

# *Sistema PTP1000*

---

## **Manuale d'uso**

versione 4.4E

**tpa** Tecnologie Prodotti per l'Automazione SpA

20099 Sesto S. Giovanni (Mi) Via Carducci, 221 tel.02/26223139 fax 02/2481008

Il presente manuale è di proprietà della TPA SpA - Tecnologie Prodotti per l'Automazione.  
Qualsiasi forma di copia dello stesso può essere effettuata solo su autorizzazione della TPA.  
La TPA si riserva il diritto di apportare modifiche in qualsiasi momento e senza preavviso.

Nell'intento di elevare al meglio la qualità e le prestazioni dei propri prodotti e per rendere un miglior servizio alla propria clientela, la TPA Vi è grata se vorrete informarla di eventuali errori riscontrati, indicando se si tratta di:

- a) errori di funzionamento
- b) errori di documentazione
- c) errori di altro tipo

Nel caso di errori del tipo a) è opportuno indicare le modalità con cui si è prodotto l'errore.

Se si tratta di errori di esecuzione del programma fornire possibilmente anche il listato del programma.

Precisare inoltre la versione Software del Sistema rilevandola dalla videata del Menu principale.

Le informazioni possono essere fornite per telefono o via telefax ai progettisti della Divisione Prodotti/PTP1000 .

# 1. INTRODUZIONE

## GENERALITA'

Il sistema **PTP1000** è un **controllo numerico** basato su 2 elementi fondamentali:

- **Unità di programmazione** e di interfacciamento Uomo-macchina costituita da un **Personal Computer (PC)**.
- una o più **Unità centrali**, costituenti la parte elettronica del controllo, denominate **Moduli**.

Ciascun modulo può contenere una o più schede denominate **Stazioni**, queste schede possono essere di due tipi:

|                |                               |
|----------------|-------------------------------|
| <b>PTP200N</b> | per controllo di 2 assi in CC |
| <b>PLC200</b>  | con funzioni di Plc           |

La scheda PTP200N supporta inoltre una scheda di espansione per il controllo di altri 3 assi in CC denominata **ESPAS**.

Ogni stazione può avere delle **espansioni di Ingressi/Uscite (I/O)**, queste vengono realizzate con le schede: **INOUTR** e **IOMOD**.

Il collegamento tra il PC e i moduli è realizzato con **linea seriale RS232** standard collegata ad **Anello** (vedere Interfaccia RS232).

Il **massimo numero** di **moduli** che possono far parte di un sistema PTP1000 è **16** e sono **numerati da 0 a 15**.

All'interno di ogni modulo possono essere equipaggiate fino a **16 stazioni** e la loro **numerazione** va **da 0 a 15**.

I moduli che compongono il sistema non devono essere necessariamente legati alla stessa macchina ma possono essere di macchine diverse; è quindi possibile controllare da un solo PC il lavoro di più macchine indipendenti l'una dall'altra.

## RAPPRESENTAZIONE DI UNA VIDEATA TIPICA

Come già detto, l'interfaccia Uomo-macchina è un PC e la gestione software del sistema è realizzata dal sistema operativo OS1000.

## 1.2 Introduzione

---

Di seguito viene mostrata una tipica videata delle operatività dell'OS1000, separata nelle varie zone operative.

| <b>Operatività o Funzionalità</b> | <b>Stazione attiva</b> | <b>Modulo attivo</b> |
|-----------------------------------|------------------------|----------------------|
| <b>Zona dati operatore</b>        |                        |                      |
| <b>Elenco tasti disponibili</b>   |                        |                      |
| <b>Tasti funzione</b>             |                        |                      |

### **MODULO E STAZIONE ATTIVI**

Nelle varie operatività, sulla riga 1 del video, vengono visualizzati:

**sulla destra** la descrizione del modulo attivo (in tutte le operatività).

**al centro** la descrizione della stazione attiva (solo in alcune operatività).

Il modulo e la stazione attivi indicano appunto il modulo e la stazione su cui agisce l'operatività in cui ci si trova.

Il cambiamento del modulo o della stazione si ottiene, in tutte le operatività, col tasto funzione F9 - CHANGE.

Quando si seleziona un nuovo modulo viene resa attiva automaticamente anche la stazione 0 dello stesso.

## DESCRIZIONE E UTILIZZO DEI CONTATORI

Ogni stazione ha a disposizione **32 contatori numerati da 0 a 31**. Questi contatori possono essere utilizzati dall'utente nei programmi come **contatori** o **timer** oppure, parte di essi, vengono usati dal programma come **indici** per le **tabelle di quote**, nel caso si faccia uso delle stesse.

La mappa dei contatori è la seguente:

*Indici: Contatori o Timer*

|    |  |
|----|--|
| 0  |  |
|    |  |
| 11 |  |

*Indici: Tabelle di quote o Contatori o Timer*

|    |                  |
|----|------------------|
| 12 | asse X tabella 0 |
| 13 | asse X tabella 1 |
| 14 | asse X tabella 2 |
| 15 | asse X tabella 3 |
| 16 | asse Y tabella 0 |
| 17 | asse Y tabella 1 |
| 18 | asse Y tabella 2 |
| 19 | asse Y tabella 3 |
| 20 | asse Z tabella 0 |
|    |                  |
| 31 | asse V tabella 3 |

Se sono utilizzati come contatori possono assumere valori da 0 a 65535.

La base tempi nell'uso come timer è di 0.01 secondi, quindi il massimo conteggio consentito è di 655.35 secondi.



Le pagine a fine capitolo mostrano la struttura delle configurazioni di I/O. I porti che non sono utilizzati da schede di espansione possono essere usati come flag interni ai programmi.

Come si nota alcuni bit sono già utilizzati per collegamenti specifici e precisamente:

|                 |  |  |
|-----------------|--|--|
| - negli Input:  | <b>ERRS</b><br><b>RDY</b><br><b>SPEX</b> | <p>per uso interno al sistema</p> <p>per uso interno al sistema</p> <p>riguarda il Setpoint e viene usato nei programmi:<br/>             se = 1 Setpoint eseguito<br/>             se = 0 inibisce il test sui fine corsa assi (vedi tabella emergenze)</p> <p>Se SPEX viene usato come definizione simbolica va messo nel file DEF.</p> <p>All'inizio del programma di Setpoint va posto a zero l'istruzione RES SPEX per disabilitare l'uso dei fine corsa assi come emergenze.</p> <p>A fine Setpoint, se questo è stato eseguito correttamente, va posto a uno con l'istruzione SET SPEX per abilitare l'uso dei fine corsa assi come emergenze.</p> <p>All'inizio del programma Automatico è possibile testare lo SPEX per sapere se il Setpoint è già stato eseguito.</p> |
| con             | <b>MSG5</b>                              | <p>per uso interno al sistema</p>  |
| - negli Output: | <b>PWON</b><br><br><b>SYNC</b>           | <p>è un segnale che viene attivato al termine della trasmissione dei parametri all'Inizializzazione del sistema.</p> <p>Può essere usato per l'accensione della potenza.</p> <p>Viene usato solo sulla stazione 0.</p> <p>Il collegamento fisico a questo Output va fatto sull'OUT 23 della scheda Master e precisamente:</p> <ul style="list-style-type: none"> <li>- num. max stazioni 8: bit 7 porto 018</li> <li>- num. max stazioni 16: bit 7 porto 010</li> </ul> <p>per uso interno al sistema.</p>   |

## SIMBOLI UTILIZZATI PER L'ACQUISIZIONE DEI DATI

Nelle finestre, o box, di acquisizione dati appaiono solitamente delle indicazioni sul tipo di dato da inserire: in alcuni casi si può scegliere tra alcune risposte fisse, in altri tra una serie di dati.

Altre informazioni sono: l'unità di misura a cui il dato si riferisce e una guida relativa al tipo di dato e cioè se si tratta di numeri interi, in virgola mobile o di caratteri alfanumerici, quindi di tipo descrittivo.

Per tutte queste informazioni sono stati adottati i seguenti segni convenzionali:

- ( /A/B)** Selezione fissa; i dati sono racchiusi tra parentesi e separati dal carattere /.  
In questo esempio si può scegliere tra i caratteri: **spazio, A e B**.  
La selezione si effettua con i tasti: **Barra Spaziatrice e Cursori sinistro/destro**.
- (1\_4)** Selezione tra una serie di dati; i dati sono racchiusi tra parentesi tonde e separati dal carattere \_.  
In questo esempio si può scegliere tra un valore compreso tra **1 e 4**.
- (0.25\_15/1,2,4,8)** Impostazione di 2 dati (caso Feed Forward punto a punto); i 2 dati sono separati dal carattere /.  
Il primo dato va impostato tra una serie di valori in **virgola mobile** che va da 0.25 a 15.  
Il secondo dato va impostato scegliendo tra i valori **1, 2, 4, 8**.
- (1\_31\*0.25\_15)** Impostazione di 2 dati (caso Feed Forward interpolazione); i 2 dati sono separati dal carattere \*.  
Il primo dato va impostato tra una serie di valori interi compresi tra **1 e 31**.  
Il secondo dato va impostato tra una serie di valori in **virgola mobile** che va da 0.25 a 15.
- [imp/mm]** Unità di misura; l'unità è racchiusa tra parentesi quadre.  
In questo esempio l'unità di misura indicata è:  
**impulsi di encoder per millimetro**.
- ±x.x** Dato in virgola mobile; i 2 caratteri **x** sono separati dal carattere **punto**.  
In questo esempio il dato può essere sia positivo che negativo, il numero di interi e di decimali è a piacere ed è limitato dal campo di acquisizione.
- x** Dato di soli interi; indicato dal carattere **x**.  
In questo esempio il dato può essere solo positivo, la sua lunghezza è limitata dal campo di acquisizione.
- bit+porto o simbolo** Indica che il dato può essere espresso in 2 modi diversi.  
In questo esempio è possibile esprimerlo con:  
**numeri** nel caso **bit+porto** (es. 1005)  
**simbolo** corrispondente (es. FC21).



## TASTI DISPONIBILI NELL'ACQUISIZIONE DEI DATI

Nelle acquisizioni dei dati, a secondo del tipo di dati da inserire, viene abilitato o meno l'uso di alcuni tasti. Ciò permette, già in prima istanza, di evitare che l'operatore inserisca dei caratteri che originerebbero degli errori sintattici.

Se, ad esempio, il dato da inserire è composto da soli numeri interi, solo i tasti da 0 a 9 saranno abilitati.

In altri casi, i caratteri alfabetici se digitati in minuscolo, vengono convertiti automaticamente in maiuscolo.

I tasti che seguono invece, agiscono sul testo ciascuno nel modo a fianco descritto:

|                                |   |
|--------------------------------|---|
| <b>ENTER</b>                   | Termina l'acquisizione dei dati, confermando quanto inserito fino alla sua premuta.   |
| <b>ESC</b>                     | Termina l'acquisizione dei dati con l'abbandono di quanto inserito.   |
| <b>BACK SPACE</b>              | Cancella il carattere a sinistra del cursore.   |
| <b>DEL</b>                     | Cancella il carattere puntato dal cursore.  |
| <b>INS</b>                     | Permette di inserire caratteri partendo dalla posizione del cursore.<br>Si disabilita ripremendolo.<br>Lo stato di INS è evidenziato dal cursore a forma di piccolo quadratino. |
| <b>Cursori sinistro/destro</b> | Spostano il cursore di una posizione a sinistra o a destra rispetto al carattere puntato.   |
| <b>Cursori su/giù</b>          | Spostano il cursore sulla linea precedente o successiva rispetto a quella puntata.  |
| <b>HOME</b>                    | Sposta il cursore sul primo carattere della linea.  |
| <b>END</b>                     | Sposta il cursore dopo l'ultimo carattere della linea.  |
| <b>CTRL+HOME</b>               | Elimina le modifiche fatte sulla linea rimettendo i dati che c'erano in precedenza.   |
| <b>CTRL+DEL</b>                | Cancella tutti i caratteri sulla linea spostando il cursore all'inizio.   |

|                  |   |
|------------------|---|
| <b>TAB</b>       | Nei casi di acquisizione su più colonne sposta il cursore sulla colonna successiva a quella puntata.  |
| <b>SHIFT+TAB</b> | Nei casi di acquisizione su più colonne, sposta il cursore sulla colonna precedente a quella puntata. |

### COME SELEZIONARE UNA VOCE IN UN MENU

In tutti i menu la selezione della voce interessata è resa possibile dai tasti:

|                                |   |
|--------------------------------|---|
| <b>cursore su/giù</b>          | per scorrere voce per voce  |
| <b>PGUP/PGDN</b>               | per scorrere pagina per pagina nel caso il menu sia composto da più pagine. In questo caso sulla destra del box compare una barra di scorrimento verticale. |
| <b>ricerca immediata</b><br>si | digitare, partendo dalle iniziali, il nome della voce desiderata; torna indietro col tasto <b>BackSpace</b> .   |
| <b>ENTER</b>                   | per selezionare la voce puntata.  |

## **2. HARDWARE DEL SISTEMA**

### **UNITA' DI PROGRAMMAZIONE**

L'unità di programmazione e di interfacciamento Uomo-macchina è costituita da un **Personal Computer** Olivetti<sup>1</sup>, IBM<sup>2</sup>, IBM compatibile o PC Industriale **COMPACT90** prodotto dalla TPA; tutti basati su sistema operativo **MS-DOS**<sup>3</sup>.

La configurazione minima del PC è la seguente:

- 640K di memoria ram
- video grafico monocromatico
- hard-disk 20 Mbyte e 1 Floppy-disk drive da 1.44Mb
- interfaccia seriale RS232 per comunicazione asincrona verso unità centrale PTP1000
- interfaccia parallela Centronics per comunicazione con stampante

Opzionalmente è possibile equipaggiare:

- hard-disk 40 Mbyte o superiore
- interfaccia seriale RS232 per comunicazione asincrona con Host-computer
- video grafico a colori CGA/EGA/VGA

### **UNITA' CENTRALE**

L'unità centrale è costituita da una struttura modulare 19" da 6U, basata su schede a norme DIN per l'interfacciamento verso la macchina, predisposta per inserimento in armadio.

I moduli hardware che la compongono sono:

- cpu **PTP200N**
- scheda espansione assi in continua **ESPAS**
- scheda interpolazione veloce **HSINT**
- scheda con funzioni di plc e gestione assi in alternata **PLC200**
- scheda espansione ingressi/uscite **INOUTR**
- scheda espansione ingressi/uscite modulare **IOMOD**
- **alimentatore**

---

<sup>1</sup>. Olivetti è un marchio della Ing. C. Olivetti & C., SpA.

<sup>2</sup>. IBM è un marchio registrato della International Business Machines Corp.

<sup>3</sup>. MS-DOS è un marchio registrato della Microsoft Corp.

### **PTP200N**

Scheda CPU basata su microprocessore 8085AH1 a 10 MHz con:

- 32K memoria RAM
- 64K memoria ROM
- circuiti per interfacciamento con 2 motori CC
- 24 input optoisolati
- 24 output optoisolati a transistor 24Vcc/1A
- interfaccia seriale RS232 per comunicazione asincrona verso PC
- interfaccia parallela per comunicazione veloce tra schede PTP200N e PLC200
- interfaccia per connessione a plug-in con scheda di espansione assi
- interfaccia locale per connessione di schede di espansione  
Ingressi/Uscite tipo INOUTR e IOMOD
- circuito con batteria per memorizzazione permanente dei dati

### **ESPAS**

Scheda di espansione assi con connessione tipo plug-in con PTP200N con:

- circuiti per interfacciamento con 3 motori CC
- uscita analogica ausiliaria  $\pm 10V$
- 4 input optoisolati

### **HSINT**

Scheda di interpolazione basata su microprocessore 80C186 a 16 MHz con:

- 64K memoria RAM
- 64K memoria ROM
- interfaccia per connessione a plug con scheda PTP200N

### **PLC200**

Scheda CPU basata su microprocessore 8085AH1 a 10 MHz con:

- 32K memoria RAM
- 64K memoria ROM
- interfaccia seriale RS232 per comunicazione asincrona verso PC
- interfaccia parallela per comunicazione veloce tra schede PLC200 e PTP200N
- 6 interfacce per connessione a plug con schede modulari di  
Ingresso/Uscita (vedi IOMOD)

- circuito con batteria per memorizzazione permanente dei dati
- interfaccia seriale RS232/RS485 per comunicazione asincrona verso dispositivi digitali di Ingresso/Uscita
- interfaccia locale per connessione di schede di espansione Ingressi/Uscite tipo INOUTR e IOMOD

## INOUTR

Scheda di espansione Ingressi/Uscite con:

- 24 input optoisolati
- 24 output a relè 24V<sub>cc</sub>/2A, 220V<sub>ac</sub>/10A

## IOMOD

Scheda espansione Ingressi/Uscite con:

- 6 interfacce per connessione a plug con schede modulari di Ingresso/Uscita

I moduli per connessione a plug su schede PLC200 e IOMOD sono:

**Modinp** 8 Ingressi optoisolati

**Modoutr** 8 Uscite a relè 24V<sub>cc</sub>/2A, 110V<sub>ac</sub>/10A

**Modainp** 2 circuiti di Ingresso con convertitore Analogico/Digitale a 8 bit

**Modaout** 2 circuiti di Uscita con convertitore Digitale/Analogico a 8 bit con uscite ±10V

**Modcont**<sup>4</sup> 3 circuiti di Conteggio encoder per pilotaggio di motori in corrente alternata.

**Modiofar** Ricetrasmittitore a fibra ottica verso schede remote di ingresso/uscita.

*I moduli remoti sono:*

**Iofar** modulo di ricetrasmisione verso Modiofar

**Inp** modulo con 8 ingressi

**Outpnp e Outnpp** modulo con 8 uscite a transistor 24V<sub>cc</sub>/1A

**Outr** modulo con 8 uscite a relè 24V<sub>cc</sub>/2A,110V<sub>ac</sub>/10A

<sup>4</sup>. Il plug Modcont può essere montato solo su scheda PLC200.

### **ALIMENTATORE**

Versione a cassetto inseribile nel cestello 19":

- +5V 10A
- $\pm 12V$  2A

Versione per montaggio a retroquadro:

- +5V 20A
- $\pm 12V$  2A

### **CESTELLO CE1004**

Cestello completo di alimentatore con 4 posizioni disponibili.

dimensioni: L. 332 H. 265 P. 275

### **CESTELLO CE1010**

Cestello completo di alimentatore con 10 posizioni disponibili.

dimensioni: L. 483 H. 265 P. 275

Il cestello, se fornito con alimentatore a retroquadro, ha 16 posizioni libere.

## **DESCRIZIONE DEI MODULI HARDWARE**

La caratteristica principale del sistema PTP1000 è costituita dalla sua modularità e configurabilità.

Dopo aver esaminato quali sono le parti hardware disponibili, di seguito vengono indicate le possibili configurazioni.

### **Configurazione con massimo 16 PTP200N o PLC200**

Questa è la massima configurazione per ogni modulo.

Questo numero è legato alla capacità di indirizzamento dell'interfaccia parallela (4 bit) utilizzata per la comunicazione veloce tra le schede.

L'indirizzo delle varie schede va da 0 (indirizzo scheda MASTER: posizione più a sinistra nel cestello) a 15.

In questo tipo di configurazione il massimo numero di Input/Output del modulo (1024 Input e 1024 Output) è diviso equamente tra le schede a blocchi di 64 Input e 64 Output. L'espansione può essere realizzata per mezzo delle schede INOUTR e IOMOD, al massimo una per stazione data la dimensione limitata del blocco di Input/Output.

### **Configurazione con massimo 8 PTP200N o PLC200**

Questa è la configurazione standard di un modulo.

In questo tipo di configurazione il massimo numero di Input/Output del modulo (1024 Input e 1024 Output) è diviso equamente tra le schede a blocchi di 128 Input e 128 Output.

Questo tipo di configurazione è consigliabile in quanto, a fronte di una riduzione del numero di schede PTP200N e PLC200, si ottiene un notevole incremento delle schede di espansione di Input/Output equipaggiabili per ogni stazione. Infatti in questa configurazione si può sfruttare la massima espansione possibile di 4 schede INOUTR o IOMOD.

### **Circuito di interfacciamento con motori CC**

Il circuito di interfacciamento con motori CC è costituito dai seguenti elementi:

- uscita analogica  $\pm 10V$  con trimmer di regolazione dell'offset e del guadagno, realizzata con un convertitore D/A a 12 bit.
- ingressi optoisolati per fasi encoder per frequenze fino a 50KHz per fase (200KHz di frequenza di lavoro con moltiplicazione \*4), adatti per **encoder** con uscita **push pull** o **open collector** ad onda quadra alimentati a 5 o 12 Vcc; la tensione di alimentazione encoder viene fornita dalla scheda: può essere selezionata tramite ponticelli a 5 o 12 Vcc/100mA.
- contatore di posizione con discriminatore di direzione (frequenza di campionamento 5 Mhz).

Sulla scheda PTP200N sono forniti i circuiti di interfacciamento per 2 motori CC. Le caratteristiche di pilotaggio sono:

- frequenza di lavoro max 200 KHz (corrisponde a 120 mt/1' con risoluzione 0.01 mm)
- tempo di accelerazione selezionabile in millisecondi
- controllo continuo di velocità e di posizione

## 2.6 Hardware del sistema

---

Su ogni scheda PTP200N può essere aggiunta una scheda di espansione assi con connessione tipo plug-in con i circuiti per altri 3 motori CC (ESPAS). Con questa espansione il numero massimo di motori CC pilotabile dal sistema è:



80 con la configurazione con massimo 16 stazioni

40 con la configurazione con massimo 8 stazioni.

## CONFIGURAZIONE CESTELLO

Come detto in precedenza, il cestello fornito è un contenitore standard formato doppio europa con 16 connettori per l'inserimento di altrettante schede, o 10 schede più l'alimentatore a cassetto.

I segnali sulla carta madre si dividono in:

- alimentazioni
- interfaccia parallela per comunicazione tra PTP200N e PLC200 (**ISB**)
- interfaccia locale per connessione tra PTP200N e INOUTR o IOMOD e tra PLC200 e INOUTR o IOMOD (**ILB**)
- interfaccia seriale **RS232**

### • Alimentazioni

I segnali di alimentazione, costituiti dalle tensioni filtrate  $+10V \pm 10\%$ ,  $\pm 20V \pm 10\%$ , sono multiplati su tutta la carta madre sul connettore (8+8): è sufficiente quindi collegare le alimentazioni al connettore posto sulla scheda madre (questa connessione non è necessaria se si utilizza l'alimentatore a cassetto).

### • Interfaccia parallela (ISB)

Viene utilizzata per il trasferimento veloce dei dati tra le schede PTP200N o PLC200. Consente l'interfacciamento di un massimo di 16 schede ed è già multiplata su tutta la scheda madre.

### • Interfaccia locale (ILB)

Viene utilizzata per l'interfacciamento tra una scheda PTP200N o PLC200 e le relative schede di espansione di I/O (INOUTR o IOMOD).

### • Interfaccia seriale RS232

Viene utilizzata per la comunicazione seriale tra il calcolatore e l'unità centrale. La connessione è di tipo ad **ANELLO**, cioè il segnale di trasmissione del PC è collegato (tramite il connettore posto sulla scheda madre) al segnale di ricezione della prima scheda

## 2.8 Hardware del sistema

---

più a sinistra (MASTER), quindi il segnale di trasmissione della scheda Master è collegato al segnale di ricezione della seconda scheda PTP200N, etc.

Il segnale viene infine rinviato sul ricevitore del calcolatore dall'ultima scheda PTP200N presente nel cestello.

### **3. SISTEMA OPERATIVO OS1000**

#### **FORNITURA DEL SISTEMA OPERATIVO OS1000**

Il sistema operativo del PTP1000 si chiama **OS1000**, viene fornito su floppy-disk da 1.44Mb (3.5") e può girare su hard-disk sotto sistema operativo MS-DOS versione 3.3 e successive.

Il dischetto non contiene file di sistema MS-DOS, pertanto è necessario procedere all'installazione descritta nelle pagine seguenti. Se il PC viene fornito dalla TPA, il sistema di norma è già installato sull'hard-disk.

#### **Composizione del disco**

Il disco è suddiviso in 4 parti, ciascuna delle quali contiene i seguenti file:

##### **1) direttorio radice**

INSTALL.EXE programma d'installazione e di setup.

##### **2) direttorio \PTP1000**

|              |   |
|--------------|---|
| MILLE.EXE    | programma principale.                             |
| MENU.EXE     | operatività menu principale e gestione programmi. |
| VIED.EXE     | operatività editor e compilatore.                 |
| CONFMAIN.EXE | operatività configurazioni.                       |
| PLANCIA.EXE  | operatività gestione macchina.                    |
| AUTOAP.EXE   | operatività programmazione.                       |
| OSCI.EXE     | operatività analizzatore.                         |
| MESSAGES.EXE | gestione messaggi residenti.                      |
| GPL1000.TPA  | tabella istruzioni linguaggio GPL1000.            |
| FONT3        | font per caratteri grafici                        |

##### **3) direttorio \MESSAGGI**

Il numero di file contenuti in questo direttorio dipende dalle lingue che si vogliono installare. I file **TSIMB.lng** e **INFO.lng** possono anche non essere presenti.

|                          |   |
|--------------------------|---|
| MESSPTP.lng              | file dei messaggi nella lingua indicata da lng. Se ad esempio lng è ITA (italiano) sarà: MESSPTP.ITA. |
| MESSOSCI.lng<br>indicata | file dei messaggi dell'operatività Analizzatore nella lingua indicata da lng.                         |

|            |   |
|------------|---|
| ERRSYS.lng | file degli errori di sistema nella lingua indicata da lng.                                  |
| TSIMB.lng  | tabella simboli function nella lingua indicata da lng.                                      |
| INFO.lng   | (opzionale) file con informazioni relative all'installazione, nella lingua indicata da lng. |
| LINGUE.MNU | file con elenco ed estensioni delle lingue selezionabili.                                   |

#### 4) direttorio \TOOLS

|              |   |
|--------------|---|
| COMUNICA.EXE | programma per inviare comandi a un modulo.  |
| SPY.EXE      | programma per intercettare ciò che passa sull'anello seriale utilizzando un altro PC. |
| SPY.HLP      | help per l'uso di SPY.  |

Il file SPY.HLP si può visualizzare o stampare con i comandi DOS: TYPE e PRINT.

## INSTALLAZIONE DEL SISTEMA OPERATIVO E SETUP

L'operatività di installazione può essere utilizzata per 2 scopi:

- installare il software di sistema OS1000 su disco rigido
- modificare i parametri ambientali (SETUP) contenuti nel file ENVIRON.TPA

Durante la prima installazione viene prodotto il file ENVIRON.TPA, descritto più avanti, che viene memorizzato sullo stesso disco su cui viene installato il sistema operativo OS1000.

Prima di procedere all'installazione o al setup si consiglia di leggere le seguenti parti del manuale:

- Struttura del file ENVIRON.TPA (cap. 3)
- Struttura e organizzazione del Disco di lavoro (cap. 4)

### Procedura di Setup e d'Installazione

Il **setup** si effettua eseguendo i punti da 1 a 10 descritti di seguito.  
L' **installazione** si effettua eseguendo tutti i punti descritti.

- 1) Porsi sotto **MS-DOS** in **A:>**.
- 2) Inserire il **disco INSTALLAZIONE** nel drive **A:**.
- 3) Battere sulla tastiera: **A:INSTALL**

- 4) Viene avviato il programma **INSTALL** sul disco e appare il menu delle lingue (se presente il file LINGUE.MNU) per permettere di selezionare la lingua in cui si vogliono avere i messaggi per l'installazione.

Viene poi viene mostrato il riquadro con la richiesta: **Sistema su disco\direttorio**.

Occorre qui indicare il disco e il direttorio su cui si vuole installare il sistema operativo OS1000 o di cui si vuole modificare i dati di setup (es.: C:\PTP1000), il nome del direttorio è a libera scelta dell'utente. Il sistema propone C:\PTP1000.

- 5) Il sistema legge il file ENVIRON.TPA sul disco selezionato in precedenza; se il file non esiste assume dei dati di default.

- 6) Vengono presentati i **Direttori di lavoro**, appaiono qui:

- il nome del direttorio di sistema
- i nomi dei direttori di lavoro di ciascun modulo

E' possibile confermare o modificare questi nomi, specie nel caso in cui siano presenti sullo stesso PC i Dischi di lavoro di macchine diverse, per accedere al Disco di lavoro desiderato.

- 7) Viene presentato il **Menu delle lingue** per la selezione della Lingua utente, quella cioè che sarà assunta all'avviamento del sistema.

- 8) Vengono presentati i **Dati di setup** per la modifica o la conferma, i dati sono:

- Tipo di video
- Lancio da AUTOEXEC.BAT

- 9) Viene eseguito l'aggiornamento dei dati di setup.

- 10) Viene posta la richiesta **Esecuzione installazione**:

rispondendo No il programma termina senza eseguire l'installazione.

Se si è risposto Si al punto precedente, viene avviata la parte di installazione che esegue la copia dei file che fanno parte del sistema operativo OS1000 sul disco destinatario.

### **Struttura del file ENVIRON.TPA**

Questo file contiene i dati relativi all'installazione del sistema operativo e altri dati utilizzati per l'ambiente di esecuzione dei programmi TPA. I dati ambientali possono essere successivamente modificati tramite il SETUP, tranne quelli riguardanti il colore per i quali è

necessario editare direttamente il file ENVIRON.TPA; a questo scopo è possibile utilizzare un Editor qualsiasi.

Nel file possono si possono inserire linee di commento, queste vanno precedute dal carattere **punto e virgola (;)**.

Il file ENVIRON.TPA è di tipo ascii sequenziale e contiene:

- **informazioni per disco di lavoro**

```
SYS = nome direttorio di sistema
  0 = nome direttorio di lavoro del modulo  0
  1 = "   "           " "   " "   1
  :   :
  :   :
 15 = "   "           " "   " "   15
```

- **informazioni per la lingua utente**

LINGUA = nome estensione del file dei messaggi MESSPTP; indica il tipo di lingua contenuta nel file ed è la lingua che viene utilizzata dal sistema.  
Se, ad esempio, **LINGUA = ITA** verrà utilizzato il file di messaggi MESSPTP.ITA.  
Le lingue disponibili sono indicate nel file LINGUE.MNU.

- **informazioni per l'installazione**

AUTOEX = indica se il sistema viene lanciato automaticamente dal PC all'accensione dal file AUTOEXEC.BAT (1) oppure no (0).  
Se **AUTOEX=1**, nel file AUTOEXEC.BAT sarà presente la stringa con il comando: PTP1000.BAT.  
Se **AUTOEX=0**, il sistema dovrà essere avviato battendo sulla tastiera il comando: PTP1000.

- **informazioni per sistema operativo OS1000**

VIDEO = tipo di video (0=Monocromatico[BW]/1=Colori[CO]).

COLORE = n. informazione, colore background seguito dal colore foreground,  
dove:

**n. informazione** è il numero dell'informazione a cui i colori si riferiscono.

**colore background** è il numero del colore dello sfondo in esadecimale.

**colore foreground** è il numero del colore in primo piano in esadecimale.

• **informazioni per i file di messaggi**

DIRLING = nome del percorso che contiene i file di messaggi nelle varie lingue, e cioè:

- MESSPTP.lng
- MESSOSCI.lng
- MESS\*.lng (eventuali altri file di messaggi)
- ERRSYS.lng
- GPL1000.TPA
- LINGUE.MNU

se DIRLING = nessun percorso, viene assunto il direttorio corrente

La **tabella dei colori base** utilizzati è la seguente:

| <b>N. info</b> | <b>Cod.</b> | <b>Col. background</b> | <b>Col. foreground</b> | <b>Descrizione</b>                   |
|----------------|-------------|------------------------|------------------------|--------------------------------------|
| 0              | 07          | NERO                   | BIANCO                 | principale                           |
| 1              | 0F          | NERO                   | BIANCO INTENSO         | alternativo                          |
| 2              | 87          | GRIGIO                 | BIANCO                 | zona tasti                           |
| 3              | 6E          | ARANCIONE              | GIALLO                 | msg errore utente                    |
| 4              | 8A          | GRIGIO                 | VERDE INTENSO          | tasti disponibili                    |
| 5              | 2F          | VERDE                  | BIANCO INTENSO         | msg da istruz. VIDEO                 |
| 6              | 4E          | ROSSO                  | GIALLO                 | msg errori sistema                   |
| 7              | 6E          | ARANCIONE              | GIALLO                 | msg errori ciclo                     |
| 8              | 13          | BLU                    | AZZURRO                | cornice box                          |
| 9              | 1E          | BLU                    | GIALLO                 | titolo box                           |
| 10             | 1A          | BLU                    | VERDE INTENSO          | col. principale box                  |
| 11             | 17          | BLU                    | BIANCO                 | col. alternativo box                 |
| 12             | 3F          | AZZURRO                | BIANCO INTENSO         | col. principale input                |
| 13             | 1F          | BLU                    | BIANCO INTENSO         | col. alternativo input               |
| 14             | 3F          | AZZURRO                | BIANCO INTENSO         | reverse                              |
| 15             | 1C          | BLU                    | ROSSO INTENSO          | primo carattere menu                 |
| 16             | 30          | AZZURRO                | NERO                   | help                                 |
| 17             | CF          | ROSSO INTENSO          | BIANCO INTENSO         | riga 1 schermo                       |
| 18             | 30          | AZZURRO                | NERO                   | titolo tasti funzione                |
| 19             | 6E          | ARANCIONE              | GIALLO                 | tasto funzione premuto               |
| 20             | 1C          | BLU                    | ROSSO INTENSO          | bottone spento                       |
| 21             | 2E          | VERDE                  | GIALLO                 | bottone acceso                       |
| 22             | 2E          | BIANCO                 | NERO                   | linee non protette in Programmazione |
| 23             | 2E          | VERDE                  | BIANCO INTENSO         | linee protette in Programmazione     |



Se, ad esempio, si vuole cambiare il colore principale (n. 0 in tabella) utilizzando un colore NERO su sfondo BIANCO, nel file ENVIRON.TPA dev'essere presente la riga:

COLORE = 0,70.

I **codici dei colori** tra cui si può scegliere sono:

|                 |   |
|-----------------|---|
| NERO            | 0 |
| BLU             | 1 |
| VERDE           | 2 |
| AZZURRO         | 3 |
| ROSSO           | 4 |
| MAGENTA         | 5 |
| ARANCIONE       | 6 |
| BIANCO          | 7 |
| GRIGIO          | 8 |
| BLU INTENSO     | 9 |
| VERDE INTENSO   | A |
| AZZURRO INTENSO | B |
| ROSSO INTENSO   | C |
| MAGENTA INTENSO | D |
| GIALLO          | E |
| BIANCO INTENSO  | F |

### Creazione del file LINGUE.MNU

Questo file contiene i dati relativi alle lingue disponibili e viene utilizzato ogni qualvolta si accede al menu delle lingue; il file può essere editato con un Editor qualsiasi. Il massimo numero di lingue supportate è 10.

Il file viene già fornito col disco OS1000 e comprende le seguenti lingue:

Italiano,ITA  
Español,ESP  
English,ENG  
Deutsch,DEU  
Français,FRA

Se il file non esiste l'unica lingua disponibile è quella indicata nel file ENVIRON.TPA.

Ogni record del file contiene due informazioni, separate tra loro dal carattere virgola:

- **nome della lingua** è il nome che apparirà poi nel menu di selezione lingua (es.: Italiano).

### 3.8 Sistema operativo OS1000

---

- **estensione della lingua** ITA). è l'estensione che contraddistingue il file messaggi (es.:

Il nome dell'estensione è a libera scelta ma dev'essere composto da 3 caratteri alfabetici.

Se ad **esempio** si vuole creare un menu con 3 lingue, si dovranno avere nel file 3 record col seguente contenuto:

Italiano,ITA  
English,ENG  
Deutsch,DEU

Di conseguenza dovranno esistere 3 file di messaggi:

MESSPTP.ITA  
MESSPTP.ENG  
MESSPTP.DEU

Se viene selezionata una lingua di cui non esiste il corrispondente file di messaggi, il sistema emette il messaggio d'errore: lingua non disponibile.

Nel caso si voglia mettere il proprio file LINGUE.MNU sul disco OS1000 fornito dalla TPA, bisogna copiarlo nel direttorio \MESSAGGI; in questo modo verrà automaticamente installato dal programma d'installazione.

## 4. DISCO DI LAVORO

### STRUTTURA E ORGANIZZAZIONE DEL DISCO DI LAVORO

Il **Disco di Lavoro (DdL)** è il disco sul quale vengono memorizzati tutti i file creati dall'utente; nel caso di PC con hard-disk è in pratica una zona del disco rigido.

Il **DdL** è suddiviso in varie parti (**Direttori**) di cui una comune a tutti i moduli denominata **Direttorio di sistema**, le altre sono dedicate ciascuna a un modulo e si chiamano **Direttorio di modulo**.

I nomi di questi direttori possono essere scelti liberamente dall'utente all'atto dell'installazione, come standard i nomi presi per default dal sistema sono:

|                |                                |
|----------------|--------------------------------|
| <b>\PTPSYS</b> | per il direttorio di sistema   |
| <b>\PTP0</b>   | per il direttorio del modulo 0 |
| <b>\PTP1</b>   | " " " " " 1                    |
| :              | " " " " " :                    |
| :              | " " " " " :                    |
| <b>\PTP15</b>  | " " " " " 15                   |

Nel caso si utilizzi un PC con hard-disk, quindi con elevata capacità di memoria, c'è la possibilità di tenere più sistemi sullo stesso calcolatore, semplicemente dando ai vari sistemi nomi di direttori diversi e attivando per il sistema quello desiderato eseguendo un SETUP.

Nel **Direttorio di sistema** vengono memorizzati i seguenti file:

|             |  |
|-------------|--|
| CONFSYS.PAR | dati di configurazione di sistema.                                       |
| OPTSYS.PAR  | dati delle opzioni di sistema.   |
| OPTMOD.PAR  | dati delle opzioni di modulo.  |
| TAMP.DAT    | maschera dello stato dei tamponamenti; utilizzata per le ritrasmissioni. |
| FKEY.DAT    | tasti funzione assegnati dall'utente per l'Editor.                       |
| BACKUP.DAT  | dati del Salvataggio automatico in Programmazione, se utilizzato.        |

Ciascun **Direttorio di modulo** contiene i seguenti file:

|              |  |
|--------------|--|
| CONFGEN.PAR  | dati di configurazione generale del modulo.  |
| CONFSTAZ.PAR | dati di configurazione stazioni del modulo.  |
| CONFCMP.PAR  | dati compilati delle configurazioni; sono i parametri che vengono trasmessi alle stazioni.                       |
| DESCNT.PAR   | dati di descrizione contatori; presente solo se è stata usata l'opzione Descrizione contatori in Configurazione. |
| CONFKEYB.PAR | dati di configurazione della tastiera di autoapprendimento CAT90.  |
| DIRSOR       | direttorio dei file creati dall'utente.  |

e gli altri file, cioè: Programmi, Definizioni I/O, Function, etc.

## COMPOSIZIONE DEI NOMI DEI FILE

Il nome di un file generato col sistema PTP1000 è composto dalla parte **Nome** seguita dal **Numero stazione** a cui il file appartiene; le due parti sono separate dal carattere **punto** (.).

**Esempio:**     SETP.0

In questo esempio la parte **nome** è **SETP** e il file appartiene alla **stazione 0**.

Il **nome** può essere **lungo** fino ad **8 caratteri alfanumerici**, compreso il carattere **meno**; non è possibile usare il carattere punto nel nome poiché questo è usato per separare il nome dal numero stazione.

Durante la memorizzazione, al nome del file il sistema aggiunge anche il **tipo**, inoltre il numero della stazione viene convertito in esadecimale.

Se si esegue un comando DIR sotto MS-DOS, il nome del file dell'esempio sarà visualizzato nel modo seguente: **SETP.0S** (S indica il tipo file Setpoint).

Se il nome fosse: **SETP.15**                     si avrebbe: **SETP.FS**.

Alcuni tipi di file necessitano di un nome fisso per essere riconosciuti dal sistema, questi sono:

|               |                         |              |   |
|---------------|-------------------------|--------------|---|
| <b>SETP</b>   | Setpoint                | (tipofile S) |   |
| <b>DEF</b>    | Definizioni I/O         | (tipofile D) |   |
| <b>ERRlng</b> | Errori di ciclo         | (tipofile E) | dove <b>lng</b> è l'estensione del file di lingua |
| <b>MSGlng</b> | Messaggi dell'operatore | (tipofile M) | dove <b>lng</b> è l'estensione del file di lingua |

I nomi degli altri tipi sono liberi, inoltre i nomi di un gruppo di programmi o di tabelle di quote da eseguire su stazioni diverse, possono essere identici o differenti per ogni stazione a secondo delle **Opzioni di modulo** relative ai nomi comuni attivate in **Configurazione**.

## TIPOLOGIA DEI FILE

I file che l'utente può creare in un sistema PTP1000 sono di vario tipo. I nomi di questi file, all'atto dell'archiviazione su disco, vengono inseriti in un altro file, denominato **Direttorio**, dove vengono aggiunte altre informazioni, quali:

- **commento** al file
- **data** dell'ultima modifica al file
- contrassegno \* per indicare, nel caso si tratti di programmi o function, se è stato **compilato**
- il **numero di byte** che il file occuperà nella memoria ram della scheda
- il **tipofile**

I **tipi di file** sono:

- A** programma **Automatico**; è il ciclo di lavoro
- S** programma di **Setpoint**
- F** file di **Function**
- D** file di **Definizioni I/O**
- Q** **Tabella di quote**
- G** file di **Disegno** generato dal CAD
- E** file degli **Errori di ciclo**
- R** file di **Report**
- T** file di **Testo generico**
- O** file generati dall'operatività **Analizzatore**
- M** file di messaggi per l'istruzione **Message**

i contrassegni del tipofile sono mostrati sotto la **colonna T** nella visualizzazione del **Direttorio**.

La suddivisione in tipi permette nelle varie operatività di accedere all'elenco dei file del tipo desiderato, tramite il seguente menu:

**Programma**<sup>1</sup>  
**Setpoint**<sup>1-2</sup>  
**Function**<sup>1</sup>  
**Definizioni I/O**<sup>1-2</sup>  
**Tabelle di quote**  
**Disegni**  
**Errori di ciclo**<sup>2</sup>  
**Report**  
**Testo**  
**Messaggi**<sup>1-2</sup>  
**Analizzatore**

<sup>1</sup>. Questi tipi di file vengono compilati automaticamente prima della memorizzazione del file sorgente; viene così prodotto un file compilato che nel caso dei file di tipo: Programma, Setpoint e Function contiene la parte di codice che viene poi trasmessa alle schede quando si esegue il programma. Per questi tipi quindi, oltre al file sorgente, si avrà anche un file compilato; la sua presenza è indicata nel Direttorio dal carattere asterisco (\*) posto nella colonna C.

<sup>2</sup>. Questi tipi di file hanno nomi fissi, pertanto la selezione del file non avviene col nome ma selezionando la stazione a cui il file appartiene.

## PROGRAMMI DI SETPOINT E DI AUTOMATICO

I programmi o cicli che l'operatore può creare sono di 2 tipi:

- programmi di Setpoint (tipofile S)
- programmi di Automatico (tipofile A)

I **programmi di Setpoint** hanno nome fisso **SETP** e ce n'è uno per ogni stazione. Vanno creati solo se nelle **Opzioni di modulo in Configurazione generale** è stata attivata l'opzione: **Uso operatività di Setpoint**; se non è stata attivata è inutile crearli poiché il sistema non li gestirebbe. Avendo nome fisso, essi vengono selezionati tramite il menu delle stazioni.

I **programmi di Automatico** hanno un nome liberamente scelto dall'operatore e possono essere anche più di uno per stazione; sarà poi l'operatore a mettere in esecuzione quello desiderato. Volendo, è possibile eseguire la parte di Setpoint in testa al programma Automatico; ecco perché l'operatività di Setpoint è opzionale.

Se in **Configurazione generale** è stata attivata l'**Opzione di modulo: Nomi programmi comuni per tutte le stazioni**, il nome dei programmi che si intende far eseguire contemporaneamente su ogni stazione deve essere uguale, ad esempio:

**AUTO.0**            per la stazione 0  
**AUTO.1**            per la stazione 1

Ogni programma è normalmente composto da:

- **parola riservata PROG**            identifica l'inizio di uno dei 4 programmi paralleli (1÷8).
- **istruzioni GPL1000**            sono le istruzioni utilizzate per comporre il programma; vedere al riguardo il capitolo dedicato al Linguaggio GPL1000.
- **richiami di function**            è la chiamata di una function che risiede in un altro file; vedere al riguardo Descrizione dei file di function.
- **commenti**                            sono dei commenti che possono essere inseriti nel testo; devono essere preceduti dal carattere **punto e virgola (;)**.

Un programma, sia esso di Setpoint o Automatico, può essere separato in **8 programmi o task** che verranno eseguiti in parallelo (**multitasking**); questi sono **numerati da 1 a 8**. Il **programma 1** è considerato il **programma principale** da cui verrà poi attivata l'esecuzione di uno degli altri programmi, se utilizzati. La gestione dell'esecuzione degli 8 programmi e dell'interazione tra gli stessi, è ottenuta tramite opportune **istruzioni GPL1000**.

Ciascuno degli 8 programmi deve essere preceduto dalla **parola riservata PROG** seguita dal numero del programma (es.: **PROG 2**); nel caso del programma principale (**PROG 1**) è sufficiente scrivere **PROG**.

**Esempio:**

```

PROG                ;dichiarazione di programma principale (1)
APROG  2            ;attiva esecuzione programma 2
|
|
APROG  2,START     ;attiva esecuzione prog. 2 da etichetta
|
END

PROG  2            ;dichiarazione del programma n. 2
|
START: EPROG  2    ;fine esecuzione programma n. 2

```

E' possibile attivare l'esecuzione di un programma parallelo da un punto qualsiasi dello stesso identificato da una etichetta; questo significa che potenzialmente ciascuno dei programmi da 2 a 8 può essere suddiviso in tanti piccoli programmi, attivati e disattivati secondo le necessità del ciclo.

Un altro sistema è quello di avere un programma principale e tanti programmi paralleli ( o task). I programmi paralleli possono essere messi in esecuzione, fino a un massimo di 8 contemporaneamente, dal programma principale che li richiamerà tramite il loro nome.

Queste task vengono eseguite solo se c'è una disponibilità di task libere pari al numero delle task che si desidera eseguire contemporaneamente, in caso contrario il programma attende che si verifichi questa condizione prima di passare all'istruzione successiva.

Ogni task inizia con la parola riservata PROG seguita dal nome della task e termina, nel punto di uscita, con l'istruzione EPROG.

**Esempio:**

```

PROG                ;dichiarazione di programma principale (1)
|
CICLO: X            30,Q
        Y 25, Z 18, Q
        APROG Chiudi, Scarico ;attiva due prog. paralleli
        X            0,Q
        WPROG Chiudi          ;attesa fine prog. parallelo
        APROG Reset          ;attiva un prog.parallelo
        Y 0, Z 0, Q
        BRA         CICLO

```

;programmi paralleli

```
    PROG    Chiudi          ;dichiarazione inizio progr. parallelo
    |
    ChiudiRiparo          ;function
    |
    EPROG                ;fine programma parallelo

    PROG    Scarico        ;dichiarazione inizio progr. parallelo
    |
    |
    |
    EPROG                ;fine programma parallelo

    PROG    Reset          ;dichiarazione inizio progr. parallelo
    |
    ResetTubi            ;function
    |
    EPROG                ;fine programma parallelo
```

## FILE DI DEFINIZIONI SIMBOLICHE

Tutti gli ingressi, le uscite, i flags e altri tipi di dati utilizzati in un sistema PTP1000 possono essere definiti con dei simboli; ciò permette di rendere poi condivisibili questi simboli ai programmi e alle function durante la compilazione. Si possono definire con simboli solo i dati che sono normalmente espressi con numeri interi.

Non è obbligatorio utilizzare i simboli, tuttavia è consigliabile farlo, almeno per gli I/O, per ragioni di chiarezza di programmazione. Le definizioni di I/O inoltre vengono visualizzate anche nell'operatività di Diagnostica, permettendo così di operare sugli I/O con maggior facilità.

Il file che contiene le definizioni ha un nome fisso che è **DEF** e se ne può creare uno per stazione, il **tipofile** è **D**. Poiché ha nome fisso, la selezione per l'edit del file avviene col menu delle stazioni una volta selezionato il tipofile **Definizioni I/O**.

La **lunghezza** di un simbolo è libera, tuttavia nell'operatività Manuale e diagnostica vengono visualizzati solo i primi 8 caratteri; il **massimo numero** di definizioni simboliche che si possono utilizzare è **500**.

Ogni **riga di testo** nel file si compone di:

**simbolo**            **codice**            **;commento**



dove:

- **simbolo** è il simbolo che si vuole attribuire al dato; può essere composto da caratteri alfanumerici e dai caratteri ÷ - < >.

- **codice** è il codice a cui si riferisce il simbolo e può appartenere a una delle seguenti categorie:

**BYTE** identifica un valore tra **0** e **255** e può indicare un **bit** (se il valore non è maggiore di 7), un **dato**, un **porto** o un **valore**; può essere:

**decimale** (es.: 7 15 121) Max: 255

**esadecimale** se preceduto dal simbolo **\$** e seguito da **H**  
(es.: \$1H \$EAH) Max: \$FFH

**binario** se preceduto dal simbolo **\$** e seguito da **B**  
(es.: \$10B \$01110011B) Max: \$11111111B

**un'altra definizione di tipo BYTE**

**BITPORTO** indica una linea di ingresso o di uscita o un flag ed è composto da un valore tra **0** e **7** e un altro tra **0** e **255**; può essere:

**4 cifre decimali** la prima indica il **bit 0÷7**, le altre il **porto 0÷255**  
(es.: 2150 0001 7255)

**un valore di tipo BYTE** non superiore a 7, il carattere + e un altro **valore di tipo BYTE**  
(es.: 5+122 3+EV BIT+\$FFH)

**un'altra definizione di tipo BITPORTO**

**DATOPORTO** identifica la composizione di un dato con un porto (vedere istruzione Gpl OUT); può essere:

**6 cifre decimali** **0÷255+0÷255**, le prime tre cifre..., le altre tre ...  
(es.: 012021)

**un valore di tipo BYTE**, il carattere + e un altro **valore di tipo BYTE**  
(es.: 57+98 \$49H+\$11B  
DATO+PORTO)

### un'altra definizione di tipo DATOPORTO

**STRINGA** identifica un messaggio che deve essere racchiuso tra doppi apici:

Se ad esempio si crea la definizione:

```
SEPT_OK    "Setpoint terminato correttamente"
```

nel programma si potrà usarla con l'istruzione VIDEO in questo modo:

```
VIDEO      SETP_OK
```

il **QUOTA** identifica una quota; la quota deve avere obbligatoriamente punto decimale.

Se ad esempio si crea la definizione:

```
RIPOSO    10.0
```

nel programma si potrà usarla con tutte le istruzioni che gestiscono quote, nel seguente modo:

```
X    RIPOSO, Y RIPOSO, Q
```

- **commento** è un commento, opzionale, che deve essere preceduto dal carattere **punto e virgola (;)**.

Nel file di Definizioni simboliche si potrà ad **esempio** mettere:

```
EV1    1010 ;bitporto
EV2    2     ;bit
PRG2   2     ;valore
NRIP1  50    ;valore
ERR1   1     ;valore
PORT10 10    ;porto
MASK   125   ;dato
ZERO   0     ;dato
```

Questi simboli potranno, ad **esempio**, essere usati nel programma nel seguente modo:

```

PROG
      APROG      PRG2
      OUT        ZERO+PORT10,MASK
      SET        EV1
      REPEAT     NRIP1
      |
      SET        EV2+PORT10
      |
      ENDREP
      |
      ERROR      ERR1

```

## FILE DI FUNCTION

### Premessa

Le function, **tipofile F**, sono delle routines o parti di programma, scritte in linguaggio GPL1000, che possono essere richiamate da un programma, sia esso di Setpoint o di Automatico.

Le function offrono il seguente vantaggio: vengono trasmesse alle stazioni all'atto dell'**Inizializzazione sistema** e rimangono residenti nella memoria delle schede. Se si cambia il programma che le utilizza non vengono ritrasmesse e possono essere richiamate dal nuovo programma.

E' comunque possibile, in qualsiasi momento, modificare le function o ricompilarle e ritrasmetterle alle schede.

Ogni function è contraddistinta da un **numero** che va da **0 a 255**, ogni stazione può quindi avere fino a **256 function**; una function può essere richiamata più volte dallo stesso programma anche tramite un simbolo che gli è stato eventualmente attribuito e la cui lunghezza massima è 15 caratteri alfanumerici più i caratteri `_ - <>`.

E' possibile creare più file di function per ogni stazione e teoricamente si potrebbe avere un file di function per ogni function, tuttavia è consigliabile raggruppare le function in file secondo un criterio che tenga conto delle operazioni da loro svolte; volendo e se le function sono poche, si può metterle anche in un unico file.

Si accede all'**edit** di un file di **function** selezionando dal menu di edit il **tipofile Function**.

L'inizio di ogni function deve essere segnalato nel file dalla parola **FUN** seguita dal **numero della function (0÷255)**, dal **nome simbolico** se si intende utilizzarlo, ed eventualmente dal **commento**.

### Esempio:

```
FUN      10,ApriPinza ;commento alla function
|
|
FRET
```

La function dell'esempio potrà essere richiamata dal programma in uno dei seguenti modi:

con il **numero**:               **FCALL     10**

con il **simbolo**:           **ApriPinza**

Il ritorno da una function al programma che l'aveva chiamata viene eseguito dall'istruzione **FRET** che deve risiedere nella function.<sup>3</sup>

E' inoltre possibile eseguire una o più function come programmi paralleli utilizzando l'istruzione **FPROG**, in questo caso vale quanto detto sulla multiprogrammazione.

Nello scrivere una function, vanno osservate le seguenti regole:

- si può innestare fino a un massimo di 4 chiamate di function e cioè: una function può richiamarne un'altra, questa un'altra ancora, fino a 4 livelli di chiamata; una function però non può richiamare se stessa.
- non si può usare l'istruzione **APROG** (attiva programma) in una function.
- la lunghezza massima del codice oggetto di ogni function è 4095 bytes.
- una function non può comparire più di una volta nello stesso file o in file di function diversi della stessa stazione.

Ogni function viene compilata come un programma a sè stante, è pertanto possibile usare lo stesso nome di etichette, per i salti, in function diverse come si può notare nelle function 10 e 20 dell'esempio successivo per quanto riguarda l'etichetta **ERRFC02**.

E' comunque consigliabile, per chiarezza di programmazione, evitare di usare le stesse etichette in function diverse, almeno per quelle che risiedono nello stesso file.

Durante la trasmissione delle function alla stazione, il sistema trasmetterà tutte le function dei file che saranno stati precedentemente compilati, mentre quelle contenute nei file non compilati non saranno trasmesse. Questo perché potreste decidere di non usare le function di un certo file e di conseguenza questo non è stato compilato.

---

<sup>3</sup>. Il simbolo, sia esso scritto in minuscolo o in maiuscolo, o composto da entrambi i tipi di carattere, è considerato sempre lo stesso, ad esempio: ApriPinza o APRIPINZA, con tutte le sue combinazioni, è considerato sempre lo stesso simbolo.

**Esempio** di un file di function:

```

; *** function 10 ***

      FUN      10,ApriPinza      ;apertura pinza

      RES      EV01
      SET      EV02
      WIZ      FC02,ERRFC02
      FRET

ERRFC02: VIDEO  FINE CORSA FC02 NON RILEVATO
      RES      EV02
      FRET

; *** function 20 ***

      FUN      20

      SET      EV02
      WIZ      FC02,ERRFC02
      Y        6
      RES      EV02
      FRET

ERRFC02: VIDEO  FINE CORSA FC02 NON RILEVATO
      RES      EV02
      FRET

```

**Function parametriche**

Le function parametriche, a differenza di quelle normali, danno la possibilità di sostituire la maggior parte degli argomenti delle istruzioni GPL1000 con dei simboli; questo permette poi di passare i valori, o parametri, da sostituire ai simboli sulla chiamata della function nel programma. I parametri vanno passati nello stesso ordine dei simboli a cui si riferiscono.

I simboli devono essere preceduti dal carattere **&** e devono essere presenti in un file denominato **TSIMB.lingua** (spiegato più avanti); questo file deve risiedere nel direttorio del sistema operativo OS1000.

Se un simbolo viene utilizzato più volte nella stessa function, assume sempre lo stesso significato e cioè si riferisce sempre allo stesso dato, pertanto sulla chiamata sarà sufficiente passarlo una sola volta.

**Esempio:**

Supponiamo di avere la seguente function parametrica in un file chiamato FUNCTION:

|      |           |                       |
|------|-----------|-----------------------|
| FUN  | 05,Fparam | ;function parametrica |
| SET  | &OUTPUT   |                       |
| X    | &QX       |                       |
| Y    | &QY       |                       |
| WEND | XY,Q      |                       |
| RES  | &OUTPUT   |                       |
| FRET |           |                       |

Nel programma si potrà richiamare la function tutte le volte che si desidera, passandole di volta in volta parametri diversi:

```
PROG  
  
Fparam    EV1,12,25  
Z         10,Q  
Fparam    EV5,120,40.5  
END
```

Come si nota, il dato riferito al simbolo OUTPUT viene passato una sola volta su ciascuna chiamata, sarà il compilatore a espanderlo. Il numero massimo di simboli utilizzabili in una function è 60.

Durante la compilazione della function, il compilatore controlla che i simboli siano presenti nel file TSIMB.lingua effettuando così un'analisi sintattica; viene inoltre verificato che il tipo di simbolo sia congruente con l'argomento dell'istruzione, tranne che per le istruzioni di interpolazione. Per ciascuno di essi viene inoltre riservato un numero di bytes predisposti al valore 0, il numero di byte è ovviamente pari alla lunghezza del tipo di parametro (es.: per bit+porto sono riservati 2 byte uguali a 0); la somma totale dei bytes di parametri in una function è 255. Per ogni function viene creata una tabella di descrittori che contiene l'indirizzo e la lunghezza in byte di ogni parametro.

### **Direttorio function**

Durante la compilazione di function viene gestito un file denominato **SYMFUN.nstazioneR**, (tipofile R), che è in pratica il direttorio delle stesse. Questo file viene aggiornato tutte le volte che si crea, si modifica, si cancella o si compila un file di function.

Nel direttorio vengono memorizzati nell'ordine:

- 1) numero della function in decimale
- 2) nome della function
- 3) commento alla function (solo i primi 40 caratteri)
- 4) nome del file che contiene la function
- 5) numero totale bytes di parametri, in decimale
- 6) simboli usati nella function

tutti separati tra loro dalla virgola.

Rifacendoci all'esempio delle function parametriche, nel file SYMFUN avremo:

```
05,Fparam,function parametrica,FUNCTION,12,OUTPUT,QX,QY,OUTPUT
```

Nelle operatività Editor e Manuale e diagnostica, è possibile vedere il direttorio delle function tramite il tasto funzione **F7 - FUN**, mentre lo stesso si può stampare col comando PRINT nell'operatività Gestione programmi.

In Editor sono visualizzati i dati da 1) a 4).

In Manuale e diagnostica sono visualizzati i dati da 1) a 3).

### Tabella dei simboli utilizzabili nelle function parametriche

I simboli usati nelle function parametriche devono risiedere in un file denominato **TSIMB** seguito dal tipo di lingua a cui questo si riferisce<sup>4</sup>; per quella italiana sarà: TSIMB.ITA.

La TPA fornisce già un file TSIMB.ITA, questo può essere utilizzato così com'è o modificato a piacere; in ogni caso si consiglia di fare una copia di salvataggio del proprio file TSIMB perchè durante l'installazione quello sul disco OS1000 TPA viene sempre copiato.

Il file deve stare nel direttorio del sistema operativo OS1000, è unico per tutto il sistema e può essere editato con un Editor qualsiasi. La prima riga deve contenere il numero totale dei simboli, mentre ciascuna delle altre contiene le informazioni relative a ciascun simbolo. Queste informazioni, separate tra loro dal carattere virgola, sono:

- |  |   |
|--|---|
| <b>1) nome del simbolo</b><br>usa            | può essere composto da un massimo di 10 caratteri alfanumerici a scelta; va scritto senza il carattere <b>&amp;</b> che lo precede quando lo si usa nella function).                |
| <b>2) messaggio</b><br>cui<br>di 35          | è un messaggio che viene visualizzato nell'acquisizione del dato a cui si riferisce il simbolo, può essere composto da un massimo di 35 caratteri alfanumerici.                     |
| <b>3) help</b><br>lunghezza                  | se presente, viene affiancato al messaggio descritto in precedenza per l'aiuto nella acquisizione del dato, può avere una massima di 40 caratteri alfanumerici.                     |
| <b>4) numero dell'argomento</b><br>rispetti- | è il numero che contrassegna il tipo di argomento di istruzione a cui il simbolo si riferisce; i tipi d'argomento e il loro codice sono elencati in questa pagina e nella seguente. |

Sempre rifacendosi all'esempio fatto in precedenza, nel file TSIMB.ITA dovrebbero esserci i seguenti dati:

```
3
OUTPUT,,,3
QX,,,50
QY,,,51
```

<sup>4</sup>. Vedere l'elenco delle lingue supportate in: Struttura del file ENVIRON.TPA nel capitolo 3.

### Tipi di argomenti delle istruzioni GPL1000

Di seguito sono elencati tutti i tipi di argomenti delle istruzioni GPL1000; la conoscenza dei tipi e del relativo codice si rende necessaria durante la creazione del file TSIMB descritto in precedenza.

Per ogni tipo, oltre al codice, vengono elencati anche il valore che può assumere e una breve descrizione; nella colonna centrale alcuni valori sono descritti con i segni convenzionali spiegati nel capitolo Introduzione.

| <b>Codice</b> | <b>Valore</b>   | <b>Descrizione argomento</b>                           |
|---------------|-----------------|--|
| 1             | 0÷7             | bit  |
| 2             | 0÷255           | dato o porto   |
| 3             | 0÷7+000÷255     | bit+porto  |
| 4             | 000÷255+000÷255 | dato+porto   |
| 5             | 0÷31            | numero contatore                                       |
| 6             | 0÷65535         | valore contatore                                       |
| 7             | 1÷65535         | numero ripetizioni                                     |
| 8             | 1÷255           | numero errore di ciclo                                 |
| 9             | 0÷31            | numero offset  |
| 10            | 0/1             | no/nc  |
| 11            | 1÷8             | numero utensile  |
| 12            | 0÷3             | numero tabella di quote                                |
| 13            | A/0÷3           | numero tabella di quote (A=All)                        |
| 14            | </=/>           | operatori di confronto                                 |
| 15            | A/R/D/F/Q       | stato assi   |
| 16            | A/R/D/F/Q/+/-   | stato assi e direzione                                 |
| 17            | -10.000÷10.000  | tensione su uscita analogica                           |
| 18            | x.x             | tempo per istruzione Delay                             |
| 19            | x               | accelerazione d'interpolazione in ms                   |
| 20            | x.x             | velocità d'interpolazione                              |
| 21            | XYZWV           | maschera composta dai nomi degli assi                  |
| 22            | XYZWV/I         | maschera composta dai nomi degli assi o interpolazione |
| 50            | ±x.x            | quota asse X o X1                                      |
| 51            | ±x.x            | quota asse Y o Y1                                      |
| 52            | ±x.x            | quota asse Z o Z1                                      |
| 53            | ±x.x            | quota asse W o W1                                      |
| 54            | ±x.x            | quota asse V o V1                                      |
| 55            | ±x.x            | quota asse U1  |
| 56            | ±x.x            | quota asse X2  |
| 57            | ±x.x            | quota asse Y2  |
| 58            | ±x.x            | quota asse Z2  |
| 59            | ±x.x            | quota asse W2  |
| 60            | ±x.x            | quota asse V2  |
| 61            | x.x             | quota asse U2  |
| 62            | ±x.x            | quota asse X3  |
| 63            | ±x.x            | quota asse Y3  |



---

|     |           |   |
|-----|-----------|---|
| 64  | $\pm x.x$ | quota asse Z3   |
| 65  | $\pm x.x$ | quota asse W3   |
| 66  | $\pm x.x$ | quota asse V3   |
| 67  | $\pm x.x$ | quota asse U3   |
| 68  | x.x       | velocità asse X   |
| 69  | x.x       | velocità asse Y   |
| 70  | x.x       | velocità asse Z   |
| 71  | x.x       | velocità asse W   |
| 72  | x.x       | velocità asse V   |
| 74  | x         | accelerazione asse X  |
| 75  | x         | accelerazione asse Y  |
| 76  | x         | accelerazione asse Z  |
| 77  | x         | accelerazione asse W  |
| 78  | x         | accelerazione asse V  |
| 200 | XYZWV     | maschera composta dai nomi degli assi che interpolano per interpolazione a 2 assi |
| 201 | XYZWV     | maschera composta dai nomi degli assi che interpolano per interpolazione a 3 assi |
| 202 | O/A       | senso d'interpolazione  |
| 203 | x.x       | velocità d'interpolazione   |
| 204 | x.x       | numero giri d'interpolazione  |
| 205 | $\pm x.x$ | quota o raggio o passo d'interpolazione   |

## TABELLE DI QUOTE

### Premessa

Le Tabelle di quote possono essere utilizzate solo su schede con assi in corrente continua.

L'uso di Tabelle di quote, **tipofile Q**, offre soprattutto i seguenti vantaggi:

- agevola l'uso del sistema da parte dell'operatore poiché può agire solo sulle quote, non essendo obbligato a conoscere il linguaggio di programmazione.
- permette di costruire parecchi file di tabelle contenenti le quote legate ad ogni lavorazione, mantenendo invariato il programma che eseguirà queste quote.

Un file di tabelle di quote può contenere fino a un **massimo di 4 tabelle**, numerate da **0** a **3**, per ogni asse.

Le quote nelle tabelle verranno poi gestite dal programma che le utilizza tramite apposite istruzioni GPL1000.

E' possibile creare uno o più file di tabelle di quote per ogni stazione e poi decidere quale abbinare al programma in esecuzione; in questo caso nell'operatività Automatico appare anche il nome della tabella in uso e il relativi-vo commento.

Nei casi in cui si renda necessario attivare o disattivare una o più uscite o flag in stretto abbinamento con l'esecuzione delle quote in tabella, è possibile configurare il tipo di uscita o flag e vincolarla ad un asse; l'uscita o le uscite verranno gestite prima di ogni movimento di quell'asse.

### Edit

Una volta selezionato il file di tabelle da editare, nuovo o esistente, appare il menu dei tasti funzione:

#### **F1 - DIM (dimensioni)**

Vanno qui indicate le dimensioni, cioè il numero di quote per ogni tabella che si intende utilizzare; è possibile dimensionare tabelle solo per gli assi esistenti nella stazione.

Il **numero totale di quote** è determinato da due fattori:

- l'area di memoria assegnata alle tabelle in Configurazione.
- il numero di byte utilizzati per esprimere la quota (vedere: **F4 - Option**).

#### **F2 - OFFSET (offset di zero)**

Permette di impostare una quota di offset di zero per ciascun asse di ogni tabella. Questo offset viene sommato all'atto dell'esecuzione dell'istruzione EQTAB alla quota puntata in tabella, nel caso sia selezionato il movimento assoluto dell'asse.

Questo consente di programmare le quote in tabella relativamente ad uno zero di lavoro diverso dalla posizione di Setpoint dell'asse.

### F3 - QUOTE (selezione tabelle)

Permette di **inserire** o **modificare** le **quote** e il dato relativo all'uscita (se presente) di una delle 4 tabelle.

Se è stata utilizzata più di una tabella bisogna selezionare quella desiderata, altrimenti appare l'unica tabella usata.

Se la tabella è nuova, le quote vengono predisposte a zero.

Nel caso la tabella venga successivamente allungata (F1 - DIM) solo le quote aggiunte vengono messe a zero, se invece viene accorciata le quote in eccesso sono rimosse.

Durante la fase di edit delle quote, il tasto funzione **F1 - EXPAND** attiva un comando che permette di creare in modo automatico un certo numero di quote che abbiano lo stesso passo in millimetri l'una dall'altra. Se si è sul campo **Out** il comando **F1 - EXPAND** esegue la stessa funzionalità per quanto concerne i dati relativi all'uscita.

### F4 - OPTION (opzioni)

Permette di attivare le opzioni legate alle tabelle, queste sono:

**numero di byte per quota** Indica quanti byte la quota occuperà nella memoria della scheda. Per ogni tabella è possibile scegliere tra tre lunghezze: **1, 2, 4**.

A ciascuna di queste corrisponde un **range** di corsa dell'asse in **impulsi** e precisamente:

**1 byte = da -128 a 127**

**2 byte = da -32768 a 32767**

**4 byte = da -2147483648 a 2147483647**

**Generalmente**, e comunque nel caso non si abbiano problemi di memoria, **va impostato 4** per avere il massimo range di corsa dell'asse; questo valore è già attivato per default dal sistema quando si edita una nuova tabella.

**rate** Utilizzato nei movimenti coordinati degli assi, determina l'intervallo di tempo nel quale viene eseguita una quota.

Questo numero, da 1 a 128 in progressione quadratica, viene utilizzato come segue:

$$\text{intervallo tempo} = \text{rate} * 1 / \text{frequenza real time}$$

dove la frequenza di real time è tipicamente pari a 200 Hz, quindi con:

$$\text{rate} = 1 \quad \text{intervallo tempo} = 5 \text{ ms}$$

rate = 128 intervallo tempo = 640 ms.

Si intuisce che a parità di spostamento incrementale, la velocità dell'asse in movimento coordinato sarà tanto più alta quanto più piccolo verrà impostato il valore di rate.

### **F5 - OUTPUT (parametri uscite)**

Consente di configurare, per ogni tabella (0÷3), il tipo di Uscita e di associarla ad un asse, l'uscita verrà attivata prima di ogni movimento dell'asse.

I parametri da impostare sono:

**Asse** E' possibile impostare uno dei seguenti valori:

**(/X/Y/Z/W/V)**

Se l'asse non è impostato, l'uscita associata alla tabella non viene gestita e nella funzionalità F3 - QUOTE non apparirà la colonna Out.

Se viene impostato il nome di un asse, l'uscita verrà attivata prima di ogni movimento di quell'asse; in questo caso nella funzionalità F3 - QUOTE apparirà la colonna Out.

**Uscita** E' possibile impostare uno dei seguenti valori:

**(bit+porto, maschera+porto, simbolo)**

Vediamo i 3 casi:

**- bit+porto** indica il bit che va attivato o disattivato; nella colonna Out della funzionalità F3 - QUOTE si potrà scegliere tra ON o OFF.  
Il valore di default per una tabella nuova è OFF.

**- maschera+porto** è la maschera dei bit utilizzati come Out; nella colonna Out della funzionalità F3 - QUOTE si potrà scegliere un valore compreso tra 0 e 255 che rappresenta la maschera dei bit che si vuole attivare o disattivare.

Il valore di default per una tabella nuova è 0.

- **simbolo** è uno dei due casi precedenti rappresentato con una definizione simbolica di I/O.

### **F10 - EXIT (salva ed esce)**

Effettua la memorizzazione del file, se modificato, e il ritorno al menu dell'Editor. Se si esce dall'operatività col tasto ESC (escape) tutte le modifiche vengono perse.

## **FILE DEGLI ERRORI DI CICLO**

Il file degli Errori di ciclo, **tipofile E**, contiene tutti i messaggi di errore previsti dall'operatore nei programmi di Setpoint, Automatici o nelle Function; ogni messaggio ha una lunghezza massima di 60 caratteri.

Il nome del file è composto dalla radice fissa **ERR** seguita dai 3 caratteri di estensione che contraddistinguono il file di lingua a cui il file si riferisce; può esserne creato uno per ogni stazione.

La selezione per l'edit avviene col menu delle stazioni una volta selezionato il tipo di file **Errori di ciclo**.

L'operatività di Editor carica il file associato alla lingua attiva in quell'istante. Se da un file con messaggi in italiano (es.:ERRITA.0E) si vuole ad esempio creare un file per la lingua inglese, si può procedere nel seguente modo: caricare il file degli errori di ciclo in italiano, modificare i messaggi con quelli in inglese, selezionare F10-exit e poi il comando Salva o Esci e alla richiesta del nome del file sostituire a ERRITA il nome ERRENG.

Nei programmi, la visualizzazione di questi errori viene attivata sia con l'istruzione **ERROR** che da altre istruzioni utilizzando il numero dell'errore come indice del messaggio.

Quando durante il ciclo viene rilevato uno di questi errori la task in cui si è verificato viene sospesa, sempre che l'istruzione che ha generato l'errore non abbia l'opzione di continuazione, nel qual caso l'esecuzione prosegue. La scheda invia al PC il numero dell'errore e questo lo preleva dal file. Se non è stato creato un file di Errori di ciclo per la stazione interessata, il PC preleverà il messaggio dal file di errori della stazione 0. Nel caso il messaggio non esista, viene visualizzato solo il numero dell'errore.

Per mantenere la compatibilità con versioni precedenti nel caso non venga trovato il file degli errori nella lingua attiva, il sistema cerca di caricare il file **ERRCYC** della stazione in questione, se non lo trova cerca quello della stazione 0.

L'errore viene visualizzato preceduto dal numero di riga che occupa nel file, questo per facilitare il riconoscimento dell'errore da parte del personale di assistenza nel caso di problemi installate all'estero.

## **FILE DI REPORT**

Questi file, **tipofile R**, contengono informazioni di tipo diagnostico e sono prodotti automaticamente dal sistema, pertanto hanno nomi fissi.

Ne esiste uno per ogni modulo e sono selezionabili solo dal relativo Direttorio.

Tipicamente possono essere presenti fino a 2 file di report, questi sono:

LSTCMP            file contenente il listato dell'ultima compilazione in cui si è            scelto di  
  produr-            re il listato su disco.

REPORT            file contenente le seguenti informazioni attivate nelle **Opzioni per Diagnostica** nella **Configurazione generale modulo**:

- errori di ciclo e di sistema
- messaggi prodotti dalla istruzione VIDEO
- contenuto dei contatori usati nel programma e descritti in Configurazione

## FILE DI TESTO

I file di Testo, **tipofile T**, sono dei file generici di testo ad uso e consumo dell'operatore.

E' possibile creare uno o più file di questo tipo per ogni modulo con nomi a libera scelta dell'utente.

## FILE DEI MESSAGGI

Il file dei Messaggi, **tipofile M**, contiene tutti i messaggi previsti dall'operatore nei programmi di Setpoint, Automatici o nelle Function; ogni messaggio ha una lunghezza massima di 32 caratteri.

Il nome del file è composto dalla radice fissa **MSG** seguita dai 3 caratteri di estensione che contraddistinguono il file di lingua a cui il file si riferisce; può esserne creato uno per modulo con un massimo di 255 messaggi.

La selezione per l'edit avviene col menu selezionando il tipo di file **Messaggi**.

L'operatività di Editor carica il file associato alla lingua attiva in quell'istante. Se da un file con messaggi in italiano (es.:MSGITA.M) si vuole ad esempio creare un file per la lingua inglese, si può procere nel seguente modo: caricare il file dei messaggi in italiano, modificare i messaggi con quelli in inglese, selezionare F10-exit e poi il comando Salva o Esci e alla richiesta del nome del file sostituire a MSGITA il nome MSGENG.

Nei programmi, la visualizzazione di questi messaggi viene attivata con l'istruzione **MESSAGE**, utilizzando il numero del messaggio come indice dello stesso. L'istruzione oltre al numero di messaggio può avere come argomento anche il numero di riga su cui visualizzare il

messaggio (1÷8), se questo numero non viene indicato il messaggio viene visualizzato sulla prima riga.<sup>5</sup>

Quando durante il ciclo viene incontrata una istruzione MESSAGE, la scheda invia al PC il numero del messaggio e questo lo preleva dal file. Se non è stato creato un file di Messaggi, o il messaggio non esiste, viene visualizzato solo il numero del messaggio.

---

<sup>5</sup>. Per ulteriori dettagli sulla visualizzazione vedere al capitolo 7 la descrizione dell'operatività Automatico.

## **5. OPERATIVITA' DEL SISTEMA**

### **MENU PRINCIPALE**

Da questo menu si accede alle **operatività principali** del sistema.

Di seguito vengono sommariamente descritte le operatività legate ad ogni voce del menu, poiché le stesse sono spiegate in dettaglio nelle parti a loro dedicate.

In questa fase è possibile eseguire una **Shell** al sistema operativo **MS-DOS** premendo contemporaneamente i tasti **ALT** e **D**. Si ritorna al sistema battendo sulla tastiera il comando **EXIT**.

Le operatività, attivate ciascuna dal **tasto funzione** corrispondente, sono:

#### **F1 - ONLINE (gestione macchina)**

Consente di tornare alla Gestione macchina, cioè all'operatività **Automatico**.  
Se il sistema non è stato inizializzato si accede all'operatività **Inizializzazione sistema**.

#### **F2 - EDITOR (manipolazione file)**

Permette l'accesso all'**Editor** per la creazione o la modifica dei file utente.

#### **F3 - COMPIL (compilazione programmi e function)**

Consente l'accesso all'operatività di **Compilazione** che permette di compilare uno o più programmi o file di function.

#### **F4 - FILE (gestione programmi)**

Permette l'accesso all'operatività **Gestione programmi** preposta ad eseguire tutte le operazioni tipiche di una gestione di file, quali: copia, cancellazione, cambio nome, stampa e altre.



### **F5 - CONFIG (configurazione)**

Si accede all'operatività **Configurazione** dove vanno fornite tutte le informazioni relative alla configurazione software del sistema, quali ad esempio: numero moduli, numero stazioni, parametri generali e degli assi e selezione delle varie opzioni utilizzate.

### **F8 - LANG (cambio lingua)**

Consente di caricare i messaggi nella lingua desiderata.

### **F9 - CHANGE (cambio modulo)**

Consente di cambiare il **modulo** attivo.

### **F10 - EXIT (fine lavoro)**

Consente di effettuare il **Fine lavoro** e cioè di uscire dal sistema OS1000 e di tornare all'MS-DOS.

## **EDITOR**

### **Premessa**

L'Editor permette all'operatore di manipolare i testi o file del sistema PTP1000.

Il **massimo numero di linee di testo** introducibili per file è **2000**.

Il modo di operare è il seguente:

- si seleziona il tipo di file che si intende editare.
  
- si inserisce il nome e il commento nel caso si tratti di un nuovo file, altrimenti se non si dà il nome si accede al direttorio per la selezione del file.  
Il nome deve essere seguito dal numero stazione a cui il file si riferisce, preceduto dal carattere punto; se il numero stazione viene omissso, il sistema assume come stazione la zero.

## Tasti funzione

Durante l'edit sono disponibili i seguenti tasti funzione:

### **F1 - EDIT (menu dei tipifile)**

Permette di selezionare un altro file da editare. Viene richiesto se si vuole salvare il file che si stava editando (solo se modificato) e in questa fase è possibile:

- cambiare il commento al file.
- dare un nuovo nome al file. Questo può essere utile per creare un'ulteriore copia del file oppure, ad esempio, se si vuole copiare un file di una stazione in un'altra stazione.

### **F2 - SEARCH (ricerca/sostituzione)**

Attiva il menu che permette di selezionare uno dei seguenti comandi:

**Trova** per cercare delle sequenze o sottosequenze di caratteri nel testo.

**Sostituisci** per sostituire delle sequenze o sottosequenze di caratteri nel testo.

### **F3 - BLOCKS (azioni su blocchi di testo)**

Attiva il menu che permette la selezione dei comandi:

**Marca** serve per marcare un blocco di linee di testo che verrà poi preso automaticamente in considerazione dai comandi successivi.

La sequenza è:

Porre il cursore sulla prima linea del blocco e selezionare Marca.

Porre il cursore sull'ultima linea del blocco e selezionare Marca.

A questo punto vengono messe in evidenza le righe di testo che fanno parte del blocco.

Per disattivare il blocco occorre selezionare nuovamente Marca.

**Copia** consente di copiare una o più linee di testo duplicando così parti di esso.

**Sposta** permette di spostare parte del testo.

**Importa** permette di includere un altro file nel testo che si sta editando.

**Stampa** consente la stampa di tutto o di parte del testo in edit.

### **F4 - DELETE (cancellazione linee di testo)**

Consente la cancellazione di una o più linee di testo.

### **F6 - TSIMB (tabella simboli function)**

Consente la visualizzazione di tutti i simboli utilizzabili con le function con parametri.

### **F7 - FUN (direttorio function)**

Visualizza l'elenco delle function già scritte e compilate presenti nei vari file di Function della stazione a cui appartiene il file in edit.

E' attiva solo per i tipifile: A, S, F, D, E.

Se il file che si sta editando è del tipo A, S, F è possibile selezionare e inserire il nome della function nel testo; l'inserimento avviene dalla posizione del cursore.

### **F8 - OPTION (opzioni)**

Permette di selezionare una serie di opzioni utilizzabili durante l'edit; le opzioni sono considerate attive quando davanti alla voce nel menu c'è il carattere •.

Le opzioni sono:

**Sintassi** abilita/disabilita l'analisi sintattica su ciascuna linea di testo per quanto riguarda i file di tipo S, A, F.

**Sola lettura** abilita/disabilita il modo di operare sul testo.

Se questo è posto in **Sola lettura** si evitano alterazioni accidentali non potendolo editare.

In alcuni casi, per certi tipi di file, questa opzione viene attivata automaticamente dall'Editor.

**Tabulazione** consente di assegnare le varie posizioni a cui andrà il cursore alla premuta dei tasti **TAB** e **SHIFT+TAB**.

**Assegnamento tasti funzione** permette di assegnare ai **tasti funzione 1÷10** delle stringhe a piacere.

Questo può essere molto comodo, per esempio, se si assegnano delle istruzioni che si usano spesso. Quindi, anziché scrivere continuamente l'istruzione, si può inserirla nel testo premendo contemporaneamente il

**tasto ALT** e il **tasto funzione** a cui è stata associata l'istruzione (es.: **ALT+F2**).

**Ignora maiuscole/minuscole** abilita/disabilita, durante la ricerca (Trova), il controllo delle stringhe tenendo conto o meno della composizione della stringa da cercare.

### **F9 - CHANGE (cambio modulo)**

Consente di **cambiare il modulo attivo**.

### **F10 - EXIT (fine edit)**

Si accede alla selezione dei seguenti comandi:

**Esci** consente l'uscita dall'Editor.  
Se il testo è stato modificato, prima di uscire si ha la stessa funzionalità descritta al punto F1 - EDIT.  
Se è attiva l'**opzione Sintassi** ed il file in edit è di tipo S, A, F, D, questo viene **compilato** prima dell'archiviazione su disco.

**Salva** permette di salvare il testo senza uscire dall'edit.  
Vengono richiesti nome e commento e si può operare come descritto al riguardo al punto F1 - EDIT.

**Precedente** se era già stato editato un testo precedentemente a quello in edit permette di tornare a editarlo senza doverlo rifelezionare.  
Se il testo in edit è stato modificato, prima di caricare il testo precedente viene attivata la funzionalità descritta al punto F1 - EDIT.

### **Tasti disponibili**

Durante l'edit del testo sono disponibili dei tasti che permettono operazioni varie; questi tasti sono:

|                         |  |
|-------------------------|--|
| <b>ALT</b>              | attiva le softkey  |
| <b>TAB</b>              | tabulazione successiva   |
| <b>SHIFT+TAB</b>        | tabulazione precedente   |
| <b>SHIFT+cursore su</b> | per selezionare righe di testo partendo dalla riga puntata dal cursore in su |

|                            |  |
|----------------------------|--|
| <b>SHIFT+cursore giù</b>   | per selezionare righe di testo partendo dalla riga puntata dal cursore in giù  |
| <b>SHIFT+PAGE UP</b>       | per selezionare righe di testo partendo dalla riga puntata dal cursore verso le pagine precedenti                                  |
| <b>SHIFT+PAGE DOWN</b>     | per selezionare righe di testo partendo dalla riga puntata dal cursore verso le pagine successive                                  |
| <b>SHIFT+HOME</b>          | per selezionare righe di testo partendo dalla riga puntata dal cursore alla prima riga   |
| <b>SHIFT+END</b>           | per selezionare righe di testo partendo dalla riga puntata dal cursore all'ultima riga   |
| <b>cursori</b>             | per spostarsi su righe e colonne   |
| <b>PAGE UP e PAGE DOWN</b> | per cambiare pagina  |
| <b>HOME</b>                | per andare a inizio riga   |
| <b>END</b>                 | per andare a fine riga   |
| <b>ENTER</b>               | per andare al primo carattere della riga successiva; se INS è attivo inserisce una nuova riga.                                     |
| <b>INS</b>                 | per inserire caratteri nella riga; per disattivare questa funzionalità ripremere INS   |
| <b>DEL</b>                 | per cancellare caratteri dalla posizione del cursore in poi  |
| <b>BACK SPACE</b>          | per cancellare il carattere precedente a quello puntato dal cursore e spostare il testo di 1 colonna verso sinistra                |
| <b>ESC</b>                 | per abbandonare le modifiche fatte sulla riga ripristinando ciò che c'era in precedenza  |
| <b>CTRL+ENTER</b>          | se è attivo INS, inserisce una riga vuota sopra quella puntata dal cursore, altrimenti va al primo carattere della riga precedente |
| <b>CTRL+HOME</b>           | per andare sulla prima riga del testo  |
| <b>CTRL+END</b>            | per andare sull'ultima riga del testo  |
| <b>CTRL+Y</b>              | per cancellare la riga di testo puntata dal cursore  |

---

|                              |  |
|------------------------------|--|
| <b>CTRL+T</b>                | per cancellare una intera parola o una sequenza di spazi. Il testo viene slittato a sinistra di tante colonne quanti sono i caratteri cancellati     |
| <b>CTRL+F</b>                | per cercare le successive ricorrenze di una stringa o sottostringa già cercate in precedenza   |
| <b>CTRL+R</b>                | per sostituire una stringa o sottostringa quando il cursore è posizionato all'inizio della stringa da sostituire (si usa in congiunzione con CTRL+F) |
| <b>CTRL+U</b>                | per reinserire l'ultima riga cancellata in precedenza; la riga viene inserita sopra quella puntata dal cursore                                       |
| <b>CTRL+cursore destro</b>   | per andare all'inizio della parola successiva  |
| <b>CTRL+cursore sinistro</b> | per andare alla parola precedente; se il cursore è all'interno di una parola va all'inizio della stessa  |
| <b>CTRL+cursore su</b>       | per scorrere in giù il testo mantenendo inalterata la posizione del cursore  |
| <b>CTRL+cursore giù</b>      | per scorrere in su il testo mantenendo inalterata la posizione del cursore   |

## COMPILAZIONE PROGRAMMI E FUNCTION

La compilazione dei programmi di Setpoint e di Automatico (tipofile S e A) e delle Function (tipofile F) è indispensabile perché venga creato il corrispondente file compilato. Questo file contiene il codice oggetto che verrà trasmesso alle schede PTP200N e PLC200 per l'esecuzione.

Il **file compilato** prende lo stesso nome del file sorgente con l'aggiunta nel suffisso del carattere C.

La **compilazione viene fatta automaticamente, nell'operatività di EDITOR**, dopo aver salvato il file su disco. Questo però accade **solo se l'opzione Sintassi è abilitata**, altrimenti la compilazione non viene fatta.

Nel **direttorio** viene segnalata la presenza del file compilato dal carattere asterisco (\*) posto sotto la **colonna C**.

Anche i **file di Definizioni I/O** (tipofile D) vanno compilati, tuttavia questo viene fatto automaticamente dal sistema quando, mentre sta compilando gli altri tipi di file, si accorge che il corrispondente file di Definizioni di quella stazione (se presente) non è già stato compilato.

## 5.8 Operatività del sistema

---

Di norma quindi l'operatività Compilazione programmi non viene usata. Tuttavia può capitare di dover ricompilare tutti i file o parte di essi perché, ad esempio, è stata cambiata la risoluzione o la velocità massima di un asse, oppure ad una definizione simbolica di I/O è stato cambiato l'ingresso o l'uscita ad essa associata.

Viene presentato il direttorio dei tipifile S, A, F, D del modulo attivo. Si può selezionare uno o più file da compilare e poi viene richiesto se si vuole ottenere un **listato della compilazione su Disco o su Stampante**; per selezionare contemporaneamente tutti i file si può utilizzare il tasto **CTRL** associato col tasto + del tastierino numerico, per deselezionarli usare **CTRL** associato al tasto -.

Il listato prodotto è contenuto nel file **LSTCMP.R** e contiene solo il listato dell'ultima compilazione effettuata.

Il listato mostra, per ogni istruzione:

- indirizzo
- codice oggetto
- numero di linea
- testo sorgente
- commento (se troppo lungo viene troncato)
- errore descritto in chiaro (se presente)

Il listato può essere visualizzato o stampato con i comandi **VIEW** e **PRINT** nell'operatività Gestione programmi o visualizzato nell'operatività Editor; va selezionato tra i tipifile **REPORT**.

L'uso del listato è utile anche nei casi in cui si debbano effettuare delle verifiche sul funzionamento del programma utilizzando l'operatività di **Monitor**. Nei tipi di file compilabili è possibile utilizzare la **parola riservata PAGE** per generare un salto pagina durante la stampa del listato di compilazione.

Durante la compilazione può succedere che un errore su una linea generi un errore su un'altra linea, anche se questo secondo errore non c'è.

Un caso, ad esempio, può essere questo:

```
1          PROG
|
6          BRA    TENSION
|
12 TENSION: SNZ   STOP
|
25         END
```

Se la definizione simbolica **STOP** non viene trovata, il compilatore darà come primo errore "campo non ammesso" sulla linea 12.

Quando il compilatore abbina le etichette darà l'errore "etichetta non trovata" sulla linea 6 poiché l'etichetta TENSION si trova sulla linea che ha generato l'errore precedente.

In questo caso è sufficiente sistemare l'errore sulla linea 12 per veder scomparire anche quello sulla linea 6.

Il **menu** dell'operatività Compilazione comprende i seguenti **tasti funzione**:

### **F1 - COMPIL (compilazione)**

Fa apparire il direttorio dei file del modulo attivo per la selezione e la conseguente compilazione.

A fine compilazione se in uno o più file sono stati riscontrati degli errori apparirà la **lista delle compilazioni errate**.

### **F9 - CHANGE (cambio modulo)**

Permette di cambiare il modulo attivo.

### **F10 - EXIT (menu principale)**

Consente l'uscita dall'operatività con ritorno al Menu principale.

## **GESTIONE PROGRAMMI**

I comandi di questa operatività rendono possibile la **supervisione dei file**, generati dall'operatore, **sul Disco di lavoro**.

Le varie funzionalità, corrispondenti ciascuna ad un tasto di funzione, sono:

### **F1 - DIR (direttorio)**

**Visualizza il direttorio completo** dei file del modulo attivo. Per ogni file vengono visualizzate, su 7 colonne, le seguenti informazioni:

- 1) **tipo** file (colonna contrassegnata da **T**)
- 2) **nome** del file
- 3) **numero** della **stazione** a cui il file appartiene
- 4) **commento** al file
- 5) **data** delle **ultime modifiche** apportate al file



- 6) **lunghezza** in **byte** del **codice oggetto**; è il numero di byte che occuperà nella memoria della stazione<sup>1</sup>
- 7) **indicatore** di **compilazione** avvenuta; è il carattere asterisco (\*) ed indica se è stato prodotto il compilato e cioè il codice oggetto di quel file<sup>1</sup> (colonna contrassegnata da **C**)
- 8) **indicatore** di **protezione**; è il carattere **P** ed indica che il file è stato protetto, pertanto non può essere modificato o cancellato (colonna contrassegnata da **P**)

**Altre due informazioni** visualizzate sono:

- numero totale dei file in direttorio
- numero di byte liberi sul disco

### **F2 - RENAME (cambio nome/commento ai file)**

Permette di cambiare nome e commento ai file del modulo attivo.

Appare il menu dei tipifile, si seleziona il tipo su cui si vuole agire e appare il direttorio dei file non protetti. Si può selezionare uno o più file da rinominare e poi si passa ad eseguire il cambio nome/commento di ciascuno tra quelli selezionati.

Per ognuno appare un riquadro con le seguenti informazioni:

- **titolo** con il nome attuale del file
- **nome** è il nome attuale del file a cui si può sostituire il nuovo nome
- **commento** è il commento attuale del file a cui si può sostituire il nuovo commento

Nel caso venga cambiato nome al programma in esecuzione sulla scheda, tornando in Automatico sul video apparirà il nuovo nome.

### **F3 - COPY (copia file)**

Consente di copiare file contenuti nel direttorio di lavoro di un modulo nel direttorio di lavoro di un altro modulo; utile anche per trasferimenti tra hard-disk e floppy-disk e viceversa, quindi anche per avere copie di backup.

Nella richiesta dei direttori sorgente e destinatario viene già presentato il nome del direttorio sorgente del modulo attivo.

Se il direttorio destinatario è la radice del disco, ad esempio A, bisogna impostare: **A:\**.

Va poi selezionato il tipofile di cui si desidera fare la copia, appare così l'elenco dei file e fatta la selezione si dà inizio alla copia.

I file copiati saranno aggiunti a quelli eventualmente già presenti nel direttorio. Nel caso venga copiato un file il cui nome è già presente nel direttorio destinatario, il file che c'era

---

1. Le informazioni descritte ai punti 6 e 7 si riferiscono ai file di tipo: A, S, F, Q.

già su quest'ultimo viene sostituito da quello copiato; nel caso il destinatario sia protetto in scrittura, viene richiesta conferma prima di procedere alla copia.

I file di configurazione non possono essere copiati perché non sono presenti nel direttorio.

#### **F4 - DELETE (cancellazione file)**

Permette di cancellare uno o più file del modulo attivo.

Appare il menu dei tipifile, si seleziona il tipo su cui si vuole agire e appare il direttorio dei file non protetti. Si può selezionare uno o più file da cancellare e poi viene richiesto se si desidera confermare o meno ogni cancellazione, quindi si passa alla cancellazione.

Se si cerca di cancellare il programma che è in esecuzione sulla scheda, viene emesso un messaggio di errore dato che questa operazione non è permessa.

#### **F5 - PRINT (stampa file e direttorio)**

Permette di stampare uno o più file, il Direttorio o l'Elenco delle function del modulo attivo.

Appare il menu dei tipifile, si seleziona il tipo desiderato e appare il direttorio dei file. Si può selezionare uno o più file da stampare, quindi si passa alla stampa.

#### **F6 - VIEW (visualizzazione file)**

Permette di visualizzare un file del modulo attivo.

Appare il menu dei tipifile, si seleziona il tipo su cui si vuole agire e appare il direttorio di quei file. Si seleziona quindi il file da visualizzare.

#### **F7 - SECURE (protezione file)**

Permette di proteggere o sprotteggere un file del modulo attivo. I file protetti non possono essere nè modificati nè cancellati e sono contrassegnati dal carattere P nel direttorio.

L'accesso a questo comando è consentito solo dietro conoscenza della chiave d'accesso richiesta.

Appare il menu dei tipifile, si seleziona il tipo su cui si vuole agire e appare il direttorio di quei file. E' possibile selezionare uno o più file da proteggere o sprotteggere.

### **F8 - TIME (cambio data e ora)**

Visualizza la data e l'ora del sistema consentendo di modificarle.

### **F9 - CHANGE (cambio modulo)**

Consente di cambiare il modulo attivo.

### **F10 - EXIT (menu principale)**

Permette l'uscita dall'operatività con ritorno al Menu principale.

## **CONFIGURAZIONE**

La fase successiva alla configurazione hardware è la configurazione software e cioè l'introduzione su calcolatore di tutti i dati caratteristici del sistema quali: numero moduli, numero stazioni, numero assi e loro caratteristiche e altri dati riguardanti le opzioni che permettono un'ulteriore personalizzazione del sistema in generale.

Se il sistema è nuovo questa operatività viene presentata automaticamente all'accensione, poiché senza questi dati il sistema non sarebbe in grado di lavorare.

L'operatività di Configurazione viene richiamata dal Menu principale; l'accesso è condizionato dalla conoscenza di una parola chiave per impedire alterazioni accidentali che comprometterebbero il corretto funzionamento del sistema.

Le varie funzionalità associate ai tasti funzione sono:

### **F1 - SYSTEM (configurazione sistema)**

Permette di fornire informazioni di carattere generale relative a tutto il sistema, quali:

- **Configurazione moduli**

Per ciascun modulo equipaggiato vanno indicati:

- lo **Stato**: indica se il modulo è inserito (**On**) o meno (**Off**) nell'anello seriale.

- il **numero di Stazioni**: è il numero (**0÷15**) di schede PTP200N o PLC200 presente nel sistema. Mettendo lo spazio il modulo viene rimosso dalla configurazione.
- la **Descrizione**: è una descrizione opzionale, composta da caratteri alfanumerici, che apparirà poi in alto a destra sullo schermo nella posizione che indica il modulo attivo.
- il campo **Exist**: se è presente il carattere asterisco (\*) indica che i dati relativi al modulo sono già stati inseriti; se vuoto significa che il modulo è nuovo e pertanto va ulteriormente configurato. Questo campo è gestito automaticamente dal sistema.

**Nota:** quando un modulo viene rimosso (numero di stazioni = spazio) i dati relativi allo stesso quali ad esempio: tipi schede e loro dati relativi, non vengono cancellati, pertanto se il modulo viene nuovamente inserito in configurazione questi dati verranno riportati. Se il numero di schede in un modulo viene diminuito e poi riportato al valore iniziale (es. si passa da 3 schede a 2 e poi nuovamente a 3) le schede che prima erano state rimosse e poi reinserite vengono predisposte di tipo PTP.

#### • Opzioni di sistema

Sono informazioni relative a tutto il sistema e comprendono:

- **Linea seriale del P.C. utilizzata**: indica la linea di comunicazione seriale del PC (**1÷4**) a cui sono collegati in anello i moduli PTP1000.
- **Baud rate**: indica la velocità di comunicazione (**19200/9600**) della linea seriale. Questo dato deve essere congruente con le memorie montate sulle schede.
- **Inizializzazione automatica**: Se abilitata (**Si**), il PC all'accensione cerca di collegarsi automaticamente con i moduli evitando così l'inizializzazione da parte dell'operatore.
- **Statistiche**: indica se viene utilizzato (**Si**) o meno (**No**) il pacchetto software per le statistiche. Questo pacchetto è opzionale e viene fornito a parte.
- **Numero colonne di stampa**: permette di ottenere delle stampe conformemente al tipo di stampante che si utilizza (**80/132 colonne**).

## F2 - GENMOD (configurazione generale modulo)

Una volta configurato il sistema si passa alla configurazione generale di ciascun modulo. Il tasto F9 - CHANGE permette di selezionare il modulo interessato.

I dati da introdurre si dividono in:

### • Parametri generali

- **Batteria:** questo dato indica se le memorie RAM sulle schede sono del tipo con mantenimento permanente dei dati.  
Si imposta **Si** se le schede sono equipaggiate in tal senso; questo consente di evitare le trasmissioni di tutti i dati ad ogni avviamento del sistema.  
Si imposta **No** se le schede non sono equipaggiate in tal senso.

- **Funzionamento con PC:** questo dato è in stretta relazione con il precedente e indica se il sistema funziona con PC oppure senza.  
Si imposta **Si** se viene utilizzato il PC.  
Si imposta **No** se non viene usato il PC; questo modo viene accettato solo se al punto precedente è impostato **SI**.  
L'uso del PC è comunque indispensabile qualora si  
utilizzi l'architettura multimodulo.

- **Numero massimo stazioni:** è il numero massimo di stazioni che si intende equipaggiare nel modulo (**8/16**).  
Da questo numero dipende la possibilità di espandere l'I/O, e cioè:  
a 1) con **max 8 stazioni** è possibile equipaggiare fino a 4 schede di espansione di I/O; è il default di sistema.  
2) con **max 16 stazioni** è possibile equipaggiare una sola scheda di espansione di I/O.  
Per maggiori spiegazioni al riguardo vedere anche:  
**Numerazione dei porti di Input/Output.**

- **Timeout su errori:** è un valore in secondi (**1÷255**) che indica il tempo di attesa di un evento. E' legato a tutte le istruzioni GPL1000 che offrono la possibilità di gestire le situazioni di errore determinate dallo scadere di questo timeout.

### • Configurazione stazioni

Per ognuna delle stazioni del modulo vanno dichiarati:

- il **Tipo scheda**: PTP se è una PTP200N.  
PLC se è una PLC200.
- la **Descrizione**: è una descrizione opzionale, composta da caratteri alfanumerici, che apparirà poi in centro alla prima riga dello schema nella posizione che indica la stazione attiva.
- il campo **Exist**: se è presente il carattere asterisco (\*) indica che i dati relativi alla stazione sono già stati creati; se vuoto significa che la stazione è nuova e va configurata.

#### • Frequenze di lavoro

Questi dati sono di capitale importanza per il funzionamento di molte parti del sistema e di norma non vanno mai toccati se non in accordo con la TPA. Per questi motivi non viene fornita alcuna spiegazione riguardo ad essi. I valori di default sono:

- **Frequenza di real time**: 200 Hz.
- **Frequenza controllo assi interpolazione**: 400 Hz.
- **Frequenza interpolazione**: 400 Hz.

Se è equipaggiata la **scheda HSINT** il parametro **Frequenza interpolazione** è: **2000 Hz**.

#### • Opzioni di modulo

Queste opzioni permettono di usare o meno, conformemente alle necessità dell'utente, alcune prestazioni e operatività del sistema.

- **Selezione programmi in inizializzazione**: se abilitata (**Si**) permette di selezionare i programmi di lavoro nella fase di Inizializzazione sistema, (vedere descrizione relativa).
- **Selezione tabelle in inizializzazione**: come per il punto precedente ma riferito alle tabelle di quote.
- **Nomi programmi comuni per tutte le stazioni**: se abilitata (**Si**) indica che i programmi da eseguire su ogni stazione del modulo hanno lo stesso nome.  
Rispondendo **No** è possibile far eseguire programmi con nomi diversi su ogni stazione.

- **Nomi tabelle comuni per tutte le stazioni:** come per il punto precedente ma riferito alle tabelle di quote.
- **Tabelle di quote:** se abilitata (**Si**) indica che si intendono utilizzare tabelle di quote.
- **Function con parametri:** se abilitata (**Si**) indica che si utilizzano function a cui si vogliono passare dei parametri, pertanto l'istruzione FCALL parametrica deve essere seguita dai parametri richiesti, altrimenti si incorre in un errore di compilazione.  
Se non abilitata (**No**) l'istruzione FCALL parametrica può non essere seguita dai parametri, in tal caso i byte dei parametri vengono messi a zero dal compilatore. Questo modo di utilizzo è legato ad applicazioni in cui i parametri in precedenza posti a zero vengono poi sostituiti con i valori reali dal pacchetto software specifico.
- **Function con tabelle di parametri:** se abilitata (**Si**) indica che si utilizzano function a cui verranno passati dei parametri sotto forma di tabelle.  
**Nota:** questa particolare operatività non è attualmente disponibile poiché sarà oggetto di implementazioni successive.
- **Uso operatività di Setpoint:** se abilitata (**Si**) permette di eseguire, il Setpoint con l'operatività ad esso dedicata.  
In caso contrario il Setpoint va eseguito nel ciclo Automatico o comunque con un ciclo dedicato.
- **Uso operatività di Edit in Automatico:** se abilitata (**Si**) permette di accedere alla operatività Editor direttamente dalla operatività Automatico.
- **Visualizzazioni quote reali:** se abilitata (**Si**) nelle operatività in cui vengono visualizzate le quote, verranno visualizzate le quote reali anziché quelle teoriche.

#### • Opzioni per diagnostica

Queste opzioni possono essere attivate nei casi in cui si voglia avere un report a livello diagnostico del funzionamento della macchina. Tutti i dati descritti sotto vengono memorizzati nel file REPORT (tipofile R).

Le opzioni sono:

- **Memorizzazione errori ciclo/sistema:** se abilitata (**Si**) attiva la memorizzazione di questi tipi di errore con ulteriori informazioni riguardo l'ora, la data, il modulo e la stazione che hanno generato l'errore.
- **Memorizzazione messaggi VIDEO:** se abilitata (**Si**) attiva la memorizzazione di tutti i messaggi nella forma descritta al punto precedente.
- **Memorizzazione contatori:** se abilitata (**Si**) attiva la memorizzazione dei valori contenuti nei contatori nella forma descritta al punto precedente.
- **Descrizione contatori:** se abilitata (**Si**) permette l'accesso al punto seguente.

- **Descrizione contatori**

Questa operatività, abilitata nelle opzioni per diagnostica, permette di descrivere fino a 32 contatori del modulo, anche di schede diverse, di cui si vuole visualizzare lo stato durante l'operatività Automatico; i contatori, utilizzabili anche come timer, sono gestiti da opportune istruzioni GPL1000.

I dati da specificare sono:

- **numero Contatore:** 1) se è un numero **da 0 a 31** è il numero del contatore interessato.  
2) se **spazio**, i dati relativi alla stazione e alla descrizione, se presenti, vengono cancellati.
- **numero Stazione:** è il numero della stazione a cui appartiene il contatore.
- **Descrizione:** è la descrizione relativa al contatore che apparirà nell'operatività Automatico quando verranno visualizzati i valori nei contatori.

### **F3 - STAT (configurazione stazione)**

Dopo aver configurato il modulo a livello generale bisogna configurare ciascuna delle stazioni che lo compongono.

Il tasto F9 - Change permette di selezionare di volta in volta la stazione che si vuole configurare.

I dati da introdurre sono:



• **Parametri assi (corrente continua)**

Un asse viene considerato configurato quando tutti i dati relativi ad esso sono stati indicati. Per rimuovere un asse occorre quindi rimuovere tutti i dati.

Questi dati sono:

- **Descrizione:** è una descrizione opzionale, composta da caratteri alfanumerici, relativa all'asse che appare nella visualizzazione delle quote e degli errori. Nel caso non sia stata impostata, nelle visualizzazioni d'errore appare il nome dell'asse.
  
- **Risoluzione:** è il numero di impulsi/mm forniti dall'encoder.  
Per esempio, per un asse centesimale va impostato 100.  
Per ottenere il numero di impulsi corretto è necessario moltiplicare per 1, 2 o 4 il numero di impulsi/giro forniti dall'encoder  
compatibil- mente con la selezione effettuata con i ponticelli di moltiplicazione delle fasi sulle schede PTP200N ed ESPAS.
  
- **Velocità massima:** indica la massima velocità, in metri/minuto, impostabile da programma.  
Qualora l'unità di programmazione non sia in millimetri, per esempio in giri o gradi, va comunque indicata la velocità come:  
$$1000 * \text{unità di programmazione} / 1'$$
  
- **Accelerazione:** indica il tempo di accelerazione/decelerazione, in millisecondi, che l'asse deve impiegare per raggiungere la velocità massima e viceversa.
  
- **Finestra:** rappresenta il massimo valore di offset sul segnale analogico accettato al sistema. Viene impostato in impulsi di encoder (**1÷255**) dove:  
$$1 \text{ impulso} = 5 \text{ mV.}$$
  
E' in pratica una tolleranza entro cui il movimento dell'asse viene considerato finito. Il dato ha valore in positivo e negativo e ad esempio una finestra di 3 equivale a una tolleranza da -3 a +3 impulsi rispetto al  
punto di arrivo.
  
- **Guadagno:** determina il guadagno proporzionale dell'anello di posizione.  
Agisce come moltiplicatore dell'errore d'anello e può essere impostato con una serie di valori da **0.25 a 15**.  
La resistenza di guadagno normalmente montata sulle schede è da 20  
quindi, ad esempio, un guadagno 2 equivale a raddoppiarne il valore  
K raddoppiando altresì il guadagno.

- **Limite positivo:** è la quota massima dell'asse nella direzione positiva.  
 Equivale in pratica ad un fine corsa che viene abilitato da programma e quindi gestito dal controllo di posizione dell'asse.  
 Qualora questo limite venga superato, durante un movimento punto-punto, provoca il blocco istantaneo dell'asse. Nel caso il movimento sia interpolato vengono automaticamente bloccati anche gli altri assi in interpolazione.
  
- **Limite negativo:** è la quota massima dell'asse nella direzione negativa.  
 Vale quanto detto al punto precedente.
  
- **Feed forward p (punto a punto):** si compone di 2 dati separati dal carattere /.  
 Serve per calcolare un valore proporzionale alla velocità che viene sommato all'errore d'anello durante il controllo di posizione nei movimenti punto-punto, e può essere impostato con un range di valori:  
 $0.25 \div 15$  (dato che precede il carattere /) e  
 $1, 2, 4$  o  $8$  (dato che segue il carattere /).  
 Il primo di questi valori costituisce un moltiplicatore ed il secondo un divisore di velocità in modo da ottenere:  

$$K_v = 1^{\circ} \text{ dato} * V / 2^{\circ} \text{ dato}$$
  
- **Feed forward i (interpolazione):** Questo parametro può essere programmato in due modi diversi. Se è equipaggiata la scheda HSINT viene impostato un numero in virgola mobile che viene usato come moltiplicatore di velocità, altrimenti viene impostato con 2 dati separati dal carattere \*; il primo dato ha un range  $1 \div 32$ , il secondo ha un range  $0.25 \div 15$ .  
 Il primo di questi valori serve per calcolare la velocità media dell'asse in un intervallo di tempo pari a:  
 $1^{\circ} \text{ dato} * 1 / \text{frequenza interpolazione}$ .  
 Il secondo valore viene utilizzato come moltiplicatore della velocità così calcolata.  
 L'effetto che si ottiene dall'uso del feed forward è il miglioramento della precisione rispetto alla posizione teorica calcolata dall'algoritmo di interpolazione.

• **Parametri assi (corrente alternata)**

Un asse viene considerato configurato quando tutti i dati relativi ad esso, tranne l'uscita velocità alta/bassa, sono stati indicati. Per rimuovere un asse occorre quindi rimuovere tutti i dati.

Questi dati sono:

- **Descrizione:** è una descrizione opzionale, composta da caratteri alfanumerici, relativa all'asse che appare nella visualizzazione delle quote e degli errori. Nel caso non sia stata impostata, nelle visualizzazioni dei messaggi d'errore appare il nome dell'asse.
- **Risoluzione:** è il numero di impulsi/mm forniti dall'encoder.  
Per esempio, per un asse centesimale va impostato 100.  
Per ottenere il numero di impulsi corretto è necessario moltiplicare per 1, 2 o 4 il numero di impulsi/giro forniti dall'encoder compatibil-  
mente con la selezione effettuata con i ponticelli di moltiplicazione delle fasi sul plug Modcont.
- **Arrivo in quota positivo:** rappresenta, in numero di impulsi, la distanza dal punto di arrivo, nei movimenti positivi, alla quale viene spento il motore.
- **Arrivo in quota negativo:** rappresenta, in numero di impulsi, la distanza dal punto di arrivo, nei movimenti negativi, alla quale viene spento il motore.
- **Uscita avanti:** va indicata l'uscita, come bit+porto o simbolo, utilizzata per il pilotaggio in direzione positiva.
- **Uscita indietro:** va indicata l'uscita, come bit+porto o simbolo, utilizzata per il pilotaggio in direzione negativa.
- **Uscita velocità alta/bassa:** va indicata, solo se utilizzata, l'uscita usata per la com-  
mutazione della velocità del motore. Questa uscita viene attivata (velocità alta) se lo spostamento che il motore deve compiere, è maggiore di 4 volte il valore di Arrivo in quota (finestra di rallentamento).  
L'uscita va indicata come bit+porto o simbolo.

• **Interpolazione (solo per assi in corrente continua)**

Questi dati si riferiscono alla scheda HSINT e sono:

- **Scheda di interpolazione:** se è equipaggiata la scheda HSINT bisogna impostare **Si**.

- **Velocità massima:** è la massima velocità, in metri/minuto, che gli assi possono assumere durante l'interpolazione; la velocità nelle istruzioni di interpolazione non può essere superiore alla velocità massima dichiarata.

- **Accelerazione:** indica il tempo di accelerazione/decelerazione, in millisecondi, che gli assi che interpolano devono impiegare per raggiungere la velocità massima e viceversa.

**Nota:** i dati Velocità massima e Accelerazione sono significativi solo se è stato impostato Si al primo punto.

### • Espansioni di input/output

Vanno qui indicati il tipo e le caratteristiche delle schede che espandono l'I/O. Le fasi sono 2:

1) indicare il tipo scheda, che può essere:

- **Inoutr** supporta 24 ingressi + 24 uscite
- **Iomod** supporta fino a 6 plug di 8 ingressi (Modinp) o 8 uscite (Modout)
- **Remote** supporta fino a 6 moduli remoti di ingresso/uscita<sup>3</sup>

2) questa fase è legata alla precedente e va completata solo se si utilizzano schede di tipo **Iomod** oppure la stazione è di tipo **PLC200**.

Vanno indicati i tipi di plug<sup>3</sup> montati nelle rispettive posizioni sulla scheda; i plug possono essere :

|               |  |
|---------------|--|
| <b>Input</b>  | 8 ingressi optoisolati                                   |
| <b>Output</b> | 8 uscite a relè  |
| <b>Aninp</b>  | 2 ingressi analogici                                     |
| <b>Anout</b>  | 2 uscite analogiche                                      |
| <b>Ac</b>     | 3 ingressi per conteggio encoder                         |
| <b>Remote</b> | ricetrasmittitore verso moduli di ingresso/uscita remoti |

**Nota:** se si rimuove una scheda, anche i plug relativi vengono cancellati.

### • Aree memoria utilizzate sulla scheda

Il **numero di byte disponibili** per l'utente sulla memoria ram di ogni scheda PTP200N o PLC200 è **27648**.

E' possibile segmentare quest'area di memoria per utilizzi diversi o dedicarla ai soli programmi, i dati sono:

<sup>3</sup>. Vedere capitolo: Hardware del sistema.

- **Programmi:** è la quantità di byte disponibili per i programmi e le function. La lunghezza degli stessi, in numero byte, è visualizzata nel direttore programmi.  
In quest'area vengono scaricate le Function (se usate), il programma di Setpoint (se usato) e il programma Automatico.
- **Tabelle:** è l'area di memoria riservata alle tabelle di quote; utilizzabili solo per assi in continua.
- **Parametri function:** è l'area riservata alle tabelle di parametri function, attualmente non implementate, pertanto va posta = 0.
- **Altre:** è l'area di memoria da utilizzare con l'operatività Analizzatore; l'entità di questa area determina infatti il numero di campionamenti che possono essere eseguiti.  
Quest'area potrà essere altresì utilizzata per usi specifici che verranno sviluppati su richiesta, pertanto, se non utilizzata, va posta = 0.

### • **Function gestioni speciali**

Queste function, che possono essere indicate col numero o col nome simbolico, vengono eseguite quando si verifica la condizione a cui sono associate e cioè:

- **Errori di sistema**
- **End ciclo**
- **Emergenza**

La **function di PLC** viene invece messa in esecuzione dopo la trasmissione delle function, all'accensione del sistema, e rimane attiva fino allo spegnimento del sistema.

**Nota:** le function che si intendono utilizzare per le gestioni speciali devono essere scritte prima di dichiararle in configurazione e il file dev'essere stato compilato.

### • **Tabella delle emergenze**

Vanno qui indicate le linee di ingresso delle schede sulle quali sono cablati eventuali segnali di emergenza (Stop da campo, Fine corsa, ect.).

Le emergenze sono 16 e a ciascuna di esse è associato un sottoprogramma Firmware che viene eseguito ad ogni commutazione da stato normale a stato di emergenza (è necessario tornare in stato normale perché venga riabilitato il controllo sullo stesso ingresso). Questo consente, nel caso del fine corsa di un asse, di poter ripristinare le condizioni normali di funzionamento per mezzo dell'operatività Manuale.

I dati da inserire sono:

- **Input:** è l'ingresso a cui è collegato il segnale di emergenza; può essere descritto come bit+porto o con la definizione simbolica.

- **Stato:** è lo stato di condizione normale di funzionamento e può essere:

**Nc** se normalmente chiuso

**No** se normalmente aperto

In un sistema PTP1000 standard alcune emergenze sono già disponibili all'utente. Ad ognuna di queste è associato un **Errore di sistema** che viene segnalato al rilevamento dell'emergenza; questo messaggio d'errore viene visualizzato nel campo **Descrizione** a fianco di ogni emergenza (solo i primi 40 caratteri del messaggio).

Queste emergenze sono:

1. **emergenza generale:** blocca il movimento su tutti gli assi e termina il programma.  
Viene eseguita la function speciale di gestione Errori di sistema, se abilitata.
2. **asse X - emergenza:** come emergenza 1.
3. **asse Y - emergenza:** come emergenza 1.
4. **asse Z - emergenza:** come emergenza 1.
5. **asse W - emergenza:** come emergenza 1.
6. **asse V - emergenza:** come emergenza 1.
7. **asse X - fine corsa zero:** come emergenza 1.
8. **asse Y - fine corsa zero:** come emergenza 1.
9. **asse Z - fine corsa zero:** come emergenza 1.
10. **asse W - fine corsa zero:** come emergenza 1.
11. **asse V - fine corsa zero:** come emergenza 1.
12. **emergenza ausiliaria:** non genera Errore di sistema, quindi non termina il programma. Viene eseguita la function speciale di Emergenza, se abilitata.
13. **emergenza ausiliaria:** come per emergenza 12.
14. **emergenza ausiliaria:** come per emergenza 12.
15. **stop da campo:** sospende l'esecuzione del programma; gli assi, se in movimento, si fermano eseguendo una rampa di decelerazione.  
Equivale allo STOP su PC.
16. **start da campo:** fa partire il programma; se questo era stato sospeso da un comando di STOP gli assi ripartono per raggiungere la quota a cui erano destinati prima dello STOP.

**Nota per le emergenze 7÷11:** queste emergenze vengono rilevate solo se il flag SPEX (bit+porto 2003) è = 1 che costituisce quindi una abilitazione da programma delle emergenze; (vedere al riguardo la parte: **Numerazione dei porti di Input/Output**. Questo consente, ad esempio, di utilizzare lo stesso input di fine corsa sia per il programma di Setpoint che come emergenza dopo che questo è stato eseguito.

• **Feed rate override (solo per assi in corrente continua)**

Va indicato, solo se utilizzato, il numero del porto di input (001÷255), o la relativa definizione simbolica, a cui è collegato il potenziometro del Feed rate override che consente all'operatore di modificare la velocità degli assi che interpolano, con escursione dallo 0.4% al 100%.

Il potenziometro può essere collegato all'ingresso di un plug MODAINP, oppure all'ingresso analogico della scheda ESPAS; in quest'ultimo caso, va indicato come numero del porto 255.

• **Parametri gestione tastierina**

Questi dati, che vanno impostati se si utilizza la tastiera CAT90, sono:

- **Function:** è il numero (1÷255) o il nome simbolico della function utilizzata per la gestione del funzionamento della tastierina.
- **Primo porto:** è il numero del primo di otto porti contigui di flag utilizzati per la gestione del funzionamento della tastierina.

**F4 - PRINT (stampa configurazioni)**

Consente di stampare i dati di configurazione con 4 modalità diverse:

- **Configurazione completa:** permette di realizzare la documentazione relativa alle configurazioni di:
  - Sistema
  - Tutti i moduli
  - Tutte le stazioni di ogni modulo

vuole Prima di procedere con la stampa viene richiesto se si effettuarla su: Stampante/Disco.  
Selezionando Disco la stampa viene dirottata nel file CONFIG.PRN nel direttorio di sistema.

- **Configurazione di sistema:** stampa i dati descritti al punto F1 - SYSTEM.
- **Configurazione modulo attivo:** stampa i dati relativi al modulo attivo e a tutte le stazioni che lo compongono.
- **Configurazione stazione attiva:** stampa i dati descritti al punto F3 - STAT riguardanti la stazione attiva.

### **F9 - CHANGE (cambio modulo/stazione)**

Consente di **cambiare il modulo e la stazione attivi.**

### **F10 - EXIT (menu principale)**

Permette di tornare al Menu principale.

## **RITRASMISSIONE PARAMETRI ALLE STAZIONI**

Se vengono modificati dei parametri a sistema già avviato, appena rientrati nell'operatività Automatico viene eseguita automaticamente la ritrasmissione parametri alle stazioni interessate.

La ritrasmissione non termina l'eventuale esecuzione dei programmi, pertanto è necessario apportare le modifiche ai parametri con cautela e, se possibile, con i programmi non in esecuzione; la modifica di parametri quali: **risoluzione assi** e **velocità** comporta inoltre la ricompilazione dei programmi eseguita da operatore nell'operatività **Compilazione programmi e function.**

E' ovvio che modifiche ai parametri di equipaggiamento del sistema, quali ad esempio: numero di moduli, numero di stazioni, numero massimo stazioni, numero assi, etc., richiedono una nuova inizializzazione del sistema.



## 6. LINGUAGGIO GPL1000

### IL LINGUAGGIO

Il linguaggio GPL1000 (General Purpose Language) è il linguaggio di programmazione del sistema PTP1000.

Il GPL1000 è composto dal set di istruzioni di seguito raggruppate secondo il tipo di funzione svolta. Tutte le istruzioni possono essere usate per i programmi delle schede PTP200N, mentre vi sono dei limiti per quanto riguarda l'uso per i programmi delle schede PLC200. A fianco di ciascuna istruzione è indicato, con il carattere **p**, se può essere utilizzata per le schede PLC200.

#### • Istruzioni varie

|         |   |                                    |
|---------|---|------------------------------------|
| DELAY   | p | ritardo programmabile              |
| END     | p | fine programma                     |
| ERROR   | p | errore di ciclo                    |
| MESSAGE | p | visualizza un messaggio            |
| NOP     | p | nessuna operazione                 |
| PAUSE   | p | sospende esecuzione programma      |
| RESTART | p | esecuzione sottoprogramma firmware |
| SPECIAL | p | esecuzione speciale                |
| VIDEO   | p | visualizza il messaggio descritto  |

#### • Istruzioni per contatori/timer

|          |   |  |
|----------|---|--|
| DCOUNT   | p | decremento contatore                         |
| HTIMER   | p | sospende un conteggio                        |
| ICOUNT   | p | incremento contatore                         |
| INCOUNT  | p | copia il contenuto contatore nel/i porto/i   |
| OUTCOUNT | p | copia il contenuto del porto/i nel contatore |
| RTIMER   | p | azzerà un timer                              |
| SCOUNT   | p | carica un contatore                          |
| STIMER   | p | attiva un timer                              |
| TCOUNT   | p | test contatore                               |
| TTIMER   | p | testa un timer                               |
| VCOUNT   | p | visualizzazione contatori                    |

### • Istruzioni di salto

|        |   |  |
|--------|---|--|
| BRA    | p | salto a etichetta                      |
| CALL   | p | esecuzione sottoprogramma              |
| ENDREP | p | fine blocco da ripetere                |
| FCALL  | p | esecuzione function                    |
| FRET   | p | ritorno da function                    |
| REPEAT | p | ripetizione di un blocco di istruzioni |
| RET    | p | ritorno da sottoprogramma              |

### • Istruzioni per la multiprogrammazione

|       |   |  |
|-------|---|--|
| APROG | p | attivazione programmi paralleli              |
| EPROG | p | fine programmi/function parallele            |
| FPROG | p | attivazione function parallele               |
| RPROG | p | ripresa programmi/function parallele         |
| SPROG | p | sospensione programmi/function parallele     |
| TPROG | p | test esecuzione programmi/function parallele |
| WPROG | p | attesa fine programmi/function parallele     |

### • Istruzioni per ingressi/uscite

|        |   |   |
|--------|---|---|
| AND    | p | prodotto logico                           |
| COMP   | p | test porto                                |
| INP    | p | lettura porto                             |
| NOT    | p | negazione logica                          |
| OR     | p | somma logica                              |
| OUT    | p | scrittura di un porto                     |
| RES    | p | disattiva un bit di un porto              |
| SET    | p | attiva un bit di un porto                 |
| SHIFTL | p | rotazione porto a sinistra                |
| SHIFTR | p | rotazione porto a destra                  |
| SKZ    | p | salta se il bit del porto è uguale a 0    |
| SNZ    | p | salta se il bit del porto è diverso da 0  |
| SYNC   | p | invio sincronizzazione                    |
| VOUT   |   | scrittura uscita analogica                |
| WIZ    | p | attesa se il bit del porto è uguale a 0   |
| WNZ    | p | attesa se il bit del porto è diverso da 0 |
| WSYNC  | p | attesa sincronizzazione                   |
| XOR    | p | or esclusivo                              |

### • Istruzioni relative agli assi

|          |   |  |
|----------|---|--|
| ABS      | p | moto assoluto assi                           |
| ACCEL    |   | accelerazione asse                           |
| ADDOFS   |   | somma due correttori asse                    |
| ADJUST   |   | compensazione offset assi                    |
| CALOFS   |   | apprendimento correttore asse                |
| CHAIN    |   | concatenamento assi                          |
| FREE     |   | assi liberi                                  |
| GAIN     |   | guadagno asse                                |
| INC      | p | moto incrementale assi                       |
| LIMITOFF |   | disattiva controllo limiti assi              |
| LIMITON  |   | attiva controllo limiti assi                 |
| MOV      | p | movimento asse a quota                       |
| MOVOFs   |   | movimento asse a quota correttore            |
| NORMAL   |   | assi controllati                             |
| SERVO    |   | attiva servo error per movimenti punto punto |
| SETFF    |   | feed forward asse                            |
| SETQ     | p | imposta quota asse                           |
| SETOFS   |   | imposta correttore asse                      |
| SETPF    |   | setpoint al volo                             |
| SKE      | p | test stato asse o interpolazione             |
| STOP     | p | arresto assi                                 |
| SUBOFS   |   | sottrae due correttori asse                  |
| TOFS     |   | test correttore asse                         |
| TQ       | p | test quota asse                              |
| USEOFS   |   | richiama correttori assi                     |
| VEL      |   | velocità asse                                |
| WEND     | p | attesa stato assi o interpolazione           |

### • Istruzioni di interpolazione

|        |                           |
|--------|---------------------------|
| CIRCLE | interpolazione circolare  |
| ECNT   | fine contornatura         |
| HELIC  | interpolazione elicoidale |
| ICNT   | inizio contornatura       |
| LINEAR | interpolazione lineare    |

### • Istruzioni per le tabelle di quote

|       |                                   |
|-------|-----------------------------------|
| DPTAB | decremento puntatore tabella      |
| EQTAB | movimento assi a quota in tabella |
| IPTAB | incremento puntatore tabella      |
| MQOFS | copia una quota in un correttore  |
| MQTAB | copia una quota in tabella        |
| SPTAB | imposta puntatore tabella         |

|       |                                      |
|-------|--------------------------------------|
| TPTAB | test puntatore tabella               |
| TQTAB | test quota asse con quota in tabella |

### IL SET DI ISTRUZIONI

Nelle pagine seguenti vengono descritte una ad una tutte le istruzioni che compongono il linguaggio GPL1000.

Per ciascuna di esse viene indicato:

- il nome
- il titolo
- i vari tipi di sintassi
- la descrizione degli argomenti
- la descrizione operativa
- eventuali esempi riguardo l'uso

I simboli usati nelle spiegazioni indicano:

|                      |  |
|----------------------|--|
| <b>[,]</b>           | argomento opzionale.   |
| <b>x.x</b>           | dato numerico in virgola mobile.   |
| <b>asse</b>          | si riferisce al nome di un asse in corrente continua:<br>X Y Z W V.  |
| <b>assealternata</b> | si riferisce al nome di un asse in corrente alternata:<br>X1 Y1 Z1 W1 V1 U1 X2÷U2 X3÷U3;<br>nei nomi degli assi X1÷U1 il numero può essere omesso poichè viene assunto automaticamente dal compilatore.  |
| <b>bitporto</b>      | si riferisce a bit+porto (bit=0÷7, porto=000÷255).   |
| <b>datoporto</b>     | si riferisce a dato+porto (dato=000÷255, porto=000÷255).   |
| <b>etichetta</b>     | si riferisce all'etichetta usata per i salti e per i nomi dei programmi paralleli; può essere composta da un max di 15 caratteri alfanumerici e dai caratteri _ - < >.<br>Nel caso si tratti di un'etichetta di testa, deve essere seguita dal carattere due punti ':' e può stare indifferentemente su una riga |

con o senza istruzione, tuttavia se si utilizza l'operatività Programmazione conviene metterla su una riga da sola.

Se l'istruzione ha più argomenti, questi devono essere separati tra loro con la virgola o con lo spazio (es.: WEND X,Q).

## **ABS**

Attiva il moto dell'asse in modo assoluto

**ABS assi**

**ABS assealternata**

assi: **X Y Z W V**                      Nomi degli assi interessati.

assealternata: **X1÷U3**              Nome dell'asse.

Dopo aver incontrato questa istruzione gli spostamenti punto a punto degli **assi** indicati vengono eseguiti in modo assoluto.

Vedere anche istruzione INC.

## ACCEL

Selezione accelerazione asse

**ACCEL asse[,valore]**

**ACCEL I[,valore]**

|                          |   |
|--------------------------|---|
| asse: <b>X Y Z W V</b>   | Nome dell'asse.                         |
| interpolazione: <b>I</b> | Indica interpolazione.                  |
| valore: <b>x</b>         | Tempo di accelerazione in millisecondi. |

Dopo l'esecuzione di questa istruzione, i successivi movimenti punto-punto dell'**asse** verranno eseguiti con il nuovo **valore** di accelerazione.

La nuova accelerazione viene accettata in condizione di asse fermo (stato Q). Qualora l'asse stia eseguendo un movimento, se ne attende la fine.

Il valore del tempo di accelerazione deve essere superiore o uguale a quello indicato in Configurazione stazione; nel caso non venga impostato viene automaticamente assunto quello indicato in Configurazione stazione.

L'accelerazione indica il tempo necessario per passare da velocità 0 alla velocità massima indicata in Configurazione stazione.

Per quanto riguarda l'accelerazione di **interpolazione** valgono le stesse considerazioni fatte per il movimento punto-punto.

## **ADJUST**

Compensazione offset assi

### **ADJUST assi**

assi: **X Y Z W V**      Nomi degli assi interessati.

Esegue la misura e relativa compensazione dell'offset sul segnale di riferimento degli **assi** indicati.

La misura viene realizzata in condizione di asse fermo (stato "Q"). Qualora l'asse stia eseguendo un posizionamento si attende la fine. Il tempo di misura è circa 200 millisecondi.

Il valore di offset rimane memorizzato e può essere verificato con ulteriori istruzioni **ADJUST**.



## ADDOFS

Somma due correttori asse

**ADDOFS** *asse1,correttore1,asse2,correttore2[,asse3,correttore3]*

|                          |                                     |
|--------------------------|-------------------------------------|
| <b>asse1: X Y Z W V</b>  | Nome dell'asse sorgente 1.          |
| <b>correttore1: 0÷31</b> | Numero del correttore sorgente 1.   |
| <b>asse2: X Y Z W V</b>  | Nome dell'asse sorgente 2.          |
| <b>correttore2: 0÷31</b> | Numero del correttore sorgente 2.   |
| <b>asse3: X Y Z W V</b>  | Nome dell'asse destinatario.        |
| <b>correttore3: 0÷31</b> | Numero del correttore destinatario. |

Questa istruzione viene utilizzata per sommare due quote contenute nei correttori asse.

Se non viene indicato il correttore destinatario il risultato viene posto nel correttore2.

L'operazione può essere eseguita anche tra correttori di assi diversi; in questo caso è necessario che le risoluzioni di encoder degli assi siano le stesse.

Vedere anche istruzioni SETOFS, USEOFS, CALOFS, SUBOFS, TOFS, MOVDFS.

## AND

Prodotto logico di 2 bit

**AND op1,op2[,result]**

op1: **bitporto** Operando 1.

op2: **bitporto** Operando 2.

result: **bitporto** Risultato.

Esegue la moltiplicazione logica degli operandi 1 e 2 (**op1** e **op2**) e pone il risultato nel secondo operando.

Il risultato viene invece messo in **result** se questo è presente.

## APROG

Attiva l'esecuzione di uno o più programmi paralleli

**APROG num**

**APROG num,nomi programmi**

**APROG nomi programmi**

num: **1÷8**                      Numeri dei programmi paralleli.

nomi programmi:                Nomi dei programmi paralleli.

Questa istruzione, come la FPROG, EPROG, SPROG, RPROG, TPROG e WPROG viene usata nella multiprogrammazione o multitasking.

Permette cioè di far eseguire in parallelo più programmi (max 4) o task contemporaneamente.

Se nell'istruzione vengono indicati i numeri (**num**) dei programmi paralleli, gli eventuali nomi vengono associati automaticamente alle task disponibili. Naturalmente se viene richiesta l'attivazione di uno o più programmi paralleli, l'istruzione attende fino a quando sono disponibili tutte le task necessarie ad iniziare l'operazione, in modo da ottenere una partenza contemporanea dei programmi.

Ogni programma viene separato dagli altri usando **PROG num** oppure **PROG nomeprogramma** e terminato con la istruzione EPROG. Il programma principale (1) va terminato con l'istruzione END.

Quando si genera un solo programma (cioè non si usa il multitasking) di fatto si usa la task 1, quindi si può considerare che il PROG, che viene inserito in testa al programma, equivale a scrivere PROG 1; infatti qualora venga omesso viene assunto come PROG 1. Di fatto in caso di multiprogrammazione il 1° programma (PROG o PROG 1) è quello principale, o main, da cui vengono poi attivati gli altri programmi. Il 1° programma viene attivato automaticamente dal sistema.

Se si tenta di attivare un programma già in esecuzione, l'istruzione ne attende la fine per poterlo attivare e solo dopo si passa all'esecuzione dell'istruzione successiva.

Vedere anche istruzioni EPROG, FPROG, RPROG, SPROG, TPROG, WPROG.



## CALL

Esecuzione di un sottoprogramma

### CALL sottoprogramma

sottoprogramma:        Nome del sottoprogramma da eseguire.

L'esecuzione del programma continua alla prima istruzione del **sottoprogramma** (o subroutine) richiamato.

Il nome del sottoprogramma è l'etichetta che viene posta sulla prima linea del sottoprogramma richiamato; l'etichetta viene definita come per l'istruzione BRA.

Ogni sottoprogramma per poter tornare all'istruzione successiva alla CALL deve terminare, nel punto di uscita, con l'istruzione RET.

E' possibile innestare un massimo di 4 sottoprogrammi.

Vedere anche istruzione RET.

### Esempio:

Ricerca della tacca di 0 asse X.

```
PROG
ABS      X
VEL      X,0.5
CALL    SETPX
END
```

\*\*\* subroutine SETPX \*\*\*

```
SETPX:   X          100
         WIZ        TC0X
         SETQ       X,0
         RET
```

## CALOFS

Apprendimento quota di correzione

**CALOFS asse,correttore,segno,bitporto,stato,quota**

|                         |   |
|-------------------------|---|
| asse: <b>X Y Z W V</b>  | Nome dell'asse.   |
| correttore: <b>0÷31</b> | Numero del correttore.  |
| segno: <b>+ -</b>       | Determina il calcolo della quota di correzione.               |
| bit: <b>0÷7</b>         | Numero del bit.   |
| porto: <b>000÷255</b>   | Numero del porto.   |
| stato: <b>0 1</b>       | Indica: 0 bitporto attivo chiuso<br>1 bitporto attivo aperto. |
| quota: <b>±x.x</b>      | Quota nominale.   |

Questa istruzione consente l'apprendimento di una quota di correzione calcolata come differenza tra la **quota nominale** indicata e la quota reale dell'**asse** sulla commutazione dello **stato** del **bitporto** da disattivo ad attivo, quindi:

$$\text{quota correzione} = \text{quota reale} - \text{quota nominale}$$

La quota risultante viene memorizzata nel **correttore** indicato in base al **segno** e cioè:

|                     |   |
|---------------------|---|
| segno positivo (+): | viene memorizzata, la quota di correzione precedentemen te calcolata. |
| segno negativo (-): | viene memorizzata la quota di correzione cambiata di segno.           |

La quota di correzione viene inoltre trasferita a tutte le schede nel correttore di tutti gli assi configurati. Questo consente, qualora le risoluzioni degli assi siano identiche, di autoapprendere una correzione con il movimento di un asse ed automaticamente poter correggere anche assi di schede diverse.

Vedere anche istruzioni SETOFS e USEOFS.

## CHAIN

Concatenamento movimento assi

### CHAIN asse,assi

asse: **X Y Z W V**      Nome dell'asse master.

assi: **X Y Z W V**      Nomi degli assi slave.

Dopo l'esecuzione di questa istruzione, gli **assi** slave eseguiranno tutti i movimenti sia punto-punto che interpolati effettuati dall'**asse** master, compensando eventuali differenze di posizione con l'asse master stesso.

Il concatenamento viene effettuato con assi in quota (stato Q). Qualora uno degli assi stia eseguendo un posizionamento, se ne attende la fine. E' inoltre indispensabile che l'asse master e gli assi slave abbiano la stessa risoluzione di encoder e che all'atto del concatenamento siano alla stessa quota.

Per disabilitare il concatenamento assi è sufficiente eseguire l'istruzione NORMAL sull'asse master.

Vedere anche istruzione NORMAL.

### Esempio:

In questo esempio vengono concatenati l'asse X come master e l'asse W come slave. Come si può notare i comandi di movimento vengono dati solo all'asse X.

```

PROG

SETQ      X,0
SETQ      W,0
CHAIN    X,W           ;abilitazione concatenamento XW
VEL       X,10
X         1000,Q
CIRCLE    XY,A,6,10
WEND      XY,Q
X         0,Q
NORMAL    X           ;disabilitazione concatenamento XW
END

```

## CIRCLE

Interpolazione circolare

**CIRCLE assi,senso,velocità,raggio[,giri]**

**CIRCLE assi,senso,velocità,raggio,qf1,qf2**

|                        |  |
|------------------------|--|
| assi: <b>X Y Z W V</b> | Nomi degli assi interessati.                       |
| senso: <b>O A</b>      | Indica: O senso orario<br>A senso antiorario.      |
| velocità: <b>x.x</b>   | Valore di velocità espresso in mt/1'.              |
| raggio: <b>±x.x</b>    | Valore del raggio.                                 |
| giri: <b>0.25÷15</b>   | Numero giri.                                       |
| qf1: <b>±x.x</b>       | Quota finale asse 1 relativa al punto di partenza. |
| qf2: <b>±x.x</b>       | Quota finale asse 2 relativa al punto di partenza. |

Questa istruzione esegue una interpolazione circolare su due **assi**, scelti tra quelli configurati; gli assi interessati devono avere la stessa risoluzione di encoder.

Il valore di **velocità** indicato è la somma vettoriale delle velocità degli assi che interpolano.

Vedere anche istruzioni LINEAR, HELIC, ICNT ed ECNT.



Sono possibili due modi diversi di programmazione:

- 1) Il primo modo è relativo all'esecuzione di un cerchio completo, eventualmente ripetuto per un numero di volte indicato dal numero giri.

Come si può notare dal disegno, semplicemente cambiando l'ordine dei nomi degli assi ed il segno del **raggio** si può scegliere tra le quattro combinazioni illustrate.

E' anche possibile programmare, in questa forma semplificata, frazioni di giro come:  $1/4$ ,  $2/4$  o  $3/4$  indicando rispettivamente 0.25, 0.5 e 0.75.

- 2) Il secondo modo indicato è relativo all'esecuzione di un qualsiasi arco di cerchio. Vengono indicati semplicemente i punti di arrivo degli assi in incrementale, cioè le quote indicate rappresentano lo spostamento a partire dalla posizione attuale degli assi, ed il raggio.

Il segno del raggio positivo indica l'arco  $\leq 180^\circ$ , mentre il segno del raggio negativo indica l'arco  $> 180^\circ$ .

## COMP

Confronta il contenuto di due porti

**COMP** *porto1,operatore,porto2[,mask,etichetta]*

|                               |  |
|-------------------------------|--|
| <b>porto1: 000÷255</b>        | Numero del primo porto da confrontare.   |
| <b>operatore: &lt; = &gt;</b> | Operatore di confronto. Si possono usare anche combinazioni di operatori, ad esempio: >= o =<. |
| <b>porto2: 000÷255</b>        | Numero del secondo porto da confrontare.   |
| <b>mask: 000÷255</b>          | Maschera dei bit da confrontare.   |
| <b>etichetta:</b>             | Nome dell'etichetta a cui saltare.   |

Confronta il contenuto del **porto1** con quello del **porto2**. Se il risultato dell'operazione coincide con l'**operatore** indicato, l'istruzione successiva non viene eseguita oppure salta a **etichetta** se questa è presente.

Se la maschera (**mask**) non è indicata, viene assunto come default il valore 255, altrimenti, se presente, la maschera rappresenta la somma delle maschere (in decimale) dei bit del porto che si vuole confrontare.

## **DCOUNT**

Decrementa di 1 un contatore

### **DCOUNT cnt**

cnt: **0÷31**      Numero del contatore.

Il valore nel contatore (**cnt**) indicato viene decrementato di una unità.

Vedere anche istruzione SCOUNT.

## **DELAY**

Tempo di ritardo

### **DELAY tempo**

tempo: **x.x**      Tempo di ritardo in secondi.

Viene eseguito un ritardo pari al **tempo** indicato prima di passare ad eseguire l'istruzione successiva.

## DPTAB

Decremento puntatore tabella

**DPTAB** *asse*,*tabella*[,*etichetta*]

*asse*: **X Y Z W V**      Nome dell'asse.

*tabella*: **0÷3**      Numero tabella.

*etichetta*:      Nome dell'etichetta a cui saltare.

Decrementa di 1 il puntatore **tabella** dell'**asse** indicato saltando l'istruzione successiva se dopo il decremento il puntatore è uguale a zero. Se è presente l'**etichetta** salta a questa se il puntatore è uguale a zero.

Vedere anche istruzione IPTAB.

**ECNT**

Fine tratto in contornatura

**ECNT**

Segnala la fine di un tratto in contornatura.

Va inserita prima dell'ultima istruzione LINEAR, CIRCLE o HELIC della sequenza.

Vedere anche istruzione ICNT.

**Esempio:**

```
PROG
    ICNT                               ;inizio contornatura
    LINEAR   XYZ,10,100,0,10           ;interp. lineare 10 mt/1'
    HELIC    XYZ,A,5,10,-50,3         ;interp. elicoidale 5 mt/1'
    ECNT
    LINEAR   XYZ,10,100,0,10           ;interp. lineare 10 mt/1'
    WEND     XYZ,Q
    END
```

Come si vede dall'esempio, qualora si debba eseguire in contornatura una interpolazione su 3 assi, in tutte le interpolazioni dello stesso tratto vanno indicati tutti gli assi interessati.

Si noti anche l'istruzione di attesa dello stato Q degli assi, necessaria poiché le istruzioni d'interpolazione terminano con gli assi interessati in stato F.

## **END**

Termina l'esecuzione del programma.

## **END**

Questa istruzione va utilizzata solo nel programma principale per terminare l'esecuzione dello stesso.



## **ENDREP**

Fine del blocco di istruzioni da ripetere

## **ENDREP**

Chiude il blocco di istruzioni da ripetere dichiarato in precedenza con REPEAT.

Quando il programma incontra questa istruzione il contatore del numero di ripetizioni viene decrementato e se non è uguale a zero il blocco di istruzioni viene rieseguito partendo dall'istruzione sulla linea successiva al REPEAT. Se le ripetizioni sono terminate viene eseguita l'istruzione successiva a ENDREP.

Vedere anche istruzione REPEAT.

## EPROG

Termina l'esecuzione di uno o più programmi o function parallele

### EPROG

**EPROG num**

**EPROG nomi programmi**

**EPROG !num**

**EPROG nomi function**

num: **1÷8**                      Numeri dei programmi paralleli.

nomi programmi:                Nomi dei programmi paralleli.

!num: **!0÷255**                Numero delle function parallele.

nomi function:                Nomi delle function parallele.

Usata in caso di multiprogrammazione, fa terminare dell'esecuzione dei programmi o delle function parallele indicate.

Se usata senza argomenti termina il programma parallelo di cui l'istruzione fa parte.

Vedere anche istruzioni APROG, FPROG, RPROG, SPROG, TPROG, WPROG.

### Esempio:

Attivazione del programma 2 con fine esecuzione se trovato un fine corsa.

```
PROG                                ; programma principale
SET        PISTONE
APROG      ATTESA                ; attiva programma parallelo
X           100,Q
WPROG      ATTESA                ; attesa fine programma parallelo
END

PROG                                ; programma parallelo
WNZ        FCPIST
EPROG                                ; fine esecuzione programma parallelo
```

## EQTAB

Esecuzione quota in tabella

### EQTAB assi,tabella[,modo]

|                        |   |
|------------------------|---|
| assi: <b>X Y Z W V</b> | Nomi degli assi interessati.  |
| tabella: <b>0÷3</b>    | Numero tabella.   |
| modo: <b>0 1 2 3</b>   | Indica: 0 movimento punto-punto<br>1 movimento coordinato.<br>2 movimento punto-punto con attivazione uscita<br>3 movimento coordinato con attivazione uscita |

Esegue le quote degli **assi** e della **tabella** indicati, nei seguenti modi:

#### **modo 0**

Movimento punto-punto: la quota di ogni asse può essere eseguita sia in modo assoluto che incrementale.

Se in **modo assoluto**, alla quota in tabella viene sommato l'offset inserito per quella tabella; questo consente di poter programmare le quote riferendole ad un punto diverso da quello di Setpoint.

Se in **modo incrementale**, lo spostamento viene eseguito sommando la quota in tabella alla posizione attuale dell'asse.

Nel caso vengano indicati più assi, il movimento viene fatto partire contemporaneamente.

Valgono in generale le indicazioni date per l'istruzione MOV.

#### **modo 1**

Movimento coordinato: quando si seleziona questo modo, ogni quota in tabella rappresenta lo spostamento incrementale che viene eseguito dall'asse nell'intervallo di tempo determinato dal parametro rate:

$$\text{intervallo tempo} = \text{rate} * 1 / \text{frequenza real time}$$

Il movimento di tutti gli assi indicati realizza quindi una traiettoria formata da tanti spostamenti incrementali eseguiti in continuo.

La tabella indicata viene eseguita a partire dal puntatore settato fino al termine della tabella.

E' inoltre facilmente realizzabile un movimento coordinato tra assi di schede diverse.

Come per il movimento punto-punto, l'esecuzione parte solo se gli assi sono in stato F o Q, quindi se c'è un movimento in corso ne attende la fine.

### **modo 2**

Come modo 0 ma con attivazione uscita: l'uscita o le uscite indicate in tabella vengono attivate prima della partenza del movimento. L'indice della tabella contenente gli stati dell'uscita è rappresentato dall'indice della tabella di quote dell'asse ad essa associato.

### **modo 3**

Come modo 1 ma con attivazione uscita: l'uscita o le uscite indicate in tabella vengono attivate all'inizio di ogni intervallo di tempo, cioè quando viene letta e posta in esecuzione una nuova quota dell'asse ad essa associato.

### **Esempio:**

| <u>Master</u> |               |        | <u>Slave 1</u> |                |
|---------------|---------------|--------|----------------|----------------|
| PROG          |               |        | PROG           |                |
| SPTAB         | XY,0,1        |        | SPTAB          | XYZ,0,1        |
| SYNC          | 1             | -----> | WSYNC          |                |
| <b>EQTAB</b>  | <b>XY,0,1</b> |        | <b>EQTAB</b>   | <b>XYZ,0,1</b> |
| WSYNC         | 1             | <----- | SYNC           |                |
| END           |               |        | END            |                |

Come si nota vengono sincronizzate 2 schede: la scheda master che muove gli assi XY e la scheda slave 1 che muove gli assi XYZ.

## ERROR

Errore di ciclo

**ERROR num[,C]**

**ERROR num[,W]**

num: **1÷255**      Numero dell'errore di ciclo

C: **C**              Se presente, l'esecuzione del programma continua dopo che l'errore si è verificato.

W: **W**              Come il precedente, ma utilizzato con il pacchetto software di statistica.

Visualizza l'errore (**num**) di ciclo generato dal programma.

L'istruzione ERROR è usata in congiunzione col file degli errori di ciclo: ERRCYC.

Il numero abbinato alla ERROR è in pratica l'indice che viene usato per prelevare il messaggio contenuto nel file ERRCYC.

Se non è presente nessuna delle due opzioni (**C** o **W**), in caso di errore la task dove è stata incontrata l'istruzione si ferma e l'esecuzione può essere ripresa premendo START.

Vedere la parte: File degli Errori di ciclo.

## FCALL

Esecuzione di una function

### FCALL num

num: **0÷255**                      Numero della function da eseguire.

L'esecuzione del programma continua alla prima istruzione della function (**num**) richiamata.

Al posto del numero è possibile usare il nome simbolico della function per richiamarla; in questo caso va omessa l'istruzione FCALL poiché va usato solo il simbolo.

Ogni function per poter tornare all'istruzione successiva alla FCALL deve terminare, nel punto di uscita, con l'istruzione FRET.

E' possibile innestare un massimo di 4 function.

Vedere anche istruzione FRET e la parte relativa ai **File di function**.

### Esempio:

```

                PROG
LOOP:X  SETPOINT          ;esecuzione function di setpoint
        10,Q
        FCALL    30       ;esecuzione function 30
        Y        2,Q
        FCALL    31       ;esecuzione function 31
        BRA     LOOP
```

## FPROG

Attiva l'esecuzione di una o più function parallele

**FPROG !num**

**FPROG num,!num**

**FPROG num,nomi function**

**FPROG nomi function**

num: **1÷8**                      Numeri dei programmi paralleli.

!num: **!0÷255**                      Numeri delle function parallele.

nomi function:                      Nomi delle function parallele.

Questa istruzione, come la APROG, EPROG, SPROG, RPROG, TPROG e WPROG viene usata nella multiprogrammazione o multitasking.

Permette cioè di far eseguire in parallelo più function contemporaneamente; le function non possono essere del tipo parametrico.

Se nell'istruzione vengono indicati i numeri (**num**) delle task, i numeri o i nomi delle function vengono associati rigidamente secondo la sequenza indicata; altrimenti l'esecuzione delle function parallele verrà avviata sulle prime task disponibili. Naturalmente se viene richiesta l'attivazione di una o più function parallele, l'istruzione attende fino a quando sono disponibili tutte le task necessarie ad iniziare l'operazione, in modo da ottenere una partenza contemporanea delle function stesse.

Se si tenta di attivare una function già in esecuzione, l'istruzione ne attende la fine per poterla attivare e solo dopo si passa all'esecuzione dell'istruzione successiva.

Vedere anche istruzioni APROG, EPROG, RPROG, SPROG, TPROG, WPROG.

## FREE

Disabilita il controllo di posizione sugli assi

### FREE assi

assi: **X Y Z W V**      Nomi degli assi interessati.

Disabilita il controllo di posizione sugli **assi** indicati.

Questa istruzione può essere usata nel caso di assi di misura per rilevamento quote, oppure su assi il cui movimento può essere forzato da organi meccanici esterni che ne altererebbero la posizione. Durante il funzionamento la quota dell'asse viene aggiornata, è possibile quindi posizionare l'asse in modo assoluto dopo la riabilitazione del controllo di posizione (istruzione NORMAL).

E' inoltre possibile effettuare movimenti degli assi senza il controllo di posizione, con l'ausilio dell'istruzione VOUT, per esempio per posizionare un asse contro un riscontro meccanico.

Vedere anche istruzione NORMAL.

### Esempio:

```
PROG
ABS      X
VEL      X,10
X        1000,Q
FREE    X
VOUT     1,0.1      ;tensione di 0.1 volt su dac
WIZ      FCRISC     ;attesa riscontro
NORMAL   X
END
```



## **FRET**

Ritorno da una function

## **FRET**

Permette l'uscita da una function; l'esecuzione del programma riprende all'istruzione successiva alla FCALL che aveva richiamato la function.

Vedere anche istruzione FCALL.

## GAIN

Attivazione guadagno asse

**GAIN** asse, **valore1**[,**valore2**]

|                         |                      |
|-------------------------|----------------------|
| asse: <b>X Y Z W V</b>  | Nome dell'asse.      |
| valore1: <b>0.25÷15</b> | Guadagno asse.       |
| valore2: <b>0.25÷15</b> | Guadagno asse slave. |

Questa istruzione serve per impostare il guadagno (**valore1**) dell'anello di posizione dell'**asse** sia in movimenti punto-punto che interpolati.

Qualora sia indicato anche il secondo valore (**valore2**), questo viene utilizzato nell'anello di compensazione delle differenze di posizione con l'asse master nella modalità di concatenamento assi.

Se non viene indicato nessun valore, viene impostato il guadagno indicato in Configurazione stazione.

Vedere anche istruzioni CHAIN e SETFF.

## HELIC

Interpolazione elicoidale

**HELIC assi,senso,velocità,passo,raggio[,giri]**

**HELIC assi,senso,velocità,passo,raggio,qf1,qf2**

|                          |  |
|--------------------------|--|
| assi: <b>X Y Z W V</b>   | Nomi degli assi interessati.                       |
| senso: <b>O A</b>        | Indica: O senso orario<br>A senso antiorario.      |
| velocità: <b>x.x</b>     | Valore di velocità espresso in mt/1'.              |
| raggio: <b>±x.x</b>      | Valore del raggio.                                 |
| passo: <b>±x.x</b>       | Spostamento lineare terzo asse.                    |
| giri: <b>0.25÷15 I F</b> | Numero giri.                                       |
| qf1: <b>±x.x</b>         | Quota finale asse 1 relativa al punto di partenza. |
| qf2: <b>±x.x</b>         | Quota finale asse 2 relativa al punto di partenza. |

Questa istruzione esegue una interpolazione elicoidale su tre **assi**, scelti tra quelli configurati; gli assi interessati devono avere la stessa risoluzione di encoder.

Il valore di **velocità** indicato è la somma vettoriale delle velocità degli assi che interpolano.

I primi due assi indicati eseguono una interpolazione circolare ed il terzo asse esegue un movimento lineare della quantità specificate come spostamento incrementale, cioè a partire dalla posizione attuale e nella direzione indicata dal segno.

Qualora venga indicato accanto al **numero giri** la lettera **I** o **F** o entrambe, il giro iniziale (I) e/o finale (F) verranno eseguiti in interpolazione circolare senza avanzamento del 3° asse.

Valgono le stesse considerazioni fatte per l'interpolazione circolare sui due differenti modi di programmazione.

Vedere anche istruzioni LINEAR, CIRCLE, ICNT ed ECNT.

**Esempio:**

Come si può vedere il **passo** indicato rappresenta lo spostamento totale eseguito nei 3 **giri** di interpolazione circolare.

## HTIMER

Sospende un conteggio

### HTIMER cnt

cnt: **0÷31**      Numero del timer.

Sospende il conteggio sul timer (**cnt**) indicato; il valore nel timer rimane fermo all'istante in cui viene eseguita questa istruzione.

L'istruzione ha effetto solo se il conteggio era in corso.

Vedere anche istruzione TTIMER.

### Esempio:

Questo esempio mostra come un timer può essere usato per visualizzare il tempo ciclo.

```
          PROG
CICLO:   RTIMER    1          ;azzeramento timer
          VCOUNT  0          ;visualizzazione timer
          STIMER   1          ;partenza timer
          |
          |
          BRA      CICLO
```



## **ICOUNT**

Incrementa di 1 un contatore

### **ICOUNT cnt**

cnt: **0÷31**      Numero del contatore.

Il valore nel contatore (**cnt**) indicato viene incrementato di una unità.

Vedere anche istruzione SCOUNT.

## **INC**

Attiva il moto dell'asse in modo incrementale

**INC assi**

**INC assealternata**

assi: **X Y Z W V**                      Nomi degli assi interessati.

assealternata: **X1÷U3**              Nome dell'asse.

Dopo aver incontrato questa istruzione gli spostamenti punto a punto degli **assi** indicati vengono eseguiti in modo incrementale.

Vedere anche istruzione ABS.



## INCOUNT

Copia il contenuto del contatore nel/i porto/i

**INCOUNT** **cnt,portol[,portoh]**

cnt: **0÷31**                      Numero del contatore.

portol: **000÷255**              Numero porto destinatario della parte bassa.

portoh: **000÷255**              Numero porto destinatario della parte alta.

Copia il contenuto del contatore (**cnt**) nel seguente modo: la parte bassa nel **portol** e la parte bassa nel **portoh**, solo se specificato.

Vedere anche istruzione OUTCOUNT.

## INP

Copia il contenuto di un porto in un altro

**INP** **porto1,porto2[,mask]**

|                        |                              |
|------------------------|------------------------------|
| porto1: <b>000÷255</b> | Numero porto sorgente.       |
| porto2: <b>000÷255</b> | Numero porto destinatario.   |
| mask: <b>000÷255</b>   | Maschera dei bit da copiare. |

Copia il contenuto di **porto1** in **porto2** permettendo di trasferire solo una parte dei bit di **porto1** utilizzando l'argomento opzionale **mask** che consente di indicare i bit interessati (i bit significativi sono = 1).

Se la maschera (**mask**) non è indicata, viene assunto come default il valore 255, altrimenti, se presente, la maschera rappresenta la somma delle maschere (in decimale) dei bit del porto che si vuole confrontare.

## **IPTAB**

Incremento puntatore tabella

**IPTAB** *asse*,*tabella*[,*etichetta*]

|                        |                                    |
|------------------------|------------------------------------|
| asse: <b>X Y Z W V</b> | Nome dell'asse.                    |
| tabella: <b>0÷3</b>    | Numero tabella.                    |
| etichetta:             | Nome dell'etichetta a cui saltare. |

Incrementa di 1 il puntatore **tabella** dell'**asse** indicato saltando l'istruzione successiva se dopo l'incremento viene superato il numero di quote in tabella. Se è presente l'**etichetta** salta a questa se viene superato il numero di quote in tabella.

Vedere anche istruzione DPTAB.

## LIMITOFF

Disattivazione controllo limiti assi

### LIMITOFF assi

assi: **X Y Z W V**      Nomi degli assi interessati.

Disattiva il controllo dei limiti positivo e negativo sugli **assi** indicati.

Le quote di limiti assi vanno specificate in **Configurazione stazione**.

Vedere anche istruzione LIMITON.

## LIMITON

Attivazione controllo limiti assi

### LIMITON assi

assi: **X Y Z W V**      Nomi degli assi interessati.

Attiva il controllo dei limiti positivo e negativo sugli **assi** indicati.

Le quote di limiti assi vanno specificate in **Configurazione stazione**.

Vedere anche istruzione LIMITOFF.

## LINEAR

Interpolazione lineare

### LINEAR assi,velocità,qf1,qf2[,qf3]

|                        |                                       |
|------------------------|---------------------------------------|
| assi: <b>X Y Z W V</b> | Nomi degli assi interessati.          |
| velocità: <b>x.x</b>   | Valore di velocità espresso in mt/1'. |
| qf1: <b>±x.x</b>       | Spostamento incrementale asse 1.      |
| qf2: <b>±x.x</b>       | Spostamento incrementale asse 2.      |
| qf3: <b>±x.x</b>       | Spostamento incrementale asse 3.      |

Questa istruzione esegue una interpolazione lineare su due o tre **assi**, scelti tra quelli configurati; gli assi interessati devono avere la stessa risoluzione di encoder.

Il valore di **velocità** indicato è la somma vettoriale delle velocità degli assi che interpolano.

Le quote (**qf1**, **qf2**, **qf3**) indicate rappresentano lo spostamento a partire dalla posizione attuale dell'asse, il segno + o - determina la direzione di spostamento.

Vedere anche istruzioni CIRCLE, HELIC, ICNT ed ECNT.

## MESSAGE

Visualizza un messaggio.

**MESSAGE num[,P]**

**MESSAGE num[,riga]**

num: 1÷255      Numero del messaggio da visualizzare.

P: P              Opzione che indica la memorizzazione del messaggio su disco.

riga: 1÷8        Numero della riga su cui visualizzare il messaggio.

Permette di emettere sul video il messaggio **num** ed opzionalmente memorizzarlo (**P**).

Se è stata utilizzata l'opzione **P** e in Configurazione è stata attivata l'opzione **Memorizzazione messaggi VIDEO**, quando il PC rileva il messaggio, lo memorizza su disco nel file REPORT.

E' possibile indicare anche il numero di **riga** su cui si vuole venga visualizzato il messaggio; nel caso il numero non sia, per errore, compreso tra 1 e 8, il messaggio viene visualizzato sulla riga 1.

Vedere anche istruzione VIDEO.

## MOV

Partenza del movimento punto a punto degli assi

**MOV asse,quota[,stato]**

**asse quota[,stato]**

**asse1 quota,asse2 quota,...[,stato]**

**MOV assealternata,quota[,stato]**

**assealternata quota[,stato]**

**assealternata1 quota,assealternata2 quota,...[,stato]**

|                             |                                |
|-----------------------------|--------------------------------|
| asse: <b>X Y Z W V</b>      | Nome dell'asse.                |
| quota: $\pm x.x$            | Quota da raggiungere (finale). |
| stato: <b>A R D F Q</b>     | Stato da attendere.            |
| assealternata: <b>X1÷U3</b> | Nome dell'asse.                |

Esegue la partenza contemporanea del movimento degli assi indicati.

La **quota** assume diverso significato a seconda del tipo di movimento selezionato:

|                               |  |
|-------------------------------|--|
| <b>incrementale</b>           | rappresenta lo spostamento a partire dalla posizione attuale dell'asse (il segno + o - determina la direzione di spostamento). |
| <b>assoluto</b><br>istruzioni | rappresenta la quota riferita al SetPoint o allo zero di lavoro (vedi di correzione quote: SETOFS, USEOFS).                    |

Se viene indicato lo **stato** questo viene atteso prima di passare all'istruzione successiva; per la descrizione sugli stati vedere istruzione WEND.

Se un asse sta eseguendo uno spostamento viene atteso lo stato F prima di eseguire lo spostamento successivo sullo stesso asse.

Nel caso di assi in corrente alternata da X1 a U1 è possibile omettere il numero dell'asse scrivendo cioè solo X, Y, etc.; il sistema assume automaticamente il numero 1.



**Esempi:****1) movimento di assi:**

```
PROG
ABS      XY      ; solo se motore in continua
VEL      X,10    ; solo se motore in continua
VEL      Y,15    ; solo se motore in continua
MOV      X,20
MOV      Y,20
WEND     XY,Q
END
```

**2) movimento contemporaneo di assi**

```
PROG
VEL      X,10    ; solo se motore in continua
VEL      Y,10    ; solo se motore in continua
VEL      Z,10    ; solo se motore in continua
X 100,Y 200,Z 10,Q
END
```

**3) sincronizzazioni sullo stato di un asse**

```
PROG
VEL      X,10    ; solo se motore in continua
X        100,D   ; attesa inizio decelerazione
SET      PISTONE ; comando pistone
END
```

## MOVOfS

Partenza del movimento punto a punto alla quota contenuta in un correttore asse

### MOVOfS assi,correttore

assi: **X Y Z W V**      Nomi degli assi interessati.

correttore: **0÷31**      Numero del correttore.

Esegue la partenza contemporanea del movimento degli assi utilizzando come quota finale quella contenuta nel correttore indicato.

Valgono le stesse considerazioni fatte per l'istruzione MOV.

Vedere anche istruzioni SETOfS, USEOfS, CALOfS, ADDOfS, SUBOfS, TOFS.

### Esempio:

```
PROG
ABS      X
VEL      X,10
SETOfS   X,0,1000
SETOfS   X,1,100
ADDOfS   X,0,X,1,X2
MOVOfS   X,2      ;movimento a quota correttore 2
WEND     X,Q
END
```

## MQOFS

Copia una quota in un correttore

**MQOFS** *asse1,tabella,asse2,correttore[,num]*

**MQOFS** *asse1,tipo,asse2,correttore[,num]*

**MQOFS** *asse1,@correttore,asse2,correttore[,num]*

|                            |  |
|----------------------------|--|
| <i>asse1: X Y Z W V</i>    | Nome dell'asse sorgente.                       |
| <i>tabella: 0÷3</i>        | Numero tabella sorgente.                       |
| <i>tipo: R T</i>           | Indica: R quota reale<br>T quota teorica.      |
| <i>@correttore: @0÷@31</i> | Numero del correttore sorgente.                |
| <i>asse2: X Y Z W V</i>    | Nome dell'asse destinatario.                   |
| <i>correttore: 0÷31</i>    | Numero del correttore destinatario.            |
| <i>num: 0÷15</i>           | Numero della stazione in cui copiare la quota. |

Questa istruzione viene utilizzata per memorizzare una quota, rilevata durante il ciclo automatico, in un correttore.

Si può copiare sia la quota attuale di un asse, **Reale (R)** o **Teorica (T)**, o di una tabella di quote, eventualmente anche su correttore di assi e di stazioni diverse; in questo caso è necessario che le risoluzioni di encoder degli assi siano le stesse.

La quota sorgente da copiare viene calcolata come segue:

*tipo = R o T:* alla quota reale o teorica viene sottratto l'eventuale correttore selezionato.

*tabella sorgente:* alla quota puntata in tabella viene sommato il correttore della tabella stessa.

Vedere anche istruzioni MQTAB, SETOFS, USEOFS, CALOFS, ADDOFS, SUBOFS, TOFS, MOVDFS.

## **MQTAB**

Copia una quota in tabella

**MQTAB asse1,tabella1,asse2,tabella2[,num]**

**MQTAB asse1,tipo,asse2,tabella2[,num]**

**MQTAB asse1,@correttore,asse2,tabella2[,num]**

|                            |  |
|----------------------------|--|
| asse1: <b>X Y Z W V</b>    | Nome dell'asse sorgente.                       |
| tabella1: <b>0÷3</b>       | Numero tabella sorgente.                       |
| tipo: <b>R T</b>           | Indica: R quota reale<br>T quota teorica.      |
| @correttore: <b>@0÷@31</b> | Numero del correttore sorgente.                |
| asse2: <b>X Y Z W V</b>    | Nome dell'asse destinatario.                   |
| tabella2: <b>0÷3</b>       | Numero tabella destinataria.                   |
| num: <b>0÷15</b>           | Numero della stazione in cui copiare la quota. |

Questa istruzione viene utilizzata per memorizzare una serie di quote, rilevate durante il ciclo automatico, in una tabella di quote.

Si può copiare sia la quota attuale di un asse, **Reale (R)** o **Teorica (T)** o una intera tabella di quote, eventualmente anche su tabelle di assi e di stazioni diverse; in questo caso è necessario che le risoluzioni di encoder degli assi siano le stesse.

La quota sorgente da copiare viene calcolata come segue:

- tipo = R o T: alla quota reale o teorica viene sottratto l'eventuale correttore selezionato.
- tabella sorgente: alla quota puntata in tabella viene sommato l'offset della tabella stessa.

La quota destinataria viene memorizzata come segue:

- alla quota precedentemente calcolata viene sottratto l'offset della tabella destinataria.

Vedere anche istruzione MQOFS.

## **NOP**

Nessuna operazione

## **NOP**

Questa istruzione non dà luogo a nessuna operazione.

## **NORMAL**

Abilita il controllo di posizione sugli assi

### **NORMAL assi**

assi: **X Y Z W V**      Nomi degli assi interessati.

Abilita il controllo di posizione sugli **assi** indicati e disabilita il concatenamento assi.

All'accensione del sistema tutti gli assi configurati vengono posti automaticamente in questo stato. E' comunque consigliabile eseguire questa istruzione prima di effettuare la procedura di azzeramento degli assi, per ripristinare eventuali condizioni di emergenza.

Vedere anche istruzioni FREE e CHAIN.

### **Esempio:**

**NORMAL XZ**

## NOT

Negazione logica di un bit

### NOT op[,result]

op: **bitporto** Operando.

result: **bitporto** Risultato.

Esegue la negazione logica dell' operando.

Il risultato viene invece messo in **result** se questo è presente.

## OR

Somma logica di 2 bit

### OR op1,op2[,result]

op1: **bitporto** Operando 1.

op2: **bitporto** Operando 2.

result: **bitporto** Risultato.

Esegue la somma logica degli operandi 1 e 2 (**op1** e **op2**) e pone il risultato nel secondo operando.

Il risultato viene invece messo in **result** se questo è presente.



## OUT

Scrive un dato su un porto

### OUT **datoporto[,mask]**

|                       |   |
|-----------------------|---|
| dato: <b>000÷255</b>  | Dato da scrivere.   |
| porto: <b>000÷255</b> | Porto su cui scrivere il dato.  |
| mask: <b>000÷255</b>  | Valore decimale di una maschera di bit interessati alla scrittura.<br>I bit interessati sono = 1. |

Questa istruzione opera la scrittura di un **dato** su un **porto** con la possibilità di mascherare (**mask**) parte dei bit del porto che si vuole o meno vengano interessati dall'istruzione.

E' utilizzata soprattutto nel caso si debbano attivare o disattivare contemporaneamente più linee di un porto, ottenendo così un' esecuzione più veloce.

### Esempio:

\* File di Definizioni I/O

|      |      |                          |
|------|------|--------------------------|
| EV01 | 0008 | ; bit 0 (maschera = 1)   |
| EV02 | 1008 | ; bit 1 (maschera = 2)   |
| EV10 | 4008 | ; bit 4 (maschera = 16)  |
| EV12 | 7008 | ; bit 7 (maschera = 128) |

\* Programma

```
PROG
OUT      001008,147 ; set EV01, reset EV02,EV10,EV12
END
```

La maschera = 147 rappresenta la somma delle maschere delle singole uscite.

## OUTCOUNT

Copia il contenuto del/i porto/i nel contatore

### OUTCOUNT **cnt,portol[,portoh]**

|                        |   |
|------------------------|---|
| <b>cnt: 0÷31</b>       | Numero del contatore.                   |
| <b>portol: 000÷255</b> | Numero porto contenente la parte bassa. |
| <b>portoh: 000÷255</b> | Numero porto contenente la parte alta.  |

Copia il contenuto di **portol** e **portoh** nel contatore (**cnt**) nel seguente modo: **portol** nella parte bassa del contatore e, solo se specificato, **portoh** nella parte alta del contatore.

Se **portoh** non viene specificato la parte alta del contatore viene posta uguale a zero.

Vedere anche istruzione INCOUNT.

## **PAUSE**

Sospende l'esecuzione del programma o task

## **PAUSE**

L'esecuzione della task in cui viene incontrata questa istruzione viene sospesa. L'esecuzione riprende quando viene premuto START.

Se usata in una function di PLC l'istruzione viene ignorata.

## REPEAT

Ripete un blocco di istruzioni

**REPEAT num**

**REPEAT #cnt**

num: **1÷65535**            Numero di ripetizioni del blocco.

#cnt: **#0÷#31**            Numero del contatore dal cui prelevare il numero di ripetizioni.

Ripete l'esecuzione delle istruzioni racchiuse tra l'istruzione REPEAT e l'istruzione ENDREP per il numero (**num**) di volte indicato o in base al valore del contatore (**#cnt**).

Il massimo numero di REPEAT/ENDREP che si possono innestare è 4.

Vedere anche istruzione ENDREP.

### Esempio:

Questo esempio mostra 2 blocchi di ripetizioni:

```
PROG
REPEAT 5 ;1a ripetiz.
SET PST ;set pistone
REPEAT 10 ;2a ripetiz.
RES PST ;reset pistone
ENDREP ;fine 2a ripetiz.
ENDREP ;fine 1a ripetiz.
END
```

|  |  |  |          |          |
|--|--|--|----------|----------|
|  |  |  |          |          |
|  |  |  |          |          |
|  |  |  | BLOCCO 2 |          |
|  |  |  |          |          |
|  |  |  |          | BLOCCO 1 |
|  |  |  |          |          |

Nell'esempio il blocco 2 viene eseguito 10 volte (REPEAT 10), mentre il blocco 1 viene eseguito 5 volte (REPEAT 5); poiché il blocco 2 è innestato nel blocco 1, verrà eseguito 50 volte (10\*5).

## RES

Disattiva un bit di un porto

### RES bitporto

bit: **0÷7**                      Numero del bit da disattivare.

porto: **000÷255**              Numero del porto a cui appartiene il bit.

Disattiva il **bit** del **porto** indicato.

L'argomento bitporto può essere definito anche con un simbolo nel file di Definizioni I/O.

Vedere anche istruzione SET.

## **RESTART**

Esecuzione di un sottoprogramma firmware

### **RESTART num**

num: **0÷255**      Numero del sottoprogramma firmware da eseguire.

Viene eseguito il sottoprogramma firmware identificato dal numero (**num**) indicato.

Questa istruzione può essere usata solo se sul firmware delle schede sono installati i programmi che gestiscono queste esecuzioni speciali opzionali, installate su richiesta del cliente.

## **RET**

Ritorno da un sottoprogramma

## **RET**

Permette l'uscita da un sottoprogramma; l'esecuzione del programma riprende all'istruzione successiva alla CALL che aveva richiamato il sottoprogramma.

Vedere anche istruzione CALL.

## **RPROG**

Riprende l'esecuzione di uno o più programmi o function parallele

**RPROG num**

**RPROG nomi programmi**

**RPROG !num**

**RPROG nomi function**

num: 1÷8                    Numeri dei programmi paralleli.

nomi programmi:            Nomi dei programmi paralleli.

!num: !0÷255                Numero delle function parallele.

nomi function:             Nomi delle function parallele.

Usata in caso di multiprogrammazione, indica la ripresa dell'esecuzione dei programmi o function parallele indicate.

L'esecuzione riprende dall'istruzione successiva a quella in cui era stata fatta la sospensione con SPROG.

Vedere anche istruzioni APROG, EPROG, FPROG, SPROG, TPROG, WPROG.



## RTIMER

Azzerare un timer

### RTIMER cnt

cnt: **0÷31** Numero del timer.

Azzerare il timer (**cnt**) indicato.

Vedere anche istruzione TTIMER, HTIMER, STIMER.

### Esempio:

Questo esempio mostra come un timer può essere usato per visualizzare il tempo ciclo.

```
          PROG
CICLO:   RTIMER   1           ;azzeramento timer
          VCOUNT  0           ;visualizzazione timer
          STIMER   1           ;partenza timer
          |
          |
          BRA      CICLO
```

## SCOUNT

Carica un valore in un contatore

**SCOUNT cnt1,valore**

**SCOUNT cnt1,#cnt2**

|                        |  |
|------------------------|--|
| cnt1: <b>0÷31</b>      | Numero del contatore.                            |
| valore: <b>0÷65535</b> | Valore che verrà caricato nel contatore.         |
| #cnt2: <b>#0÷#31</b>   | Numero del contatore da cui prelevare il valore. |

Carica il contatore (**cnt1**) con il **valore** indicato o col valore contenuto nel secondo contatore (**#cnt2**).

I 32 contatori (0÷31) possono essere usati liberamente nel programma.

Questa istruzione viene usata in associazione con le istruzioni:

ICOUNT, DCOUNT, TCOUNT, VCOUNT.

Uno degli usi che si possono fare con i contatori è ad esempio quello di contapezzi, oppure di conteggio di situazioni di errore, etc.

Vedere anche la parte relativa alla **Descrizione contatori** nelle opzioni di **Configurazione generale modulo**.

## SERVO

Imposta il limite di servoerror di un asse

### SERVO asse[,quota]

|                        |                    |
|------------------------|--------------------|
| asse: <b>X Y Z W V</b> | Nome dell'asse.    |
| quota: <b>x.x</b>      | Limite servoerror. |

Questa istruzione permette di modificare il limite di servoerror<sup>2</sup> di un asse che all'accensione del sistema è posto a  $\pm 2047$  passi di encoder ( $\pm 20.47$  mm per un asse con risoluzione 0.01 mm).

Questo limite costituisce di fatto la massima differenza tra la quota teorica e la quota reale accettata dal sistema durante un movimento e dipende dalla velocità dell'asse, dalla sua taratura, etc.

Il valore di servoerror non può essere maggiore di **32768/(Gain\*4)** passi; se viene impostato un valore maggiore, questo viene limitato al massimo valore possibile in funzione del guadagno.

Qualora non venga impostata la quota verrà ripristinato il valore imposto all'accensione.

---

<sup>2</sup>. Vedere capitolo Errori di sistema.

## SET

Attiva un bit di un porto

### SET bitporto

bit: **0÷7**                      Numero del bit da attivare.

porto: **000÷255**              Numero del porto a cui appartiene il bit.

Attiva il **bit** del **porto** indicato.

L'argomento bitporto può essere definito anche con un simbolo nel file di Definizioni I/O.

Vedere anche istruzione RES.

### Esempi:

1) esempio di programma con definizioni simboliche:

\* File di Definizioni I/O

```
;Inputs
  FCPIST  0000      ;fine corsa pistone
;Outputs
  PISTONE0008      ;pistone
```

\* Programma

```
PROG
  SET      PISTONE  ;pistone avanti
  WIZ      FCPIST   ;attesa fine corsa pistone avanti
  END
```

2) la stessa parte di programma potrebbe essere scritta così:

```
PROG
  SET      0008     ;pistone avanti
  WIZ      0000     ;attesa fine corsa pistone avanti
  END
```

Risulta evidente dagli esempi che è molto più comodo e facile usare per gli I/O dei nomi simbolici, soprattutto se vengono usati parecchie volte nel programma.

**SETFF**

Attivazione feed forward asse

**SETFF asse,0**

**SETFF asse[,num1/num2]**

|                        |   |
|------------------------|---|
| asse: <b>X Y Z W V</b> | Nome dell'asse.                                   |
| num1: <b>0.25÷15</b>   | Moltiplicatore velocità per calcolo feed forward. |
| num2: <b>1 2 4 8</b>   | Divisore velocità per calcolo feed forward.       |

Questa istruzione serve per attivare il feed forward dell'**asse** nei movimenti punto-punto.

Qualora sia indicato **zero (0)**, il contributo del feed forward nell'anello di posizione viene azzerato.

Se non viene impostato nessun valore, viene assunto il valore di feed forward indicato in Configurazione stazione.

Vedere anche istruzione GAIN.

## SETOFS

Imposta il valore di un correttore asse

### SETOFS asse,correttore,quota

|                         |                        |
|-------------------------|------------------------|
| asse: <b>X Y Z W V</b>  | Nome dell'asse.        |
| correttore: <b>0÷31</b> | Numero del correttore. |
| quota: <b>±x.x</b>      | Quota correttore.      |

Carica nel **correttore** dell'**asse** indicato una **quota** che può essere utilizzata per traslare il punto zero macchina rilevato in Setpoint. Infatti una volta richiamata la quota di correzione con l'istruzione USEOFS, questa viene sommata alle quote richieste in tutti gli spostamenti assoluti dell'asse.

I correttori possono essere utilizzati anche come registri per memorizzare quote autoapprese, eseguire calcoli, movimenti dell'asse, etc...

Vedere anche istruzione USEOFS, CALOFS, ADDOFS, SUBOFS, TOFS, MOVOFs, MQOFS.

### Esempio:

In questo esempio viene caricata e richiamata (USEOFS) una quota di correzione. Sul successivo spostamento l'asse verrà posizionato alla quota richiesta (100) più la quota di correzione (1000), quindi a 1100.

```
PROG
SETOFS X,0,1000
USEOFS X,0
VEL X,10
X 100,Q
END
```

## SETPF

Azzeramento valore di un asse

### SETPF asse,bitporto,stato

|                        |   |
|------------------------|---|
| asse: <b>X Y Z W V</b> | Nome dell'asse.   |
| bit: <b>0÷7</b>        | Numero del bit.   |
| porto: <b>000÷255</b>  | Numero del porto.   |
| stato: <b>0 1</b>      | Indica: 0 bitporto attivo chiuso<br>1 bitporto attivo aperto. |

Questa istruzione speciale consente l'azzeramento di un **asse** sulla commutazione dello **stato** del **bitporto** da disattivo ad attivo.

Può essere utilizzata per eseguire uno zero pezzo o un normale Setpoint.

La particolarità di questa istruzione è che la scansione degli ingressi è effettuata ogni millisecondo anziché ogni 5, come avviene abitualmente; questo consente velocità dell'asse molto più elevate nella fase di azzeramento (es.: 6 mt/1' con precisione di 0.1 mm).

E' comunque possibile eseguire questa procedura su di un asse per volta.

## SETQ

Imposta la quota di un asse

**SETQ asse,quota**

**SETQ assealternata,quota**

asse: **X Y Z W V**                      Nome dell'asse.

quota: **±x.x**                              Quota da settare.

assealternata: **X1÷U3**                      Nome dell'asse.

Questa istruzione viene normalmente utilizzata per l'azzeramento delle **quote** in Setpoint o per eseguire uno zero pezzo. Se l'**asse** è in movimento viene fermato e posto in stato 'Q'.

### Esempio:

```
PROG
VEL      X,0.1
X        1000
WNZ      TC0X      ;attesa tacca encoder
SETQ    X,0
END
```



## SHIFTL

Rotazione di un porto a sinistra

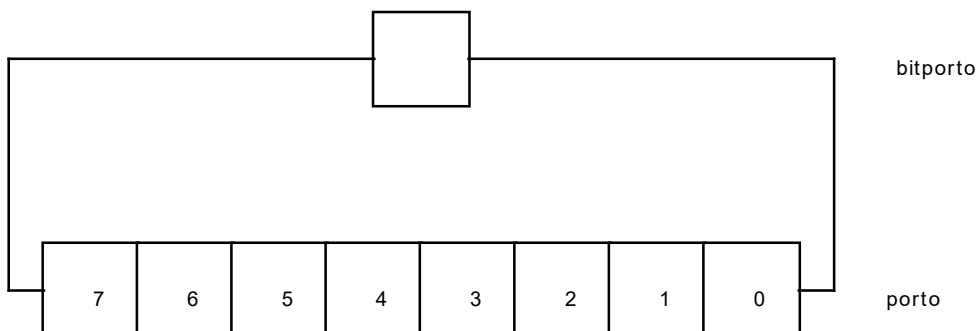
### SHIFTL **porto,bitporto**

porto: **000÷255**          Numero del porto.

bit: **0÷7**                  Numero del bit.

Esegue la rotazione a sinistra del **porto** indicato. Il **bit** del secondo argomento entra nel bit 0 del porto, il bit 7 del porto entra nel secondo argomento.

Lo schema dell'operazione è il seguente:



Vedere anche istruzione SHIFTR.

## SHIFTR

Rotazione di un porto a destra

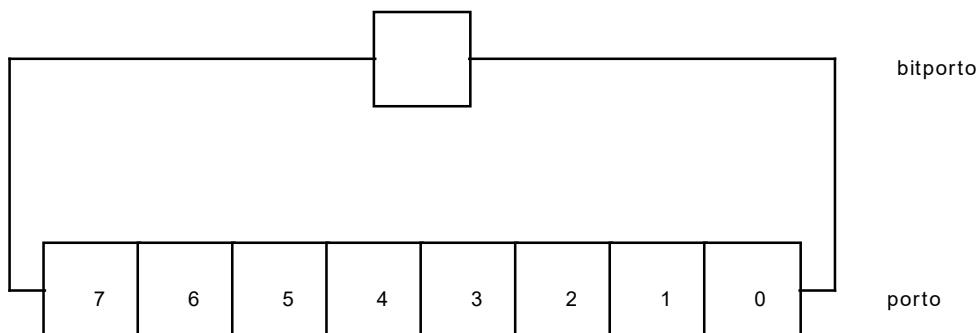
### SHIFTR **porto,bitporto**

porto: **000÷255**          Numero del porto.

bit: **0÷7**                  Numero del bit.

Esegue la rotazione a destra del **porto** indicato. Il **bit** del secondo argomento entra nel bit 7 del porto, il bit 0 del porto entra nel secondo argomento.

Lo schema dell'operazione è il seguente:



Vedere anche istruzione SHIFTL.

**SKE**

Testa lo stato o la direzione degli assi o lo stato dell'interpolazione

**SKE assi,stato[,etichetta]**

**SKE assi,direzione[,etichetta]**

**SKE assealternata,stato[,etichetta]**

**SKE assealternata,direzione[,etichetta]**

**SKE interpolazione,stato[,etichetta]**

|                             |                                    |
|-----------------------------|------------------------------------|
| assi: <b>X Y Z W V</b>      | Nomi degli assi interessati.       |
| stato: <b>A R D F Q</b>     | Stato degli assi.                  |
| direzione: + -              | Direzione assi.                    |
| etichetta:                  | Nome dell'etichetta a cui saltare. |
| assealternata: <b>X1÷U3</b> | Nome dell'asse.                    |
| interpolazione: <b>I</b>    | Indica interpolazione.             |

Confronta lo **stato** o la **direzione** degli **assi** indicati con l'argomento dell'istruzione e se uguale salta l'istruzione successiva oppure salta a **etichetta** se questa è presente.

Confronta lo **stato** dell'**interpolazione** con l'argomento dell'istruzione e se uguale salta l'istruzione successiva oppure salta a **etichetta** se questa è presente; gli stati possibili sono: **R, D, F**.

**Esempio:**

```

          PROG
          VEL      X,10
          X        1000
LOOP:     SNZ     FC,FCON
          SKE    X,Q,FINE
          BRA     LOOP

FCON:     STOP    X
          ERROR   1
          END

FINE:     VIDEO   MOVIMENTO TERMINATO
          END
```

## SKZ

Salta se il bit del porto è 0

### SKZ bitporto[,etichetta]

|                       |   |
|-----------------------|---|
| bit: <b>0÷7</b>       | Numero del bit.                           |
| porto: <b>000÷255</b> | Numero del porto a cui appartiene il bit. |
| etichetta:            | Nome dell'etichetta a cui saltare.        |

Esegue il test del **bitporto** e se questo è zero (0) salta l'istruzione successiva (vedi esempio 1).

L'argomento bitporto può essere definito anche con un simbolo nel file di Definizioni I/O.

Se è presente l'opzione **etichetta** esegue il salto alla stessa (vedi esempio 2).

Vedere anche istruzione SNZ.

### Esempi:

#### 1) test con salto dell'istruzione successiva

|         |            |                      |
|---------|------------|----------------------|
|         | PROG       |                      |
|         | SET        | PISTONE              |
|         | <b>SKZ</b> | <b>FCPIST</b>        |
|         | BRA        | ATTIVO               |
|         | VIDEO      | FINE CORSA DISATTIVO |
|         | END        |                      |
| ATTIVO: | VIDEO      | FINE CORSA ATTIVO    |
|         | END        |                      |

2) test con salto a etichetta

```
PROG
SET      PISTONE
SKZ      FCPIST,DISATT
VIDEO    FINE CORSA ATTIVO
END
```

```
DISATT:  VIDEO    FINE CORSA DISATTIVO
          END
```

## SNZ

Salta se il bit del porto è 1

### SNZ bitporto[,etichetta]

|                       |   |
|-----------------------|---|
| bit: <b>0÷7</b>       | Numero del bit.                           |
| porto: <b>000÷255</b> | Numero del porto a cui appartiene il bit. |
| etichetta:            | Nome dell'etichetta a cui saltare.        |

Esegue il test del **bitporto** e se questo è uno (1) salta l'istruzione successiva.

L'argomento bitporto può essere definito anche con un simbolo nel file di Definizioni I/O.

Se è presente l'opzione **etichetta** esegue il salto alla stessa.

Vedere anche istruzione SKZ.

## **SPECIAL**

Esecuzione di un programma firmware speciale

**SPECIAL num**

**SPECIAL #cnt**

num: **1÷255**      Numero del programma speciale da eseguire.

#cnt: **#0÷#31**      Numero del contatore da cui prelevare il numero diprogramma da eseguire.

Viene eseguito il programma speciale identificato dal numero (**num**) indicato, oppure dal contenuto del contatore (**#cnt**).

Questa istruzione può essere usata solo se sul firmware delle schede sono installati i programmi che gestiscono queste esecuzioni speciali opzionali, installate su richiesta del cliente.



## **SPROG**

Sospende l'esecuzione di uno o più programmi o function parallele

### **SPROG**

**SPROG num**

**SPROG nomi programmi**

**SPROG !num**

**SPROG nomi function**

num: **1÷8**                    Numeri dei programmi paralleli.

nomi programmi:            Nomi dei programmi paralleli.

!num: **!0÷255**                Numero delle function parallele.

nomi function:             Nomi delle function parallele.

Usata in caso di multiprogrammazione, indica la sospensione dell'esecuzione dei programmi o delle function parallele indicate.

I programmi o le function terminano l'istruzione in corso e si arrestano. L'esecuzione può essere ripresa con l'istruzione RPROG, in questo caso riprende dall'istruzione successiva all'ultima che era stata eseguita prima della sospensione.

Se usata senza argomenti sospende il programma parallelo di cui l'istruzione fa parte.

Vedere anche istruzioni APROG, EPROG, FPROG, RPROG, TPROG, WPROG.

## SPTAB

Imposta puntatore tabella

**SPTAB assi,tabella,num**

**SPTAB assi,tabella,#cnt**

|                        |  |
|------------------------|--|
| assi: <b>X Y Z W V</b> | Nomi degli assi interessati.   |
| tabella: <b>0÷3 A</b>  | Numero tabella: se 0÷3 è la tabella indicata<br>se A (All) tutte le tabelle. |
| num: <b>1÷8000</b>     | Numero quota.  |
| #cnt: <b>#0÷#31</b>    | Numero del contatore che contiene il numero della quota.                     |

Setta i puntatori **assi** della **tabella** indicata (0÷3), o di tutte (**A**) le tabelle, sulla quota **num** descritta o in base al contenuto del contatore (**#cnt**) indicato.

Il valore indicato da **num** deve essere compreso tra 1 e il numero di quote dimensionato per quella tabella. Ciò va tenuto in considerazione anche quando si utilizza un contatore (**#0÷#31**) per il caricamento del puntatore.

### Esempio:

In questo esempio si suppone di utilizzare due tabelle (0 e 1) per depalletizzare e palletizzare su 2 pallet con numero di quote differenti; il numero di quote è determinato dal dimensionamento effettuato nella Tabella di quote.

```

PROG
SCOUNT 0,0 ;reset contatore pallet prelievo
SCOUNT 1,0 ;reset contatore pallet deposito
VCOUNT
CICLOP: SPTAB XY,A,1 ;reset puntatori di tutte le tab.
VIDEO PRELIEVO
EQTAB XY,0 ;posizionamento per prelievo
Prelievo ;function operazione di prelievo
IPTAB X,0 ;incred. puntatori tab. prelievo
NOP
IPTAB Y,0 ;fine tabella ?
BRA CICLOD ;no
VIDEO PALLET PRELIEVO COMPLETO
ICOUNT 0 ;incred. cont. pallet prelievo
VCOUNT
CambioPalPre ;function cambio pallet prelievo
CICLOD: SPTAB XY,0,1 ;reset puntatori tab. prelievo
VIDEO DEPOSITO
EQTAB XY,1 ;posizionamento per deposito
Deposito ;function operazione di deposito
IPTAB X,1 ;incred. puntatori tab. deposito
NOP
IPTAB Y,1 ;fine tabella ?
BRA CICLOP ;no
VIDEO PALLET DEPOSITO COMPLETO
ICOUNT 1 ;incred. cont. pallet deposito
VCOUNT
CambioPalDep ;function cambio pallet deposito
SPTAB XY,1,1 ;reset puntatori tab. deposito
BRA CICLOP

```

## STIMER

Attiva un timer

### STIMER cnt

cnt: **0÷31** Numero del timer.

Fa partire il conteggio del timer (**cnt**) indicato che verrà incrementato ogni 0.01 secondi; il conteggio massimo è quindi 655.35 secondi.

Qualora il conteggio sia stato fermato con l'istruzione HTIMER, questo riparte dal valore corrente.

Vedere anche istruzione TTIMER, HTIMER e RTIMER.

### Esempio:

Questo esempio mostra come un timer può essere usato per visualizzare il tempo ciclo.

```
          PROG
CICLO:   VCOUNT  0          ;visualizzazione timer
          RTIMER   1          ;azzeramento timer
          STIMER  1          ;partenza timer
          |
          |
          BRA     CICLO
```

## STOP

Arresto assi o programma

### STOP

#### STOP assi

#### STOP interpolazione

#### STOP assealternata

|                             |                              |
|-----------------------------|------------------------------|
| assi: <b>X Y Z W V</b>      | Nomi degli assi interessati. |
| interpolazione: <b>I</b>    | Indica interpolazione.       |
| assealternata: <b>X1÷U3</b> | Nome dell' asse.             |

Quando è utilizzata come **STOP** l'istruzione produce lo stesso effetto che si ottiene premendo il tasto STOP in Automatico o lo **Stop da campo** e cioè: gli assi in movimento, sia punto-punto che interpolazione, rallentano e il programma si ferma. Premendo il tasto START vengono ripresi sia i movimenti degli assi che il programma.

Se usata come **STOP assi** viene interrotto il movimento sugli assi interessati eseguendo una rampa di decelerazione, ovviamente solo se c'è un movimento in corso e l'asse è in stato di Accelerazione (A) o di regime (R).

Se usata come **STOP I** l'interpolazione in corso viene sospesa e gli assi interessati vengono posti in gestione punto-punto.

### Esempio:

```

PROG
ABS      X
VEL      X,10
X        1000
WNZ      FC      ; attesa fine corsa
STOP    X      ; stop asse
END

```

## SUBOFS

Sottrae due correttori asse

**SUBOFS** **asse1,correttore1,asse2,correttore2[,asse3,correttore3]**

|                          |                                     |
|--------------------------|-------------------------------------|
| asse1: <b>X Y Z W V</b>  | Nome dell'asse sorgente 1.          |
| correttore1: <b>0÷31</b> | Numero del correttore sorgente 1.   |
| asse2: <b>X Y Z W V</b>  | Nome dell'asse sorgente 2.          |
| correttore2: <b>0÷31</b> | Numero del correttore sorgente 2.   |
| asse3: <b>X Y Z W V</b>  | Nome dell'asse destinatario.        |
| correttore3: <b>0÷31</b> | Numero del correttore destinatario. |

Questa istruzione viene utilizzata per sottrarre la quota contenuta nel **correttore2** alla quota contenuta nel **correttore1**, il risultato viene posto nel **correttore3**.

Se non viene indicato il correttore destinatario il risultato viene posto nel correttore2.

L'operazione può essere eseguita anche tra correttori di assi diversi; in questo caso è necessario che le risoluzioni di encoder degli assi siano le stesse.

Vedere anche istruzioni SETOFS, USEOFS, CALOFS, SUBOFS, TOFS, MOVOFs.

## SYNC

Invio sincronizzazione

**SYNC A**

**SYNC num**

A: **A** Sincronizzazione di tutte le stazioni (All).

num: **0÷15** Numero della stazione da sincronizzare.

Permette la sincronizzazione della stazione 0 (Master) con una (**num**), o tutte (**A**), le stazioni Slave o di una stazione Slave con la stazione Master.

Questa istruzione assume due diversi significati se utilizzata su stazione Master o Slave:

- **su stazione Master:** sincronizza la stazione Slave indicata; su quest'ultima deve essere presente la relativa istruzione WSYNC (attesa sincronismo).  
Il sincronismo funziona nel seguente modo:

SYNC (partenza stazione)

Master --> Slave

RDY (stazione pronta)

fine sincronismo

Slave --> Master

- **su stazione Slave:** sincronizza la stazione Master indicata; su quest'ultima deve essere presente la relativa istruzione WSYNC num (attesa da stazione num).  
Il sincronismo funziona nel seguente modo:

RDY (stazione pronta)

fine sincronismo Slave

Slave --> Master

fine sincronismo Master

I segnali SYNC e RDY sono definiti dal sistema (non occupano Input/Output fisici).

Se usata su stazione 0 può essere:

```

SYNC A
SYNC num
    
```

Se usata su stazioni Slave può essere:

```

SYNC (lo 0 viene preso per default)
SYNC 0
    
```

Vedere anche istruzione WSYNC.

**Esempio:**

| <u>Master</u>        | <u>Slave 1</u> |                            |
|----------------------|----------------|----------------------------|
| PROG                 | PROG           |                            |
| <b>SYNC 1</b> -----> | WSYNC          | <b>; inizio operazione</b> |
|                      | X              | 100,Q                      |
|                      | SET            | 0008                       |
|                      | X              | 0,Q                        |
| WSYNC 1 <---         | <b>SYNC</b>    | <b>; fine operazione</b>   |
| END                  | END            |                            |



## TCOUNT

Testa il contenuto di un contatore

**TCOUNT cnt1,operatore,valore[,etichetta]**

**TCOUNT cnt1,operatore,#cnt2[,etichetta]**

|                        |  |
|------------------------|--|
| cnt1: <b>0÷31</b>      | Numero del contatore.  |
| operatore: < = >       | Operatore di confronto. Si possono usare anche combinazioni di operatori, ad esempio: >= o =<. |
| valore: <b>0÷65535</b> | Valore di confronto.   |
| #cnt2: <b>#0÷#31</b>   | Numero del contatore che contiene il valore di confronto.                                      |
| etichetta:             | Nome dell'etichetta a cui saltare.   |

Confronta il contenuto del contatore 1 (**cnt1**) con il **valore** impostato o col valore contenuto nel contatore 2 (**#cnt2**). Se il risultato del confronto coincide con l'**operatore** indicato, l'istruzione successiva non viene eseguita oppure salta a **etichetta** se questa è presente.

Vedere anche istruzione SCOUNT.

## TOFS

Confronta la quota di un correttore asse

**TOFS** *asse1,correttore1,operatore,asse2,correttore2[,etichetta]*

**TOFS** *asse1,correttore1,operatore,asse2,tipo[,etichetta]*

|                               |   |
|-------------------------------|---|
| asse1: <b>X Y Z W V</b>       | Nome del primo asse.  |
| correttore1: <b>0÷31</b>      | Numero del correttore del primo asse.   |
| operatore: <b>&lt; = &gt;</b> | Operatore di confronto. Si possono usare anche combinazioni di operatori, ad esempio: <b>&gt;=</b> o <b>=&lt;</b> . |
| asse2: <b>X Y Z W V</b>       | Nome del secondo asse.  |
| correttore2: <b>0÷31</b>      | Numero del correttore del secondo asse.   |
| tipo: <b>R T</b>              | Indica: R quota reale<br>T quota teorica.   |
| etichetta:                    | Nome dell'etichetta a cui saltare.  |

Confronta il contenuto del **correttore** dell'**asse1** con il contenuto del **correttore** dell'**asse2** o con la quota **Reale (R)** o **Teorica (T)** dell'asse2.

Se il risultato del confronto coincide con l'operatore indicato, l'istruzione successiva non viene eseguita oppure salta a **etichetta** se questa è presente.

Vedere anche istruzioni SETOFS, USEOFS, CALOFS, ADDOFS, SUBOFS, MOVOFs.

## TPROG

Testa se i programmi o le function parallele sono in esecuzione

**TPROG num[,etichetta]**

**TPROG nome programma[,etichetta]**

**TPROG !num[,etichetta]**

**TPROG nome function[,etichetta]**

num: **1÷8** Numeri dei programmi paralleli.

etichetta: Nome dell'etichetta a cui saltare.

nome programma: Nome del programma parallelo.

!num: **!0÷255** Numero della function parallela.

nome function: Nome della function parallela.

Usata in caso di multiprogrammazione. Viene usata per testare se uno o più **programmi** o **function parallele** sono in esecuzione. Se questi non sono in esecuzione viene eseguito il salto a **etichetta** oppure, se questa non è indicata, l'istruzione successiva non viene eseguita.

Vedere anche istruzioni APROG, EPROG, FPROG, RPROG, SPROG, WPROG.

### Esempio:

```

          PROG
          SET      PISTONE      ;attivazione pistone
          STIMER   1
          APROG    ATTESA       ;attivazione prog. parallelo
LOOP:     TTIMER  1,>,10,ERRORE
          TPROG    ATTESA       ;prog. ATTESA in esecuzione ?
          BRA     LOOP         ;si
          END     ;no
ERRORE:  VIDEO   FINE CORSA NON RILEVATO
          END
          PROG    ATTESA       ;programma parallelo
          WIZ     FCPIST       ;attesa fine corsa pistone
          EPROG

```

## TPTAB

Testa puntatore tabella

**TPTAB** *asse*,*tabella*,*operatore*,*num*[*etichetta*]

**TPTAB** *asse*,*tabella*,*operatore*,*#cnt*[*etichetta*]

|                               |  |
|-------------------------------|--|
| asse: <b>X Y Z W V</b>        | Nome dell'asse.  |
| tabella: <b>0÷3</b>           | Numero tabella.  |
| operatore: <b>&lt; = &gt;</b> | Operatore di confronto. Si possono usare anche combinazioni, ad esempio: <b>&gt;=</b> o <b>=&lt;</b> . |
| num: <b>1÷8000</b>            | Numero quota.  |
| <i>#cnt</i> : <b>#0÷#31</b>   | Numero del contatore che contiene il numero della quota.   |
| etichetta:                    | Nome dell'etichetta a cui saltare.   |

Confronta il puntatore **tabella** con il valore (**num**) impostato o col contenuto del contatore (**#cnt**). Se il risultato del confronto coincide con l'**operatore** indicato, l'istruzione successiva non viene eseguita, oppure salta a **etichetta** se questa è presente.

**TQ**

Testa la quota di un asse

**TQ asse,operatore,quota,tipo[,etichetta]**

**TQ assealternata,operatore,quota,R[,etichetta]**

|                               |  |
|-------------------------------|--|
| asse: <b>X Y Z W V</b>        | Nome dell'asse.  |
| operatore: <b>&lt; = &gt;</b> | Operatore di confronto. Si possono usare anche combinazioni, ad esempio: <b>&gt;=</b> o <b>=&lt;</b> . |
| quota: <b>±x.x</b>            | Quota da confrontare.  |
| tipo: <b>R</b> o <b>T</b>     | Tipo di quota: <b>R</b> indica quota asse reale<br><b>T</b> indica quota asse teorica.                 |
| etichetta:                    | Nome dell'etichetta a cui saltare.   |
| assealternata: <b>X1÷U3</b>   | Nome dell'asse.  |

Confronta la **quota Reale (R)** o **Teorica (T)** dell'**asse** con la quota indicata. Se il risultato del confronto coincide con l'**operatore** indicato, l'istruzione successiva non viene eseguita oppure salta a **etichetta** se questa è presente.

Il confronto viene eseguito considerando le quote dal minimo negativo al massimo positivo, ad esempio la quota +10 è maggiore di -10000.

Per quanto riguarda gli assi in alternata il test viene sempre eseguito sulla quota reale (R) letta dall'encoder; il tipo di quota impostato viene pertanto ignorato.

**Esempio:**

In questo esempio il movimento dell'asse Y viene eseguito solo quando la quota teorica dell'asse X è **>=50**.

```

          PROG
          SETQ      X,0
          VEL       X,10
          X         1000
LOOP:  TQ        X,<,50,T,LOOP    ; attende se quota teorica < 50
          Y         100,Q

```

END

## TQTAB

Confronta la quota di un asse con la quota in tabella

**TQTAB asse,tabella,operatore,tipo[,etichetta]**

|                               |  |
|-------------------------------|--|
| asse: <b>X Y Z W V</b>        | Nome dell'asse.  |
| tabella: <b>0÷3</b>           | Numero tabella.  |
| operatore: <b>&lt; = &gt;</b> | Operatore di confronto. Si possono usare anche combinazioni, ad esempio: <b>&gt;=</b> o <b>=&lt;</b> . |
| tipo: <b>R T</b>              | Indica: R quota reale<br>T quota teorica.  |
| etichetta:                    | Nome dell'etichetta a cui saltare.   |

Confronta la quota puntata della **tabella** specificata con la quota **Reale (R)** o **Teorica (T)** dell'asse. Se il risultato del confronto coincide con l'**operatore** indicato, l'istruzione successiva non viene eseguita, oppure salta a **etichetta** se questa è presente.

Il confronto viene eseguito considerando le quote dal minimo negativo al massimo positivo, ad esempio la quota +10 è maggiore di -10000.

## TTIMER

Testa il valore di un timer

**TTIMER** *cnt,operatore,tempo[,etichetta]*

|                                      |   |
|--------------------------------------|---|
| <b>cnt:</b> <b>0÷31</b>              | Numero del timer.   |
| <b>operatore:</b> <b>&lt; = &gt;</b> | Operatore di confronto. Si possono usare anche combinazioni di operatori, ad esempio: <b>&gt;=</b> o <b>=&lt;</b> . |
| <b>tempo:</b> <b>x.x</b>             | Tempo in secondi.   |
| <b>etichetta:</b>                    | Nome dell'etichetta a cui saltare.  |

Confronta il contenuto del timer (**cnt**) con il **tempo** indicato. Se il risultato del confronto coincide con l'**operatore** indicato, l'istruzione successiva non viene eseguita, oppure salta a **etichetta** se questa è presente.

Vedere anche istruzioni STIMER, HTIMER, RTIMER



## USEOFS

Richiamo di un correttore asse

### USEOFS assi,correttore

assi: **X Y Z W V**      Nomi degli assi interessati.

correttore: **0÷31**      Numero del correttore.

Richiama uno dei **correttori** precedentemente impostati dall'istruzione SETOFS.

Infatti una volta richiamata la quota di correzione con l'istruzione USEOFS, questa viene sommata alle quote richieste in tutti gli spostamenti assoluti dell'asse.

Vedere anche istruzione SETOFS, CALOFS, ADDOFS, SUBOFS, TOFS, MOVOFs.



**VEL**

Selezione velocità asse

**VEL asse[,valore]**

**VEL interpolazione[,valore]**

|                          |                                       |
|--------------------------|---------------------------------------|
| asse: <b>X Y Z W V</b>   | Nome dell'asse.                       |
| valore: <b>x.x</b>       | Valore di velocità espresso in mt/l'. |
| interpolazione: <b>I</b> | Indica interpolazione.                |

Dopo aver incontrato questa istruzione il movimento dell'**asse** viene effettuato al nuovo **valore** di velocità.

Il valore di velocità impostato deve essere inferiore o uguale a quello di velocità massima indicato in Configurazione stazione; nel caso non venga impostato viene automaticamente assunto quello di velocità massima.

La velocità impostata indica il valore di regime al quale si porta l'asse dopo la rampa di accelerazione. E' comunque possibile durante uno spostamento variare questo valore per ottenere un profilo di velocità diverso dal normale profilo trapezoidale.

Se è equipaggiata la **scheda HSINT** questa istruzione serve per limitare la velocità di interpolazione di un tratto eseguito in contornatura. Anche in questo caso qualora non venga specificato il **valore** viene assunta la velocità massima impostata in Configurazione stazione.

**Esempi:**

1) movimento punto-punto normale

velocità indicata

0

accelerazione

decelerazione

```
PROG
VEL      X,10
X        100,Q
END
```

**2) movimento punto-punto con variazione del profilo di velocità**

velocità a

velocità b

0 accelerazione

FCRAL

```
PROG
VEL      X,10      ; velocità a
X        100
WNZ      FCRAL    ; attesa fine corsa rallentamento
VEL      X,5       ; velocità b
WEND     X,Q       ; attesa arrivo in quota
END
```

**3) movimento interpolato**

```
PROG
VEL      I,1
ICNT
CIRCLE   XY,0,6,100,0.5
ECNT
LINEAR   XY,6,0,100
END
```

## VIDEO

Visualizza un messaggio dell'operatore

### VIDEO messaggio[,P]

**messaggio:** Messaggio da visualizzare.  
La lunghezza massima del messaggio è di 32 caratteri alfanumerici.

**P: P** Opzione che indica la memorizzazione del messaggio su disco.

Permette di emettere sul video del PC un **messaggio** posto in qualsiasi punto del programma ed opzionalmente memorizzarlo (**P**).

Se il messaggio è più lungo di 32 caratteri, quelli in eccedenza vengono troncati dal compilatore.

Se l'istruzione VIDEO non è seguita dal messaggio verrà cancellata la zona di video relativa.

Se è stata utilizzata l'opzione **P** e in Configurazione è stata attivata l'opzione **Memorizzazione messaggi VIDEO**, quando il PC rileva il messaggio, lo memorizza su disco nel file REPORT.

Vedere anche istruzione MESSAGE.

### Esempio:

|                |                          |                                  |
|----------------|--------------------------|----------------------------------|
| PROG           |                          |                                  |
| <b>VIDEO</b>   | <b>SET PISTONE</b>       | <b>;messaggio</b>                |
| SET            | PST                      | ;setta pistone                   |
| <b>VIDEO</b>   | <b>ATTESA FINE CORSA</b> | <b>;messaggio</b>                |
| WIZ            | FCPST,MSGERR             | ;attesa fine corsa               |
| <b>VIDEO</b>   |                          | <b>;cancella messaggio</b>       |
| END            |                          | ;fine programma                  |
| <b>MSGERR:</b> | <b>VIDEO</b>             | <b>FINE CORSA NON RILEVATO,P</b> |
|                | END                      |                                  |

## VOUT

Attiva una tensione su un'uscita analogica

### VOUT **dac,tensione**

dac: **1÷6**                                      Numero dell'uscita analogica.

tensione: **-10.000÷10.000**                      Valore di tensione in Volt.

Permette di attivare una **tensione** analogica su uno dei 6 convertitori digitale/analogico (**dac**), che sono da considerare nel modo seguente:

- 1 uscita analogica asse X
- 2 uscita analogica asse Y
- 3 uscita analogica asse Z
- 4 uscita analogica asse W
- 5 uscita analogica asse V
- 6 uscita analogica ausiliaria (U)

## WEND

Attesa stato assi o stato dell'interpolazione

**WEND assi,stato**

**WEND assealternata,stato**

**WEND interpolazione,stato**

|                             |                              |
|-----------------------------|------------------------------|
| assi: <b>X Y Z W V</b>      | Nomi degli assi interessati. |
| stato: <b>A R D F Q</b>     | Stato da attendere.          |
| assealternata: <b>X1÷U3</b> | Nome dell'asse.              |
| interpolazione: <b>I</b>    | Indica interpolazione.       |

Attende che gli **assi** arrivino nello **stato** indicato prima di passare ad eseguire l'istruzione successiva.

Gli stati possibili sono:

- A** - accelerazione (solo per assi in continua)
- R** - regime
- D** - decelerazione
- F** - movimento teorico concluso
- Q** - asse in quota

Nell'istante in cui viene eseguita una istruzione di movimento (MOV) l'asse viene posto in stato '**A**', quindi al termine della rampa di accelerazione in stato '**R**', a meno che lo spostamento sia inferiore allo spazio necessario per concludere la rampa, in tal caso l'asse passerà direttamente in stato '**D**' ed inizierà la rampa di decelerazione. Al termine di quest'ultima il movimento teorico (comando) si conclude e l'asse viene posto in stato '**F**'. Da questo momento il controllo attende che la posizione reale dell'asse diventi uguale alla posizione teorica e quindi pone l'asse in stato '**Q**'.

Se usata per attendere uno stato d'interpolazione, gli stati possibili sono: **R,D,F**.

## WIZ

Attesa se il bit del porto è 0

**WIZ bitporto**

**WIZ bitporto,etichetta**

**WIZ bitporto,num[,C]**

**WIZ bitporto,num[,W]**

|                       |  |
|-----------------------|--|
| bit: <b>0÷7</b>       | Numero del bit.  |
| porto: <b>000÷255</b> | Numero del porto a cui appartiene il bit.  |
| etichetta:            | Nome dell'etichetta a cui saltare.   |
| num: <b>1÷255</b>     | Numero dell'errore di ciclo.   |
| C: <b>C</b>           | Se presente l'esecuzione del programma continua dopo che l'errore si è verificato. |
| W: <b>W</b>           | Come il precedente, ma utilizzato con il pacchetto software di statistica.         |

Se il **bit** del **porto** indicato è zero (0) attende che diventi uno (1) prima di passare all'istruzione successiva.

L'argomento bitporto può essere definito anche con un simbolo nel file di Definizioni I/O.

Se è presente l'argomento **etichetta** l'esecuzione di questa istruzione fa partire il Timeout stabilito in Configurazione generale modulo, allo scadere del quale viene eseguito automaticamente il salto all'etichetta (vedi esempio 1).

Se è presente l'argomento **num** l'esecuzione di questa istruzione fa partire il Timeout stabilito in Configurazione generale modulo, allo scadere del quale viene inviato il messaggio di errore di ciclo sul video del PC (vedi esempio 2).

Se non è presente nessuna delle due opzioni (**C** o **W**), in caso di errore il programma parallelo dove è stata incontrata l'istruzione si ferma e l'esecuzione può essere ripresa premendo START; l'esecuzione riprende dall'istruzione WIZ.



Se è presente una delle due opzioni (**C** o **W**), in caso di errore il programma parallelo dove è stata incontrata l'istruzione invia il messaggio di errore di ciclo al PC e riprende automaticamente l'esecuzione dall'istruzione WIZ.

Vedere anche: istruzione RES e File errori di ciclo (cap. 4).

### **Esempi:**

#### **1) attesa con salto a etichetta**

```
      PROG
      SET      PISTONE
      WIZ      FCPIST,ERR1      ; attesa fine corsa pistone
      END

ERR1:  VIDEO      FINE CORSA PISTONE NON RILEVATO
      END
```

#### **2) attesa con generazione di errore di ciclo**

```
      PROG
      SET      PISTONE
      WIZ      FCPIST,25      ; attesa fine corsa pistone
      END
```

## WNZ

Attesa se il bit del porto è 1

**WNZ bitporto**

**WNZ bitporto,etichetta**

**WNZ bitporto,num[,C]**

**WNZ bitporto,num[,W]**

|                       |  |
|-----------------------|--|
| bit: <b>0÷7</b>       | Numero del bit.  |
| porto: <b>000÷255</b> | Numero del porto a cui appartiene il bit.  |
| etichetta:            | Nome dell'etichetta a cui saltare.   |
| num: <b>1÷255</b>     | Numero dell'errore di ciclo.   |
| C: <b>C</b>           | Se presente l'esecuzione del programma continua dopo che l'errore si è verificato. |
| W: <b>W</b>           | Come il precedente, ma utilizzato con il pacchetto software di statistica.         |

Se il **bit** del **porto** indicato è uno (1) attende che diventi zero (0) prima di passare all'istruzione successiva.

L'argomento bitporto può essere definito anche con un simbolo nel file di Definizioni I/O.

Se è presente l'argomento **etichetta** l'esecuzione di questa istruzione fa partire il Timeout stabilito in Configurazione generale modulo, allo scadere del quale viene eseguito automaticamente il salto all'etichetta.

Se è presente l'argomento **num** l'esecuzione di questa istruzione fa partire il Timeout stabilito in Configurazione generale modulo, allo scadere del quale viene inviato il messaggio di errore di ciclo sul video del PC.

Se non è presente nessuna delle due opzioni (**C** o **W**), in caso di errore, la task dove è stata incontrata l'istruzione si ferma e l'esecuzione può essere ripresa premendo START; l'esecuzione riprende dall'istruzione WNZ.

Se è presente una delle due opzioni (**C** o **W**), in caso di errore, la task dove è stata incontrata l'istruzione invia il messaggio di errore di ciclo al PC e riprende automaticamente l'esecuzione dall'istruzione WNZ.

Vedere anche: istruzione SET e File errori di ciclo (cap. 4).

## **WPROG**

Attende la fine di uno o più programmi o function parallele

**WPROG num**

**WPROG nomi programmi**

**WPROG !num**

**WPROG nomi function**

num: **1÷8**            Numeri dei programmi paralleli.

nomi programmi: Nomi dei programmi paralleli.

!num: **!0÷255**            Numero delle function parallele.

nomi function:            Nomi delle function parallele.

Usata in caso di multiprogrammazione, indica l'attesa della fine dell'esecuzione dei programmi o function parallele indicate per poter passare all'istruzione successiva.

Vedere anche istruzioni APROG, EPROG, FPROG, RPROG, SPROG, TPROG.

## WSYNC

Attesa sincronizzazione

**WSYNC A**

**WSYNC num**

**A: A**            Attesa sincronismo da tutte le stazioni (All).

**num: 0÷15**       Numero della stazione da cui attendere il sincronismo.

Permette di attendere il segnale di sincronismo da parte della stazione 0 (Master) di una (**num**) o tutte (**A**) le stazioni Slave o di una stazione Slave dalla stazione Master.

Questa istruzione assume due diversi significati se utilizzata su stazione Master o Slave:

- **su stazione Master:** attende il sincronismo dalla stazione Slave indicata;            su quest'ultima deve essere presente la relativa istruzione SYNC (invio sincronismo).
- **su stazione Slave:** attende il sincronismo dalla stazione Master (0); su quest'ultima deve essere presente la relativa istruzione SYNC num (invio sincronismo).

Se usata su stazione 0 può essere:

WSYNC A  
WSYNC num

Se usata su stazioni Slave può essere:

WSYNC (lo 0 viene preso per default)  
WSYNC 0

Vedere anche istruzione SYNC.

## **XOR**

Or esclusivo di 2 bit

**XOR op1,op2[,result]**

op1: **bitporto**            Operando 1.

op2: **bitporto**            Operando 2.

result: **bitporto**        Risultato.

Esegue l'OR esclusivo degli operandi 1 e 2 (**op1** e **op2**) e pone il risultato nel secondo operando.

Il risultato viene invece messo in **result** se questo è presente.

## **7. GESTIONE MACCHINA**

### **AVVIAMENTO**

L'avviamento del sistema deve essere eseguito nella sequenza:

- accensione del PTP1000
- accensione del Personal Computer

Se è stata abilitata l'**Opzione: Inizializzazione automatica in Configurazione sistema**, il PC cerca di collegarsi con i moduli e se si tratta del primo collegamento, trasmette nell'ordine a ciascuna stazione:

- i Parametri
- le Function (se presenti)
- il Programma di Setpoint (se usato)
- il Programma di Automatico
- la Tabella di quote (se usata)

Durante la trasmissione il sistema emette messaggi di avvertimento nel caso non abbia trovato dei file che doveva trasmettere.<sup>1</sup>

Se il PC non è riuscito a collegarsi, o non è riuscito a trasmettere tutto ciò che doveva alle stazioni, oppure l'opzione Inizializzazione automatica è **No**, si pone nell'operatività **Inizializzazione sistema**.

Nel caso venga collegato un PC a un sistema già inizializzato, il PC tenta automaticamente il collegamento, indipendentemente dall'opzione Inizializzazione automatica, e se questo non è andato a buon fine si pone in Inizializzazione sistema.

### **INIZIALIZZAZIONE SISTEMA**

E' l'operatività preposta alla partenza del sistema, e cioè al collegamento tra il PC e i moduli, comandata da operatore.

Sono disponibili le seguenti funzionalità:

---

<sup>1</sup>. Se le schede sono equipaggiate con batteria il PC richiede loro lo stato dei checksum, se è corretto non ritrasmette nulla; ciò evita, su sistemi complessi, perdite di tempo in inizializzazione.

### **F1 - INIT (inizializzazione)**

Permette di avviare l'inizializzazione del sistema dietro comando dell'operatore. Può servire quando, ad esempio, si accede all'Inizializzazione dal menu dell'operatività Automatico; si spegne la parte elettronica ma non si vuole spegnere e riaccendere il PC.

### **F2 - CHGPRG (cambio programmi)**

Permette di cambiare i programmi da trasmettere. I programmi che compaiono nella selezione sono solo quelli compilati e si riferiscono al modulo attivo, pertanto è necessario eseguire il **Cambio modulo (F9 - CHANGE)** per attivare il modulo desiderato.

### **F3 - CHGTAB (cambio tabelle)**

Se sul modulo attivo è previsto l'uso di Tabelle di quote, questa funzionalità consente di selezionare le tabelle da trasmettere alle stazioni interessate. Il Cambio modulo (F9 - CHANGE) permette di attivare il modulo di cui si vuole effettuare la selezione di tabelle.

### **F4 - RETRY (ritrasmissione totale)**

Questa funzionalità permette di ritrasmettere tutti i dati alle stazioni di tutti i moduli. E' utile quando, ad esempio, si hanno sistemi equipaggiati con batteria o disco e si vuole ritrasmettere tutto.

### **F5 - AUTO (automatico)**

Questa funzionalità è attiva solo a sistema già inizializzato e permette di tornare all'operatività Automatico.

### **F9 - CHANGE (cambio modulo)**

Permette di attivare il modulo di cui si vuole effettuare il **Cambio programmi (F2 - CHGPRG)** o il **Cambio tabelle (F3 - CHGTAB)**. L'ultimo modulo selezionato prima di eseguire l'**Inizializzazione (F1 - INIT)**, sarà quello attivo nell'operatività **Automatico**.

### **F10 - MAIN (menu principale)**



Consente l'accesso al Menu principale del sistema.

## **AUTOMATICO**

L'operatività Automatico è quella che permette all'operatore di sovrintendere al lavoro della macchina o delle macchine controllate dal sistema.

Alcuni tasti funzione dell'operatività sono dei comandi che agiscono direttamente sull'esecuzione dei cicli macchina, altri permettono l'accesso ad alcune operatività accessorie.

In questa operatività il PC interroga continuamente tutti i moduli hardware per conoscere il loro stato di funzionamento, permettendo così di rilevare i vari messaggi provenienti dagli stessi, siano essi messaggi di errore o di avvertimento per l'operatore.

Sullo schermo appaiono:

- un riquadro con le quote degli assi della stazione attiva; nel caso di assi in alternata vengono visualizzate le quote di 6 assi per volta.
- un riquadro con le informazioni sui file in esecuzione quali il nome e il commento del programma e della tabella in esecuzione (se usata) per la stazione attiva.
- i messaggi generati dalle istruzioni VIDEO e MESSAGE poste nei programmi; sul video vi sono 8 righe disponibili per la visualizzazione:
  - i messaggi dell'istruzione VIDEO vanno sempre sulla riga 8
  - i messaggi dell'istruzione MESSAGE possono essere inviati sulle righe da 1 a 8 dandone indicazione nell'istruzione stessa.
- i messaggi di errore con indicato se si tratta di errori di Ciclo o di Sistema.
- i contatori e timer, solo se configurati.

I **tasti funzione** dell'operatività sono:

### **F1 - START (avvia esecuzione)**

Avvia l'esecuzione dei programmi di Automatico su tutte le stazioni di tutti i moduli.

Nel caso sia stato premuto uno STOP a esecuzione già in corso lo START fa ripartire i programmi dal punto in cui si erano fermati.

Il comando START viene accettato solo se i programmi delle stazioni dei moduli in stato On, inclusi cioè nell'anello seriale, sono tutti presenti nelle schede.

### **F2 - STOP (sospende esecuzione)**

Sospende l'esecuzione di tutti i programmi, gli assi in movimento si fermano eseguendo una rampa di decelerazione.  
Si riparte premendo START.

### **F3 - END (termina esecuzione)**

Interrompe l'esecuzione di tutti i programmi.  
Premendo START i programmi riprendono dall'inizio.

### **F4 - LOCAL (modo locale)**

Se attivato, i comandi START, STOP e END vengono diretti solo al modulo attivo.  
Per disattivare questa funzionalità ripremere LOCAL.

### **F5 - SETP (setpoint)**

Questo tasto funzione è presente solo se in **Configurazione** è stata attivata l'**Opzione Uso operatività di Setpoint** e permette appunto l'accesso alla stessa.

I tasti funzione disponibili per l'esecuzione del Setpoint sono:

#### **F1 - START (avvia setpoint)**

Avvia l'esecuzione dei programmi di Setpoint su tutte le stazioni di tutti i moduli.  
Il comando START viene accettato solo se i programmi di Setpoint delle stazioni dei moduli in stato On, inclusi cioè nell'anello seriale, sono tutti presenti nelle schede.  
Se il setpoint termina correttamente, non sono intervenuti cioè errori di sistema o non è stato interrotto con F3-end, il sistema torna direttamente in Automatico.

#### **F3 - END (interrompe setpoint)**

Interrompe l'esecuzione di tutti i programmi.  
Premendo START i programmi ripartono dall'inizio.

#### **F4 - LOCAL (modo locale)**

Se attivato dirige i comandi START e END solo al modulo attivo.  
Ripremendolo questa funzionalità viene disattivata.

**F9 - CHANGE (cambio modulo/stazione)**

Consente di cambiare il modulo e la stazione attivi.

**F10 - EXIT (ritorno in automatico)**

Permette di tornare all'operatività Automatico.

Se qualche programma di Setpoint è ancora in esecuzione questo comando non viene accettato.

**F8 - MENU (operatività accessorie)**

Questo menu comprende tutte le operatività accessorie all'Automatico che sono:

- Manuale e diagnostica
- Ritrasmissioni
- Programmazione
- Inizializzazione sistema
- Edit

Queste operatività sono descritte più avanti.

**F9 - CHANGE (cambio modulo/stazione)**

Permette di cambiare il modulo e la stazione attivi.

**F10 - MAIN (menu principale)**

Consente l'uscita dall'operatività Automatico andando al Menu principale.

## MANUALE E DIAGNOSTICA

Questa operatività è utilizzata per il **movimento manuale assi**, il **controllo** degli **ingressi/uscite** e l'**esecuzione di function non parametriche** della stazione attiva.

Entrando nell'operatività l'esecuzione del programma della stazione attiva non viene terminata, pertanto è necessario agire con cura evitando di fare delle operazioni che potrebbero interagire col ciclo causando danni.

Sullo schermo appaiono due riquadri principali: uno dedicato al movimento assi, l'altro agli ingressi e alle uscite digitali:

- **quadro Movimento assi:** questo riquadro contiene informazioni relative agli assi esistenti sulla scheda, viene cioè riportato per ogni asse:

- il nome
- la descrizione
- la quota
- lo stato attuale

Uno degli assi presenti è evidenziato rispetto agli altri per indicare che è quello attivo, le informazioni di **Passo** e **Velocità** rappresentate sono naturalmente riferite all'asse attivo; nel caso di assi in alternata si avrà solo l'informazione relativa al passo.

Per cambiare asse è sufficiente premere la lettera corrispondente al nome dell'asse da selezionare, in questo modo viene modificato l'asse evidenziato e nel contempo i dati di passo e velocità relativi assumono gli ultimi valori impostati del nuovo asse scelto; nel caso di assi in alternata per la selezione dell'asse ci si può avvalere anche del tasto funzione F4 - AXIS.

Per muovere l'asse selezionato è sufficiente premere il tasto + o il tasto - ottenendo uno spostamento dell'asse nella direzione corrispondente al tasto premuto. Il movimento sarà eseguito tenendo conto dei parametri impostati per l'asse selezionato, cioè dello stato, passo e velocità attuali indicati.

- **quadro Input/Output:** in questo riquadro sono rappresentati gli ingressi e le uscite digitali della scheda di controllo. Questi sono ordinati per righe e colonne, dove ogni colonna rappresenta un porto della scheda, cioè un gruppo di otto bit, ed ogni riga rappresenta uno di questi per un totale di otto righe (da 0 a 7).

Nel riquadro sono rappresentati 3 porti per volta per un totale di 24 bit di I/O; l'indirizzo di ogni bit si ottiene dalla combinazione del numero di bit e del numero di porto interessati; ad esempio il primo bit più in alto a sinistra ha indirizzo 0000 poiché si tratta del bit 0 del porto 0, il secondo bit del primo porto avrà indirizzo 1000, ed il terzo bit del secondo porto invece 2001. Per ogni bit di I/O sono disponibili otto caratteri di descrizione la quale, se è stata definita tramite l'operatività

di editor, comparirà nella posizione corrispondente del riquadro I/O; in caso non vi sia definizione per uno dei bit rappresentati comparirà una riga vuota.  
Lo stato di ogni bit (attivo/disattivo o alto/basso) viene rappresentato tramite l'evidenziamento dell'area relativa, corrispondente con l'eventuale descrizione.

I **tasti funzione** disponibili sono:

### **F1 - JOG/ST (tipo di movimento asse)**

Premendo questo tasto si ottiene la commutazione dello stato attuale dell'asse selezionato fra gli stati JOG e STEP.

Nello **stato JOG** il movimento dell'asse inizia premendo il tasto di movimento (+ o -) e continua fino a quando viene rilasciato.

Nello **stato STEP** viene eseguito un movimento di lunghezza specificata come passo di quota incrementale a partire dalla posizione attuale. Per **interrompere un movimento in STEP** è sufficiente premere il tasto **Barra spaziatrice**.

### **F2 - FREE (disabilitazione controllo asse)**

Valido solo per assi in continua.

Premendo questo tasto si ottiene la commutazione dello stato attuale dell'asse selezionato fra lo **stato FREE** ed il precedente stato dell'asse (jog o step).

Quando l'asse è in stato FREE è impossibile muoverlo da tastiera mentre è possibile spostare fisicamente l'asse meccanico, mantenendo comunque la posizione dell'asse.

### **F3 - AXPART (parametri asse)**

Serve ad impostare i dati di **passo** e **velocità** dell'asse attivo, se l'asse è in stato free non è possibile modificare tali dati. I dati per ogni asse vengono memorizzati e sono riproposti ad ogni nuova selezione.

Nel caso di assi in alternata si ha solo il dato del passo.

### **F4 - AXIS (selezione asse)**

Serve a selezionare uno degli assi rappresentati nel riquadro, il risultato è lo stesso della selezione tramite i nomi fisici degli assi.

Nel caso di assi in alternata questo tasto esegue la commutazione sui sei assi successivi puntando al primo dei sei.

In questa fase vengono aggiornati solo i dati nel riquadro MOVIMENTO ASSI e non quelli del riquadro I/O.

### **F6 - IN/OUT (selezione e test I/O)**

Premendo questo tasto vengono aggiornati solo i dati relativi al riquadro I/O e non le quote degli assi.

In questa fase è possibile cambiare i porti rappresentati nel riquadro I/O per vedere lo stato degli ingressi, delle uscite e dei flag.

E' anche possibile modificare lo stato di ogni singolo bit rappresentato, cioè attivare e disattivare delle uscite o dei flag ad ogni premuta del tasto Barra spaziatrice.

Una volta usciti da questa funzionalità resteranno visualizzati gli ultimi porti di I/O correnti anche durante le altre operazioni, eccetto quella di cambio modulo e stazione (vedi F9 - CHANGE).

Accanto all'indirizzo di ogni porto, viene posta tra parentesi una lettera minuscola la quale indica con una 'i' che si tratta di un porto dell'area ingressi oppure con una 'o' che si tratta di un porto dell'area uscite.

### **F7 - FUN (esecuzione function)**

Questo tasto serve per l'esecuzione immediata di una function senza parametri; se ve ne sono compare la lista delle function in ordine di numerazione ed è possibile avviare l'esecuzione di una di queste. Accanto al nome di ogni function compaiono, se definiti, il nome simbolico ed il commento.

Durante l'esecuzione della function vengono aggiornati i dati nei vari riquadri; è possibile **interrompere l'esecuzione** della function premendo il tasto **Barra spaziatrice**.

### **F8 - VOUT (attivazione uscita analogica)**

Consente di impostare su una delle 6 uscite analogiche un valore di tensione da -10.0 a +10.0 volt.

Valido solo per assi in continua.

### **F9 - CHANGE (cambio modulo/stazione)**

Consente di cambiare il modulo e la stazione attivi.

Una volta selezionata la nuova stazione, vengono aggiornati il riquadro degli assi e quello dell'I/O.

### **F10 - EXIT (ritorno in Automatico)**

Consente l'uscita dall'operatività tornando all'Automatico.

## PROGRAMMAZIONE

L'operatività di **Programmazione** consente in modo guidato e interattivo di collaudare, creare o modificare un programma, eseguire singole linee o blocchi di programma e autoapprendere il movimento degli assi.

L'operatività è descritta nel capitolo 8.

## RITRASMISSIONI<sup>2</sup>

Questa operatività rende possibile la ritrasmissione dei Programmi, delle Function e delle Tabelle al modulo attivo.

Viene utilizzata quando si vuole cambiare i programmi di lavoro di una o più stazioni senza necessariamente tornare a Inizializzazione sistema.

Serve anche nel caso si voglia ritrasmettere un programma dopo averlo modificato.

## INIZIALIZZAZIONE SISTEMA

E' l'operatività descritta all'inizio del capitolo.

L'accesso a questa operatività direttamente dall'Automatico è utile quando si vuole reinizializzare il sistema dopo aver spento e riaccessato la parte elettronica senza spegnere il PC.

## EDIT

Questa voce del Menu è presente solo se in **Configurazione** è stata attivata l'**Opzione di modulo: Uso operatività di Edit in Automatico**.

Consente di accedere direttamente all'**Editor** senza passare dal Menu Principale.

L'Editor carica automaticamente il programma della stazione attiva; nel caso si utilizzino tabelle di quote carica la tabella in esecuzione. Eseguite le modifiche desiderate, è possibile apportare modifiche anche ad altri file.

In uscita dall'Editor il sistema torna automaticamente all'operatività Automatico effettuando la ritrasmissione di tutti i programmi, le function e le tabelle eventualmente modificati.

---

<sup>2</sup>. La ritrasmissione provoca la fine dei programmi eventualmente in esecuzione. Nel caso di ritrasmissione di Function vengono ritrasmessi anche i programmi di Setpoint e Automatico perché le function sono allocate prima di essi nella memoria della scheda.

## **8. PROGRAMMAZIONE**

### **GENERALITA'**

L'operatività **Programmazione** consente, in modo guidato e interattivo, di:

- Creare un nuovo programma.
- Modificare un programma esistente.
- Spostarsi all'interno del programma.
- Eseguire una alla volta le istruzioni del programma.
- Eseguire contemporaneamente un blocco di istruzioni.
- Provare le istruzioni prima del loro inserimento nel programma.
- Modificare linee di programma direttamente o in autoapprendimento.
- Inserire nuove linee di programma direttamente o in autoapprendimento.
- Spostare linee di programma.
- Disabilitare linee di programma.
- Riabilitare linee di programma.
- Proteggere linee di programma.
- Cancellare linee di programma.
- Stampare il programma.

L'uso del **tastierino CAT90**<sup>1</sup>, abbinato all'operatività, rende inoltre più agevole l'apprendimento delle posizioni degli assi, che possono essere inserite nel programma direttamente da CAT90. Da tastierino è anche possibile eseguire delle function.

E' possibile intervenire sul programma in esecuzione, oppure su uno dei programmi già realizzati per la macchina.

---

<sup>1</sup>. Il tastierino CAT90 è opzionale, la sua descrizione è alla fine di questo capitolo.



La caratteristica principale dell'operatività è quella di mettere a disposizione dell'operatore un menu di guida per ogni tipo di operazione quali, ad esempio, l'introduzione di:

- movimento degli assi
- funzioni di ingresso/uscita di base della macchina
- lavorazioni speciali
- ripetizioni di parti di programma
- istruzioni GPL1000

A tale scopo, ogni movimento del cursore sulle linee del programma viene accompagnato dalla visualizzazione in chiaro della **finestra di dialogo** contenente gli argomenti dell'istruzione puntata.

### FINESTRA DI DIALOGO

La finestra di dialogo è il solo mezzo per introdurre o modificare gli argomenti delle istruzioni. Nella finestra, chiamata anche per semplicità **box**, ci si sposta sugli argomenti dell'istruzione usando i **cursori su** e **giù**, gli argomenti vanno digitati aiutandosi anche coi tasti di edit: INS, HOME, END, DEL, cursori sinistro e destro, per maggiori dettagli sugli stessi vedere al capitolo Operatività del sistema la descrizione dei tasti disponibili nell'operatività Editor. Alcuni argomenti non vanno digitati ma vanno scelti tra quelli proposti usando il tasto **Barra spaziatrice**.

Nel box compaiono insieme agli argomenti dell'istruzione anche i comandi, o **bottoni**, il passaggio dagli argomenti ai bottoni si effettua coi tasti **TAB** o **SHIFT+TAB** il campo o il bottone attivi sono evidenziati dal reverse; l'esecuzione del bottone attivo si effettua premendo il tasto **ENTER**.

I bottoni presenti nel box dipendono dall'istruzione e consentono di operare sulla stessa. Questi sono:

- |                      |   |
|----------------------|---|
| <b>Esecuzione</b>    | Consente di eseguire l'istruzione per provarla prima di inserirla nel testo.  |
| <b>Inserimento</b>   | Presente solo nel box Inserimento, consente appunto l'inserimento dell'istruzione nel testo sotto la linea puntata. Può essere sostituito dai tasti <b>CTRL+INS</b> .   |
| <b>Cancellazione</b> | Presente nel box Modifica, consente la cancellazione della linea puntata; in alcuni tipi di box cancella l'argomento puntato, quando è rimasto un solo argomento cancella la linea.<br>In modalità modifica può essere sostituito dai tasti <b>CTRL+DEL</b> , in questo caso viene cancellata tutta la linea. |

|                  |   |
|------------------|---|
| <b>Abbandono</b> | Consente di abbandonare le modifiche fatte nel box.<br>Può essere sostituito dal tasto <b>ESC</b> .   |
| <b>Aggiunta</b>  | Presente solo in alcune istruzioni del box Modifica, consente di aggiungere argomenti all'istruzione. |

Nelle pagine dedicate al **Menu di programmazione**, vengono illustrati tutti i tipi di box con i relativi bottoni.

## COME OPERARE SUL PROGRAMMA

I principali modi di operare sul programma sono: l'**Inserimento**, la **Modifica**, la **Cancellazione**, lo **Spostamento** e l'**Esecuzione** di linee di programma, altre modalità, accessorie di queste, sono il **Puntamento alla linea** sotto cui si vuole inserire o da modificare, etc.; la finestra di dialogo assume un colore diverso nelle modalità di Inserimento e Modifica.

### Puntamento alla linea programma

Il puntatore di linea è rappresentato sul video dal carattere >.

Per puntare alla linea desiderata si utilizzano i tasti:

**CTRL+cursore su** per puntare alla linea precedente

**CTRL+cursore giù** per puntare alla linea successiva.

Le pagine di testo sul video si possono scorrere velocemente con i tasti:

**CTRL+PGUP** per visualizzare la pagina precedente

**CTRL+PGDN** per visualizzare la pagina successiva.

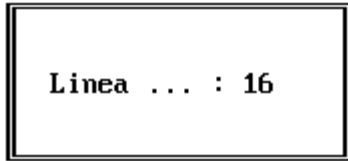
Per puntare direttamente alla prima e all'ultima linea di testo usare:

**CTRL+HOME** per puntare alla prima linea

**CTRL+END** per puntare all'ultima linea

Il **puntamento immediato**, a una linea di programma, si realizza premendo contemporaneamente i tasti **CTRL+L**.

Appare il box:



Digitare il numero di linea e confermare col tasto **ENTER**; l'operazione può essere abbandonata col tasto **ESC**.

### Inserimento linea programma

L'Inserimento consiste nel creare una nuova linea di programma che verrà inserita sotto quella puntata. Si accede a questa modalità dal **Menu di programmazione**, attivabile col tasto **ALT**, e si seleziona poi la voce interessata; un altro modo è **ALT+tasto scorciatoia**, per accedere direttamente all'operatività desiderata; il tasto scorciatoia corrisponde al primo carattere su ogni voce del Menu di programmazione. Per l'inserimento di istruzioni di spostamento asse il tasto scorciatoia è **ALT+nomeasse** (es.: ALT+X nel caso dell'asse X).

Il box di inserimento consente:

- l'introduzione degli argomenti dell'istruzione
- l'esecuzione dell'istruzione in modo da provarne il comportamento
- l'inserimento dell'istruzione nel programma

L'**inserimento**, oltre che col **botone Inserimento**, può essere effettuato anche premendo i tasti **CTRL+INS**.

L'**abbandono** dall'inserimento può essere effettuato col botone Abbandono oppure col tasto **ESC**.

### Modifica linea programma

La Modifica consente di modificare una linea di programma esistente.

Il box di modifica compare automaticamente spostando il cursore sulla linea di programma da modificare e permette:

- la modifica degli argomenti dell'istruzione
- l'esecuzione dell'istruzione in modo da provarne il comportamento
- l'eliminazione dell'istruzione dal programma

- il ripristino degli argomenti originali dopo una modifica

La linea del programma verrà modificata definitivamente quando ci si sposta su un'altra linea, oppure quando si sceglie una nuova istruzione dal menu di programmazione o si esegue la linea puntata, in questo caso infatti prima verrà modificata la linea programma e poi eseguita.

L'abbandono delle modifiche può essere effettuato col bottone Abbandono oppure col tasto **ESC**.

E' possibile **disabilitare** temporaneamente il box di modifica premendo i tasti **CTRL+T**, per **riabilitarlo** ripremere CTRL+T; questo consente di vedere tutte le istruzioni della pagina visualizzata. In questo caso i dati modificati nel box non vengono inseriti nel testo perchè si tratta di uno spegnimento temporaneo del box di modifica, la linea di testo verrà modificata quando si effettua una delle manovre descritte sopra.

### **Cancellazione linea o blocco**

La cancellazione, effettuabile coi tasti **CTRL+DEL**, consente di rimuovere una istruzione o un blocco di istruzioni precedentemente selezionato; prima di cancellare viene richiesta una conferma. La cancellazione della linea puntata è anche effettuabile col bottone Cancellazione del box di modifica.

### **Selezione linea o blocco**

La **selezione** di una linea o di un blocco di istruzioni, su cui agire successivamente con altri comandi, va fatta premendo i tasti **SHIFT+cursore su** o **SHIFT+cursore giù**; i tasti **SHIFT+HOME** consentono di selezionare dalla linea puntata a inizio testo, mentre i tasti **SHIFT+END** selezionano dalla linea puntata a fine testo. Le linee selezionate sono evidenziate da un colore diverso da quello di fondo.

Per **deselezionare** una linea o un blocco premere il tasto **ESC**.

### **Spostamento linea o blocco**

Per spostare una linea o un blocco di istruzioni, precedentemente selezionato, premere i tasti **CTRL+P**, portarsi sulla linea di programma dopo la quale si desidera inserire la linea o il blocco da spostare e ripremere CTRL+P.

Per abbandonare lo spostamento premere il tasto **ESC**.

### Duplicazione linea o blocco

E' possibile duplicare una linea o un blocco di istruzioni, precedentemente selezionato, nel seguente modo:

Per **duplicare una linea**, puntare alla stessa e premere i tasti **CTRL+INS**. Verrà creata una nuova linea uguale a quella puntata, sotto la stessa.

Per **duplicare un blocco** di istruzioni, selezionare il blocco, portarsi sulla linea di programma sotto la quale si desidera inserire e premere i tasti **CTRL+INS**. Verrà inserito un blocco di istruzioni uguale a quello selezionato.

### Esecuzione linea o blocco

La linea puntata o il blocco selezionato possono essere eseguiti premendo i tasti **CTRL+ENTER**; questa modalità è tuttavia inibita quando è attivo un box di inserimento.

Per interrompere l'esecuzione premere il tasto **ESC**.

Nel caso di sistemi con più stazioni è possibile, durante l'esecuzione del blocco, cambiare stazione coi tasti **CTRL+S**; questo consente di fare delle verifiche sul programma della nuova stazione selezionata.

### Gestione task

Consente di vedere, all'interno di un box, i nomi dei task in esecuzione o sospesi a causa di un errore di ciclo/sistema, di uno stop da campo o di un'istruzione Pause.

Si accede premendo contemporaneamente i tasti **CTRL+X**, si esce premendo **ESC**.

All'interno del box sono disponibili i seguenti bottoni:

|           |   |
|-----------|---|
| VAI       | Per puntare al task selezionato                     |
| TERMINA   | Per terminare l'esecuzione del/i task selezionato/i |
| AGGIORNA  | Per aggiornare l'elenco dei task nel box            |
| ABBANDONO | Per uscire dalla gestione task                      |

Il task su cui viene intrapresa l'azione alla premuta del bottone, si seleziona utilizzando i tasti cursore su/giù; nel caso nessun task sia selezionato, viene assunto quello puntato dal cursore.

Volendo terminare contemporaneamente tutti i task, è possibile selezionarli tutti premendo contemporaneamente i tasti **CTRL++**.

Con questo box aperto il cursore di esecuzione, se va oltre i limiti del testo sul video, non viene più visualizzato. Per vederlo bisogna uscire dal box.

## Disabilitazione e Riabilitazione linea

E' possibile escludere dall'esecuzione una o più linee all'interno del blocco operando una disabilitazione, questa si ottiene selezionando dal Menu di programmazione la voce Gestione programma e successivamente la voce **Disabilitazione**, davanti alla linea disabilitata apparirà un carattere a forma di rombo.

La disabilitazione è effettuabile solo su una linea per volta, le linee di commento, quelle con etichetta, quelle protette e quelle con l'istruzione PROG, non si possono disabilitare.

La **Riabilitazione** della linea viene fatta col bottone **Abilita** del box di modifica che compare puntando alla linea stessa. Le linee disabilitate restano tali anche uscendo dalla Programmazione, pertanto non vengono eseguite in Automatico.

Le operazioni di disabilitazione e riabilitazione possono essere effettuate anche premendo contemporaneamente i tasti **CTRL+D**.

Nel file sorgente le linee disabilitate sono contraddistinte dal carattere **punto e virgola** seguito da **parentesi tonda chiusa (;)**, posti all'inizio della linea.

## Protezione e Sprotezione linea o blocco

E' possibile proteggere una o più linee per evitare modifiche accidentali delle stesse, le linee protette infatti non possono essere spostate, cancellate o modificate (il box di modifica non compare) e tantomeno è possibile inserire nuove linee all'interno di un blocco di queste. Questo consente anche di focalizzare l'attenzione dell'operatore solo su quelle parti di programma che realmente necessitano di essere modificate.

La **protezione** viene attivata nel seguente modo:

- puntare alla linea iniziale della linea o del blocco da proteggere
- premere i tasti **CTRL+K** per attivare la funzionalità di protezione
- premere il tasto **B** per indicare l'inizio protezione
- spostare il cursore sulla linea finale del blocco proteggere
- premere i tasti **CTRL+K** per attivare la funzionalità di protezione
- premere il tasto **E** per indicare la fine protezione

Le linee protette verranno evidenziate con un colore diverso da quello usato per le linee selezionate. Nel file sorgente le linee protette sono contrassegnate dal carattere **punto e virgola (;)** posto immediatamente dopo all'ultimo argomento nella linea.

La **sprotezione** si effettua procedendo come per la protezione.

L'abbandono da queste modalità è reso possibile dal tasto **ESC**.

### Cambio stazione automatico e ritorno

Nel caso di sistema con più stazioni, è possibile effettuare un cambio stazione automatico per passare dalla stazione zero ad un'altra stazione, tornando poi alla stazione zero. Il funzionamento dipende dalla stazione ed è il seguente:

|                             |   |
|-----------------------------|---|
| Su stazione zero            | Premere contemporaneamente i tasti <b>CTRL+SHIFT+ENTER</b> per passare all'altra stazione; nel caso si abbiano più di due stazioni  |
| viene                       | richiesto il cambio stazione del modulo corrente. Viene caricato il programma della stazione selezionata che ha lo stesso nome di quello della  |
| zero.                       |   |
| Su stazione uno o superiore | Premere contemporaneamente i tasti <b>CTRL+SHIFT+ENTER</b> per tornare alla stazione zero. Il sistema seleziona automaticamente tutto il programma, lo mette in esecuzione e ricarica il programma della stazione zero, puntando alla riga successiva a quella puntata quando si è effettuato il cambio stazione. |

### Come selezionare una voce in un menu

In tutti i menu la selezione della voce interessata è resa possibile dai tasti:

- **cursore su/giù** per scorrere voce per voce
- **PGUP/PGDN** per scorrere pagina per pagina nel caso il menu sia composto da più pagine. In questo caso sulla destra del box compare una barra di scorrimento verticale.
- **ricerca immediata** digitare, partendo dalle iniziali, il nome della voce desiderata; si torna indietro col tasto  **BackSpace**.
- **ENTER** per selezionare la voce puntata.

### Tasti disponibili nell'operatività

La lista dei tasti di maggior utilizzo è sempre visualizzata nella zona dell'elenco dei tasti disponibili (ultime righe del video), tuttavia il numero di tasti disponibili è superiore. Di seguito sono elencati tutti i tasti e il loro utilizzo, il carattere + tra i tasti indica che vanno premuti insieme iniziando dal primo descritto.

|                         |  |
|-------------------------|--|
| <b>CTRL+cursore su</b>  | Va alla linea precedente.  |
| <b>CTRL+cursore giù</b> | Va alla linea successiva.  |
| <b>CTRL+PGUP</b>        | Va alla pagina precedente.   |
| <b>CTRL+PGDN</b>        | Va alla pagina successiva.   |
| <b>CTRL+HOME</b>        | Va alla prima linea del testo.   |
| <b>CTRL+END</b>         | Va all'ultima linea del testo.   |
| <b>CTRL+C</b>           | Per creare una linea di commento.  |
| <b>CTRL+D</b>           | Per abilitare/disabilitare una linea.                                    |
| <b>CTRL+E</b>           | Per cercare un'etichetta, solo col nome completo della stessa.           |
| <b>CTRL+F</b>           | Per cercare una stringa.   |
| <b>CTRL+L</b>           | Per puntare alla linea desiderata.                                       |
| <b>CTRL+P</b>           | Per spostare una linea o il blocco selezionato.                          |
| <b>CTRL+S</b>           | Per cambiare stazione durante l'esecuzione di un blocco.                 |
| <b>CTRL+T</b>           | Per attivare/disattivare la finestra di dialogo nella modalità Modifica. |
| <b>CTRL+X</b>           | Per attivare l'operatività Gestione task.                                |
| <b>CTRL+K+B</b>         | Per segnalare l'inizio del blocco da proteggere/sproteggere.             |
| <b>CTRL+K+E</b>         | Per segnalare la fine del blocco da proteggere/sproteggere.              |
| <b>CTRL+INS</b>         | Equivale al bottone inserimento.   |
| <b>CTRL+DEL</b>         | Per cancellare una linea o il blocco selezionato.                        |
| <b>CTRL+ENTER</b>       | Per eseguire la linea o il blocco selezionato.                           |
| <b>CTRL+ALT+O</b>       | Per accedere all'operatività Analizzatore.                               |
| <b>CTRL+SHIFT+ENTER</b> | Per eseguire il Cambio automatico stazione e ritorno.                    |



|                            |   |
|----------------------------|---|
| <b>CTRL+F5</b>             | Esegue il programma dalla riga puntata a fine programma, selezionando automaticamente l'area di esecuzione.   |
| <b>CTRL+F7</b>             | Esegue il programma dalla riga in esecuzione alla riga precedente a quella puntata, selezionando automaticamente l'area di esecuzione, solo dall'alto in basso. |
| <b>CTRL+F8</b>             | Esegue la riga puntata dal puntatore di linea, ignorando e deselegionando eventuali blocchi selezionati.  |
| <b>CTRL+F10</b>            | Esegue la riga puntata dal puntatore di linea, ignorando e deselegionando eventuali blocchi selezionati. Se l'istruzione è una CALL, la esegue completamente.   |
| <b>F2</b>                  | Riporta il puntatore di linea alla posizione precedente all'ultima modifica.  |
| <b>F3</b><br>fino          | Cerca l'ultima stringa dalla riga successiva a quella puntata a fine programma.   |
| <b>SHIFT+F3</b><br>fino    | Cerca l'ultima stringa dalla riga precedente a quella puntata a inizio programma.   |
| <b>SHIFT+5</b> (tast. NUM) | Seleziona l'intero programma. Premendo CTRL+ENTER esegue il programma dall'inizio.  |
| <b>SHIFT+cursore su</b>    | Per selezionare la linea corrente e le precedenti.  |
| <b>SHIFT+cursore giù</b>   | Per selezionare la linea corrente e le successive.  |
| <b>SHIFT+HOME</b>          | Per selezionare dalla linea corrente a inizio testo.  |
| <b>SHIFT+END</b>           | Per selezionare dalla linea corrente a fine testo.  |
| <b>ESC</b>                 | In esecuzione: ferma l'esecuzione.<br>Se blocco selezionato: deselegiona.<br>In forma più generale: abbandona ciò che si è fatto.                               |
| <b>ENTER</b>               | Esegue il bottone attivo nella finestra di dialogo.   |
| <b>TAB e SHIFT+TAB</b>     | Per muoversi sui campi e i bottoni della finestra di dialogo.   |
| <b>ALT</b>                 | Attiva il Menu di Programmazione.   |

|                              |  |
|------------------------------|--|
| <b>ALT+H</b>                 | Attiva la finestra di aiuto.<br>Questa funzionalità è in fase di implementazione e verrà introdotta successivamente. |
| <b>ALT+F4</b>                | Uscita immediata dall'operatività Programmazione.<br>Se il programma è stato modificato, chiede di salvare.          |
| <b>ALT+nomeasse (XYZWV)</b>  | Attiva la gestione dell'asse corrispondente in modalità Inserimento.   |
| <b>ALT+tasto scorciatoia</b> | Per accedere direttamente, in modalità Inserimento, all'operatività legata al tasto scorciatoia.                     |

## RAPPRESENTAZIONE DEL VIDEO

Sul video, oltre alle istruzioni del programma, compaiono altre informazioni. Di seguito viene mostrata una tipica videata di Programmazione, in particolare va precisato che:

- le **quote assi** sono sempre riferite alla posizione attuale degli stessi.
- la **riga 2** contiene un messaggio che indica se si sta operando sul programma principale o su un programma parallelo, in questo caso è presente anche il nome dello stesso.
- il **puntatore di linea** (carattere >) indica la linea attiva; ricordiamo che il box di modifica fa sempre riferimento all'istruzione su questa linea.
- il **puntatore di esecuzione** (cambia il colore del numero di linea) indica qual'è la prossima linea che verrà eseguita o quella di cui è stata interrotta l'esecuzione; durante l'esecuzione del programma o di un blocco di istruzioni, questo cursore si sposta continuamente sulla linea in esecuzione.
- l'**indicatore di linea disabilitata** (carattere a forma di rombo) indica che la linea è disabilitata e non verrà messa in esecuzione.
- il **messaggio d'errore** viene visualizzato quando si cerca di compiere operazioni errate sul programma o quando uno dei dati digitati è sbagliato.
- i **tasti disponibili** visualizzati dipendono dal contesto operativo.
- il **messaggio all'operatore** appare quando viene eseguita una linea con l'istruzione VIDEO o MESSAGE.

| PROGRAMMAZIONE: DEMO  |                | station 0 - ptp 2 assi |     | Modulo 0   |  |
|---|----------------|------------------------|-----|--|--|
| X 0   | Y 0            | Z 0                    | W 0 |  |  |
| Programma   |                |                        |     |  |  |
| 1   | prog           |                        |     | Function   |  |
| 2   | gira:          |                        |     | <div style="border: 1px solid black; padding: 5px; display: inline-block;">           FunNormale         </div> <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-left: 20px;">           Esecuzione<br/>Abbandono<br/>Cancellazione         </div> |  |
| 3   | ◆ DELAY        | 2.5                    |     |  |  |
| 4   | FPROG          | FunNorma               |     |  |  |
| 5   | WIZ            | INP00                  |     |  |  |
| 6   | ▶ FunNormale   |                        |     |  |  |
| 7   | FunParametrica |                        |     |  |  |
| 8   | X              | 20 Q                   |     |  |  |
| 9   | VIDEO          | tutto bene !!!!!       |     |  |  |
| 10  | DELAY          | 2.5                    |     |  |  |
| 11  | VIDEO          |                        |     |  |  |
| 12  | BRA            | gira                   |     |  |  |
| ALT+H aiuto                      ALT menu                      CTRL+←  esegue<br>CTRL+T toggle box              ESC abbandono              CTRL+DEL cancella<br>CTRL+P sposta                    TAB campo successivo |                |                        |     |  |  |

*Rappresentazione del video nell'operatività Programmazione*

## INGRESSO IN PROGRAMMAZIONE

L'operatività Programmazione viene richiamata dal Menu dell'operatività Automatico. E' possibile entrare in Programmazione anche col programma automatico in esecuzione, in questo caso se si vuole interrompere l'esecuzione basta premere il tasto **ESC**. Se si vuole tornare in Automatico senza interrompere l'esecuzione del programma, basta premere contemporaneamente i tasti **ALT+F4**.

Entrando in Programmazione col programma automatico in esecuzione, se il sistema si accorge che il programma è stato modificato ma non ritrasmesso, rilevando quindi un'incongruenza tra il programma sulla scheda e quello su PC, il testo non viene evidenziato e il programma viene terminato.

Se si entra in Programmazione e il programma non è in esecuzione appare il seguente box:

| Programma    |   |
|--------------|---|
| Nome .....   | : |
| Commento ... | : |

### Creazione di un nuovo programma

Se si vuole creare un nuovo programma è necessario scrivere il nome del programma ed eventualmente il commento allo stesso, come indicato in questo box:

| Programma    |                          |
|--------------|--------------------------|
| Nome .....   | : DEMO                   |
| Commento ... | : programma dimostrativo |

Il sistema, in questo caso, crea un finto programma anche sulle altre stazioni del modulo, composto dalle istruzioni:

```

                PROG
LOOP:          BRA      LOOP
    
```

### Selezione di un programma esistente o di quello attivo

Se si vuole agire su un programma esistente premere il tasto **ENTER** sul primo box, apparirà la lista dei programmi della stazione attiva con evidenziato quello attivo:

| Direttorio |    |                        |          |
|------------|----|------------------------|----------|
| Nome       | St | Commento               | Data     |
| DEMO       | Ø  | programma dimostrativo | 10/11/93 |
| PROVA      | Ø  | programma di prova     | 04/11/93 |

Selezionare il programma desiderato e premere **ENTER**.

A questo punto è possibile agire sul programma con le modalità operative descritte in precedenza.

## USCITA DALLA PROGRAMMAZIONE

L'uscita dalla Programmazione si ottiene in due modi diversi:

- premendo contemporaneamente i tasti **ALT+F4**, se non si è in modalità Inserimento.
- selezionando dal **Menu** la voce **Gestione programma** e poi la voce **Automatico**.

Se il programma è stato modificato, appare un box che richiede se si vuole o meno salvarlo. Rispondendo **Sì** appare il box col nome e il commento del programma, a questo punto è possibile salvare il programma con un altro nome, qualora si volesse crearne uno nuovo derivato da quello modificato; in questo caso il sistema produce automaticamente una nuova copia dei programmi anche delle altre stazioni.

Prima del rientro nell'operatività Automatico il programma viene ritrasmesso alla scheda; se si è prodotto un nuovo programma dalle modifiche di quello selezionato entrando in Programmazione questo diventa il programma attivo su tutte le schede del modulo.

## MENU DI PROGRAMMAZIONE

Il Menu di Programmazione contiene tutte le voci per accedere alle operazioni di:

- inserimento di istruzioni relative a Lavorazioni speciali
- inserimento di nuove istruzioni nel programma
- accesso alle voci della Gestione programma
- cambio stazione
- test degli ingressi e uscite

Il menu si attiva premendo il tasto **ALT** e appare:

```
— Menu di programmazione —
1-Gestione assi
2-Dispositivi
3-Messaggio
4-Salto
5-Ripeti
6-Attesa function parallela
7-Istruzione Gpl
8-Gestione programma
9-Cambio stazione
T-Test ingressi/uscite
```

Nelle pagine seguenti sono descritte una per una tutte le voci che compongono il Menu di programmazione, nelle varie sezioni vengono illustrate sia le modalità di Inserimento, accessibili da questo Menu, che quelle di Modifica di linee precedentemente inserite.

## 1 - GESTIONE ASSI

La gestione assi consente di operare direttamente sugli assi, apprendere le posizioni desiderate e inserirle nel programma.

Alle istruzioni sugli assi corrispondono le istruzioni Gpl **X Y Z W V**<sup>2</sup>.

### Selezione asse

Selezionare nel menu degli assi l'asse desiderato.

```

┌ Selezione asse ───────────┐
│ X asse X                   │
│ Y asse Y                   │
│ Z asse Z                   │
│ W asse W                   │
│ V asse V                   │
└──────────────────────────┘

```

La selezione può essere fatta anche coi tasti scorciatoia, premendo contemporaneamente i tasti **ALT+nomeasse** (es.: ALT+X), in questo caso si apre immediatamente il box di Inserimento.

### Inserimento

Il box di inserimento contiene tutti i dati necessari per eseguire il movimento dell'asse, il titolo è composto dal nome e dalla descrizione dell'asse:

```

────────── X asse X ───────────
│
│ Passo ..... : 0.1
│ Quota assoluta : 20
│
│ Velocità ..... : 1.5
│ Arresto ..... : IN QUOTA
│
│ [mm] (0.01/0.1/1/5/x.x)
│
└──────────────────────────┘

```

Quota assoluta  
Abbandono  
Inserimento

-      +

Per muoversi sui campi digitabili utilizzare i tasti  **cursore su/giù**; nell'ultima riga del box appare un'informazione di aiuto relativa al campo puntato.

<sup>2</sup>. Per ulteriori dettagli sulle istruzioni GPL vedere il capitolo: Linguaggio GPL1000.

Per raggiungere la quota che si desidera inserire nel programma, si possono effettuare due tipi di spostamento:

- Passo impostato** Premendo i tasti più (+) o meno (-) del tastierino numerico, si effettuano spostamenti positivi o negativi di entità pari alla quota Passo impostata. Sono possibili cinque passi diversi, quattro preimpostati e uno impostabile a piacere. La selezione del passo desiderato si effettua col tasto **Barra spaziatrice**, per impostare un proprio passo è sufficiente digitarne il valore nel campo Passo.
- Quota assoluta** Premendo il bottone Quota assoluta si effettua lo spostamento quota assoluta impostata.

In entrambi i casi la **Velocità** di spostamento è quella impostata nel box.

Il movimento dell'asse può essere interrotto col tasto **ESC**.

Il dato **Arresto** viene considerato solo all'atto dell'inserimento dell'istruzione nel programma e indica quale tipo di attesa si vuole avere per considerare terminato il movimento dell'asse durante l'esecuzione del programma. Gli stati di attesa possibili sono tutti quelli dell'istruzione Gpl WEND<sup>2</sup>, più uno che consente di non avere attesa sulla fine dell'istruzione per passare all'esecuzione di quella successiva:

|          |                        |
|----------|------------------------|
| ACCELER. | Accelerazione          |
| A REGIME | Regime                 |
| DECELER  | Decelerazione          |
| TEORICO  | Fine movimento teorico |
| IN QUOTA | Arrivo in quota        |
| PROSEGUE | Prosegue               |

La selezione del tipo di arresto si effettua col tasto **Barra spaziatrice**.

Dopo aver portato l'asse alla quota desiderata, è possibile inserire l'istruzione nel programma.

Nel caso preso ad esempio, supponendo di aver portato l'asse alla quota 20.4, si avrà la linea:

X 20.4 Q

Se il tipo di Arresto fosse "PROSEGUE", la linea sarebbe:

X 20.4

Per aggiungere altri assi alla linea ci si avvale della modalità Modifica.



## Modifica

Puntando alla prima delle linee inserite nell'esempio precedente appare il box:

| X asse X                 |  |
|--------------------------|--|
| Passo ..... : 0.1        | Quota assoluta<br>Abbandono<br>Cancellazione |
| Quota assoluta : 20.4    |  |
| Velocità ..... : 1.5     | -      +                                     |
| Arresto ..... : IN QUOTA |  |
| [mm] (0.01/0.1/1/5/x.x)  | X  |

Come si può notare la Quota assoluta e l'Arresto sono quelli della della linea programma; nella zona di aiuto del box appare anche il nome dell'asse sulla linea, aggiungendo altri assi anch'essi verranno visualizzati qui.

E' possibile eseguire spostamenti per modificare la quota programmata, aggiungere spostamenti di altri assi sulla stessa linea programma e cambiare il tipo di Arresto; le operazioni sono le stesse descritte per l'Inserimento.

Per aggiungere lo spostamento di un'altro asse premere il tasto corrispondente al nome dello stesso (es.: per aggiungere Y premere tasto Y). Il titolo del box sarà composto dal nome del nuovo asse e dalla sua descrizione. Nella zona di aiuto apparirà oltre a X anche Y.

Supponendo di muovere Y alla quota 10 e di sostituire al tipo di Arresto IN QUOTA quello TEORICO, quando si rendono effettive le modifiche, la linea di programma sarà:

X      20.4 Y 10 F

Il bottone **Cancellazione** opera nel seguente modo:

Se nella zona di aiuto è presente più di un asse, cancella l'asse a cui il box si riferisce in quell'istante.

Se nella zona di aiuto è presente un solo asse, cancella la linea di programma.

## 2 - DISPOSITIVI

Il menu dei Dispositivi macchina, attivabile anche coi tasti scorciatoia **ALT+2**, è il seguente:

```

┌ Dispositivi macchina ───┐
│ Function                │
│ Function parallele      │
│ Ingressi                │
│ Uscite                  │
└──────────────────────────┘
    
```

Selezionare nel menu il tipo di dispositivo desiderato.

Ogni tipo di dispositivo ha un suo sottomenu nel quale sono elencati i dispositivi che ne fanno parte.

Una volta inserito un dispositivo nel testo, riappare il menu dei dispositivi per permettere inserimenti successivi. Per abbandonare l'inserimento di dispositivi, premere il tasto **ESC** su questo menu.

### Function

Le function sono parti di programma, scritte anch'esse in linguaggio GPL1000, solitamente create dal costruttore per gestire un gruppo meccanico e utilizzabili dall'utente tramite il loro nome; nel caso, ad esempio, di una function per l'apertura di una pinza questa potrebbe chiamarsi ApriPinza.

Al dispositivo Function corrisponde l'istruzione Gpl **FCALL**<sup>2</sup>.

Selezionare nel menu che contiene l'elenco delle Function quella desiderata.

```

┌ Function ───┐
│ FunNormale  │
│ ApriPinza   │
│ FunParametrica │
└──────────┘
    
```

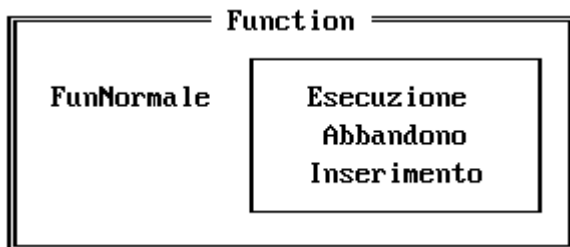
Le Function possono essere di due tipi, quelle **senza parametri** e quelle **con parametri (o parametriche)**.

### Function senza parametri

Una function senza parametri è una function che non ha bisogno di argomenti che ne modificano il comportamento.

#### Inserimento

Nel caso sia stato selezionato il nome di una function senza parametri, appare il seguente box:



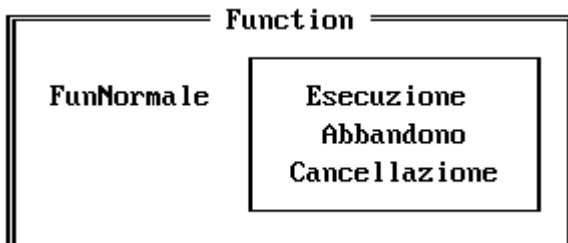
Inserendo la function nel testo si otterrà in questo caso la linea:

FunSenzaParametri

Come si può notare l'istruzione FCALL non compare e questo dipende dal fatto che è concesso richiamare la function col solo nome della stessa, omettendo così l'istruzione FCALL.

#### Modifica

Puntando a una linea di testo che richiama una function senza parametri, appare il box:



Qualora si volesse sostituire la function sulla linea con un'altra, è necessario Cancellare la linea ed introdurre una nuova ripetendo le operazioni di Inserimento descritte sopra.

## Function parametrica

Una function parametrica si distingue per la presenza di argomenti, detti parametri, che seguono il nome della function e verranno utilizzati dalla stessa durante la sua esecuzione modificandone il comportamento.

### Inserimento

Nel caso sia stato selezionato il nome di una function parametrica appare, ad esempio, il seguente box:

| FunParametrica           |           |             |
|--------------------------|-----------|-------------|
| Velocità X ..... : 9     |           |             |
| Quota asse X ... : 10.52 |           |             |
| [mt/1' ] x.x             |           |             |
| Esecuzione               | Abbandono | Inserimento |

Digitare i parametri spostandosi sugli stessi coi tasti  **cursore su/giù**; ad ogni parametro è associato un aiuto che fornisce indicazioni sul tipo di dato da digitare (vedere la sezione: "Simboli utilizzati per l'acquisizione dei dati" nel capitolo "Introduzione").

Inserendo la function nel testo, si otterrà in questo caso la linea:

FunParametrica      10.52,9

Come si può notare l'istruzione FCALL non compare, questo dipende dal fatto che è concesso richiamare la function col solo nome della stessa, omettendo così l'istruzione FCALL. Sono inoltre presenti i due parametri digitati.

### Modifica

Puntando a una linea di testo che richiama la function parametrica dell'esempio, appare il box:

| FunParametrica           |           |               |
|--------------------------|-----------|---------------|
| Velocità X ..... : 9     |           |               |
| Quota asse X ... : 10.52 |           |               |
| [mt/1' ] x.x             |           |               |
| Esecuzione               | Abbandono | Cancellazione |

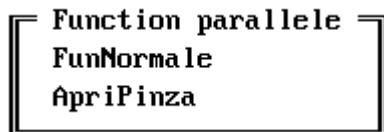
Si possono qui modificare i parametri con nuovi valori.

## Function parallele

Le function parallele sono function senza parametri che possono essere eseguite in parallelo all'esecuzione del programma principale, di programmi paralleli o di altre function parallele; il numero massimo complessivo di processi paralleli è 8.

Al dispositivo Function parallela corrisponde l'istruzione Gpl **FPROG**<sup>2</sup>.

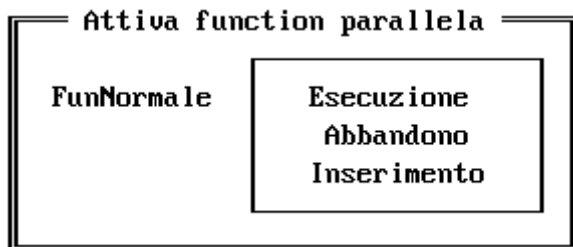
Selezionare nel menu che contiene l'elenco delle Function parallele quella desiderata.



Possono essere attivate fino a tre function parallele sulla stessa linea di testo, tuttavia il sistema per poterle eseguire verifica che vi siano almeno tre processi paralleli disponibili, in caso contrario attende la loro disponibilità.

## Inserimento

Una volta selezionato il nome della function parallela desiderata, appare il seguente box:



L'Esecuzione di prova della function non occupa in questo caso uno degli otto processi paralleli ma viene effettuata su un altro processo dedicato.

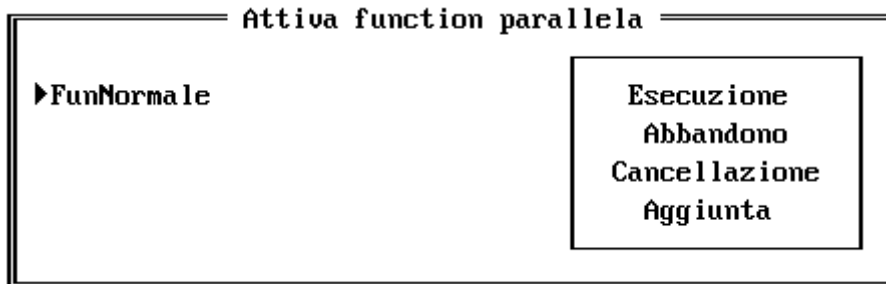
In questa modalità è possibile inserire la function su una nuova linea di programma, le altre potranno essere aggiunte con la modalità Modifica.

In questo caso verrà inserita la linea:

FPROG FunSenzaParametri

## Modifica

Puntando a una linea contenente una function parallela, appare il box:

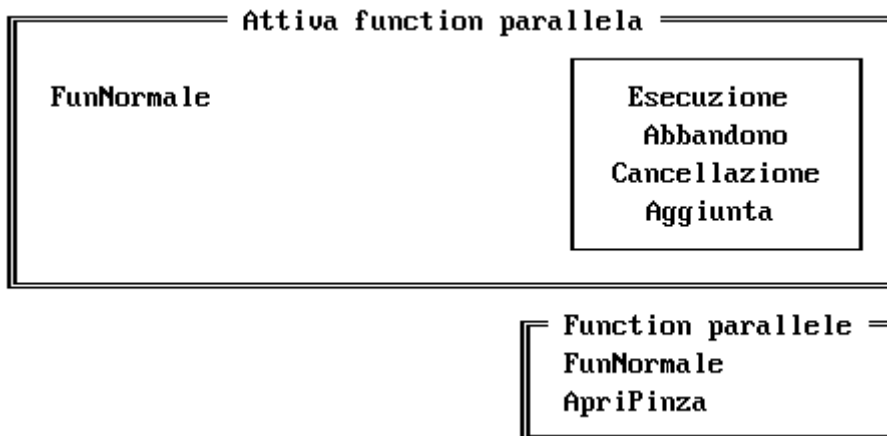


Nel box compaiono i nomi delle function parallele presenti sulla linea di programma. Sulla prima è visibile il cursore di selezione che corrisponde al carattere >.

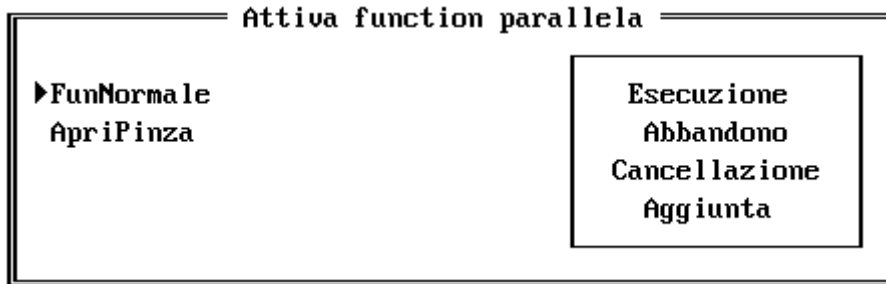
Volendo aggiungere una seconda function sulla linea è necessario premere il bottone Aggiunta.

## Aggiunta

La premuta di Aggiunta fa comparire il menu delle function parallele, permettendo così di selezionare il nome della nuova function:



La function aggiunta viene visualizzata sotto quella già presente:



Se nel box ci sono già tre function il bottone Aggiunta non produce più nessun effetto, inoltre il sistema non permette di aggiungere una function già presente nel box.

Coi tasti  **cursore su/giù**  è possibile puntare alla function desiderata per le operazioni di Esecuzione o Cancellazione; se è presente una sola function, l'azione sul bottone Cancellazione fa sì che venga cancellata la linea di programma.

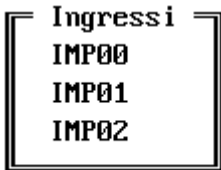
Una volta confermata la modifica nell'esempio appena descritto, la linea di programma sarà:

FPROG FunSenzaParametri,ApriPinza

Vale qui la pena di ricordare che la linea di programma verrà eseguita solo se ci sono due processi liberi sugli otto possibili, in caso contrario l'esecuzione della linea verrà tenuta in sospenso finché i due processi non si liberano.

## Ingressi

Selezionare nel menu che contiene l'elenco degli I/O l'ingresso desiderato.



La funzionalità consente l'inserimento delle istruzioni Gpl **WIZ**, **WNZ**, **SNZ** e **SKZ** relative ai test sullo stato degli ingressi<sup>2</sup>.

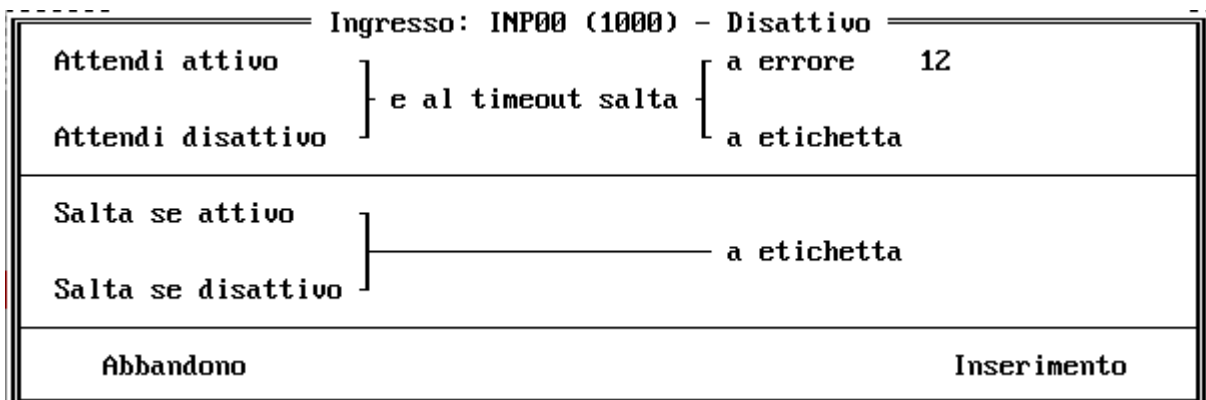
## Inserimento

Questo box è caratterizzato dalle seguenti informazioni nel titolo:

- la definizione simbolica                      es.: INP01
- il bitporto                                        es.: 1000
- lo stato attuale dell'ingresso            es.: Disattivo

Lo stato iniziale è **Attendi attivo** e corrisponde all'istruzione **WIZ**, la selezione dello stato desiderato si effettua con i tasti  **cursore su/giù**.

Premendo il bottone Inserimento viene inserita l'istruzione corrispondente a quella selezionata.



In questo caso nel testo verrà inserita la linea:

```
WIZ INP01,12
```

Per le istruzioni **Attendi attivo** (**WIZ**) e **Attendi disattivo** (**WNZ**), è consentito, allo scadere del timeout, di saltare a uno dei seguenti argomenti:



- **a errore**

l'esecuzione

va indicato il numero dell'errore di ciclo aggiungendo opzionalmente l'argomento **,C** se si vuole che il programma continui

senza arrestarsi dopo aver emesso il messaggio di errore.

Es.: WIZ INP01,12,C.

- **a etichetta** va indicato il nome dell'etichetta a cui saltare.

Es.: WIZ INP01,CICLO.

Per le istruzioni **Salta se attivo** (SNZ) e **Salta se disattivo** (SKZ), è consentito, allo scadere del timeout, di saltare a una etichetta.

Nel caso il nome dell'etichetta a cui saltare non fosse ancora presente nel testo, una nuova linea con etichetta viene creata sotto a quella dell'istruzione, appena viene effettuato l'inserimento. Il cursore punta alla linea con etichetta per consentire, volendo, lo spostamento della stessa.

### Modifica

Il box evidenzia lo stato attuale dell'ingresso nel titolo e lo stato corrispondente all'istruzione Gpl nella linea di testo, all'interno del box.

|                                    |                        |
|------------------------------------|------------------------|
| ingresso: INP00 (1000) - Disattivo |                        |
| Attendi attivo                     | } e al timeout salta { |
| Attendi disattivo                  |                        |
|                                    | a errore               |
|                                    | a etichetta            |
| Salta se attivo                    | } _____ a etichetta    |
| Salta se disattivo                 |                        |
| Abbandono                          | Cancellazione          |

Le operazioni che si possono eseguire sono identiche alla modalità Inserimento.

Se viene modificato il nome dell'etichetta a cui saltare e questo non è ancora presente nel testo, una nuova linea con etichetta viene creata sotto a quella dell'istruzione, appena viene effettuata la modifica.



|                                  |               |
|----------------------------------|---------------|
| Uscita: OUT02 (2000) - Disattiva |               |
| Attiva                           | Esecuzione    |
| ▶ Disattiva                      | Abbandono     |
|                                  | Cancellazione |

Le operazioni che si possono eseguire sono identiche alla modalità Inserimento.

### 3 - MESSAGGIO

Questa funzionalità permette l'inserimento di un messaggio per l'operatore, che verrà visualizzato nella zona dedicata quando l'istruzione viene eseguita; se l'argomento non viene digitato, quando l'istruzione viene eseguita, la zona messaggio sul video viene ripulita da un eventuale messaggio presente.

L'istruzione Messaggio corrisponde all'istruzione Gpl VIDEO<sup>2</sup>.

#### Inserimento

Attivabile anche coi tasti scorciatoia ALT+3, fa comparire il box:

|                            |           |             |
|----------------------------|-----------|-------------|
| <b>Messaggio</b>           |           |             |
| Messaggio per l'operatore_ |           |             |
| Esecuzione                 | Abbandono | Inserimento |

Digitare il testo del messaggio nel campo argomento e inserire; in questo caso verrà inserita la linea:

VIDEO      Messaggio per l'operatore

#### Modifica

Puntando a una linea con istruzione VIDEO appare il box di modifica:

|                            |           |               |
|----------------------------|-----------|---------------|
| <b>Messaggio</b>           |           |               |
| Messaggio per l'operatore_ |           |               |
| Esecuzione                 | Abbandono | Cancellazione |

Modificare il testo del messaggio nel campo argomento.

## 4 - SALTO

Consente l'inserimento di un'istruzione di salto a etichetta; quando l'istruzione viene eseguita l'esecuzione continua sulla linea dove è posta l'etichetta.

Dopo l'inserimento o la modifica se l'etichetta non esiste viene creata automaticamente sulla linea sottostante a quella inserita o modificata. E' possibile posizionarla poi nel punto desiderato con la funzionalità di Spostamento.

La linea contenente l'etichetta può essere cancellata coi tasti **CTRL+DEL**, tuttavia la cancellazione viene effettuata solo se nessuna linea del programma contiene istruzioni di salto alla stessa, in caso contrario appare il messaggio: **Etichetta ancora in uso**.

L'istruzione Salto corrispondente all'istruzione Gpl **BRA**.

### Inserimento

Attivabile anche coi tasti scorciatoia **ALT+4**, fa comparire il box:

|                      |             |
|----------------------|-------------|
| <b>Salto</b>         |             |
| Salta a ... : CICLO_ |             |
| Abbandono            | Inserimento |

Digitare il nome dell'etichetta (es.: CICLO) nel campo argomento e inserire.

In questo caso verranno inserite le linee:

```

                BRA  CICLO
CICLO:
    
```

; solo se non è già presente

### Modifica

Puntando a una linea con istruzione BRA appare il box di modifica:

|                      |           |               |
|----------------------|-----------|---------------|
| <b>Salto</b>         |           |               |
| Salta a ... : CICLO_ |           |               |
| Esecuzione           | Abbandono | Cancellazione |

## 8.32 Programmazione

---

Modificare il nome dell'etichetta nel campo argomento.

## 5 - RIPETI

Consente l'inserimento dell'istruzione di ripetizione di blocco di istruzioni; la fine del blocco di istruzioni da ripetere coincide con l'istruzione ENDREP.

L'istruzione Ripeti corrispondente all'istruzione Gpl **REPEAT**<sup>2</sup>.

E' possibile impostare come numero di ripetizioni anche il contenuto di un contatore, la sintassi è: **#cnt**, dove cnt è il numero del contatore.

### Inserimento

Attivabile anche coi tasti scorciatoia **ALT+5**, fa comparire il box:

| Ripeti                                   |             |
|--|-------------|
| Numero di ripetizioni ...(1÷65535) : 12_ |             |
| Abbandono                                | Inserimento |

Dopo aver digitato nel campo argomento il numero di ripetizioni desiderato, inserire l'istruzione nel testo; una nuova linea, contenente l'istruzione ENDREP, verrà automaticamente creata sulla linea sottostante a quella inserita. Posizionare questa linea nel punto desiderato con la funzionalità di Spostamento.

In questo caso verranno inserite le linee:

```
REPEAT    12
ENDREP
```

### Modifica

Puntando a una linea con istruzione REPEAT appare il box di modifica:

| Ripeti                                   |               |
|--|---------------|
| Numero di ripetizioni ...(1÷65535) : 12_ |               |
| Abbandono                                | Cancellazione |



## 8.34 Programmazione

---

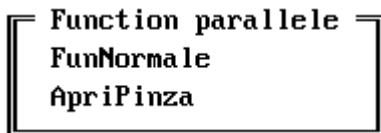
La **Cancellazione** della linea contenente l'istruzione comporta la cancellazione automatica della linea contenente l'istruzione ENDREP associata; cancellando la linea con ENDREP viene invece cancellata la corrispondente linea con l'istruzione REPEAT.

## 6 - ATTESA FUNCTION PARALLELA

Questa istruzione consente di testare se le function indicate sono in esecuzione, nel qual caso ne attende la fine per passare ad eseguire la linea di programma successiva.

Al dispositivo Attesa function parallela corrisponde l'istruzione Gpl **WPROG<sup>2</sup>**.

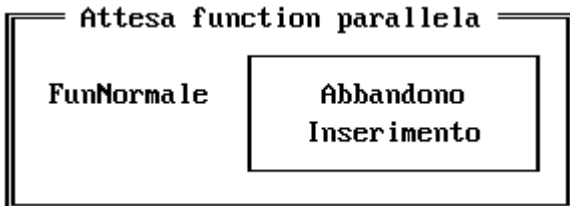
Selezionare nel menu che contiene l'elenco delle Function parallele quella desiderata.



Si può attendere il termine dell'esecuzione di un massimo di tre Function parallele sulla stessa linea di testo.

### Inserimento

Una volta selezionato il nome della function parallela desiderata, appare il seguente box:



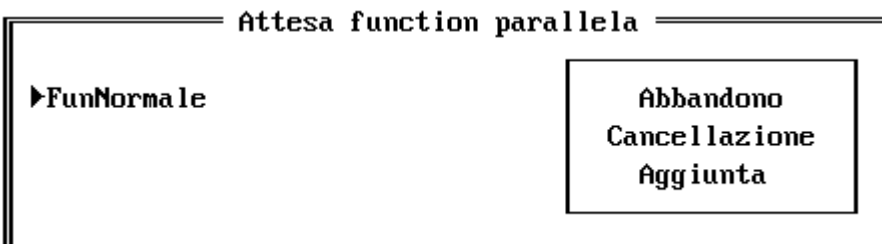
In questa modalità è possibile inserire l'attesa di una sola function su una nuova linea di programma, le altre potranno essere aggiunte con la modalità Modifica.

In questo caso verrà inserita la linea:

WPROG      FunSenzaparametri

### Modifica

Puntando a una linea contenente l'attesa di una function parallela, appare il box:

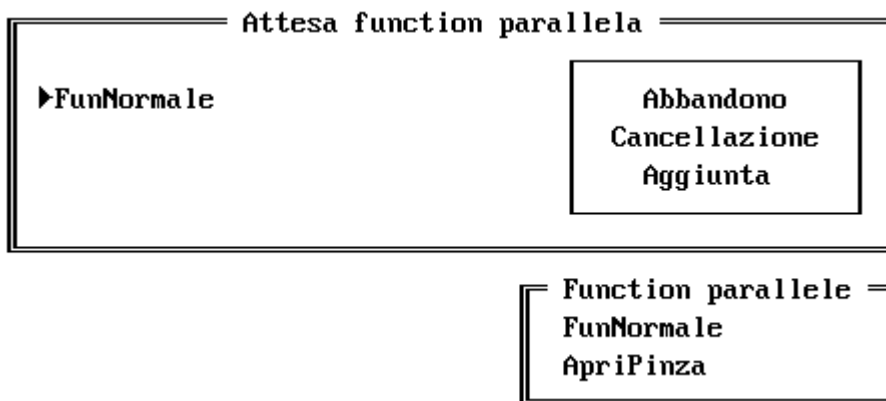


Nel box compaiono i nomi delle function parallele presenti sulla linea di programma. Sulla prima è visibile il cursore di selezione che corrisponde al carattere >.

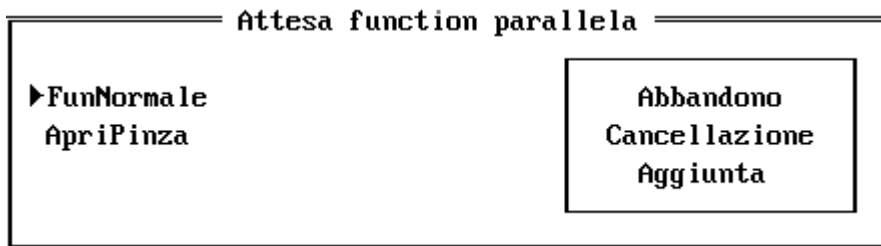
Volendo aggiungere una seconda function sulla linea è necessario premere il bottone Aggiunta.

### Aggiunta

La premuta di Aggiunta fa comparire il menu delle function parallele, permettendo così di selezionare il nome della nuova function:



La function aggiunta viene visualizzata sotto quella già presente:



Se nel box sono già presenti tre function il bottone Aggiunta non produce più nessun effetto, il sistema non permette inoltre di aggiungere una function già presente nel box.

Coi tasti  **cursore su/giù**  è possibile puntare alla function desiderata per l'operazione di **Cancellazione**; se è presente una sola function, l'azione sul bottone Cancellazione fa sì che venga cancellata la linea di programma.

Una volta confermata la modifica, nell'esempio appena descritto, la linea di programma sarà:

WPROG      FunSenzaParametri,ApriPinza

Vale qui la pena di ricordare che la linea di programma verrà eseguita solo se è terminata l'esecuzione di tutte le function parallele presenti sulla linea, in caso contrario l'esecuzione della linea verrà tenuta in sospenso finché tutte le function non saranno terminate.

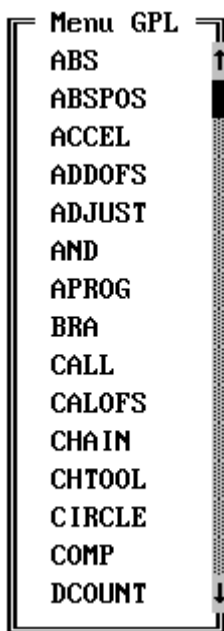
## 7 - ISTRUZIONE GPL

Consente l'inserimento di un'istruzione GPL1000 qualsiasi, le istruzioni sono descritte nel capitolo **Linguaggio GPL1000**.

Le istruzioni sono di due tipi, quelle con argomento e quelle senza.

### Menu GPL

Una volta attivata questa funzionalità, anche coi tasti scorciatoia **ALT+7**, appare il menu che contiene i nomi di tutte le istruzioni:



Selezionare nel menu l'istruzione desiderata.

Nei box di Inserimento e Modifica, il bottone Esecuzione compare solo per certe istruzioni.

Come detto in precedenza il box che appare dipende dal tipo di istruzione, il titolo del box è il nome dell'istruzione.

### Istruzione GPL con argomento

Sono istruzioni che richiedono, nella quasi totalità dei casi, la presenza di uno o più argomenti, separati tra loro dal carattere virgola o spazio. Per alcune istruzioni l'argomento è significativo anche se non c'è.

Se tra gli argomenti dell'istruzione è presente un'etichetta, se questa è nuova viene creata automaticamente una riga col suo nome sotto a quella puntata.

### Inserimento

Appare il seguente box:

|            |           |             |
|------------|-----------|-------------|
| ABS        |           |             |
| X_         |           |             |
| Esecuzione | Abbandono | Inserimento |

Dopo aver digitato l'argomento o gli argomenti dell'istruzione, inserire la stessa nel testo.

### Modifica

Puntando a una linea con istruzione GPL con argomenti, se la stessa non fa parte di quelle per cui si apre un box di modifica dedicato, appare il box di modifica:

|            |           |               |
|------------|-----------|---------------|
| ABS        |           |               |
| X_         |           |               |
| Esecuzione | Abbandono | Cancellazione |

### Istruzione GPL senza argomento

Sono istruzioni che non richiedono nessun argomento.

### Inserimento

Appare il seguente box:

|             |
|-------------|
| ECNT        |
| Inserimento |
| Esecuzione  |
| Abbandono   |

Inserire l'istruzione nel testo.

**Modifica**

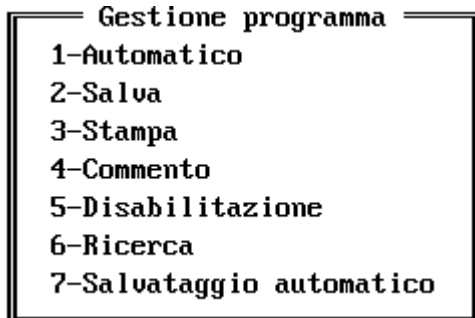
Puntando a una linea con istruzione GPL senza argomento, appare il box di modifica:



In questo caso non c'è nulla da modificare.

## 8 - GESTIONE PROGRAMMA

Questo menu consente l'accesso alle funzionalità elencate in questo menu:



### 1 - Automatico

Consente l'uscita dall'operatività Programmazione per tornare a quella di Automatico. Questa funzionalità è già stata descritta in precedenza nella sezione Uscita dalla Programmazione.

Se il programma è stato modificato, appare un box che richiede se si vuole o meno salvarlo. E' possibile salvare il programma con un altro nome qualora si volesse crearne uno nuovo.

Ricordiamo che questa funzionalità è attivabile anche premendo contemporaneamente i tasti **ALT+F4**.

### 2 - Salva

Consente di salvare il lavoro fatto fino a questo istante senza uscire dall'operatività; il suo funzionamento è identico a quello descritto nella sezione **Uscita dalla Programmazione**.

Se prima di salvare erano state fatte delle modifiche ai dati nel box di Modifica, queste vengono prima inserite nel programma, poi viene effettuata la memorizzazione.

Se è attiva la funzionalità **Salvataggio automatico**, prima di salvare viene fatta una copia di riserva e poi viene fatto il salvataggio. Se il tempo di salvataggio è uguale a 0, magari perchè durante il lavoro la funzionalità è stata disabilitata, l'eventuale copia di riserva viene cancellata.

### 3 - Stampa

Permette di stampare tutto il programma su cui si sta lavorando; prima di avviare la stampa appare un box con la richiesta di conferma per consentire l'abbandono dalla funzionalità.

#### 4 - Commento

Questa funzionalità, selezionabile anche premendo i tasti **CTRL+C**, permette di inserire una linea di commento; appare il box:

| Commento             |             |
|----------------------|-------------|
| questo è un commento |             |
| Abbandono            | Inserimento |

Puntando a una linea di commento il box di Modifica non compare, pertanto per modificare un commento è necessario cancellare la linea e crearne una nuova.

Nel file sorgente le linee di commento sono contraddistinte dal carattere **punto e virgola (;)**, posto all' inizio della linea.

#### 5 - Disabilitazione

Questa funzionalità, già descritta nella sezione **Disabilitazione e Riabilitazione linea programma**, è attivabile anche coi tasti **CTRL+D**; si rimanda alla sezione citata per il suo funzionamento.

Il box di Modifica mostrato di seguito compare solo puntando a una linea disabilitata e consente di riabilitarla o di cancellarla dal programma.

| Linea disabilitata |
|--------------------|
| Abbandono          |
| Abilita            |
| Cancellazione      |

#### 6 - Ricerca

Attivabile anche coi tasti **CTRL+F**, consente la ricerca di sequenze o sottosequenze di caratteri nel programma. Appare il box nel quale va digitata la sequenza di caratteri che si vuole cercare; la ricerca viene avviata premendo il tasto **ENTER**. Durante la ricerca non viene fatta nessuna distinzione tra caratteri maiuscoli e minuscoli.

|                    |
|--------------------|
| Cerca ... : CICLO_ |
|--------------------|



Se la stringa viene trovata, il puntatore di linea viene spostato sulla linea che la contiene. Ripremendo i tasti CTRL+F si ripresenta il box e premendo ENTER si passa a cercare la successiva e così via; arrivato a fine testo, il sistema ricomincia la ricerca dall'inizio del testo.

### 7 - Salvataggio automatico

Prima di procedere alla spiegazione di questa funzionalità vanno chiariti questi termini:

- **copia di riserva** è una copia di scorta del programma su cui si intende operare; questa viene effettuata prima del caricamento dello stesso solo se è attivo un tempo di salvataggio.
- **salvataggio** è la memorizzazione dello stato attuale del programma che viene effettuata dalla funzionalità Salvataggio automatico.

La funzionalità consente di effettuare una copia di salvataggio del programma su cui si sta operando, inoltre il sistema, allo scadere di un tempo stabilito dall'operatore, chiede di effettuare un salvataggio temporaneo del programma. Questo consente al sistema, in caso di cadute di tensione, di permettere all'operatore di tornare al programma su cui stava operando.

Una volta selezionata dal menu Gestione programma la voce Salvataggio automatico, appare il box:

```
Salvataggio automatico programma
Tempo in minuti ...[0=disabilita] : 10
```

Digitare il tempo in minuti e confermare col tasto ENTER. Il tempo viene fatto partire dopo che è stata fatta la prima modifica al programma.

In questo caso, attivando cioè la funzionalità, viene fatta una copia di riserva del programma.

Allo scadere del tempo appare il box:

```
Vuoi salvare il file ? ...(Si/No) : Si
```

Rispondendo **Si** viene salvato il lavoro fatto fino a questo momento, se la risposta non viene data entro 30 secondi, il sistema effettua automaticamente la memorizzazione; il timeout viene fatto ripartire alla successiva modifica.

Rispondendo **No** il timeout riparte immediatamente.

Se prima del salvataggio erano state fatte delle modifiche ai dati nel box di Modifica, queste non vengono prese in considerazione; questo è dovuto al fatto che può darsi che si stessero facendo solo delle prove, a fine salvataggio riapparirà il box con gli stessi dati.

La modifica del tempo ha effetto immediato, vale a dire che se era impostato un tempo di 10 minuti e questo stava per scadere, se lo si porta a 20 minuti, verranno sommati altri 10 minuti al tempo passato fino a quell'istante.

Per disabilitare la funzionalità digitare come tempo di salvataggio 0 o nessun dato, in questo caso il sistema cancella la copia di riserva fatta in precedenza.

Uscendo dalla Programmazione, rispondendo **Si** alla richiesta di salvare o meno il programma, il programma viene salvato e la copia di riserva cancellata, rispondendo **No** viene ripresa la copia di riserva.

Quando si entra in Programmazione ed è già attiva questa funzionalità, viene fatta una copia di riserva del programma selezionato.

Se si verifica uno **spegnimento accidentale** del sistema e la funzionalità era attiva, il sistema alla riaccensione si pone direttamente nell'operatività Programmazione sul programma su cui si stava operando, permettendo così di continuare il lavoro o di eliminarlo riprendendo la copia di riserva; in questo caso un messaggio avverte l'operatore di questa situazione. Al momento dello spegnimento accidentale sono valide le modifiche fatte con l'ultimo salvataggio.

Con salvataggio automatico attivo, e supponendo di lavorare sul programma PROVA.0A, si avranno i seguenti file:

| <u>File sorgenti</u> | <u>Copie di riserva</u> | <u>Descrizione</u>              |
|----------------------|-------------------------|---------------------------------|
| C:\PTP0\PROVA.0A     | C:\PTPSYS\PROVA.0AB     | programma sorgente              |
| C:\PTP0\PROVA.0AC    | C:\PTPSYS\PROVA.0AD     | programma compilato             |
|                      | C:\PTPSYS\DIRSOR.BAK    | informazioni nel direttorio     |
|                      | C:\PTPSYS\BACKUP.DAT    | dati relativi alla funzionalità |

Nell'esempio i nomi dei direttori di lavoro sono solo indicativi.

## 9 - CAMBIO STAZIONE

Questa funzionalità, attivabile anche coi tasti **ALT+9**, è utilizzabile nei sistemi con più stazioni e consente di passare al programma di un'altra stazione; viene automaticamente caricato il programma che ha lo stesso nome di quello su cui si stava agendo.

Nel caso il sistema sia multimodulo viene richiesto prima di selezionare il modulo desiderato, poi la stazione; prima di caricare il programma della stazione selezionata il sistema chiede se si vuole salvare o meno quello su cui si stava lavorando.

## T - TEST INGRESSI/USCITE

Consente la verifica degli stati degli I/O della stazione su cui si sta lavorando, permettendo anche di comandare le uscite; è attivabile anche coi tasti **ALT+T**. Appaiono i seguenti box:

| Ingresso | Simbolo |
|----------|---------|
| INP00    | 1000    |
| INP01    | 1001    |
| INP02    | 1002    |

| Uscita  | Simbolo |
|---------|---------|
| ▶ INP00 | 1000    |
| INP01   | 1001    |
| INP02   | 1002    |

In entrambi sono visualizzati sia le **Definizioni simboliche** che il **Bitporto** di appartenenza di tutti gli Ingressi e le Uscite, le righe illuminate indicano lo stato attivo.

Col tasto **TAB** ci si sposta sul box su cui si vuole agire. Sul box delle Uscite è possibile attivare o disattivare l'uscita premendo il tasto **Barra spaziatrice**.

Il tasto **ESC** consente l'abbandono da questa funzionalità.

## Tastiera di programmazione CAT90

### PREMESSA

La tastiera CAT90 viene utilizzata per l'esecuzione del movimento degli assi e di funzioni di ingresso/uscita.

La tastiera è composta da 27 tasti e 16 led ed è divisa in 3 zone:

1. tasti operativi
2. tasti di gestione assi
3. tasti funzione

La tastiera interagisce direttamente con l'operatività **Programmazione** consentendo l'apprendimento di movimenti asse e l'inserimento o la modifica delle istruzioni relative nel programma.

La gestione delle operatività legate a ciascun tasto è affidata a una function<sup>3</sup>.

### GENERALITA'

La tastiera viene **abilitata**, (LED ON acceso) in alcuni casi dell'operatività Programmazione, ed in particolare quando si è in modalità **Modifica** o durante un **Inserimento** di quote.

Viene invece **disabilitata**, (LED ON spento), quando si compie una esecuzione di linea o blocco di programma (**CTRL+ENTER**), da box col bottone **Esecuzione** o se si è in modalità **Inserimento** di una istruzione che non sia quella della gestione assi.

### DESCRIZIONE TASTI

Di seguito viene descritta la funzionalità dei vari gruppi di tasti.

#### Tasti operativi

|            |  |
|------------|--|
| <b>INS</b> | Viene usato per inserire una istruzione di movimento assi nella riga successiva a quella puntata dal cursore.<br>Viene inserita una istruzione di movimento con le quote attuali degli assi che sono stati mossi. Viene inoltre accodata all'istruzione l'attesa di Arresto assi IN QUOTA (stato Q). |
|------------|--|

---

<sup>3</sup>. Vedi **Parametri gestione tastierina** in Configurazione stazione.

|             |  |
|-------------|--|
| <b>MOD</b>  | Se si è puntati su una istruzione di movimento, permette di modificare le quote assi del programma con quelle attuali. Vengono modificate solo le quote degli assi mossi; eventuali assi mossi non presenti nell'istruzione, vengono aggiunti alla stessa. |
| <b>EXEC</b> | Viene usato per mettere in esecuzione l'istruzione puntata dal programma o il blocco selezionato (equivale a CTRL+ENTER).  |
| <b>STOP</b> | Interrompe l'esecuzione in corso, fermando gli eventuali assi in movimento con una rampa di decelerazione. Equivale al tasto ESC del PC.   |

I led di questi tasti vengono accesi dal PC quando viene accettata la premuta del tasto e spenti al suo rilascio.

#### Tasti di gestione assi

|                 |  |
|-----------------|--|
| <b>STEP/JOG</b> | Selezione del movimento STEP (led spento) o JOG (led acceso) degli assi.   |
| <b>PITCH</b>    | Ad ogni pressione di questo tasto si seleziona un diverso passo di spostamento per il modo STEP.   |
| <b>VEL</b>      | Permette di selezionare una velocità di movimento bassa (led spento) o alta (led acceso).  |
| <b>X+</b>       | Tasti per il movimento assi:<br>Se è selezionata la modalità JOG l'asse si muove alla velocità selezionata fino a quando non viene rilasciato il tasto.<br>Se è selezionata la modalità STEP ad ogni premuta corrisponde un movimento pari al passo selezionato. |
| <b>X-</b>       |  |
| <b>Y+</b>       |  |
| <b>Y-</b>       |  |
| <b>Z+</b>       |  |
| <b>Z-</b>       |  |
| <b>W+</b>       |  |
| <b>W-</b>       |  |
| <b>V+</b>       |  |
| <b>V-</b>       |  |



- La nuova linea verrà inserita nel programma.
- Alla linea verrà accodata anche l'attesa di Arresto assi IN QUOTA (stato Q).

### **Modifica istruzioni sugli assi**

Per modificare da CAT90 una linea di programma contenente uno o più spostamenti, bisogna procedere nel seguente modo:

- Puntare alla linea che si desidera modificare.
- Muovere l'asse o gli assi alle quote desiderate.
- Premere il tasto MOD.
- La linea di programma verrà modificata con le nuove quote.
- L'eventuale attesa di Arresto assi nella linea, non verrà alterata.