziare la ricerca'
N52[
N53[      PROG
N54[***** SEZIONE VELOCE *****]
N55[      ..... lettura potenziometri .....
N56[      POFO=ANI(1)      [avanzamento in automatico
N57[      [se zeri non fatti riduco la velocita' in manuale ad 1/5
N58[      IF(ZERIOK) $
N59[      POMO(1)=ANI(2); $
N60[      POMO(2)=POMO(1); $
N61[      POMO(3)=POMO(2); $
N62[      ELSE $
N63[      POMO(1)=ANI(2)/5; $
N64[      POMO(2)=POMO(1); $
N65[      POMO(3)=POMO(1)
N66[      END
N67[***** SEZIONE LENTA *****]
N68[      ..... decodifica delle funzioni ausiliarie .....

```

1. Esempi di programmazione

```

N69 BURDY=0 [ ... acquisizione funzioni da CN
N70 [
N71 [ ..... abilitazione assi .....
N72 UMOVE1=MOVCN(1)
N73 UMOVE2=MOVCN(2)
N74 UMOVE3=MOVCN(3)
N75 RDMOV=MOVCN
N76 [ ..... jog .....
N77 [in stato di ricerca 0 sono abilitati solo JOG positivi
N78 L1=(P1&"L7)~RIC0X [lampada softk jog x+
N79 L2=P2&"L7 [lampada softk jog x-
N80 L3=(P3&"L7)~RIC0Y [lampada softk jog y+
N81 L4=P4&"L7 [lampada softk jog y-
N82 L5=(P5&"L7)~RIC0Z [lampada softk jog z+
N83 L6=P6&"L7 [lampada softk jog z-
N84 [
N85 JOGP(1)=L1
N86 JOGM(1)=L2
N87 JOGP(2)=L3
N88 JOGM(2)=L4
N89 JOGP(3)=L5
N90 JOGM(3)=L6
N91 MOVMA=JOGP~JOGM [selezione manuale in JOG
N92 [
N93 [ ..... ricerca zeri .....
N94 [Il ciclo inizia in stato manuale premendo P7 (softk)
N95 [comando di ricerca 0
N96 ZERIOK=(MIZEA(1)&MIZEA(2)&MIZEA(3))
N97 L7=FF(P7),(P7&L7~(NCMD<>5)~BRKA~ZERIOK~(JOGIN=0))
N98 [
N99 [memorie autoretenute per ciclo zero automatico
N100 RIC0Z=FF(P5&L7),("L7~MIZEA(3))
N101 RIC0X=FF(MIZEA(3)),("L7~MIZEA(1))
N102 RIC0Y=FF(MIZEA(3)),("L7~MIZEA(2))
N103 [
N104 [selezione ricerca zero con micro presente
N105 MICZE(1)=L7
N106 MICZE(2)=L7
N107 MICZE(3)=L7
N108 [assegnazioni micro di zero
N109 MIZER(1)=IMZX
N110 MIZER(2)=IMZY
N111 MIZER(3)=IMZZ
N112 [
N113 [selezione ricerca zero senza micro
N114 [sostituire MARK() con MICZE() e non assegnare MIZER()
N115 [MARK(1)=L7
N116 [MARK(2)=L7
N117 [MARK(3)=L7
N118 [ ..... generali .....
N119 FHOLD=(NCMD<>5)&"ZERIOK
N120 DHOLD=0 [ ...
N121 END
N122 [***** SEZIONE SUPERLENTA *****
N123 IF ("ZERIOK) DISPL, 0, MSG1; ELSE CLR, 0 [messaggio zeri
N124 IF ("ZERIOK&L7) DISPL, 1, MSG2; ELSE CLR, 1 [messaggio premi Z+
N125 [
N126 END
N127 [ ..... fine programma .....

```

ESRNDU - Cambio utensile random con presa e posa in tempo mascherato

```

N1 [*****]
N2 [* CAMBIO UTENSILE RANDOM ASINCRONO CON S3000 ]
N3 [* CATENA con 24 utensili e ricerca più breve ]
N4 [* ]
N5 [* ESRNDU 9401008 ]
N6 [*****]
N7 [
N8 [***** PROGRAMMI COM x MOVIMENTI ASSI LANCIATI DA PLC *****]
N9 [CUAUTO:
N10 [P1=100 [Quota X di cambio utensile
N11 [P2=-100 [Quota Y di cambio utensile
N12 [P3=150 [Quota Z di sicurezza
N13 [P4=50 [Quota Z di cambio utensile
N14 [
N15 [
N16 [O0 [origine assoluta
N17 [M26 [Su sequenza 4 posa manuale a terra
N18 [M62 [apre riparo magazzino
N19 [ZP3RM19 [Z sicurezza + orienta mandrino
N20 [XP1YP2R [X Y in posizione
N21 [ZP4R [Z di cambio
N22 [M... [Funzioni M di cambio utensile
N23 [G4K5 [0.5 sec
N24 [...
N25 [O-1 [ripristina origine
N26 [M29 [attiva correttore
N27 [M63 [chiude riparo magazzino
N28 [M34 [fine cambio utensile
N29 [..... fine programma .....
N30 [ fine CUAUTO
N31 [
N32 [CUMANU:
N33 [M26 [cambiare utensile manualmente
N34 [M29 [attivazione correttore
N35 [M34 [fine cambio utensile
N36 [
N37 [
N38 [CORR:
N39 [M29 [attivazione correttore
N40 [M34 [fine cambio utensile
N41 [
N42 [
N43 INP
N44 IAUXON [ 1 Ausiliari inseriti
N45 IZERM [ 2 Micro di zero magazzino
N46 IRIMAA [ 3 Riparo magazzino aperto
N47 IRIMAC [ 4 Riparo magazzino chiuso
N48 [... [ altri ...
N49 [
N50 OUT
N51 ABX [ 1 abilitazione asse X
N52 ABY [ 2 abilitazione asse Y
N53 UABMAG [ 3 abilitazione magazzino
N54 ABZ [ 4 abilitazione asse Z
N55 UARIMA [ 5 output apertura riparo magazzino
N56 UCRIMA [ 6 output chiusura riparo magazzino
N57 [... [ altri ...
N58 RAM,16
N59 PORIT [posizione finale per movimenti magazzino
N60 [
N61 RAM,1
N62 RICUT [ciclo di posizionamento Magazzino in corso
N63 INPOS [magazzino in posizione valida
N64 ERRM06 [M6 programmata senza T
N65 [
N66 MM26 [cambiare utensile manualmente
N67 MM62 [apre riparo magazzino
N68 MM63 [chiude riparo magazzino

```

1. Esempi di programmazione

```

N69      MM66                                [sospensione sequenza per inizio deposito
N70      CIM6                                [ciclo di M06 in corso
N71      [...      [altre ...
N72      STR
N73      MSG(10)                                [testo messaggi ed allarmi
N74      [
N75      STIMER
N76      TIRIC,TURIC,TDRIC,TARIC,TWRIC      [validifica di SGLP2P
N77      [
N78      SOFTK,1
N79      [
N80      P1,CUAUT,      'cambio utensile AUTOMAT.'
N81      P2,CUMAN,      'cambio utensile MANUALE'
N82      P3,L3,
N83      P4,L4,      'fine CU manuale'
N84      P5,L5,
N85      P6,L6,
N86      P7,L7,      'RESET CU'
N87      P8,L8,
N88      [
N89      [
N90      INIT      [SEZIONE INIZIALIZZAZIONE
N91      [
N92      MSG(1)= 'VERIFICARE TABELLA UTENSILI E RESETTARE IL CU'
N93      MSG(2)= 'cambiare utensile manualmente'
N94      MSG(3)= 'M6 programmata senza Txx'
N95      MSG(4)= 'Attesa apertura riparo'
N96      MSG(5)= 'Attesa chiusura riparo'
N97      [
N98      SSA=00000111B                                [XYZ sempre attivi
N99      [
N100     [***** DEFINIZIONE SEQUENZE CAMBIO UTENSILE *****
N101     [... SEQUENZA CU PER CARICO UT DA TERRA CON MANDRINO VUOTO ...
N102     DEF SEQCU(1)=-6,-16,-34,COM,1,'CUMANU'
N103     [
N104     [... SEQUENZA CU PER SCARICO MANDRINO A TERRA (T0M6) ...
N105     DEF SEQCU(2)=-6,-10,-34,COM,1,'CUMANU'
N106     [
N107     [... SEQUENZA CU PER SCAMBIO TRA MANDRINO E TERRA ...
N108     DEF SEQCU(3)=-6,-10,-16,-34,COM,1,'CUMANU'
N109     [
N110     [... SEQUENZA CU PER POSA MANDRINO TERRA E CARICO DA MAG. ...
N111     DEF SEQCU(4)=-1,901,-5,-6,-10,-17,-34,COM,1,'CUAUTO'
N112     [
N113     [... SEQUENZA CU PER POSA MANDR. IN MAGAZ. E PRESA DA TERRA ...
N114     DEF SEQCU(5)=-23,923,-6,-12,-16,66,26,-27,-34,COM,1,'CUAUTO'
N115     [
N116     [... SEQUENZA CU PER SCAMBIO UTENS.CON MANDR. CARICO ...
N117     DEF SEQCU(6)=-1,901,-5,-6,-12,-17,66,-23,923,-27,-34, $
N118     COM,1,'CUAUTO'
N119     [
N120     [... SEQUENZA CU PER CARICO DA MAGAZZ. CON MANDR.SCARICO ...
N121     DEF SEQCU(7)=-1,901,-5,-6,-17,-34, $
N122     COM,1,'CUAUTO'
N123     [
N124     [... SEQUENZA CU PER SCARICO UTENS.DA MANDR. IN MAGAZZ....
N125     DEF SEQCU(8)=-23,923,-6,-12,66,-27,-34, COM,1,'CUAUTO'
N126     [
N127     [... SEQUENZA CU PER CARICO UTENS.= UTENS. NEL MANDR. ...
N128     DEF SEQCU(11)=-6,-34,COM,1,'CORR'
N129     [
N130     [... T programmata dopo una T (nella fase di attesa M06) ...
N131     [... deposito PINZA in magazzino e rianalizza situazione ...
N132     DEF SEQCU(19)=923,-23,-31,0
N133     [
N134     [NOTA: se si da' la possibilità di muovere il magazzino
N135     [      in JOGCU già dopo che il magazzino si e'
N136     [      posizionato in automatico; occorre rimettere una
N137     [      fase di ricerca posto (901) o (923) dopo la -6.
N138     [
N139     PROG      [SEZIONE VELOCE

```

```

N140 END [SEZIONE LENTA
N141 ABX=MOVCN(1) [abilitazione assi
N142 ABY=MOVCN(2)
N143 ABZ=MOVCN(3)
N144 RDMOV=MOVCN
N145 POFO=ANI(1) [potenziometro avanzamento assi
N146 [
N147 [
N148 [-----PARTE SINCRONA-----
N149 [
N150 IF("BURDY)ASINC
N151 FHOLD=1; DHOLD=1
N152 [Nella decodifica ci vuole sempre prima T e poi M
N153 IF(STROT)CALL GEFUT
N154 IF(STROM)CALL GEFUM
N155 BURDY=0
N156 ASINC:$
N157 [-----PARTE ASINCRONA-----
N158 [*****
N159 [ GESTIONE MODULO CU AUTOMATICO *
N160 [*****
N161 CALL CUAUTO [routine gestione CU automatico
N162 [
N163 CALL POSMAG
N164 [..... attuazioni fisiche per cambio ut .....
N165 [le sicurezze di movimentazione vanno messe direttamente
N166 [sui comandi (uscite), per esempio:
N167 [out =((comando_auto) ~ (comando_manu)) & secur_mecc.
N168 [
N169 UARIMA=MM62 [&... secur.
N170 UCRIMA=MM63 [&... secur.
N171 [...
N172 L4=MM26 [cambio ut. manuale in corso
N173 [...
N174 [
N175 [reset memorie per fine comando (comandi eseguiti)
N176 IF(IRIMAA&"IRIMAC) MM62=0 [sportello aperto
N177 IF(IRIMAC&"IRIMAA) MM63=0 [sportello chiuso
N178 IF(P4) MM26=0 [ok fine CU manuale
N179 [*****
N180 [ ALTRE GESTIONI ASINCRONE *
N181 [*****
N182 [
N183 [...
N184 [...
N185 [
N186 [*****
N187 [ ALLARMI,CONSENSI E SICUREZZE *
N188 [*****
N189 [relativi al CN
N190 DHOLD=EMACU~MM26~MM62~MM63~EMAP2P(1) [ ~...
N191 FHOLD=DHOLD [ ~...
N192 REME=FF("IAUXON),(EMEA) [ ~... [Rich. di emergenza al CN
N193 [
N194 END
N195 [----- SEZIONE SUPERLENTA -----
N196 [..... visualizzazione messaggi .....
N197 IF(EMACU) DISPL,1,MSG(1); ELSE CLR,1 [CU in emergenza
N198 IF(MM26) DISPL,2,MSG(2); ELSE CLR,2 [cambiare ut. manu
N199 IF(ERRM06) DISPL,3,MSG(3); ELSE CLR,3 [M6 senza T pronta
N200 IF(MM62) DISPL,4,MSG(4); ELSE CLR,4 [attesa riparo ap.
N201 IF(MM63) DISPL,5,MSG(5); ELSE CLR,5 [attesa riparo ch.
N202 [
N203 WINDOW=IFP(UTSPCU) [Visualizzazione Tool nel mandrino
N204 ASCW=116 [Codice carattere 't'
N205 [La visualizzazione è molto utile se si usano correttori
N206 [alternativi (la finestra T sul video è il correttore
N207 [attivo e non l' utensile).
N208 [
N209 END
N210 [

```

1. Esempi di programmazione

```

N211 [----- SEZIONE ROUTINES -----]
N212 [
N213 [*****]
N214 [      FUNZIONE T      *]
N215 [*****]
N216 GEFUT:$
N217 [.....Attivazione correttore alternativo .....]
N218 [Ha senso solo se non si usano le famiglie utensili:
N219 [si possono interpretare, per esempio,
N220 [i codici maggiori di 100 (che andranno quindi inseriti
N221 [nella tabella utensili):
N222 IF(TOOL>100) OFST=TOOL; INTOF=1; RTS
N223 [
N224 [..... CHIAMATA DEL CAMBIO UTENSILE .....]
N225 UTECU=TOOL      [comunico l' utensile voluto al modulo CU
N226 NEWCU=1          [richiedo attivazione modulo CU
N227 RTS
N228 [
N229 [*****]
N230 [      FUNZIONI M      *]
N231 [*****]
N232 GEFUM:$
N233 WNDINT(1)=AUXM
N234 IF(AUXM=6) M06
N235 IF(AUXM=30) CALL RESET; RTS
N236 IF("CUATT) RTS
N237 IF(AUXM=62) MM62=1; RTS
N238 IF(AUXM=63) MM63=1; RTS
N239 IF(AUXM=26) M26
N240 IF(AUXM=29) INTOF=1; RTS
N241 IF(AUXM=34) CUATT=0; CIM6=0; RTS
N242 RTS
N243 [
N244 M06:$
N245 IF("CUATT) ERRM06=1; RTS      [M6 senza T
N246 M6PGM=1; CIM&=1
N247 RTS
N248 [
N249 M26:$
N250 IF(NSEQCU<5) MM26=1; RTS      [CU manuale solo in SEQ 1,2,3,4
N251 RTS
N252 [
N253 [*****]
N254 [      GESTIONE CAMBIO UTENSILE AUTOMATICO *]
N255 [*****]
N256 [..... selezione modo CU .....]
N257 CUAUTO:$
N258 IF(CUATT) NOSELE
N259 IF(P1) SELECU=0      [CU automatico (default)
N260 IF(P2) SELECU=1      [CU manuale (magazzino escluso)
N261 NOSELE:$
N262 [
N263 [lampade softkeys selezione modo
N264 CUAUT=(SELECU=0)
N265 CUMAN=(SELECU=1)
N266 [
N267 [*****]
N268 [... interruzione sequenza, cancellazione emergenze ....]
N269 [
N270 [Il CU viene interrotto solo se:
N271 [- si spengono gli ausiliari durante un CU (non in attesa M6)
N272 [- viene dato un BREAK durante la sequenza di cambio
N273 [
N274 [L' interruzione viene fatta con REMCU in quanto
N275 [il CU risponde attivando EMACU
N276 REMCU=FF(((BRKA&CIM6)~("IAUXON&CUATT))&(OPERCU<>-6)),(EMACU)
N277 [
N278 [La softkey P7 fa RBKCU per uscire da EMACU (emergenza)
N279 IF(P7&EMACU) RBKCU=1; RBRK=1 [cancellazione emergenza CU
N280 [
N281 [In caso di interruzione e' necessario resettare il CU

```



```

N282 [con l' apposita softkey dopo avere VERIFICATO LA TABELLA UTE.
N283 L7=EMACU [lampada CU in emergenza
N284 [
N285 IF(EMACU) CALL RESECU [resetto comunque i comandi PLC
N286 [
N287 [*****
N288 [..... decodifica codici fasi sequenza intrapresa .....
N289 IF ("BRDYCU) NOCU
N290 MAPRCU=0 [arresto avanzamento fasi
N291 CALL OPER [gestione fasi cambio ut.
N292 BRDYCU=0 [fase CU acquisita
N293 NOCU:$
N294 [
N295 [*****
N296 [..... consenso avanzamento fasi .....
N297 IF("CUATT) MM66=0 [terminata parte sincrona con M6
N298 [ [ok inizio deposito tempo mascherato
N299 MAPRCU="MM66&"RICUT [&"... &"...
N300 [
N301 RTS
N302 [***** RITORNO DA GESTIONE CUAUTO *****
N303 [
N304 [*****
N305 [ ROUTINE DECODIFICA OPERAZIONI CU e RESET *
N306 [*****
N307 [caso di reset CU
N308 RESECU:$
N309 MM26=0 [reset cambio utensile
N310 MM62=0
N311 MM63=0
N312 MM66=0
N313 RICUT=0
N314 CIM6=0
N315 [reset normale (M30 o break)
N316 RESET:$
N317 WNDINT(1)=30 [visualizza M30
N318 ERRM06=0 [cancello errore su M6 (M6 senza T pronta)
N319 RTS
N320 [-----
N321 [gestione OPERAZIONI cu
N322 OPER:$
N323 IF(OPERCU=26) CU26
N324 IF(OPERCU=66) CU66
N325 IF(OPERCU=901) CU901
N326 IF(OPERCU=923) CU923
N327 [...
N328 RTS
N329 [cambio utensile manuale (solo sequenza 5)
N330 CU26:$
N331 MM26=1
N332 RTS
N333 [
N334 [attesa fine cambio parte sincrona
N335 CU66:$
N336 MM66=1
N337 RTS
N338 [
N339 [cerca posto per prelievo
N340 CU901:$
N341 PORIT=PPRECU
N342 RICUT=1
N343 RTS
N344 [cerca posto per deposito
N345 CU923:$
N346 PORIT=PPOSCU
N347 RICUT=1
N348 RTS
N349 [
N350 [*****
N351 [ POSIZIONAMENTO MAGAZZINO: ASSE PUNTO PUNTO *
N352 [*****

```

1. Esempi di programmazione

```

N353 [se posizione ok viene resettato RICUT
N354 POSMAG:$
N355 SSAP2P(1)=1 [magazzino sempre abilitato
N356 UABMAG=MOV2P(1) [abilitazione asse magaz.
N357 RDMP2P(1)=MOV2P(1) [risposta asse abilitato
N358 INPOS=SGLP2P(1)&MZAP2P(1)&"RUNP2P(1)&"RICUT&"EMAP2P(1)[pos. ok
N359 [
N360 IF("RICUT) RTS [non necessita posizionamento
N361 POTP2P(1)=1 [potenziometro velocità
N362 MIZP2P(1)=IZERM [micro zero magazzino
N363 IF("MZAP2P(1)) ZEMAG [test asse azzerato
N364 JGPP2P(1)=0 [se azzerato reset JOG
N365 MCZP2P(1)=0 [se azzerato reset ricerca zero
N366 [calcolo posizione da raggiungere (via più breve)
N367 PFNP2P(1)=IFP(PORIT)-NEI((IFP(PORIT)-NEI(POAP2P(1)))/24)*24
N368 IF(RICUT) RUNP2P(1)=1 [lancio il movimento
N369 TIRIC(5)=RUNP2P(1)-TDRIC [segnale x accendere INPOS
N370 [notare che qui entro solo se MZAP2P e' presente
N371 IF(SGLP2P(1)&"TDRIC) RICUT=0 [movimento terminato
N372 RTS
N373 ZEMAG:JGPP2P(1)=RICUT [ricerca zero
N374 MCZP2P(1)=RICUT [setto la ricerca zero
N375 INPOS=0 [resetto posizione OK
N376 RTS
N377 [..... fine programma .....

```

SCROLLIN - Gestione fino a 128 messaggi con scrolling su video

```

N1 [*****
N2 [* *
N3 [* Programma di scrolling messaggi *
N4 [* SCROLLIN 940516 *
N5 [* *
N6 [*****
N7 [
N8 [VISUALIZZA UN MASSIMO DI 16 MESSAGGI CONTEMPORANEI.
N9 [IMPACCANDO IN SEQUENZA SOLO QUELLI ATTIVI
N10 [(nell' esempio NMAX=48) DICHIARATI
N11 [Settando il bit n-esimo di SG visualizza il
N12 [messaggio n-esimo con scrolling automatico.
N13 [
N14 INP
N15 I1 [ingresso di attivazione messaggio 1
N16 I2 [ingresso di attivazione messaggio 10
N17 I3 [ingresso di attivazione messaggio 47
N18 [
N19 RAM,8
N20 NMSG [indice conteggio Numero MeSsaGgio
N21 NMAX [numero MAX di messaggi
N22 NRIGA [numero della riga del messaggio
N23 [Dichiarare fino a SGxx dove (xx) >NMAX/8
N24 SG [flag messaggi 1 - 8
N25 SG2 [flag messaggi 9 - 16
N26 SG3 [flag messaggi 17 - 24
N27 SG4 [flag messaggi 25 - 32
N28 SG5 [flag messaggi 33 - 40
N29 SG6 [flag messaggi 41 - 48
N30 [
N31 STR
N32 MSG(48) [vettore per i testi dei 48 messaggi
N33 [ dichiarare NMAX elementi
N34 [
N35 INIT
N36 NMAX=48 [Numero Max MESSAGGI
N37 [
N38 [Scrittura messaggi utilizzati
N39 MSG(1)= 'ALLARME ASSI: CONTROLLARE FUSIBILI AZIONAMENTO'
N40 MSG(2)= 'SOVRACCARICO MOTORE LUBRIFICAZIONE SLITTE'

```

```

N41 MSG(3)= 'SOVRACCARICO MOTORE REFRIGERANTE'
N42 MSG(4)= 'SOVRACCARICO MOTORE VENTILATORE MANDRINO'
N43 MSG(10)= 'SOVRATEMPERATURA MOTORE MANDRINO'
N44 MSG(17)= 'SOVRACCARICO AZIONAMENTO ASSI'
N45 MSG(18)= 'DISFUNZIONAMENTO ARIA COMPRESSA'
N46 MSG(19)= 'ASSI IN OLTRECORSO'
N47 MSG(47)= 'AZIONAMENTO MANDRINO NON PRONTO'
N48 MSG(48)= '--'
N49 [
N50 [
N51 [ .....PROGRAMMA.....
N52 PROG
N53 END
N54 END
N55 [ .....sezione superlenta.....
N56 [ Attivazione messaggi
N57 SG(1)=I1
N58 SG(10)=I2
N59 SG(47)=I3
N60 [
N61 CALL SCROLL [ Chiamata alla routine di gestione messaggi
N62 [
N63 END
N64 [
N65 [..... sezione routines .....
N66 [ ..... GESTIONE MESSAGGI SU VIDEO .....
N67 SCROLL:$
N68 NMSG=1; NRIGA=1 [inizializzazione variabili
N69 LOOVIS: IF(NMSG > NMAX) CLRSCR [se fine scansione CLR righe
N70 IF(NRIGA>16) RTS [pi— di 16 messaggi
N71 IF(SG(NMSG)) DISPL, NRIGA, MSG(NMSG); NRIGA=NRIGA+1 [DISPL
N72 NMSG=NMSG+1; LOOVIS [testo altri SG
N73 CLRSCR: IF(NRIGA>16) RTS [altre righe da cancellare
N74 CLR,(NRIGA); NRIGA=NRIGA+1; CLRSCR [cancello righe success.
N75 [ ..... fine programma .....

```

SHIFTZ - Esempio compensazione caduta Y in funzione di Z

```

N1 [*****]
N2 [* ESEMPIO COMPENSAZIONE CADUTA Y IN FUNZIONE DI Z ]
N3 [* ]
N4 [* SHIFTZ 940516 ]
N5 [*****]
N6 [
N7 [Compensazione asse verticale Z in funzione
N8 [della caduta dello slittone Y orizzontale
N9 [Naturalmente la compensazione viene attuata solo se
N10 [gli assi sono asserviti; in caso contrario la compensazione
N11 [comporta uno shift della quota asse e la compensazione
N12 [verra' attuata successivamente ad asse abilitato.
N13 [
N14 [***** SEZIONE DICHIARATIVA *****]
N15 [ input fisici
N16 INP
N17 [ output fisici
N18 OUT
N19 UMOVE1 [abilitazione asse 1
N20 UMOVE2 [abilitazione asse 2
N21 UMOVE3 [abilitazione asse 3
N22 [
N23 [ variabili interne
N24 SRAM,32
N25 TABCOZ(11) [tabella dei valori con cui compensare Z
N26 [
N27 RAM,32
N28 FCYP [quota del finecorsa Y positivo
N29 FCYN [quota del finecorsa Y negativo
N30 NCAMPY [numero degli intervalli

```

1. Esempi di programmazione

```

N31 STEPY [valore dell' intervallo]
N32 QUOYI [quota di Y riferita al FC negativo]
N33 COMPZ [valore di compensazione corrente]
N34 IND [numero dell' intervallo corrente]
N35 [
N36 RAM,8
N37 IND8 [numero dell' intervallo corrente in formato byte]
N38 [
N39 INIT
N40 FCYP=100 [quota di finecorsa software Y+]
N41 FCYN=-200 [quota di finecorsa software Y-]
N42 NCAMPY=10 [numero degli intervalli di compensazione]
N43 STEPY=(FCYP-FCYN)/NCAMPY [calcolo valore intervallo]
N44 [
N45 PROG
N46 [***** SEZIONE VELOCE *****]
N47 POFO=ANI(1) [potenziometro assi]
N48 [
N49 UMOVE1=MOVCN(1) [abilitazione assi]
N50 UMOVE2=MOVCN(2)
N51 UMOVE3=MOVCN(3)
N52 RDMOV=MOVCN
N53 END
N54 [***** SEZIONE LENTA *****]
N55 [..... funzioni ausiliarie .....]
N56 [
N57 BURDY=0 [...acquisizione funzioni da CN]
N58 [
N59 [..... compensazione caduta .....]
N60 QUOYI=POA(2)-FCYN [quota relativa rispetto al fc soft -]
N61 IND=INT(QUOYI/STEPY) [numero dell' intervallo corrente]
N62 IND8=FPI(IND) [intervallo in formato byte]
N63 COMPZ=((QUOYI-STEPY*IND)*(TABCOZ(IND8+2)-TABCOZ(IND8+1)))/$
N64 STEPY)+TABCOZ(IND8+1) [interpolazione dentro all' intervallo]
N65 [limitazioni al di fuori dei finecorsa]
N66 IF(POA(2)<=FCYN) COMPZ=TABCOZ(1)
N67 IF(POA(2)>FCYP) COMPZ=TABCOZ(FPI(NCAMPY+1))
N68 SHIFT(3)=COMPZ [attuo la compensazione]
N69 [
N70 END
N71 [..... fine programma .....]

```

AXBLOC1 - Bloccaggio assi con attesa a tempo

```

N1 [*****]
N2 [* GESTIONE ASSE CON BLOCCAGGIO/SBLOCCAGGIO A TEMPO ]
N3 [* ]
N4 [* AXBLOC1 941010 ]
N5 [*****]
N6 [
N7 INP
N8 IMUON [ 1 ausiliari inseriti]
N9 IDRAOK [ 2 drives assi OK]
N10 [
N11 OUT
N12 UMOVE1 [1 abilitazione asse 1]
N13 TERM,5
N14 USFREX [6 sbloccaggio asse X]
N15 [
N16 [ variabili interne]
N17 RAM,8
N18 MOVCPN [copia dei MOVCN vecchi per variazioni]
N19 [
N20 STIMER
N21 TISBX,TUSBX,TDSBX,TASBX,TWSBX[sbloccaggio asse X]
N22 TIBLX,TUBLX,TDBLX,TABLX,TWBLX [bloccaggio asse X]
N23 [

```

```

N24  PROG
N25  END
N26  [***** SEZIONE LENTA *****]
N27  [..... funzioni ausiliarie .....]
N28  BURDY=0 [ ... acquisizione funzioni da CN
N29  POFO=ANI(1) [potenziometro avanzamento assi
N30  [
N31  [..... gestione assi.....]
N32  TISBX(3)=MOVCN(1) [timer sbloccaggio
N33  TIBLX(5)=("MOVCN(1)&MOVCNP(1))~TDBLX [timer bloccaggio
N34  [
N35  UMOVE1=(MOVCN(1)~TDBLX)&IMUON&IDRAOK [abilitazione assi
N36  USFREX=MOVCN(1)&IMUON&IDRAOK [sbloccaggio
N37  [
N38  RDMOV(1)=(MOVCN(1)&"TDSBX)~TDBLX [risposte al CN
N39  MOVCNP=MOVCN [per derivata MOVCN
N40  [
N41  REME=FF("IMUON~"IDRAOK),(EMEA) [Richiesta emergenza
N42  END
N43  IF(RDMOV<>MOVCN) DISPL,1,'ATTESA BLOCCAGGIO/SBLOCCAGGIO ASSI';$
N44  ELSE CLR,1
N45  IF("IMUON) DISPL,2,'AUSILIARI NON INSERITI'; ELSE CLR,2
N46  IF("IDRAOK) DISPL,3,'ANOMALIA AZIONAMENTI ASSI'; ELSE CLR,3
N47  END
N48  [..... fine programma .....]

```

AXBLOC2 - Sbloccaggio assi con consenso esterno

```

N1  [*****]
N2  [* GESTIONE BLOCCAGGIO/SBLOCCAGGIO ASSE con PRESSOSTATO ]
N3  [* ]
N4  [* AXBLOC2 941010 ]
N5  [*****]
N6  [Si prevede lo sbloccaggio con pressostato
N7  [ed il bloccaggio con un tempo aggiuntivo.
N8  [
N9  [ input fisici
N10 [
N11 INP
N12 IMUON [ 1 ausiliari inseriti
N13 IDRAOK [ 2 drives assi OK
N14 ISBLOX [ 3 asse X sbloccato (pressostato)
N15 [
N16 [ output fisici
N17 OUT
N18 UMOVE1 [1 abilitazione asse 1
N19 TERM,5
N20 USFREX [6 sbloccaggio asse X
N21 [
N22 [ variabili interne
N23 RAM,8
N24 MOVCNP [copia dei MOVCN vecchi per variazioni
N25 [
N26 STIMER
N27 TIBLX,TUBLX,TDBLX,TABLX,TWBLX [bloccaggio asse X
N28 [
N29 PROG
N30 END
N31 [***** SEZIONE LENTA *****]
N32 [..... varie .....]
N33 POFO=ANI(1) [override avanzamenti
N34 BURDY=0 [ ... acquisizione funzioni da CN
N35 [
N36 [..... gestione assi.....]
N37 TIBLX(5)=("MOVCN(1)&RDMOV(1)&"ISBLOX) [timer bloccaggio X
N38 [
N39 UMOVE1=(MOVCN(1)~RDMOV(1))&IMUON&IDRAOK [abilitazione X

```

1. Esempi di programmazione

```

N40  USFREQ=MOVCN(1)&IMUON&IDRAOK                                [sbloccaggio X
N41  [
N42  RDMOV(1)=(MOVCN(1)&ISBLOX)~RDMOV(1)&"(TUBLX-EMEA)[risposte CN
N43  [
N44  END
N45  [***** SEZIONE SUPER LENTA *****]
N46  IF(RDMOV<=>MOVCN) DISPL,1,'ATTESA BLOCCAGGIO/SBLOCCAGGIO ASSI';$
N47  ELSE CLR,1
N48  IF("IMUON) DISPL,2,'AUSILIARI NON INSERITI'; ELSE CLR,2
N49  IF("IDRAOK) DISPL,3,'ANOMALIA AZIONAMENTO ASSI'; ELSE CLR,3
N50  END
N51  [..... fine programma .....

```

ESSNCU - Cambio utensile sincrono a rastrelliera

```

N1  [*****]
N2  [* CAMBIO UTENSILE POSTI FISSI SINCRONO A RASTRELLIERA ]
N3  [*          ESSNCU          941010 ]
N4  [*          _____ ]
N5  [* ]
N6  [*****]
N7  [
N8  [***** PROGRAMMI COM x MOVIMENTI ASSI LANCIATI DA PLC *****]
N9  [CUAUTO:
N10 [P1=4          [numero utensili su riga
N11 [P2=6          [numero utensili su colonna
N12 [P3=10        [interasse ut. su riga
N13 [P4=20        [interasse ut. su colonna
N14 [P5=0          [posizione X 1^ utensile
N15 [P6=0          [posizione Y 1^ utensile
N16 [P7=150       [posizione Z alto
N17 [P8=100       [posizione Z di cambio
N18 [
N19 [P34=1        [parametro sempre a 1
N20 [[P10         [caricato da PLC: posizione di presa
N21 [[P11         [caricato da PLC: posizione di posa
N22 [[P13         [caricato da PLC: numero della sequenza
N23 [[P14,P15,P16 [parametri temporanei
N24 [[P17         [quota X utensile richiesto
N25 [[P18         [quota Y utensile richiesto
N26 [_____
N27 [M62          [apre riparo magazzino
N28 [O0           [origini assolute
N29 [testa quale caso:
N30 [sono salti e non Call !
N31 [{P13=6} L6    [scambio con magazzino
N32 [{P13=7} L7    [carico utensile da magazzino
N33 [{P13=8} L8    [scarica utensile in magazzino
N34 [{P13=4} L4    [posa mandrino a terra e carica da magaz.
N35 [{P13=5} L5    [posa mandrino in magaz. e presa terra
N36 [{P34=1} L34   [vai a fine CU (solo per sicurezza)
N37 [_____
N38 [caso 6
N39 [L=6
N40 [posa: _____
N41 [ZP7RM19      [Z sicurezza
N42 [P14=P11      [carica posizione per posa
N43 [L99          [chiama routine per X, Y utensile
N44 [XP17YP18R    [va su posto posa
N45 [ZP8R         [Z di cambio
N46 [M64          [sblocca utensile
N47 [G4K5         [0.5 sec
N48 [ZP7R         [Z sicurezza
N49 [
N50 [prelievo: _____
N51 [P14=P10      [carica posizione per prelievo

```

```

N52 [L99 [chiama routine per X, Y utensile
N53 [XP17YP18R [va su posto prelievo
N54 [ZP8R [Z di cambio
N55 [M65 [blocca utensile
N56 [G4K5 [0.5 sec
N57 [ZP7R [Z sicurezza
N58 [{P34=1} L34 [vai fine _____
N59 [
N60 [... altri casi (L=...)
N61 [L=7
N62 [M0 [Sequenza da definire
N63 [{P34=1} L34 [vai fine _____
N64 [
N65 [L=8
N66 [M0 [Sequenza da definire
N67 [{P34=1} L34 [vai fine _____
N68 [
N69 [L=4
N70 [M0 [Sequenza da definire
N71 [{P34=1} L34 [vai fine _____
N72 [
N73 [L=5
N74 [M0 [ Sequenza da definire
N75 [{P34=1} L34 [vai fine _____
N76 [
N77 [parte comune a tutti i casi:
N78 [L=34
N79 [O-1 [ripristina origine
N80 [M29 [attiva correttore
N81 [M63 [chiude riparo magazzino
N82 [M34 [fine cambio utensile
N83 [G32 [fine programma
N84 [_____ routine di calcolo X, Y utensile _____
N85 [L=99
N86 [P15=INT(P14-1)/P1
N87 [P16=P14-(P1*P15)-1
N88 [P17=P5+P16*P4
N89 [P18=P6+P15*P3
N90 [G32
N91 [_____ fine CUAUTO _____
N92 [CUMANU:
N93 [P90=1
N94 [{P13=1} L1 [presa da terra
N95 [{P13=2} L1 [posa a terra
N96 [{P13=3} L1 [scambio con terra
N97 [{P13=11} L11 [Tprog. = Tmandrino
N98 [{P90=1} L90
N99 [
N100 [casi 1, 2, 3
N101 [L=1
N102 [M26 [cambiare utensile manualmente
N103 [M29 [attivazione correttore
N104 [{P90=1} L90
N105 [
N106 [caso 11
N107 [L=11
N108 [M29 [attivazione correttore
N109 [
N110 [L=90
N111 [M34 [fine cambio utensile
N112 [_____
N113 [
N114 [INP
N115 [IAUXON [ 1 Ausiliari inseriti
N116 [IRIMAA [ 2 Riparo magazzino aperto
N117 [IRIMAC [ 3 Riparo magazzino chiuso
N118 [... [ altri ...
N119 [
N120 [OUT
N121 [ABX [ 1 abilitazione asse X
N122 [ABY [ 2 abilitazione asse Y

```

1. Esempi di programmazione

```

N123 ABZ [ 4 abilitazione asse Z
N124 UARIMA [ 7 output apertura riparo magazzino
N125 UCRIMA [ 8 output chiusura riparo magazzino
N126 [... [ altri ...
N127 [
N128 RAM,1
N129 ERRM06 [M6 programmata senza T
N130 [ [memorie comandi automatici CU
N131 MM26 [cambiare utensile manualmente
N132 MM62 [apre riparo magazzino
N133 MM63 [chiude riparo magazzino
N134 [... [altre ...
N135 [
N136 STR
N137 MSG(10) [testo messaggi ed allarmi
N138 [
N139 SOFTK,1
N140 [ '+-----+'
N141 P1,CUAUT, 'cambio utensile AUTOMAT.'
N142 P2,CUMAN, 'cambio utensile MANUALE'
N143 P3,L3, '
N144 P4,L4, 'fine CU manuale'
N145 P5,L5, '
N146 P6,L6, '
N147 P7,L7, 'RESET CU'
N148 P8,L8, '
N149 [
N150 [
N151 INIT [SEZIONE INIZIALIZZAZIONE
N152 [
N153 MSG(1)= 'VERIFICARE TABELLA UTENSILI E RESETTARE IL CU'
N154 MSG(2)= 'cambiare utensile manualmente'
N155 MSG(3)= 'M6 programmata senza Txx'
N156 MSG(4)= 'Attesa apertura riparo'
N157 MSG(5)= 'Attesa chiusura riparo'
N158 [
N159 [
N160 [***** DEFINIZIONE SEQUENZE CAMBIO UTENSILE *****
N161 [... SEQUENZA CU PER CARICO UT DA TERRA CON MANDRINO VUOTO ...
N162 DEF SEQCU(1)=-6,-16,-34,COM,1,'CUMANU'
N163 [
N164 [... SEQUENZA CU PER SCARICO MANDRINO A TERRA (T0M6) ...
N165 DEF SEQCU(2)=-6,-10,-34,COM,1,'CUMANU'
N166 [
N167 [... SEQUENZA CU PER SCAMBIO TRA MANDRINO E TERRA ...
N168 DEF SEQCU(3)=-6,-10,-16,-34,COM,1,'CUMANU'
N169 [
N170 [... SEQUENZA CU PER POSA MANDRINO TERRA E CARICO DA MAG. ...
N171 DEF SEQCU(4)=-6,-10,-1,-4,-34,COM,1,'CUAUTO'
N172 [
N173 [... SEQUENZA CU PER POSA MANDR. IN MAGAZ. E PRESA DA TERRA ...
N174 DEF SEQCU(5)=-6,-23,-13,-16,-34,COM,1,'CUAUTO'
N175 [
N176 [... SEQUENZA CU PER SCAMBIO UTENS.CON MANDR. CARICO ...
N177 DEF SEQCU(6)=-6,-23,-13,-1,-4,-34,COM,1,'CUAUTO'
N178 [
N179 [... SEQUENZA CU PER CARICO CON MANDR.SCARICO ...
N180 DEF SEQCU(7)=-6,-1,-4,-34,COM,1,'CUAUTO'
N181 [
N182 [... SEQUENZA CU PER SCARICO UTENS.DA MANDR. IN MAGAZZ...
N183 DEF SEQCU(8)=-6,-23,-13,-34,COM,1,'CUAUTO'
N184 [
N185 [... SEQUENZA CU PER CARICO UTENS.= UTENS. NEL MANDR. ...
N186 DEF SEQCU(11)=-6,-34,COM,1,'CUMANU'
N187 [
N188 [
N189 PROG [SEZIONE VELOCE
N190 [abilitazione assi
N191 ABX=MOVCN(1)
N192 ABY=MOVCN(2)
N193 ABZ=MOVCN(3)

```



```

N194 RDMOV=MOVCN
N195 POFO=ANI(1) [potenziometro avanzamento assi
N196 END [SEZIONE LENTA
N197 [
N198 [-----PARTE SINCRONA-----
N199 [
N200 IF("BURDY)ASINC
N201 FHOLD=1; DHOLD=1
N202 [Nella decodifica ci vuole sempre prima T e poi M
N203 IF(STROT)CALL GEFUT
N204 IF(STROM)CALL GEFUM
N205 BURDY=0
N206 ASINC:$
N207 [-----PARTE ASINCRONA-----
N208 [*****
N209 [ GESTIONE MODULO CU AUTOMATICO *
N210 [*****
N211 CALL CUAUTO [routine gestione CU automatico
N212 [
N213 [..... attuazioni fisiche per cambio ut .....
N214 [le sicurezze di movimentazione vanno messe direttamente
N215 [sui comandi (uscite), per esempio:
N216 [out =((comando_auto) ~ (comando_manu)) & sicur_mecc.
N217 [
N218 UARIMA=MM62 [&... sicur.
N219 UCRIMA=MM63 [&... sicur.
N220 [...]
N221 L4=MM26 [cambio ut. manuale in corso
N222 [...]
N223 [
N224 [reset memorie per fine comando (comandi eseguiti)
N225 IF(IRIMAA&"IRIMAC) MM62=0 [sportello aperto
N226 IF(IRIMAC&"IRIMAA) MM63=0 [sportello chiuso
N227 IF(P4) MM26=0 [ok fine CU manuale
N228 [*****
N229 [ ALTRE GESTIONI ASINCRONE *
N230 [*****
N231 [
N232 [...]
N233 [...]
N234 [
N235 [*****
N236 [ ALLARMI, CONSENSI E SICUREZZE *
N237 [*****
N238 [relativi al CN
N239 DHOLD=EMACU~MM26~MM62~MM63 [ ~...
N240 FHOLD=DHOLD [ ~...
N241 REME=FF("IAUXON),(EMEA) [ ~... [Rich. di emergenza al CN
N242 [
N243 END
N244 [----- SEZIONE SUPERLENTA -----
N245 [..... visualizzazione messaggi .....
N246 IF(EMACU) DISPL,1,MSG(1); ELSE CLR,1 [CU in emergenza
N247 IF(MM26) DISPL,2,MSG(2); ELSE CLR,2 [cambiare ut. manu
N248 IF(ERRM06) DISPL,3,MSG(3); ELSE CLR,3 [M6 senza T pronta
N249 IF(MM62) DISPL,4,MSG(4); ELSE CLR,4 [attesa riparo
N250 IF(MM63) DISPL,5,MSG(5); ELSE CLR,5 [attesa riparo
N251 [
N252 END
N253 [
N254 [----- SEZIONE ROUTINES -----
N255 [
N256 [*****
N257 [ FUNZIONE T *
N258 [*****
N259 GEFUT:$
N260 [..... CHIAMATA DEL CAMBIO UTENSILE .....
N261 UTECU=TOOL [comunico l' utensile voluto al modulo CU
N262 NEWCU=1 [richiedo attivazione modulo CU
N263 RTS
N264 [

```

1. Esempi di programmazione

```

N265 [*****
N266 [      FUNZIONI M      *
N267 [*****
N268 GEFUM:$
N269 WNDINT(1)=AUXM
N270 IF(AUXM=6) M06
N271 IF(AUXM=30) CALL RESET; RTS
N272 IF("CUATT) RTS
N273 IF(AUXM=62) MM62=1; RTS
N274 IF(AUXM=63) MM63=1; RTS
N275 IF(AUXM=29) INTOF=1; RTS
N276 IF(AUXM=34) CUATT=0; RTS
N277 RTS
N278 [
N279 M06:$
N280 IF("CUATT) ERRM06=1; RTS      [M6 senza T
N281 M6PGM=1
N282 RTS
N283 [
N284 [*****
N285 [      GESTIONE CAMBIO UTENSILE AUTOMATICO *
N286 [*****
N287 [..... selezione modo CU .....
N288 CUAUTO:$
N289 IF(CUATT) NOSELE
N290 IF(P1) SELECU=0 [CU automatico (default)
N291 IF(P2) SELECU=1 [CU manuale (magazzino escluso)
N292 NOSELE:$
N293 [
N294 [lampade softkeys selezione modo
N295 CUAUT=(SELECU=0)
N296 CUMAN=(SELECU=1)
N297 [
N298 [*****
N299 [... interruzione sequenza, cancellazione emergenze ....
N300 [
N301 [Il CU viene interrotto solo se:
N302 [- si spengono gli ausiliari durante un CU
N303 [- viene dato un BREAK durante la sequenza CU
N304 [
N305 [L' interruzione viene fatta con REMCU in quanto
N306 [il CU risponde attivando EMACU
N307 REMCU=FF((BRKA&CUATT)~("IAUXON&CUATT)),(EMACU)
N308 [
N309 [La softkey P7 fa RBKCU per uscire da EMACU (emergenza)
N310 IF(P7&EMACU) RBKCU=1 [cancellazione emergenza CU
N311 [
N312 [In caso di interruzione e' necessario resettare il CU
N313 [con l' apposita softkey dopo avere VERIFICATO LA TABELLA UTE.
N314 L7=EMACU      [lampada CU in emergenza
N315 [
N316 IF(EMACU) CALL RESECU [resetto comunque i comandi PLC
N317 [
N318 [*****
N319 [Passaggio parametri alla COM
N320 P(10)=IFP(PPRECU)      [posizione di presa
N321 P(11)=IFP(PPOSCU)      [posizione di posa
N322 P(13)=IFP(NSEQCU)      [sequenza intrapresa
N323 [
N324 [..... decodifica codici fasi sequenza intrapresa .....
N325 IF ("BRDYCU) NOCU
N326 MAPRCU=0      [arresto avanzamento fasi
N327 CALL OPER      [gestione fasi cambio ut.
N328 BRDYCU=0      [fase CU acquisita
N329 NOCU:$
N330 [
N331 [*****
N332 [..... consenso avanzamento fasi .....
N333 MAPRCU=1      [&" ... &"...
N334 [
N335 RTS

```

```

N336 [****** RITORNO DA GESTIONE CUAUTO *****]
N337 [
N338 [******]
N339 [ROUTINE DECODIFICA OPERAZIONI CU e RESET *]
N340 [******]
N341 [caso di reset CU]
N342 RESECU:$
N343 MM26=0 [reset cambio utensile manuale]
N344 MM62=0
N345 MM63=0
N346 [
N347 [reset normale (M30 o break)]
N348 RESET:$
N349 WNDINT(1)=30 [visualizza M30]
N350 ERRM06=0 [cancello errore su M6 (M6 senza T pronta)]
N351 RTS
N352 [-----]
N353 [gestione OPERAZIONI cu]
N354 OPER:$
N355 [IF(OPERCU=...) OPCUX]
N356 [...]
N357 RTS
N358 [
N359 [OPCUX: ...; RTS]
N360 [
N361 [..... fine programma .....]

```

AXP2P - Movimentazione asse magazzino utensili da PLC

```

N1 [******]
N2 [*POSIZIONAMENTO MAGAZZINO UTENSILI COME ASSE INDIPENDENTE]
N3 [*]
N4 [*AXP2P 941008]
N5 [******]
N6 [
N7 [****** SEZIONE DICHIARATIVA *****]
N8 [Si considera un magazzino con 24 posizioni]
N9 [l' algoritmo posiziona con il percorso piu' breve.]
N10 [Si considera il caso di trasduttore non assoluto.]
N11 [In modo manuale il posizionamento termina sempre]
N12 [su una stazione.]
N13 [Il segnale INPOS significa ultima posizione raggiunta.]
N14 [
N15 INP
N16 IZERM [micro zero magazzino]
N17 IRIPM [micro riparo magazzino]
N18 [
N19 OUT
N20 UMOVEX [abilitazione asse X]
N21 UMOVEY [abilitazione asse Y]
N22 UMOVEZ [abilitazione asse Z]
N23 UABMAG [abilitazione magazzino]
N24 INPOS [asse posizionato sulla quota comandata]
N25 [
N26 RAM,16
N27 PORIT [posizione richiesta per il magazzino]
N28 [
N29 RAM,1
N30 RICUT [richiesta posizionamento magazzino ut.]
N31 [
N32 STIMER
N33 [temporizzatore tolleranza posizione magazzino]
N34 TIRIC,TURIC,TDRIC,TARIC,TCRIC
N35 [menu di softk gestiti da PLC]
N36 SOFTK,1
N37 P1,L1,1,' JOG + maggazz.'
N38 P2,L2,1,' JOG - maggazz.'

```

1. Esempi di programmazione

```

N39  [
N40  PROG
N41  END
N42  [***** SEZIONE LENTA *****]
N43  [ ..... decodifica delle funzioni ausiliarie .....]
N44  IF("BURDY)ASINC
N45  DHOLD=1; FHOLD=1
N46  IF(STROT) CALL GEFUT
N47  IF(STROM) CALL GEFUM
N48  BURDY=0
N49  ASINC:$
N50  [
N51  [----- PARTE ASINCRONA -----]
N52  UMOVEX=MOVCN(1)      [abilitazione azion. X]
N53  UMOVEY=MOVCN(2)      [abilitazione azion. Y]
N54  UMOVEZ=MOVCN(3)      [abilitazione azion. Z]
N55  RDMOV=MOVCN          [assi abilitati per il CN]
N56  [
N57  [ ..... posizionamento magazzino .....]
N58  IF (NCMD<>5) NOJOG
N59  IF (P1) PORIT=FPI(NEI(POAP2P)+1); RICUT=1; L1=1
N60  IF (P2) PORIT=FPI(NEI(POAP2P)-1); RICUT=1; L2=1
N61  NOJOG:$
N62  IF("RICUT) L1=0; L2=0
N63  CALL POSMAG
N64  [
N65  [.....generali.....]
N66  FHOLD=RICUT
N67  DHOLD=RICUT          [arresto avanzamento blocchi]
N68  REME=FF(EMAP2P(1)),(EMEA) [emergenza MU su emerg. asse]
N69  [
N70  IF(BRKA) CALL RESET      [reset funzioni PLC da CN]
N71  [
N72  END
N73  [..... sezione superlenta .....]
N74  WINDOW=NEI(POAP2P(1)) [visualizzazione posto corrente]
N75  ASCW=109
N76  END
N77  [
N78  [***** SEZIONE ROUTINES *****]
N79  [*****]
N80  [ POSIZIONAMENTO MAGAZZINO: ASSE INDIPENDENTE *]
N81  [*****]
N82  POSMAG:$
N83  POTP2P(1)=1          [potenziometro velocita']
N84  SSAP2P(1)=1          [magazzino sempre abilitato]
N85  MIZP2P(1)=IZERM      [micro zero magazzino]
N86  UABMAG=MOVP2P(1)     [abilitazione asse magazz.]
N87  RDMP2P(1)=MOVP2P(1) [risposta asse abilitato]
N88  INPOS=SGLP2P(1)&MZAP2P(1)&"RUNP2P(1)&"RICUT&"EMAP2P(1) [in posiz]
N89  [
N90  [anomalie e reset ...]
N91  IF(EMAP2P(1)) RICUT=0; RTS      [anomalie resetto comando]
N92  [su EMAP2P si attiva REME]
N93  [
N94  [se asse azzerato ...]
N95  IF("MZAP2P(1)) ZEMAG          [test asse azzerato]
N96  JGPP2P(1)=0                  [cancello JOG]
N97  MCZP2P(1)=0                  [cancello modo ricerca zero]
N98  [
N99  [calcolo della posizione con algoritmo percorso piu' breve]
N100 PFNP2P(1)=IFP(PORIT)-NEI((IFP(PORIT)-NEI(POAP2P(1)))/24)*24
N101 IF(RICUT) RUNP2P(1)=1          [attivo posizionamento]
N102 TIRIC(5)=RUNP2P(1)-TDRIC      [segnale di sincronismo x INPOS]
N103 [notare che qui entro solo se MZAP2P e' presente]
N104 IF (SGLP2P(1)&"TDRIC) RICUT=0 [movimento terminato]
N105 RTS
N106 [
N107 [asse da azzerare ...]
N108 ZEMAG: $
N109 JGPP2P(1)=RICUT[per ricerca zero forzo JOG+

```

```

N110 MCZP2P(1)=RICUT      [selezione modo ricerca
N111 INPOS=0              [tolgo immediatamente eventuale INPOS
N112 RTS
N113 [
N114 [ ..... decodifica funzioni M & T .....
N115 GEFUT:$
N116 PORIT = TOOL        [preparo posizione da ricercare
N117 RTS
N118 [
N119 GEFUM:$
N120 WNDINT(1)=AUXM[visualizzazione M
N121 IF (AUXM=6) RICUT=1; RTS [posiziona magazzino su ultima T
N122 RTS
N123 [
N124 [..... routine di reset .....
N125 RESET:$
N126 IF(EMAP2P(1)) RBKP2P(1)=1; RICUT=0      [recupero emergenza P2P
N127 WNDINT(1)=30                          [visualizzazione M30
N128 RTS
N129 [..... fine programma .....

```

COMMUCM - Commutazione mandrino con asse C

```

N1  [*****]
N2  [*]
N3  [*      COMMUTAZIONE ASSE C e MANDRINO      ]
N4  [*]
N5  [*      COMMUCM      940516      ]
N6  [*]
N7  [*****]
N8  [Nei dati di configurazione l' asse C e' il 4.
N9  [La commutazione con il mandrino (1) viene tramite
N10 [M21 da mandrino ad asse C
N11 [M20 da asse C a mandrino
N12 [E' importante usare M di fine blocco in modo
N13 [che la commutazione da asse C a mandrino
N14 [non possa avvenire con l' asse in moto.
N15 [L' asse C e mandrino hanno stessi canali di I/O
N16 [il trasduttore e' un encoder
N17 [nell' esempio non e' previsto il micro di zero asse C
N18 [
N19 [dichiarazione INPUT fisici
N20 INP
N21 [
N22 [dichiarazione OUTPUT fisici
N23 OUT
N24 ABILX      [ 1 abilitazione asse X
N25 ABILY      [ 2 abilitazione asse Y
N26 ABILM      [ 3 abilitazione mandrino o asse C
N27 ABILZ      [ 4 abilitazione asse Z
N28 [
N29 [dichiaraz. var. BIT retentive alla accensione
N30 SRAM,1
N31 CICM20      [Commutazione da asse C a mandrino
N32 CICM21      [Commutazione da mandrino a asse C
N33 ASSEC      [setta modo di lavoro asse C
N34 ASSEM      [setta modo di lavoro mandrino
N35 [
N36 [dichiarazione variabili BIT non retentive
N37 RAM,1
N38 ABMAN      [abilitazione mandrino
N39 ABC        [abilitazione asse C
N40 [
N41 STR
N42 MSG1      [messaggi
N43 MSG2      [messaggi
N44 [

```

1. Esempi di programmazione

```

N45 [***** INIZIALIZZAZIONE *****]
N46 INIT
N47 MSG1='Commutazione in corso da asse C a mandrino'
N48 MSG2='Commutazione in corso da mandrino a asse C'
N49 [
N50 [***** INIZIALIZZO MODO MANDRINO *****]
N51 [
N52 IF ("ASSEC&"ASSEM) CALL RESCM [se nessun modo
N53 IF (CICM20~CICM21) CALL RESCM [se interruzione
N54 [
N55 SPGAM(1)=1 [gamma 1 per il mandrino
N56 [
N57 [***** LOGICA VELOCE (ogni 10 mS) *****]
N58 PROG
N59 ABILX = RDMOV(1)
N60 ABILY = RDMOV(2)
N61 ABILZ = RDMOV(3)
N62 RDMOV = MOVCN [Move di risposta al C.N.
N63 [
N64 [***** gestione potenziometri *****]
N65 POFO=ANI(1)
N66 POMO(1)=ANI(2); POMO(2)=ANI(2); POMO(3)=ANI(2)
N67 END
N68 [**** decodifica funzioni ausiliarie dal CN *****]
N69 IF ("BURDY) ASINC
N70 DHOLD=1; FHOLD=1
N71 IF (STROM) CALL GEFUM
N72 BURDY=0
N73 ASINC: $
N74 [
N75 IF(BRKA) CALL LM05 [fermo mandrino su break
N76 [
N77 [***** GESTIONE ASSE C *****]
N78 [gestione del reset sequenze (interruzione)
N79 IF(BRKA&(CICM20~CICM21)) CALL RESCM
N80 [
N81 [gestione potenziometro
N82 IF(CICM21) POMO(4)=.1; ELSE POMO(4)=ANI(2)
N83 [
N84 [..... commutazione da asse C a mandrino .....]
N85 [sequenza: - DISRQ(4)=1
N86 [ - SPDRQ(1)=0 e SPDIS(1)=0
N87 [ - ASSEC=0; ASSEM=1
N88 IF("CICM20) NOCM
N89 IF("SPDRQ(1)) ASSEC=0; ASSEM=1; CICM20=0; NOCM
N90 IF(DISRQ(4)) SPDRQ(1)=0; SPDIS(1)=0; NOCM
N91 DISRQ(4)=1; SSA(4)=0
N92 NOCM: $
N93 [
N94 [..... commutazione da mandrino a asse C .....]
N95 [sequenza: - attesa "SPMOT(1)
N96 [ - SPDRQ(1)=1; SPDIS(1)=1
N97 [ - DISRQ(4)=0
N98 [ - FOMAN(4)=1; MARK(4)=1; JOGP(4)=1
N99 [ - attesa MIZEA(4)
N100 [ - JOGP(4)=0; MARK(4)=0; FOMAN(4)=0
N101 [ - attesa "JOGIN(4)
N102 [ - SSA(4) = 1 (se necessario)
N103 [ - ASSEC=1; ASSEM=0
N104 IF("CICM21) NOMC
N105 IF(SSA(4)&MIZEA(4)) ASSEC=1; ASSEM=0; CICM21=0;NOMC [end ciclo
N106 IF(MIZEA(4)&"JOGIN(4)) SSA(4)=1; NOMC [SSA
N107 IF(MIZEA(4)) FOMAN(4)=0;MARK(4)=0;JOGP(4)=0; NOMC [zero fatto
N108 [fai lo zero
N109 IF("MIZEA(4)&"DISRQ(4)) FOMAN(4)=1; MARK(4)=1; JOGP(4)=1; NOMC
N110 IF(SPDRQ(1)) DISRQ(4)=0; NOMC
N111 IF("SPMOT(1)) SPDRQ(1)=1; SPDIS(1)=1
N112 NOMC: $
N113 [
N114 [..... gestione mandrino .....]
N115 [

```

```

N116 [velocita' e potenziometro
N117 SPSSO(1)=ANI(3)
N118 SPVEL(1)=SPEED
N119 [
N120 ABMAN=SPMOV(1) [memoria abil. MANDRINO
N121 ABC=MOVCN(4) [memoria abil. asse C
N122 ABILM=ABMAN~ABC [
N123 [
N124 [***** GESTIONE DEI CONSENSI VERSO IL CN *****
N125 DHOLD = CICM20~CICM21
N126 FHOLD = DHOLD
N127 [
N128 END
N129 GIRMI=INT(ABS(SPTCH)) [visual. speed effettiva
N130 WINDOW=PASP [visual. posizione mandrino
N131 ASCW=109
N132 IF(ASSEC) DISPL,0,'ASSE C ATTIVO'; ELSE CLR,0
N133 END
N134 [
N135 GEFUM:$
N136 IF ((AUXM = 3)&ASSEM) SPROT(1)=1; SPD(1)=0; RTS
N137 IF ((AUXM = 4)&ASSEM) SPROT(1)=1; SPD(1)=1; RTS
N138 IF (AUXM = 5) LM05
N139 IF (AUXM = 20) LM20
N140 IF (AUXM = 21) LM21
N141 RTS [Programmata funzione NoN gestita o non attuabile
N142 [
N143 LM05: SPROT(1)=0; RTS
N144 LM20: IF(ASSEC) CICM20=1; RTS; ELSE RTS [da C a M
N145 LM21: IF(ASSEM) CALL LM05; CICM21=1; RTS; ELSE RTS [da M a C
N146 [
N147 [Ripristino in stato MANDRINO su interruzione
N148 RESCM: $
N149 JOGP(4)=0; MARK(4)=0; FOMAN(4)=0; DISRQ(4)=1
N150 SPDRQ(1)=0; SPD(1)=0
N151 CICM20=0; CICM21=0
N152 ASSEC=0; ASSEM=1
N153 RTS
N154 [..... fine programma .....

```

NEWFILT - Filtro numerico

```

N1 [*****]
N2 [* FILTRO NUMERICO (ESEMPIO SU INGRESSO ANALOGICO) ]
N3 [*          940930          NEWFILT ]
N4 [*****]
N5 INP
N6 OUT
N7 [
N8 RAM,32
N9 SOMMA [accumulo delle ultime letture
N10 ELE(30) [tabella delle ultime letture
N11 MEDIA [risultato filtrato
N12 [
N13 RAM,8
N14 MAXELE [numero massimo di letture
N15 IELE [indice lemento corrente
N16 [
N17 INIT
N18 MAXELE=30 [numero di letture per media
N19 [
N20 PROG
N21 IELE=IELE+1 [elemento corrente
N22 IF(IELE>MAXELE) IELE=1 [controllo su massimo numero
N23 SOMMA=SOMMA-ELE(IELE) [elimino da accumulo elemento vecchio
N24 ELE(IELE)=ANI(1) [leggo nuovo elemento
N25 SOMMA=SOMMA+ELE(IELE) [metto elemento nuovo al suo posto

```

1. Esempi di programmazione

```

N26  MEDIA=SOMMA/IFP(MAXELE)      [divido accumulo per numero medie
N27  END
N28  [..... fine programma .....

```

TABUTE1 - Riordino dei posti utensile nella tabella

```

N1  [*****]
N2  [          RIORDINO TABELLA UTENSILI          ]
N3  [          TABUTE1          940908          ]
N4  [*****]
N5  [
N6  RAM,16
N7  IND          [Indice elemento corrente
N8  [
N9  RAM,1
N10 MM1234          [Ciclo di reset in corso
N11 [
N12 PROG
N13 END
N14 [
N15 IF("BURDY) ASINC
N16 DHOLD=1; FHOLD=1
N17 IF(STROM&(AUXM=1234)) CALL GEFUM
N18 BURDY=0
N19 ASINC:$
N20 [
N21 DHOLD=MM1234
N22 FHOLD=DHOLD
N23 [
N24 [..... RESET TABELLA UTENSILI .....
N25 [Il ciclo riordina i posti utensili
N26 [da 1 al numero di posti magazzino.
N27 IF("MM1234) SKIP          [Non e' attivo il ciclo M1234
N28 IF(UTEFRE<=0) SKIP          [Non c' e' più disponibilità scritt.
N29 EXEC=UTEFRE          [Scrivi per il numero di scritt. poss.
N30 IF(IND>MAGNPO) MM1234=0;NOWRI [Ciclo terminato
N31 UTPOS(IND)=IND          [Carica posto
N32 IND=IND+1          [Incrementa indice posto
N33 NOWRI:$
N34 ENDE
N35 SKIP:$
N36 [
N37 END
N38 END
N39 [
N40 GEFUM:$
N41 MM1234=1          [Attiva ciclo
N42 IND=1          [Inizializza indice
N43 RTS

```


TESTAR - Testa indexata mossa con motore mandrino

```

N1      [*****]
N2      [ ]
N3      [      ESEMPIO COMMUTAZIONE MANDRINO CON TESTA INDEXATA ]
N4      [ ]
N5      [      TESTAR 941010 ]
N6      [*****]
N7      [Questo esempio mostra la tecnica di commutazione per gestire
N8      [mandrino e testa con lo stesso motore e stesso trasduttore.
N9      [Nei parametri di configurazione si definiranno gli stessi
N10     [canali e che il programma plc deve alternativamente
N11     [abilitare.
N12     [Si è preferito sfruttare due sequenze comandate da plc, con
N13     [le funzioni M20 e M21, per facilitare l'inserimento di
N14     [altri comandi necessari alla meccanica e l'estensione a due
N15     [assi.
N16     [L'asse testa utilizza il trasduttore mandrino in modo
N17     [incrementale quindi per l'asse encoder deve essere
N18     [specificata la modalità di azzeramento condizionata dalla
N19     [presenza micro di zero per evitare l'azzeramento al
N20     [passaggio sul marker.
N21     [All'accensione viene fatta una inizializz. divisa in due fasi:
N22     [1-aggiornamento posizione testa
N23     [2-commutazione in stato mandrino
N24     [
N25     [programma PROM21 che commuta mandrino su asse testa
N26     [
N27     [M5                [arresta mandrino (orienta se richiesto)
N28     [M101             [disabilita lettura e controllo mandrino
N29     [M102             [abilita lettura asse testa
N30     [M103             [aggiorna posizione corrente testa
N31     [M104             [abilita controllo asse testa
N32     [
N33     [programma PROM20 che commuta asse testa su mandrino
N34     [
N35     [M112             [disabilita lettura e controllo testa
N36     [M113             [abilita lettura e controllo mandrino
N37     [*****]
N38     [
N39     INP
N40     OUT
N41     TERM,3
N42     ABM                [abilitazione azionamento mandrino
N43     [
N44     SRAM,32
N45     MEMTA              [memoria testa A
N46     RAM,1
N47     ROTMA              [comando di rotazione
N48     [
N49     PULSE
N50     PFASE2             [pulse 2a fase inizializzazione
N51     INIT
N52     SPGAM(1)=1         [gamma 1 (unica)
N53     [
N54     CALL INTSTA        [inizializza testa
N55     [
N56     PROG
N57     END
N58     IF(PFASE2) CALL FASE2
N59     IF("BURDY) ASINC
N60     IF(STROM) CALL GEFUM
N61     BURDY=0
N62     ASINC:$
N63     [
N64     [***** gestione asse testa A *****]
N65     [
N66     RDMOV(4)=MOVCN(4)
N67     IF(MOVCN(4)&RDMOV(4)) MEMTA=POO(4)
N68     POFO=ANI(1)

```

1. Esempi di programmazione

```

N69  [***** gestione mandrino *****]
N70  SPVEL(1)=SPEED
N71  SPSSO(1)=0.7 + ANI(3)*0.6
N72  SPROT(1)=ROTMA&"HOLDA           [comando di rotazione e HOLD
N73  ABM=SPMOV(1)-RDMOV(4)]&...      [abilitazione + consensi
N74  [
N75  END
N76  [ ..... sezione super lenta .....
N77  GIRMI=INT(ABS(SPTCH))           [visualizzazione S
N78  END
N79  [
N80  [— ROUTINES —————]
N81  GEFUM: $
N82  WNDINT(1)=AUXM                 [visualizzazione M
N83  IF(AUXM=3) M03
N84  IF(AUXM=4) M04
N85  IF(AUXM=5) M05
N86  IF(AUXM=20) M20
N87  IF(AUXM=21) M21
N88  IF(AUXM=101) M101
N89  IF(AUXM=102) M102
N90  IF(AUXM=103) M103
N91  IF(AUXM=104) M104
N92  IF(AUXM=112) M112
N93  IF(AUXM=113) M113
N94  RTS
N95  M03: SPDIR(1)=0; ROTMA=1; RTS
N96  M04: SPDIR(1)=1; ROTMA=1; RTS
N97  M05: ROTMA=0; RTS
N98  M20: COM,1,'PROM20';RTS
N99  M21: COM,1,'PROM21';RTS
N100 M101: SPDRQ(1)=1;SPDIS(1)=1;RTS      [disabilita lettura
N101 [                                     e controllo mandrino
N102 M102: DISRQ(4)=0;RTS                 [abilita lettura asse testa
N103 [
N104 M103: SHIFT(4)=SHIFT(4)+POO(4)-MEMTA;RTS [agg. testa
N105 [
N106 M104: DSERV(4)=0;RTS                 [abilita controllo asse testa
N107 [
N108 M112: DISRQ(4)=1;DSERV(4)=1;RTS      [disabilita lettura
N109 [                                     e controllo testa
N110 M113: SPDRQ(1)=0;SPDIS(1)=0;RTS      [abilita lettura e
N111 [                                     controllo mandrino
N112 [
N113 INTSTA: SPDRQ(1)=1                   [fase 1 inizializz. testa
N114 SPDIS(1)=1
N115 DSERV(4)=1
N116 DISRQ(4)=0
N117 SHIFT(4)=SHIFT(4)+POO(4)-MEMTA
N118 PFASE2=1                             [setta pulse 2a fase di inizializz.
N119 RTS
N120 [
N121 FASE2: ROTMA=0                       [fase 2 inizializz. testa
N122 SPDIS(1)=0
N123 SPDRQ(1)=0
N124 DISRQ(4)=1
N125 DSERV(4)=1
N126 RTS
N127 [..... fine programma .....

```

APPENDICI

APPENDICE A - TABELLA DEI CODICI ASCII

DEC	HEX	CARATTERE	DEC	HEX	CARATTERE	DEC	HEX	CARATTERE	DEC	HEX	CARATTERE
000	00	(NULL)	016	10	► (DLE)	032	20	BLANK (SPAZIO)	048	30	0
001	01	☺ (SOH)	017	11	◄ (DC1)	033	21	!	049	31	1
002	02	☻ (STX)	018	12	↕ (DC2)	034	22	"	050	32	2
003	03	♥ (ETX)	019	13	!! (DC3)	035	23	#	051	33	3
004	04	♦ (EOT)	020	14	¶ (DC4)	036	24	\$	052	34	4
005	05	♣ (ENQ)	021	15	§ (NACK)	037	25	%	053	35	5
006	06	♠ (ACK)	022	16	■ (SYN)	038	26	&	054	36	6
007	07	● (BEL)	023	17	↕ (ETB)	039	27	'	055	37	7
008	08	◼ (BS)	024	18	↑ (CAN)	040	28	(056	38	8
009	09	◯ (HT)	025	19	↓ (EM)	041	29)	057	39	9
010	0A	◐ (LF)	026	1A	→ (SUB)	042	2A	*	058	3A	:
011	0B	◯ [↑] (VT)	027	1B	← (ESC)	043	2B	+	059	3B	;
012	0C	◯ [♀] (FF)	028	1C	└ (FS)	044	2C	,	060	3C	<
013	0D	♪ (CR)	029	1D	↔ (GS)	045	2D	-	061	3D	=
014	0E	🎵 (SO)	030	1E	▲ (RS)	046	2E	.	062	3E	>
015	0F	☼ (SI)	031	1F	▼ (US)	047	2F	/	063	3F	?

Appendice A - Tabella dei codici ASCII

DEC	HEX	CARATTERE	DEC	HEX	CARATTERE	DEC	HEX	CARATTERE	DEC	HEX	CARATTERE
064	40	@	080	50	P	096	60	`	112	70	p
065	41	A	081	51	Q	097	61	a	113	71	q
066	42	B	082	52	R	098	62	b	114	72	r
067	43	C	083	53	S	099	63	c	115	73	s
068	44	D	084	54	T	100	64	d	116	74	t
069	45	E	085	55	U	101	65	e	117	75	u
070	46	F	086	56	V	102	66	f	118	76	v
071	47	G	087	57	W	103	67	g	119	77	w
072	48	H	088	58	X	104	68	h	120	78	x
073	49	I	089	59	Y	105	69	i	121	79	y
074	4A	J	090	5A	Z	106	6A	j	122	7A	z
075	4B	K	091	5B	[107	6B	k	123	7B	{
076	4C	L	092	5C	\	108	6C	l	124	7C	
077	4D	M	093	5D]	109	6D	m	125	7D	}
078	4E	N	094	5E	^	110	6E	n	126	7E	~
079	4F	O	095	5F	_	111	6F	o	127	7F	△

DEC	HEX	CARATTERE	DEC	HEX	CARATTERE	DEC	HEX	CARATTERE	DEC	HEX	CARATTERE
128	80	Ç	144	90	Ê	160	A0	á	176	B0	⌘
129	81	ü	145	91	æ	161	A1	í	177	B1	⌘
130	82	é	146	92	Æ	162	A2	ó	178	B2	⌘
131	83	â	147	93	ô	163	A3	ú	179	B3	
132	84	ä	148	94	ö	164	A4	ñ	180	B4	†
133	85	à	149	95	ò	165	A5	Ñ	181	B5	‡
134	86	â	150	96	û	166	A6	ä	182	B6	‡
135	87	ç	151	97	ù	167	A7	ö	183	B7	π
136	88	ê	152	98	ÿ	168	A8	¿	184	B8	¶
137	89	ë	153	99	Ö	169	A9	¬	185	B9	‡
138	8A	è	154	9A	Ü	170	AA	¬	186	BA	
139	8B	ï	155	9B	ç	171	AB	½	187	BB	¶
140	8C	î	156	9C	£	172	AC	¼	188	BC	¶
141	8D	ì	157	9D	¥	173	AD	¡	189	BD	¶
142	8E	Ä	158	9E	₤	174	AE	«	190	BE	¶
143	8F	Å	159	9F	f	175	AF	»	191	BF	¶

DEC	HEX	CARATTERE	DEC	HEX	CARATTERE	DEC	HEX	CARATTERE	DEC	HEX	CARATTERE
192	C0	Ł	208	D0	Ƨ	224	E0	α	240	F0	≡
193	C1	⌞	209	D1	⌚	225	E1	β	241	F1	±
194	C2	⌟	210	D2	⌘	226	E2	Γ	242	F2	≥
195	C3	⌠	211	D3	Ƨ	227	E3	π	243	F3	≤
196	C4	—	212	D4	Ƨ	228	E4	Σ	244	F4	∫
197	C5	†	213	D5	Ƨ	229	E5	σ	245	F5	∫
198	C6	Ƨ	214	D6	⌘	230	E6	μ	246	F6	÷
199	C7	⌠	215	D7	⌠	231	E7	τ	247	F7	≈
200	C8	Ƨ	216	D8	Ƨ	232	E8	φ	248	F8	°
201	C9	Ƨ	217	D9	Ƨ	233	E9	θ	249	F9	•
202	CA	Ƨ	218	DA	Ƨ	234	EA	Ω	250	FA	•
203	CB	⌠	219	DB	■	235	EB	δ	251	FB	√
204	CC	⌠	220	DC	■	236	EC	∞	252	FC	∞
205	CD	=	221	DD	■	237	ED	∅	253	FD	²
206	CE	⌠	222	DE	■	238	EE	€	254	FE	■
207	CF	⌞	223	DF	■	239	EF	∩	255	FF	BLANK "FF"

APPENDICE B - TABELLA FUNZIONI AUSILIARIE

La tabella che segue contiene le principali funzioni ausiliarie definite nello standard ISO RS-274 D, la cui emissione verso il PLC è riferita al blocco programmato.

CODICE	EMESSA AD INIZIO BLOCCO	EMESSA A FINE BLOCCO	TRATTATA DA CN	DESCRIZIONE FUNZIONE
M00 - M01 - M02		X	X	Arresto programma
M03	X			Mandrino orario
M04	X			Mandrino antiorario
M05		X		Arresto mandrino
M06		X		Cambio utensile
M07 - M08	X			Refrigerante ON
M09		X		Refrigerante OFF
M10	X			Bloccaggio assi
M11	X			Sbloccaggio assi
M12		X		Sincronizzazione
M13	X			Mandrino orario e refrigerante
M14	X			Mandrino antiorario e refrigerante
M15 - M18	X			Non assegnate
M19		X		Orientamento mandrino
M20 - M29		X		Non assegnate
M30		X	X	Fine programma
M31 - M39	X			Non assegnate
M40 - M44	X			Cambio gamma
M45	X			Ripristino abbandono assi
M46	X			Disabilita abbandono assi
M47	X			Non assegnata
M48	X			Inibisce override rapidi
M49	X			Applica override ai rapidi (default)
M50 - M8999	X			Non assegnate
M9000 - M9999		X		Non assegnate
H0 - H9999	X			Non assegnate
T0 - T9999	X		X	Utensili o correttori lunghez.
SO - S99999	X		X	Velocità di rotazione mandrino

APPENDICE C – MESSAGGI DIAGNOSTICI

E18:	numero di utensile diverso da T mandrino
E19:	valore di correzione troppo grande (> 2 mm)
E20:	numero di origine o di utensile non previsti
E21:	manca incremento (funzione I)
E22:	cambio piano seguito da funzioni incompatibili
E23:	correzioni parassiali applicate a quote polari
E24:	funzione O incompatibile con cambio uten. tipo S1200
E25:	G duplicata
E26:	quota duplicata
E27:	L duplicata
E28:	P duplicata
E29:	R duplicata
E30:	S duplicata
E31:	F duplicata
E32:	M duplicata
E33:	prestazione non presente
E34:	manca la dist. minima dal centro per G202
E35:	manca l'ascissa nella definizione della macro
E36:	manca l'ordinata nella definizione della macro
E37:	manca numero di loops nella definiz. dei supercicli
E38:	manca distanza nella definiz. dei supercicli
E39:	manca raggio cerchio nella definiz. dei supercicli
E40:	funzione di salto non ammessa in esec. da periferica
E41:	richiamo di funzione L (Lxx) non presente o duplicata
E42:	richiamo di sequenza memorizzata (*) non definita
E43:	funzione L non ammessa in blocco singolo
E44:	richiamo di funzione L in un file troppo grande
E45:	memoria esaurita in compilazione o digitalizzazione
E46:	funzioni non ammesse tra G754 e G753 (invers. prof.)
E47:	punti coincid. o fuori piano lavoro nella cava
E48:	manca di funzioni apertura/chiusura
E49:	richiamo di origine o correttore non valorizzati
E50:	funzione G32 all'interno di un ciclo ripetitivo
E51:	livello di annidamento di sottoprogrammi maggiore di 8
E52:	livello di annidamento di cicli ripetit. maggiore di 8
E53:	punti coincid. nella definiz. di curva per punti G27
E54:	suddivisione errata profili verticali

E55:	il profilo non e' chiuso
E59:	errore di parita' o di linea
E60:	errore di lettura programma
E62:	richiamo di un programma non esistente in memoria
E63:	ciclo fisso non eseguibile con i parametri dati: S,F,J,Z
E64:	ciclo fisso programmato senza funz. M di rotaz. mandr.
E65:	tastatore non qualificato
E66:	ricerca memorizzata di un blocco inesistente
E67:	manca la quota di inizio foro (J) in def. ciclo fisso
E68:	ciclo G88 seguito da coord. diverse dall'asse mandr.
E69:	cava con troppe passate (> 65535)
E70:	errore in una definizione geometrica
E71:	nel controllo collisioni dell'utensile col profilo
E72:	troppi punti o enti
E73:	cava poligonale con meno di tre punti
E74:	cave programmate senza definizione del raggio utensile
E75:	le rette sono parallele, manca intersezione
E76:	in sgrossatura tra profilo piano e profili sezione
E77:	manca intersezione tra la retta e il cerchio
E78:	cave profilate con passate parallele al profilo
E79:	gestione delle isole delle cave profilate
E80:	lunghezza ente troppo grande (> 131071 mm)
E81:	cerchi concentrici
E82:	cerchi esterni
E83:	cerchi coincidenti
E84:	cerchi tangenti
E85:	cerchi interni
E86:	errore nella definizione di enti geometrici
E88:	divisione per zero
E89:	radice quadrata di un numero negativo
E90:	operazioni tra parametri P con risultato troppo grande
E91:	errore nella definizione dei parametri programma
E92:	assi fuori posizione
E93:	asse in fine corsa
E94:	quota negativa non ammessa
E95:	inver. del senso di percorrenza di un ente del profilo
E96:	valore errato o mancanza segmento in fly
E97:	blocco memorizzato da perifer. con errore di sintassi
E98:	uscita dai limiti del campo operativo
E99:	errore di sintassi nel blocco
E200:	uscita dai limiti di copiatura
E201:	urto del tastatore in copiatura
E202:	perdita del contatto del tastatore in copiatura
E206:	guasto hardware al tastatore digitale
E207:	tastatore digitale sconnesso
E208:	oltrecorsa hardware al tastatore digitale
E209:	deflessione del tastatore di copiatura sui limiti max
E210:	urto del tastatore on/off di misura
E211:	inizio ciclo di misura con probe deflesso
E212:	copia nel semispazio non consentito
E213:	deflessione azzeratore utensile sui limiti max
E253:	errore di scrittura su file di digitalizzazione
E254:	manca funzione di apertura limiti G877 in copiatura
E255:	manca funzione di chiusura limiti G879 in copiatura

E300:	richiesta di aggancio tra assi non azzerati
E301:	richiesta di aggancio tra assi già agganciati
E400:	funzioni non previste dalla macro
E401:	blocco macro in ordine sbagliato
E402:	memoria interna insufficiente per eseguire la macro
E403:	mancano parametri obbligatori
E404:	parametri errati nella chiamata a macro
E405:	profilo errato richiamato nella macro
E406:	angoli utensile non compatibili col profilo
E407:	troppe passate di filettatura
E408:	numero di passate di filettatura insuff. (min. di 4)
E409:	filettatura di un cerchio
E410:	profilo non monotono sull'asse di avanzamento
E411:	profondità di passata nulla o negativa
E412:	il sovrametallo provoca interferenza tra le passate
E413:	diam. max del profilo finito sup. a quello del pezzo
E414:	elementi del profilo non connessi
E415:	elementi del profilo che si intersecano
E416:	profilo della gola errato
E417:	larghezza della gola inferiore a quella dell'utensile
E418:	numero passate di filettatura nullo o negativo
E419:	mancano orientamento e angoli utensile
E420:	incompatibilità attacco profilo/direz. lavorazione
E421:	incompatibilità tra profilo e parametri definiti
E422:	manca memoria per memorizzare zone in ombra
E423:	numero di enti superiore al consentito
E424:	insufficiente lunghezza del profilo
E425:	i profili giacciono sullo stesso piano
E426:	profilo della zona limite concavo
E427:	isola esterna al profilo
E428:	la macro non trova enti nel profilo
E429:	definiz. di profilo finito solo con enti orizzontali
E430:	diametro min. del profilo super. a quello del grezzo
E431:	smussi e raccordi definiti contemporaneamente
E432:	inclinazione scorretta primo o ultimo ente della gola
E433:	sottosquadro nel profilo della gola
E434:	cerchio di raggio zero nel profilo della gola
E435:	lunghezza di uscita superiore a lunghezza del filetto
E437:	raggio utensile senza orientamento
E438:	orient. utensile incompatibile con la direz. lavoro
E439:	controllo zone in ombra con orientamento errato
E440:	controllo zone ombra con angoli utensili errati
E441:	raggio utensile diverso dai valori standard
E442:	orientamento utensile errato
E443:	manca larghezza utensile
E444:	profondità massima dell'utensile nulla o negativa
E445:	larghezza e raggio utensile incompatibili
E446:	punti estremi dei profili finito e grezzo non coinc.
E500:	presente sia in pinza che in magazzino l'utensile..
E501:	presente sia in s.i. che in magazzino l'utensile..
E502:	presente sia in mandr. che in magazzino l'utensile..
E503:	posto utensile già occupato per utensile..
E504:	posti anteriori insuff. per taglia utensile..
E505:	posti posteriori insuff. per taglia utensile..

E506:	dimensione non coerente per planare, utensile..
E507:	assente in tabella l'utensile richiesto..
E508:	utensile non abilitato..
E509:	assente in tabella l'utensile da posare..
E510:	assente in magazzino l'utensile da scaricare
E511:	gia' presente in magazzino l'utensile da caricare
E512:	manca posto in magaz. per carico utensile dal mandr.
E513:	manca posto in magaz. per carico utensile da T prog.
E514:	manca posto in magaz. per carico da staz. intermedia
E515:	manca posto in magazzino per carico utensile da pinza
E516:	ciclo cambio utens. interrotto causa spegnimento M.U.
E518:	tabella utensile con dato incoerente..
E519:	posizione errata associata ad utensile..
E520:	carico manuale di utensile anche presente in magazzino
E521:	richiesto carico/scarico utensile T0
E522:	non permesso carico random-fisso: num.utensile..
E951:	errore in DDI Procedure Command
E990:	errore di sintassi nel file CAMME alla riga..
E991:	numero tabella errato nel file CAMME alla riga..
E992:	troppi valori nel file CAMME alla riga..
E993:	numero valori insuff. nel file CAMME alla riga..
E1001:	anomalie codice Gray su trasdutt. assoluto asse..
E1002:	segnale troppo alto trasdutt. analog. dell'asse..
E1003:	segnale troppo basso trasdutt. analog. dell'asse..
E1004:	discontinuità di lettura quota dell'asse..
E1005:	errore servomeccanismo dell'asse..
E1006:	numero impulsi errati trasdutt. increment. asse..
E1007:	anomalie al trasduttore dell'asse..
E1008:	posizionamento fuori tolleranza dell'asse..
E1009:	mancanza contatto tra testa di foratura e piastra
E1010:	errore piano destinazione di foratura
E1011:	coordinate di foratura esterne all'area di lavoro
E1012:	combinazione non ammessa di comandi sz
E1013:	valore di quik superiore alla quota di sicurezza programmata
E1032:	segnale troppo alto trasdutt. analogico mandrino
E1033:	segnale troppo basso trasdutt. analogico mandrino
E1034:	discontinuità di lettura quota dell'asse mandrino
E1036:	numero impulsi errati trasdutt. increment. mandrino
E1037:	anomalie al trasduttore del mandrino
E1061:	anomalie Gray trasd. assoluto asse punto-punto..
E1062:	segnale troppo alto trasdutt. asse punto-punto..
E1063:	segnale troppo basso trasdutt. asse punto-punto..
E1064:	discontinuità di lettura quota asse punto-punto
E1065:	errore servomeccanismo dell'asse punto-punto..
E1066:	num. impulsi errati da trasdutt. asse punto-punto..
E1067:	anomalie al trasduttore dell'asse punto-punto..
E1068:	segnale trasd. second. troppo alto asse punto-punto
E1069:	segnale trasd. second. troppo basso asse punto-punto
E1070:	anomalie al trasduttore secondario asse punto-punto
E1080:	anomalie al comparatore potenziometrico..
E1108:	overrun interpol. per blocco successivo non pronto
E1113:	errore memoria ROM modulo Inductosyn
E1116:	errore memoria RAM modulo Inductosyn
E1130:	tempo insuff. per gli assi del modulo Inductosyn

E1158:	errore in acquisizione termocoppia controllo
E1159:	segnale termocoppia interrotto controllo
E1160:	segnale termocoppia troppo alto controllo
E1161:	segnale termocoppia troppo basso controllo
E1162:	anomalie al trasduttore termocoppia controllo
E1163:	segnale troppo alto giunto-freddo
E1164:	segnale troppo basso giunto-freddo
E1165:	anomalie al trasduttore giunto-freddo
E1200:	overrun CPU master: lavorazione simulata
E1202:	overrun CPU master: visualizzazione quote
E1204:	overrun CPU master: campionatura secondaria
E1206:	overrun CPU master: campionatura primaria
E1208:	overrun CPU master: timer di sistema
E1210:	overrun CPU master: debugger PLC
E1212:	overrun CPU master: assi punto-punto
E1214:	overrun CPU master: controlli temperatura
E1216:	overrun CPU master: interpolatore
E1218:	overrun MODIND
E1220:	cicli veloci troppo lunghi alla riga PLC..
E1222:	cicli ultra-veloci troppo lunghi alla riga PLC..
E1224:	troppe scritture in tabella utensili
E1226:	troppe scritture in tabella utensili, riga PLC..
E1300:	malfunzionamento su I/O MIX..
E1302:	Watch Dog espansioni digitali su I/O MIX..
E1304:	Watch Dog su I/O MIX..
E1306:	overrun su I/O MIX..
E1310:	errore sugli output digitali I/O MIX, ottetto..
E1312:	caduta alimentazione +24V I/O MIX..
E1314:	caduta alimentazione +24V I/O MIX, espansione..
E1316:	attesa alimentazione +24V I/O
E1318:	manca alimentazione +5V encoder I/O MIX..
E1320:	manca alimentazione +15V volantini I/O MIX..
E1322:	manca alimentazione +-15V esterni I/O MIX..
E1324:	manca alimentazione potenziometri I/O MIX..
E1421:	DDI C1D Error,drive #..., IDN000BH= ...H, IDN0081H= ...H
E1422:	DDI C2D Error,drive #..., IDN000CH= ...H, IDN00B5H= ...H
E1450:	errore DDI,piastra #..., SRCERM= ...H, SRCERR=...H
E1994:	accesso a componente assente, riga PLC..
E2000:	stack overflow in riga PLC..
E2001:	CCL troppo grande in riga PLC..
E2002:	troppe CALL annidate in riga PLC..
E2004:	RTS sbilanciate in riga PLC..
E2006:	troppe EXEC annidate in riga PLC..
E2008:	ENDE sbilanciate in riga PLC..
E2010:	PLC non attivo
E2012:	PLC non eseguibile
E2014:	DEF SEQCU(n) con numero errato in riga PLC..
E2016:	DEF SEQCU(n)=a,b, errata (ordine a,b,) riga PLC..
E2018:	DEF SEQCU(n)=a,b, incompleta in riga PLC..
E2019:	c.u.a. NON config.: impossibile DEF SEQCU riga PLC..
E2020:	configurato c.u.a. senza posti magazzino
E2021:	parametri vita utensile incongruenti..
E2022:	modo cambio utensile errato: SELECU=..
E2024:	sequenza c.u.a. non gestita da PLC: NSEQCU=..

E2026:	stringa troppo lunga in riga PLC..
E2028:	DISPL su riga inesistente in riga PLC..
E2030:	CLR su riga inesistente in riga PLC..
E2032:	stringa non esistente in riga PLC..
E2034:	indice variabile errato in riga PLC..
E2040:	branch/set unordered condition in riga PLC..
E2041:	not a number float.point in riga PLC..
E2042:	float.point operand error in riga PLC..
E2043:	float.point overflow in riga PLC..
E2044:	float.point underflow in riga PLC..
E2045:	divisione per zero float.point in riga PLC..
E2046:	fpu inexact operation in riga PLC..
E2047:	fpu inexact decimal input in riga PLC..
E2048:	errato uso maschera FPERMK in PLC
E2100:	COMR di un file inesistente in area robot..
E2101:	errore di sintassi in area robot..
E2500:	espressione non compilabile
E2501:	errore di sintassi
E2502:	operando non valido
E2503:	simbolo ASCII troppo lungo
E2504:	operatore non ammesso
E2505:	label non dichiarata
E2506:	richiamo a label tra sezioni diverse
E2507:	riga logica troppo lunga
E2508:	simbolo riservato
E2509:	simbolo già definito
E2510:	sezione già definita
E2511:	indirizzi variabili non pareggiati
E2512:	simbolo non definito
E2513:	errore di dimensione
E2514:	troppi I/O su modulo
E2515:	esaurite PULSE
E2516:	esauriti TIMER
E2517:	esauriti COUNTER
E2518:	esaurite SOFTKEY
E2519:	esaurite HARDKEY
E2525:	troppe HARDKEY per menu
E2526:	richiesto menu HARDKEY inesistente
E2530:	definite troppe variabili
E2532:	non generato codice
E2534:	fatal error: operazione impossibile
E2560:	espressione troppo complessa
E2562:	operandi non congruenti
E2563:	parentesi sbilanciate
E2564:	uso errato di una variabile
E2570:	troppe EXEC annidate
E2571:	EXEC senza ENDE
E2572:	ENDE senza EXEC
E2580:	troppe variabili numeriche da visualizzare
E2581:	troppe variabili stringa da visualizzare
E2590:	troppi segnali digitali da tracciare
E2591:	troppi segnali analogici da tracciare
E32102:	M.U. spenta per interruzione della comunicazione con PC
E10000:	Time-out di attesa risposta da piastra #...
E10001:	Errore su master RIO, piastra #...

- E10002:** Manca file BINario per gestione piastra #...
- E10004:** Nessuno slave rilevato su master RIO,piastra #...
- E10010:** Malfunzionamento slave RIO,piastra #... slave #...
- E10011:** Tipo slave RIO ignoto,piastra #... slave #...
- E10015:** Watch-dog slave RIO,piastra #... slave #...
- E10016:** Errore di ricezione RIO,piastra #... slave #...
- E10017:** Mancata risposta slave RIO,piastra #... slave #...
- E10018:** Errore output RIO,piastra #... slave #... ottetto #...
- E10020:** Errore alim.24V RIO,piastra #... slave #... base
- E10021:** Errore alim.24V RIO,piastra #... slave #... espansione #...

