

M A N U A L E D ' U S O P T P 3 0 0

T.P.A. Tecnologie e Prodotti per l'Automazione

INTRODUZIONE

La console dell'unita' PTP 300 si puo' suddividere in cinque zone operative :

- Video a 9" (28 caratteri - 16 righe) con visualizzazione delle operativita' della macchina, dei blocchi in esecuzione, dei blocchi sotto controllo, delle posizioni degli assi x, y e z, dei dati di editing, dei messaggi in allarme.
- Tastiera completa alfanumerica.
- Pulsanti funzionali programmabili (tipo soft-key).
- Zona "cliente" con pulsanti di controllo movimento assi, pulsanti di emergenza, pulsanti di start - stop e reset.
- Minicassetta (256 Kbytes).

GESTIONE TASTIERA ED ANALISI COMANDI

L'unita' di controllo PTP 300 e' dotata di una interfaccia video-tastiera verso l'operatore orientata ad una programmazione guidata attraverso l'uso di tasti di significato variabile e dipendenti dal modo di funzionamento (struttura ad albero di soft-keys) che indicano all'utilizzatore i comandi e le operativita' di volta in volta disponibili.

Quando i comandi disponibili superano la capacita' rappresentativa dello schermo (al massimo 5 soft-keys) il tasto di "etc" permette di accedere ad un secondo gruppo, di ritornare al primo; il tasto di "END" permette di uscire da una certa operativita'.

Lo schermo video e' strutturato in diversi campi, di formato fisso, atti a :

- 1) visualizzazione quote, stato dell'operativita' della macchina, valore delle funzioni di macchina
- 2) messaggi diagnostici di errori e di allarmi
- 3) soft-keys.

ACCENSIONE

All'accensione lo schermo video dell'unita' di controllo PTP 300 viene azzerato con visualizzato, inizialmente, il messaggio *RAM TEST*, ad indicazione che il controllo sta eseguendo il test della RAM abilitata. Se tale stato iniziale non fosse superato, cio' indica un esito negativo del test stesso, con guasto presumibile sui componenti di RAM CMOS.

Normalmente, invece, la fase di inizializzazione viene superata in pochi secondi, ed il video impasinato con dati di quote e messaggi inizialmente nulli.

All'accensione del controllo, superato il test della RAM, possono essere visualizzate due segnalazioni di stato delle memorie permanenti gestite dal controllo :

- memoria RAM CMOS tamponata
- memoria EAROM

Come memoria RAM CMOS tamponata si intende la ram di *EDIT* (ram di programmi). In caso di controllo nuovo o di edit comunque non inizializzato (accesso all'operativita' di *EDIT* mai effettuato), o di perdita di dati sulla ram tamponata, il controllo scrive il messaggio *MEMORY EMPTY*. Tale stato richiede un RESET da plancia.

Come memoria EAROM si intende la memoria su cui vengono scritti i dati inseriti in Parametrica e nelle Function. Nel caso in cui non siano stati inseriti dati in EAROM o i dati inseriti siano stati perduti il controllo scrive il messaggio *EAROM EMPTY*. Tale stato richiede un RESET da plancia.

EDITING DELLA MEMORIA PROGRAMMI UTENTE

EDIT : Premendo la soft-key "EDIT" si abilita la
----- correzione di un programma precedentemente
 introdotta (da cassetta magnetica o manualmente)
 o l'introduzione di un nuovo programma.
 Il video viene cancellato e compare un "*" per
 indicare che il controllo e' "in attesa di un
 comando" da scegliere fra i seguenti :

1) DIR : [(DIR) (ENTER)]

 Sul video viene scritto l'elenco delle
 prime righe di tutti i programmi presenti
 in memoria.

 Per abilitare successive operazioni dobbiamo
 necessariamente selezionare il programma su
 cui lavorare, utilizzando il comando

 [(P) () (:) (nome del programma) (:) (ENTER)]

2) PROG : [(P) () (:) (nome del programma) (:) (ENTER)]

 E' obbligatorio scrivere l'intera frase (mostrata
 qui sopra tra le due parentesi quadrate),
 per avere a disposizione il programma su
 cui operare.

 Se il programma e' presente in memoria,
 compare la scritta "* nome del programma" e
 sotto, la prima riga del programma.

 Se, invece, il programma e' nuovo compare la
 scritta "* nome del programma" e sotto, la
 riga *END OF PROGRAM*.

 N.B. Lo spazio e' obbligatorio dopo il
 comando (P).

- 3) UP : [(U) () (XX) (ENTER)]
--
Selezione del blocco posto di "XX"
posizioni sopra quello attuale.

N.B. Scrivendo solamente (U) (ENTER)
si seleziona il blocco precedente.
- 4) DOWN : [(D) () (XX) (ENTER)]
--
Selezione del blocco posto di "XX"
posizioni sotto quello attuale.

N.B. Scrivendo solamente (D) (ENTER)
si seleziona il blocco seguente.
- 5) BEGIN : [(B) (ENTER)]
--
Selezione del blocco posto all'inizio
del programma.
- 6) END : [(EN) (ENTER)]
--
Selezione del blocco posto alla fine
del programma.
- 7) LIST : [(L) () (XX) (ENTER)]
--
Esegue la visualizzazione su video degli
"XX" blocchi a partire da quello selezionato.

N.B. L'ultimo blocco visualizzato non e', pero', quello su cui operare; questo infatti rimane quello da cui e' iniziata la visualizzazione.

La visualizzazione e' completa sugli "XX" blocchi e, a video completo, la visualizzazione di un successivo blocco avviene mediante l'eliminazione della prima riga di video (operazione di SCROLLING del video). E' possibile arrestare l'operazione premendo il tasto di SPAZIO (primo tasto in basso, a sinistra, sulla tastiera). Premendo successivamente lo stesso tasto viene aggiornata la visualizzazione di un solo blocco del listato, fino al termine degli "XX" blocchi; la premuta di uno qualsiasi degli altri tasti abilita la ripresa del listing free-run. Se gli "XX" blocchi esauriscono il programma, dopo l'ultimo blocco viene visualizzato il messaggio *END OF PROGRAM*. Le modalita' di arresto, di esecuzione blocco per blocco e di ripresa free-run della funzione di "LIST", sono valide anche per la funzione di direttorio dei programmi in memoria (comando DIR).

8) REVISE : [(R) (ENTER)]

Permette di modificare il blocco visualizzato. Ci si posiziona sul carattere desiderato spostando il cursore in avanti (con il tasto "spazio"). A questo punto tale carattere e la parte di linea successiva puo' essere modificata digitando la stringa desiderata.

9) INSERT : [(INS) (ENTER)]

Serve per inserire da 1 ad N righe di un testo. (In pratica per scrivere il testo del programma). L'inserzione avviene sempre sopra l'ultimo blocco selezionato. Una volta scritti (INS) (ENTER), il controllo attende l'inserzione dei blocchi. Ad ogni *ENTER* la scritta viene traslata in alto di una posizione.

N.B. Per concludere l'inserzione del testo si battono due *ENTER* consecutivi. Sulla riga a capo compare un "*". Per eventuali correzioni da apportare sul blocco in scrittura e' possibile utilizzare il tasto *DEL*, che sposta il cursore a ritroso sulla linea, cancellando un carattere per volta.

10) CLIN : [(C) (ENTER)]

Cancella l'ultimo blocco selezionato.

11) DEL : [(DEL) (ENTER)]

Cancella il programma dall'ultimo blocco selezionato (compreso) fino alla fine del programma stesso.

Per cancellare tutto un programma occorre posizionare il cursore sul primo blocco (blocco di intestazione) e poi selezionare il comando. Un esempio e' riportato qui di seguito :

```
* P : P01 : (ENTER) <-- selezione del programma
P01 <-- visualizza il blocco di
X100 <-- intestazione piu' il primo
      blocco di programma

* UP (ENTER) <-- posiziona sul blocco di
      intestazione

P01 <-- visualizza blocco di intesta-
      zione

* DEL (ENTER) <-- cancella il programma P01

* <-- "*" piu' cursore indica
      attesa di un nuovo comando.
```

12) CMEM : [(CMEM) (ENTER)]

Cancella tutta la memoria di EDIT rendendola completamente disponibile.

13) FIND : [(F) () (:) (testo qualunque) (:) (ENTER)]

Il testo puo' essere al massimo di 8 caratteri.

Con questa operazione viene ricercato in memoria, a partire dal blocco visualizzato, il primo blocco che contiene il testo tra i ":".

14) RP : [(RP) (ENTER)]

Permette la revisione del blocco di intestazione del programma (blocco di identificazione).

15) EXIT : [(E) (ENTER)]

E' il comando di uscita dall'operativita' di *EDIT*. Riporta il controllo alla configurazione-video iniziale.

La memoria RAM gestita in operativita' di edit viene tamponata e testata, ad ogni accensione del controllo, al fine di verificarne la tenuta dei programmi inseriti. In caso di controllo con RAM di edit non ancora inizializzata (accensione di controllo nuovo o con funzione di *EDIT* non ancora utilizzata) all'accensione viene segnalato il messaggio di *MEMORY EMPTY*, per indicare che non e' ancora stato compiuto alcun accesso alla memoria di editor.

Se tale messaggio compare invece con memoria di edit gia' inizializzata, o con eventuali programmi gia' inseriti, indica perdita di dati sulla RAM CMOS.

NOTE :

- Nel caso venga impostato un comando di programma senza aver richiesto accesso ad un programma specifico, viene diagnosticato errore con messaggio di * ILLEGAL COMMAND * ; Con il termine di "comandi di programmi" si intendono tutti i comandi elencati dopo il comando di (PROG), ad eccezione dei comandi :
 - . (CMEM)
 - . (EXIT)

- La segnalazione di * ILLEGAL COMMAND * e' data su ogni errore sintattico compiuto sia a livello di comandi, sia a livelli di blocchi di programmi.

UNITA' CASSETTA MAGNETICA

Premendo la soft-key "TAPE" si perviene all'operativita' di gestione dell'unita' a cassetta magnetica che costituisce la memoria di massa del controllo.

Sul video, a questo punto, comparira' sull'ultima riga, dedicata alle soft-keys, la dicitura "SIDE A SIDE B" come in figura 2.

Premendo "SIDE A" o "SIDE B" si seleziona il lato della cassetta su cui si vuole operare.

Il controllo verifica la presenza della cassetta nel driver ed il lato selezionato segnalando eventualmente gli errori con appositi messaggi di allarme :

- a) "CASSETTE MISSING" (cassetta mancante)
- b) "WRONG SIDE" (lato sbagliato)

In caso di selezione corretta compare la nuova configurazione di soft-keys (vedi figura 2).

Premendo uno qualsiasi di questi tasti si ha sul video l'indicazione "FUNCTION TAPE" ("comando selezionato").

La macchina a questo punto e' pronta a ricevere un altro comando.

Per uscire dall'operativita' di cassetta magnetica bisogna premere il tasto "END"

Se una delle operazioni non viene eseguita comparira' sulla riga degli allarmi il relativo errore :

- a) "FILE PROTECTED" (cassetta magnetica protetta)
- b) "TAPE DAMNGE" (cassetta magnetica danneggiata)
- c) "TEXT NOT FOUND" (testo non trovato)

I comandi previsti eseguono le operazioni :

- REC : (registrazione) Ricopia i programmi in memoria sulla cassetta, registrandoli in coda a quelli eventualmente presenti.
- REW : (riavvolgimento) Riavvolge il nastro.
- READ : (lettura) Con questo tasto viene caricato in memoria un programma o un gruppo di programmi registrati precedentemente sulla cassetta.
Prima di dare il comando READ bisogna scrivere da tastiera il nome del programma che si vuole leggere.
Se il programma non viene trovato ci sara' una segnalazione di allarme : "TEXT NOT FOUND".
Se, invece, interessa prendere visione del contenuto della cassetta, di tutti i "files" occupati, basta richiedere comandi di READ, senza alcuna specificazione, intervallati da accessi di operativita' di EDIT. Con tale procedura, il controllo rendera' accessibile, uno ad uno, in memoria di EDIT tutti i "files" su cassetta, nello stesso ordine di registrazione.
- FOR : (formattazione) Serve per inizializzare una cassetta nuova o per cancellare una cassetta usata.
Dopo questa operazione la cassetta e' predisposta per essere registrata con il contenuto della memoria
- PUT : (sostituzione) Registra un "file" sullo spazio di cassetta occupato, in precedenza, dal file medesimo. In tale senso, viene accettato solo dopo un comando di "READ" altrimenti da' segnalazione di *COMMAND ERROR*.
Tale comando permette, percio', di apportare correzioni su programmi gia' registrati su cassetta, riutilizzando lo stesso spazio di cassetta gia' registrato.
Bisogna fare molta attenzione all'uso di questa operativita' che in caso di un uso errato puo' danneggiare i programmi presenti in cassetta.

N.B. Con il termine "file" si intende definire l'area elementare di memoria gestita in operativita' di cassetta magnetica. La dimensione di un file e' pari a quella della memoria di edit, e cio' serve a definire una corrispondenza binaria fra la memoria di edit e le aree individuate su ogni cassetta.

La gestione della cassetta magnetica non e' effettuata consequenzialmente sulla memoria disponibile, ma ad aree successive di dimensione fissa appunto pari ad un file. In tali termini tutti i comandi operano su files e non su singoli programmi. La lettura di un programma, ad esempio, rende disponibile in memoria non il singolo programma ma l'intero file a cui lo stesso appartiene.

ESECUZIONE

Premendo la soft-key *ESEC* si entra nella operativita' esecutiva della macchina.

Sul video, a questo punto, compariranno le scritte

MAN AUTO SEMI SETP

(sull'ultima riga dedicata alle soft-keys come da figura 4).

MAN : operativita' manuale
AUTO : operativita' automatica
SEMI : operativita' semiautomatica
SETP : operativita' di set-point

Premendo una delle soft-keys si entrera' in un particolare ciclo esecutivo, contemporaneamente comparira' su video la scritta "FUNCTION ESEC" (tipo di operativita').

Per uscire dalla fase esecutiva della macchina occorre premere il tasto "END".

MODO MANUALE

Si entra in modo MANUALE premendo la soft-key *MAN*. Sul video apparirà, sull'ultima riga dedicata alle soft-keys, la scritta :

XY Z .

A questo punto si selezionano le coppie di assi che si vogliono muovere.

Premendo di nuovo la soft-key opportuna comparirà una nuova scritta (sempre sull'ultima riga) :

"Vmax Vmed Vmin".

Premendo una delle soft-keys si può scegliere il tipo di velocità con cui spostare gli assi. Sul video, a fianco delle scritte "X AXIS Y AXIS Z AXIS", compariranno le quote in tempo reale degli assi (XXXXX.XXX la cifra meno significativa è in millesimi di millimetro).

Si abilita il movimento free-run degli assi (uno per volta) attraverso i pulsanti di direzione A1+, A1-, A2+, A2-, (A1 --> X , A2 --> Y, A1 --> Z).

L'asse si muove solo tenendo premuto il pulsante, l'eventuale rilascio ferma il movimento.

MODO AUTOMATICO

Si abilita questo tipo di funzionamento premendo la corrispondente soft-key.
I modi di funzionamento automatico sono definiti in relazione ad un programma-utente (altrimenti definito main-program), con specifiche modalita' di abilitazione. Tali modalita' verranno precisate in un successivo paragrafo, mentre ora verranno precisati i termini generali di accesso ed interpretazione delle funzioni di esecuzione automatica.
Nel modo di esecuzione automatico e' possibile scegliere tra due distinte modalita' :

- SINGLE
- CONTINUE

In modalita' "SINGLE" viene abilitata l'esecuzione in ciclo singolo di un programma-utente. In tal modo ogni singola esecuzione deve essere abilitata dallo START da plancia.

In modalita' "CONT" viene abilitata l'esecuzione in ciclo continuo di un programma-utente : il solo consenso iniziale da START da plancia avvia percio' l'esecuzione in loop (ciclo) continuo.

Nel modo automatico (singolo o continuo) si puo' sospendere il funzionamento del controllo in un qualsiasi punto del programma, premendo il pulsante di *HOLD*. L'arresto avviene al termine dell'operazione in corso di esecuzione. Per ripartire, dal punto in cui l'esecuzione e' stata sospesa, basta ripremere il pulsante di *START*. La soft-key di "END" interrompe invece totalmente la esecuzione, senza possibilita' di ripresa da un punto intermedio del programma.

MODO SEMIAUTOMATICO

Si accede a questo tipo di funzionamento premendo la soft-key di * SEMI * .
In modo semiautomatico, viene abilitata la esecuzione di un programma con sospensione ad ogni istruzione. Il pulsante di START da' il consenso all'esecuzione dell'istruzione successiva. In tal modo, l'esecuzione di un programma viene abilitata, "passo-passo", mediante consenso manuale.

SET POINT ASSI

Viene usato per cercare i proximity di zero delle quote X, Y e Z .

Il ritrovamento del punto zero viene segnalato dall'azzeramento automatico delle quote X, Y e Z e dalla comparsa della scritta "SET POINT OK".

Il moto di set-point avviene con avviamento degli assi in direzione negativa fino all'impegno di un proximity. A questo punto la direzione del moto viene invertita e la velocità ridotta, fino al disimpegno del proximity. Una volta liberato si ricerca a velocità ridottissima la tacca di zero degli encoders.

L'esecuzione dell'operatività avviene premendo il pulsante di START. L'azzeramento delle quote video degli assi indica la fine dell'operazione. L'operatività di SET-POINT viene eseguita in modo indipendente su ogni asse.

GESTIONE PULSANTI

La plancia PTP 300 e' predisposta con vari pulsanti, alcuni relativi allo spostamento degli assi, altri di controllo.

I primi sono adibiti agli spostamenti MANUALI e vengono gestiti solo in quella particolare situazione. Servono per selezionare l'asse che si vuol muovere.

Il secondo gruppo di pulsanti comprende :

START

HOLD

RESET.

FUNZIONE TASTO *START*

Abilita l'esecuzione di un programma in automatico. Con lo stesso si fa partire la funzione di SET-POINT, ed avvia l'esecuzione di ogni singola istruzione di programma in operativita' di semiautomatico

E' sentito, e quindi gestito, solo nella fase di esecuzione del programma da eseguire o dopo una conferma di HOLD.

In parallelo al tasto di START da plancia viene gestita una linea di START come input da campo.

FUNZIONE TASTO *HOLD*

HOLD equivale al termine STOP; e' attivo solo in fasi esecutive e serve per sospendere un programma ad un certo blocco.

Dopo aver premuto *HOLD* si puo' ripartire dal punto attuale premendo *START*.

FUNZIONE TASTO *RESET*

Il tasto di *RESET* toglie la macchina da una situazione di errore in cui la stessa si e' venuta a trovare.

TASTO DI EMERGENZA

In qualunque caso di emergenza bisogna premere questo interruttore "a fungo". Questo toglie potenza agli azionamenti tenendo l'alimentazione solo all'elettronica di controllo.

Per ripristinare la condizione normale di funzionamento basta ruotare la parte superiore del "fungo".

DIAGNOSTICA

DIAGNOSTICA SUI MOTI

Sono previste due situazioni di diagnostica relative al movimento degli assi :

- "SERVOERROR" : segnala uno stato di errore di posizionamento durante un moto
- "NOT ENDED MOVEMENT" : segnala un errore di posizionamento al termine di un moto.

In tali situazioni di errore viene resettata ogni condizione di moto, con perdita delle quote reali. Da tale situazione si esce premendo il tasto *RESET*.

DIAGNOSTICA DI CASSETTA

Sono qui elencati i messaggi di errore nella gestione della cassetta magnetica :

- a) "CASSETTE MISSING"
- b) "WRONG SIDE"
- c) "FILE PROTECTED"
- d) "COMMAND ERROR"
- e) "CASSETTE DANMAGED"
- f) "TEXT NOT FOUND"

Tutti questi messaggi di errore si possono togliere premendo il tasto di *RESET*.

In caso di errata struttura del main program vi sono errori nella gestione della preparazione ad un'esecuzione automatica.

Il messaggio di errore usato e' quello indicato dalla lettera d).

EMERGENZE

Il fungo di emergenza gestisce un contatto accessibile all'operatore, per eventuale abilitazione di una linea di emergenza sul riaristino della potenza ai circuiti elettromeccanici gestiti dal controllo.

Il controllo gestisce inoltre un segnale di emergenza unificato, che risulta dalla serie delle emergenze presenti sulla macchina. Tale segnale viene gestito sulla linea di input al controllo *EMERG*.

In presenza di uno stato di emergenza il controllo segnala un messaggio di allarme su video, visualizzando *EMERGENCY* sulla riga di *WARNING:*. Sono inoltre resettate eventuali condizioni di moto e di esecuzioni automatiche (esecuzione di programma utente, di set-point) in corso. E' anche tolto il segnale di consenso alla potenza, gestito sulla linea di output *POWER*.

Il messaggio di allarme permane sul video fino al RESET da plancia.

Sono gestite altre linee di emergenza, solo nelle condizioni e secondo le modalita' che seguono.

Con il controllo in stato di esecuzione non di set-point e con assi in moto viene eseguito il test sui fine corsa di emergenza sugli assi.

LINGUAGGIO DI PROGRAMMAZIONE

I programmi di esecuzione automatica sono digitati da tastiera alfanumerica del PTP 300 in operativita' di *EDIT*, e salvati in memoria RAM CMOS tamponata.

Il programma di lavoro individuato da un nome univoco, risulta composto da una successione di blocchi, terminati dalla premuta del tasto *ENTER*. Ogni blocco di programma puo' contenere fino ad un massimo di 40 caratteri, utilizzabili sui due campi, esecutivo e di comando, separati dal carattere ":". Il campo di commento e' opzionale.

Il campo esecutivo e' costituito da un insieme di funzioni eseguite contemporaneamente o con una sequenzialita' fissa, costituite essenzialmente da codici alfanumerici e caratteri numerici. L'esecuzione in sequenza di piu' comandi e' assicurata solo su blocchi consecutivi.

N.B. E' opportuno inserire alcune definizioni, corrispondenti a termini utilizzati nel manuale.

- FASE DI EDIT : definisce la fase di inserzione dei programmi di esecuzione automatica. I comandi riconosciuti sono quelli definiti nel linguaggio di programmazione; il testo editato viene anche definito con la dicitura di "programma sorgente".
- FASE DI COMPILAZIONE : definisce la fase di traduzione del programma richiesto in esecuzione automatica, in forma direttamente interpretabile dalle procedure di esecuzione. Il programma tradotto secondo tali termini e' anche definito "programma oggetto".
- FASE DI ESECUZIONE : definisce la fase di esecuzione del programma-utente. Le modalita' di esecuzione sono definite dal programma e dall'operativita' di *EXEC*.

FORME DI ANALISI

L'analisi sintattica di ogni singolo blocco di programma e' compiuta direttamente in fase di inserzione dei blocchi (fase di EDIT). Se il blocco non e' corretto sintatticamente, viene visualizzato il messaggio di errore * SYNTAX ERROR * ed il blocco non e' accettato.

L'analisi logica di ogni blocco viene invece compiuta in fase di compilazione (fase di traduzione del programma sorgente). Se il blocco non e' logicamente corretto viene visualizzato un messaggio di errore (sulla riga di * WARNING : *) e la traduzione del programma interrotta. Il messaggio di errore e' in questo caso differenziato per il tipo di errore diagnosticato.

Unitamente al codice di errore, viene inoltre visualizzato il blocco del programma a cui la compilazione e' stata interrotta, permettendo cosi' una piu' diretta individuazione della causa di errore. Tale visualizzazione compare sulla riga video * MESSAGE *.

Su questa condizione di diagnostica il controllo risulta in stato di arresto di ogni operativita', con attesa di ripristino del funzionamento normale mediante reset da plancia.

COMANDI PREPARATORI

I comandi preparatori hanno lo scopo di impostare in fase di compilazione, dei parametri tipici del sistema. Sono abilitati due comandi di questo tipo, ad impostazione di un unico parametro:

ASS	:	quote in coordinate assolute
INC	:	quote in coordinate incrementali.

Nel caso non si sia specificato nessuno di questi comandi, viene assunta come condizione iniziale ASS, cioe' programmazione delle quote secondo coordinate assolute, su tutti gli assi.

L'operatore puo' impostare un comando preparatorio per tutti gli assi, oppure solo per alcuni. Nel primo caso e' sufficiente impostare il comando.

Esempio:

ASS (ENTER)

INC (ENTER)

Nel secondo caso e' necessario impostare il comando di selezione dell'asse richiesto (o degli assi richiesti).

I comandi riconosciuti in tal senso sono i seguenti:

ASSEX / ASSEY / ASSEZ

Esempio:

ASSEX () ASSEY () ASSEZ () ASS (ENTER)

FUNZIONI DI ATTESA

COMANDO : TIME

COMANDO : DEL

Il comando TIME permette di impostare una tabella di ritardi programmabili.

Il formato del comando e' il seguente :

TIME () XXXXX ; YYYYY ; (ENTER)

dove :

() indica il carattere di SPAZIO. Verra' sempre riportato, dove necessario;

(XXXXX), (YYYYY) sono numeri, con formato massimo di 5 cifre e di valore massimo pari a 65535. I valori impostati sono espressi in unita' di misura temporale convenzionale.

I ritardi programmabili sono in numero di 8, con possibilita' di impostarne un numero inferiore. Come delimiter fra i vari valori numerici viene riconosciuto il carattere di ";".
Con piu' comandi di TIME impostati, nel caso di un programma, in esecuzione risultano validi i ritardi impostati nell'ultimo comando tradotto.
In tal senso un uso corretto del comando prevede un'unica impostazione nell'ambito di un programma.

Il comando TIME deve ultimare con un (ENTER) e non puo' coesistere in un blocco con altri comandi.

L'operatore puo' fare riferimento ai valori impostati in modo sequenziale, con codici di identificazione numerici dal valore 0 al valore 7, a scansione della tabella degli 8 ritardi programmabili.

Il comando utilizzato e' :

DEL () n

con n = numero di identificazione del ritardo.

Esempio :

```

-
-
TIME ( ) 10;20;30;40      impostazione di 4 ritardi
-
-
DEL ( ) 2                  richiamo del ritardo
                           impostato con valore : 30.
```

COMANDO : VEL

L'operatore puo' richiamare i valori impostati di velocita' con il comando di formato :

VELX () n

con n = numero di identificazione della velocita'; sono riconosciuti valori da 0 a 7. E' qui stato utilizzato il comando di selezione velocita' sull'asse X; analogamente sugli altri assi sono riconosciuti VELY e VELZ.

In assenza di comandi di tipo VELi sono assunte come velocita' in presa quelle corrispondenti al comando :

VELi 0 ;

dove con "i" si intende indicare il generico i-esimo asse su cui non e' stata specificata una selezione di velocita'.

Esempio :

```

-
-
VELX 0                      Blocco A
-
-
VELX 2                      Blocco B
```

Blocco A : e' selezionata la prima velocita' (2700 giri motore/minuto).

Blocco B : e' selezionata la terza velocita' (giri motore/minuto).

Il formato del comando e' il seguente :

OFFS () X () +/- XXXX.XX; +/- YYYY.YY (ENTER)

dove :

OFFS e' il codice mnemonico fisso del comando;

X indica qui il carattere di identificazione dell'asse;

+/- attribuisce il segno proprio al correttore impostato;

XXXX.XX, YYYY.YY sono i valori numerici impostati. Il formato accettato e' da 1 a 4 cifre intere e da 0 a 2 cifre decimali; sulla parte intera l'unita' di misura e' in millimetri. Il punto e' richiesto solo in presenza di cifre decimali.

Come delimiter fra i valori numerici e' riconosciuto il carattere ";".

Su di un comando OFFS sono accettati valori numerici fino ad un massimo pari ai correttori definiti, in operativita' di parametrica, su ogni asse.

Il comando "OFFS" deve costituire un intero blocco di programma : deve perciò ultimare con un carattere di "ENTER".

In modo analogo a quanto gia' precisato per gli altri comandi di impostazione di valori tabulati, un corretto uso dell'istruzione "OFFS" ne prevede una impostazione unica, su ogni asse abilitato.

L'operatore puo' fare riferimento ai correttori di quote con il comando di formato :

CORX () nn

dove :

"nn" e' il numero di identificazione del correttore ed e' accettato con formato fisso di due cifre.

X e' qui utilizzato come indice di identificazione dell'asse. Con tale comando e' possibile selezionare dei correttori indipendentemente su ogni asse. Su controllo con tre assi abilitati vengono riconosciuti i comandi "CORX", "CORY" e "CORZ".

Per una selezione globale su tutti gli assi e' disponibile il comando :

COR () nn

in cui "nn" seleziona l'nn-esimo correttore su tutti gli assi.

In assenza di comandi tipo "COR" vengono assunti come correttori in presa quelli corrispondenti al comando "CORi 00", con "i" che indica il generico i-esimo asse.

Un caso tipico corrispondente ad impostare correttore nullo con primo correttore su tutti gli assi e' per esempio :

```

-
-
-
OFFS X 0:64:128 (ENTER)          Blocco A
-
-
CORX  0                          Blocco B
-
-
CORX  3                          Blocco C
```

Blocco A : vengono impostati i primi due correttori sull'asse X

Blocco B : e' selezionato il correttore corrispondente al valore numerico 0 (primo correttore impostato)

Blocco C : il correttore identificato come "CORX 3" non e' stato alterato in caso di programma, e cio' porta a selezionare il correttore impostato in operativita' di parametrica come "CORX/2".

TIPI DI LAVORAZIONE

I tipi di lavorazione riconosciuti sono tre e corrispondono ai casi esposti di seguito.

CASO 1) N : NORMALE

Nella lavorazione definita normale le quote degli assi sono riferite ad un'origine traslata, rispetto all'origine di set-point degli assi (zero macchina), degli offsets tipo "N". Tali offsets sono stati definiti in parametrica, uno per asse, come "OFFSET/N" (ciclo 6). A tale origine "NORMALE", le quote programmate sui due assi sono sommate, con il relativo segno proprio impostato.

CASO 2) M : MIRROR (speculare)

Nella lavorazione definita speculare, le quote dei tre assi sono riferite ad un'origine traslata, rispetto all'origine di set-point, degli offsets tipo "M". In parametrica gli offsets sono definiti "OFFSET/M". Rispetto tale origine "SPECULARE", le quote programmate sui tre assi sono :

- sottratte, con il relativo segno impostato, sull'asse X.
- sommate, con il relativo segno proprio impostato, sull'asse Y
- sommate, con il relativo segno impostato, sull'asse Z.

CASO 3) T : TRASLATA

Nella lavorazione definita traslata le quote dei due assi sono riferite ad un'origine traslata, rispetto all'origine di set-point, degli offsets tipo "T". In parametrica tali offsets sono definiti come "OFFSET/T". Rispetto tale origine "TRASLATA", le quote programmate sui tre assi sono sommate, con il relativo segno proprio impostato.

PARAMETRI DI MACCHINA

Con la dicitura "parametri di macchina" si intende definire l'insieme dei dati necessari alla definizione meccanica e funzionale di un controllo del tipo in considerazione.

Tale insieme di dati rimane memorizzato su EAROM, mediante la procedura selezionata dalla soft-key "WPAR". All'accensione del controllo, il contenuto della EAROM viene trasferito su RAM, a disposizione dell'operatore.

L'accesso diretto a tali parametri avviene mediante l'operativita' di parametrica, abilitata da soft-key "PAR".

All'accensione di un controllo nuovo (con parametri non inizializzati su EAROM) viene diagnosticato *EAROM EMPTY*, ad indicare l'esito negativo di test sul contenuto della EAROM. Tale diagnostica avverte della necessita' di impostare i parametri di macchina.

Una tale diagnostica su controllo gia' configurato, segnala invece una perdita di dati in EAROM.

In tale caso e' necessario un nuovo caricamento di parametri o da sostituzione di EAROM gia' inizializzata.

Non e' superfluo ribadire il fatto che la memorizzazione dei parametri su EAROM non e' automatica, all'uscita della parametrica (operativita' "PAR"), ma e' eseguita su richiesta specifica, tramite soft-key "WPAR".

OPERATIVITA' DI PARAMETRICA

Si accede all'operativita' di parametrica premendo la soft-key "PAR".
Per cancellare globalmente la parametrica occorre accedere all'operativita' dopo aver digitato da tastiera la parola "CLEAR" : tale condizione inizializza tutti i parametri al valore 0.
Analogamente e' possibile azzerare i parametri di un solo ciclo digitando "CLEAR" in accesso di ciclo medesimo.

Accedendo alla parametrica si presenta sul video lo elenco dei cicli disponibili.

Esempio :

CD1 RESOLUTION XX

XX e' lo stato di occupazione del ciclo : 00 in caso di ciclo sia' impostato o all'accensione del controllo;
01 in caso di parametrica (o ciclo) azzerato dallo utente.

RESOLUTION : nome del ciclo

CD1 : manca di identificazione del ciclo, trova corrispondenza in una soft-key di selezione del ciclo.

In caso di ciclo indicato in stato di occupazione "00", e' possibile accedere al ciclo seguendo una delle due operazioni seguenti :

- 1) - digitare "1"
- Premere la soft-key corrispondente alla marca di identificaione del ciclo.
- 2) - digitare "CLEAR"
- Premere la soft-key corrispondente alla marca di identificazione del ciclo.

Nel caso 1) si accede ai parametri sia' impostati, con possibilita' di visualizzare e/o modificare i parametri.

Nel caso 2) qualora non fosse digitato "1", apparirebbe la segnalazione di diagnostica *END OF BLOCKS*, resettabile da RESET di plancia.

Un parametro puo' essere definito di formato :

- NUMERICO
- ALFANUMERICO.

Nel primo caso viene richiesto di impostare valori solamente numerici, con il formato massimo comunque visualizzato a fianco del nome del programma.

Esempio :

```

-----
PD1 X OFFSET/N          0000.00 -----! formato numerico
                                     ! massimo definito
                                     ! in 4 cifre intere
                                     ! e 2 cifre decimali

```

I valori riconosciuti validi sono ad esempio i seguenti:

```

1000      ---->          1000.00
  0.01    ---->          0000.01
 10.01    ---->          0010.01
  100     ---->          0100.00

```

valore digitato

valore visualizzato a fianco del
nome del parametro

I valori non accettati sono invece ad esempio i seguenti:

```

  .01
10000
  0.001

```

Un parametro numerico puo' inoltre essere dotato di segno proprio (+/-). Nel tal caso cio' e' specificato nelle parti del manuale inerenti la definizione, in termini dettagliati, dei parametri stessi.

Nel secondo caso (parametro di formato alfanumerico) a fianco di parametro non impostato, non risulta visualizzato nulla, come indicazione del formato corrispondente. In tal caso risulta definito in modo risido anche il set dei valori impostabili.

I parametri di macchina sono organizzati in 6 cicli, accessibili tramite soft-key :

- CICLO 1 (C01) : parametri di risoluzione degli assi comandati in posizione del controllo
- CICLO 2 (C02) : velocita' base degli assi
- CICLO 3 (C03) : correttori base delle quote in asse X
- CICLO 4 (C04) : correttori base delle quote in asse Y
- CICLO 5 (C05) : correttori base delle quote in asse Z
- CICLO 6 (C06) : offset di traslazione delle quote di lavoro.

Per ogni asse controllato in posizione sono definiti tre parametri :

- 1) fattore di cifra base
- 2) fattore moltiplicativo
- 3) fattore divisivo.

Tali fattori permettono di definire la risoluzione su di ogni asse, nel modo illustrato di seguito. Come unita' di misura fondamentale sugli spostamenti si assume il millimetro, unita' che viene ridotta nei tre sottomultipli : decimo, centesimo, millesimo.

Il "fattore di cifra base" deve permettere di individuare la cifra base sottomultipla del millimetro, a cui la risoluzione dell'asse fa riferimento in modo conveniente.

Il "fattore di cifra base" deve essere specificato con una lettera (formato non numerico), con i seguenti significati riconosciuti :

- "D" : fattore di cifra base sul DECIMO
- "C" : fattore di cifra base sul CENTESIMO
- "M" : fattore di cifra base sul MILLESIMO.

I fattori moltiplicativo e divisivo sono da definire in numeri positivi di formato intero, fino a due cifre. Tali fattori devono permettere la conversione fra la posizione dell'asse data in unita' di cifra base e la posizione dell'asse data in numero di unita' di risoluzione (esempio numero di passi di encoder).

Per esempio si consideri un asse definito con i seguenti parametri :

- fattore di cifra base : C
- fattore moltiplicativo : 1
- fattore divisivo : 2

ad uno spostamento di 10 mm corrisponde :

$$(10 \text{ mm} * 100 * 1) / 2 = 500$$

|-----|
numero centesimi

|-----|
numero unita' di risoluzione

I dati dell'esempio corrispondono ad una trasmissione con risoluzione pari a :

2 centesimi (= 0.02 mm).

I parametri di risoluzione vengono impostati nel ciclo 1, identificato come :

C01 RESOLUTION.

I parametri sono specificati nei termini seguenti :

- P01; P02; P03 definiscono il fattore di cifra base rispettivamente sugli assi X, Y e Z.
- P04; P05; P06 definiscono il fattore moltiplicativo rispettivamente sugli assi X, Y e Z.
- P07; P08; P09 definiscono il fattore divisivo rispettivamente sugli assi X, Y e Z.

CORRETTORI DI QUOTE

Per ogni asse viene definito un set di correttori di quote, in numero massimo definito sulla base delle esigenze specifiche.

I correttori hanno formato massimo fino a (4.2) cifre, con parte intera in unita' di millimetri. Sono impostabili con segno proprio (+/-), in modo da caratterizzare il modo di correzione richiesto.

I correttori di quote risultano operativi in 'operativita' di esecuzione automatica.

I correttori di quote sono impostati nei cicli identificati come :

CO3	TOOL	OFFSET	X
CO4	TOOL	OFFSET	Y
CO5	TOOL	OFFSET	Z.

Ogni ciclo e' dedicato ad uno degli assi gestiti. Su ogni ciclo sono impostabili 38 correttori di quote, identificati con le diciture :

da "CORX/0" a "CORX/37"

da "CORY/0" a "CORY/37"

da "CORZ/0" a "CORZ/37".

Analogamente a quanto precisato per i parametri di velocita', i numeri che compaiono dopo il carattere di barra (da 0 a 37) corrispondono alle selezioni effettuabili, in programmazione, con i comandi CORX, CORY e CORZ.

OFFSET DI QUOTE

Per ogni asse viene definito un set di 3 offsets di quote. Questi parametri hanno formato massimo fino a (4.2) cifre, con parte intera in unita' di millimetri. Sono impostabili con segno proprio (+/-), analogamente ai correttori.

Gli offsets sono impostati nel ciclo 6, identificati come :

C06 AXIS OFFSET.

I parametri sono specificati nei termini seguenti :

- P01; P02; P03 definiscono i 3 offsets dell'asse X
- P04; P05; P06 definiscono i 3 offsets per l'asse Y
- P07; P08; P09 definiscono i 3 offsets per l'asse z.

Su ogni asse gli offsets sono identificati con le diciture :

OFFSET/N

OFFSET/M

OFFSET/T.

I caratteri che compaiono dopo il carattere di barra corrispondono alle selezioni effettuabili, in richiesta di esecuzione automatica, del tipo di esecuzione. I tre diversi caratteri individuano i seguenti tipi di lavorazione :

- NORMALE : N
- MIRROR (speculare) : M
- TRASLATO : T.

FUNZIONI DI INPUT / OUTPUT

COMANDO : OUT

Il formato del comando OUT e' il seguente :

OUT () XYYY

dove :

XX e' il numero di identificazione del porto. Sono validi valori da 00 a 05, per sei porti di output gestiti dal controllo PTP 300. Il formato XX e' di due cifre obbligatorie.

YYY e' il dato da scrivere sul porto. Sono riconosciuti validi valori da 000 a 255, di formato fino a 3 cifre. Il dato e' inteso in formato decimale.

Esempio :

-
-
-
OUT () 01 165
-
-
-

Sul porto di output "01" viene scritta la parola decimale "165". I pilotaggi sui porti avvengono comunque in formato esadecimale, e cio' porta ad impostare, come parola di comando, il valore esadecimale A5

La conversione (dato decimale) --> (dato esadecimale), avviene secondo il procedimento seguente :

dato decimale /16 = parte intera . parte decimale

$$\begin{array}{r} \text{--- parte intera (E1)} \\ \text{-----} \\ \text{. parte decimale} \\ \\ \text{X} \quad \quad 16 \\ \text{-----} \\ \text{(EO)} \end{array}$$

da cui :

dato esadecimale = E1EO
con E1 ed EO espressi in base 16, utilizzando le cifre esadecimali dallo 0 ad F.

Nell'esempio riportato, il pilotaggio avviene secondo lo schema seguente :

CODICE LINEA	STATO LOGICO	COMANDATO
001	1	
101	0	--- 5
201	1	
301	0	
401	0	
501	1	--- A
601	0	
701	1	

COMANDI : SET/RES

Sono comandi di pilotaggio di una singola linea di output; hanno formato :

SET () XYY

RES () XYY

dove :

X indica la linea del porto (da 0 a 7)

YY indica il numero di identificazione del porto di output. Sono accettati valori da 00 a 05, con formato fisso di due cifre.

I codici "XYY" di identificazione di tutte le linee di output, gestite dal controllo, sono riportate nelle tabelle di mappatura dell'input/output.

I comandi SET/RES sono utilizzati anche nella definizione dei comandi di pilotaggio su linee di output in modo temporizzato.

Il formato dei comandi e' il seguente :

SET () XYY () TZZZ

RES () XYY () TZZZ

con :

SET/RES comandi di pilotaggio di una linea di output.

TZZZ comando di impostazione del tempo di monostabile. Viene accettato un unico valore numerico "ZZZ", di formato fisso 3 cifre e di valore massimo pari a 255. Il valore "ZZZ" impostato esprime un tempo in unita' di misura convenzionale.

COMANDI DI ATTESA CONDIZIONATA

COMANDI : WIZ/WNZ

Sono programmabili 2 funzioni di attesa condizionata sullo stato logico di una linea di input. Il formato dei comandi e' il seguente :

WIZ () XYY

WNZ () XYY

dove :

X indica la linea del Porto (da 0 a 7), come per i comandi SET/RES.

YY e' il numero di identificazione del porto di input. Sono accettati valori da 00 a 03, con formato fisso di due cifre.

I codici "XYY" di identificazione di tutte le linee di input gestite dal controllo, sono riportati nelle tabelle di mappatura dell'input-output.

Il codice di comando WIZ indica attesa dello stato logico basso (wait if zero); per il comando WNZ e' attesa dello stato logico alto (wait not zero).

COMANDI DI VISUALIZZAZIONE

COMANDO : VIDEO

E' possibile inserire blocchi di lancio di messaggi-video, visualizzati su riga *MESSAGE*.

Il formato del comando e' :

VIDEO ().AAAAA. (ENTER)

con :

"AAAAA" messaggio di visualizzazione. Deve essere composto da un massimo di 20 caratteri, delimitati da un "." (punto) iniziale ed uno finale.

Il comando VIDEO deve essere unico su di un blocco di programma.

COMANDI DI IDENTIFICAZIONE DI BLOCCO

Ogni blocco di programma puo' essere identificato con la funzione di formato :

NXXXX

con :

"XXXX" numero di 4 cifre obbligatorie (da 0000 a 9999).

La funzione "N" diventa obbligatoria, su di un blocco, se questo deve essere riferito ad un'istruzione di salto tipo "BRA" :

BRA () PXXXX

dove :

"XXXX" indica il blocco "NXXXX" a cui il programma in esecuzione salta.

Il numero massimo di comandi di identificazione di riga utilizzabili in un programma (principale o sottoprogramma) e' di 30. Nessun limite e' invece posto sul valore numerico XXXX.

Esempio :

```
-  
-  
-  
BRA P0125 ----  
- |  
- | SALTO  
- |  
N0125 X100 <----  
-  
-  
-
```

COMANDI DI SALTO CONDIZIONATO

Sono programmabili due funzioni di salto (skip) di un blocco condizionato allo stato logico di una linea di input, identificata nel modo usuale.

Il formato dei comandi e' :

SKZ () XYY

SNZ () XYY

dove :

XYY identifica la linea di input, come nei comandi di attesa condizionata ad una linea di input o di pilotaggio di una linea.

I codici mnemonici dei due comandi sono :

SKZ : skip if zero

SNZ : skip not zero.

Esempio :

```

-
-
SKZ XYY          skip se la linea e' bassa  _
BRA P0801
SET ZWW          <-----|

```

Se la linea "XYY" e' bassa, lo skip e' eseguito scavalcando l'istruzione successiva di "BRA"; se la linea "XYY" e' alta, il programma prosegue senza salti, eseguendo percio' l'istruzione di salto non condizionato.

Dopo un'istruzione di skip viene riconosciuto obbligatoriamente un comando di salto incondizionato (BRA). Con tale gestione di condizionamenti ed organizzazione del programma, si intende realizzare una generica struttura di controllo.

Sulle istruzioni di salto incondizionato utilizzabili in un programma e' posto il limite massimo di 15 comandi; tale limite massimo vale anche per le istruzioni di salto condizionato (SKZ, SNZ).

FUNZIONI DI RIPETIZIONE DI CICLI

E' possibile programmare una sequenza ripetitiva di istruzioni (loop) mediante le istruzioni di fine ed inizio del loop, di formato :

```
REP ( ) XX          repeat
ENDR                end of repeat
```

L'istruzione di repeat abilita il loop di esecuzione sulle istruzioni comprese fra il "REP" e l'"ENDR"; "XX" e' il contatore associato al loop.

Esempio :

```

-
-
REP  XX          XX = numero di ripetizioni
-          -----
-          | "loop"
-          |
-          -----
ENDR
```

E' possibile programmare fino a 3 blocchi di repeat innestati. Un nesting maggiore determina segnalazione di errore.

Esempio : nestings di 3 loop

```

-
-
REP  XX  <-----
-
-
REP  YY  <-----
-
-
REP  ZZ  <-----
-
-
ENDR    <-----
-
-
ENDR    <-----
-
-
ENDR    <-----
```

COMANDI DI ARRESTO PROGRAMMA

COMANDO : HALT

E' comando di fine assoluta di un programma, e come tale deve essere l'ultima istruzione. Dopo l'istruzione di "HALT" vengono tradotti solamente i sottoprogrammi eventualmente associati al programma principale.

COMANDO : STOP

E' comando di sospensione di esecuzione, e come tale agisce come uno "stop programmato".
Con tale comando, il controllo assume uno stato di attesa di conferma di "START" da plancia.
A livello di funzionamento in esecuzione automatica (a ciclo singolo o continuo) gestisce dei blocchi di comandi in operativita' semi-automatica, abilitata da programma.
Un uso corretto dei comandi di fine (assoluta e di stop programmato) e' previsto che occupino un intero blocco di programma.

SOTTOPROGRAMMI

Con la definizione di SOTTOPROGRAMMA si intende un blocco di istruzione che deve iniziare con l'istruzione "SUB XX" e terminare con l'istruzione "RET". Un sottoprogramma puo' essere richiamato, in programma principale, mediante l'istruzione di "CALL XX".

Comandi :

SUB () XX	XX = numero di identificazione del sottoprogramma. Vengono accettati da 00 a 19, con possibilita' di programmare 20 sottoprogrammi.
RET	comando di chiusura di sottoprogramma.
CALL () XX	comando di chiamata del sottoprogramma identificato "XX".

I sottoprogrammi sono da considerare associati ad uno specifico programma principale, e devono perciò essere scritti in coda al programma stesso, dopo l'istruzione di HALT. Le istruzioni di "CALL" a sottoprogramma devono perciò fare riferimento a sottoprogrammi associati al programma.

Le eventuali istruzioni di "BRA" devono fare riferimento ad etichette del sottoprogramma stesso. Nell'ambito di un sottoprogramma e' possibile introdurre uno qualunque dei comandi di esecuzione :

- comandi di impostazione
- comandi preparatori.

Le eventuali istruzioni di moto abilitano solo moti in incremento di quote, al momento della chiamata al sottoprogramma.

Esempio : programma P01 con chiamata a 2 sottoprogrammi

```
      P:P01                intestazione del programma P01
      -
      -
      CALL  10
      -
      -
      CALL  01
      -
      -
      HALT                chiusura programma P01
      SUB   10            elenco sottoprogrammi associati
      VIDEO.AAAAAA.      a P01
      RET
      SUB   01
      VIDEO.BBBBBB.
      RET
      -
      -
      -
```

In fase di "EDIT", il programma P01 viene chiuso dopo l'inserimento dei sottoprogrammi ad esso annessi.

FUNZIONI

Con il termine di "funzione" si intende definire un insieme di blocchi di istruzioni impostabili, nella versione piu' generale, in forma parametrica e richiamabili con il comando di formato :

F () nn : A = XXX () B = YY C = ZZZ.Z (ENTER)

dove :

nn = e' il numero di identificazione della funzione richiamata. Ha formato di 2 cifre obbligatorie.

A, B, C indicano i parametri da specificare nella funzione.

(XXX), (YY), (ZZZ.Z) indicano i valori numerici attribuiti ai parametri (A, B, C, ...) rispettivamente.

Il comando "F" deve costituire un intero blocco di programma.

La parte mnemonica del comando e' obbligatoria fino ai ":" compresi, nel caso di funzione con parametri; i ":" sono opzionali per funzione senza parametri.

OPERATIVITA' DI FUNCTION

Il testo relativo al blocco di istruzioni di una funzione viene impostato con modalita' analoga a quelle gia' esaminate in "EDIT" di programmi, mediante l'operativita' di "FUNCTION". Tale operativita' e' abilitata dalla soft-key corrispondente. L'operativita' di "FUNCTION" e' gestita, a livello formale, in modo analogo all'"EDIT" di programmi. Vengono infatti utilizzati gli stessi comandi, con le eccezioni precisate di seguito.

- 1) FUNC : e' comando riconosciuto in sostituzione del programma PROG di "EDIT".

Il formato e' :

FUNC () : Fnn : (ENTER)

La lettera "F" sottolineata e', al solito, necessaria al controllo per l'identificazione del comando, mentre la parte restante del codice (UNC) e' facoltativa.

"Fnn" sta ad indicare il codice di identificazione della function (nome della function), "nn" indica un numero di 2 cifre ad identificazione della function.

- 2) WRITE : [(W) (ENTER)]

- E' comando di uscita dall'operativita' di function, con scrittura della function su "EPROM".

- 3) EXIT : [(E) (ENTER)]

- E' comando di uscita dall'operativita' di function, senza scrittura delle function su "EPROM".

ISTRUZIONI DI FUNZIONI

Distinti dai comandi di operativita' di "FUNCTION" sono i comandi di esecuzione, utilizzabili nella stesura di funzioni. Tali comandi sono estratti dalle istruzioni utilizzabili in edit di programmi, con l'opzione di una programmazione parametrica. Piu' precisamente e' possibile suddividere i comandi in due gruppi :

- 1) comandi non parametrici
- 2) comandi parametrizzabili.

I comandi parametrici possono richiedere l'introduzione di parametri simbolici, da specificare in dati numerici alla chiamata della funzione, nell'ambito di un programma utente (comando "F").

COMANDI NON PARAMETRICI

ASS : quote in coordinate assolute
INC : quote in coordinate incrementali
X Y Z : comandi di selezione degli assi.

TIPO COMANDO	COMANDO	FORMATO PARAMETRI	
AD UN PARAMETRO	DEL VELX, VELY CORX, CORY COR	1 CIFRA ; VALORE DA 0 A 7 " " " " " "	1 CIFRA ; VALORE DA 0 A 34 " " " " " "
	T	3 CIFRE ; VALORE DA 0 A 255	
	X, Y	MASSIMO : [4.3] CIFRE	
A DUE PARAMETRI		PRIMO PARAMETRO	SECONDO PARAMETRO
	OUT	2 CIFRE DA 0 A 05	3 CIFRE DA 0 A 255
	SET/RES	1 CIFRA DA 0 A 7	2 CIFRE DA 0 A 05
	WIZ, WNZ	1 CIFRA DA 0 A 7	2 CIFRE DA 0 A 03

COMANDI PARAMETRICI

Tali comandi risultano a loro volta distinti in piu' gruppi, in base al numero di parametri impostabili ed al formato utilizzabile per i parametri. Tale suddivisione e' illustrata nella tabella che segue.

ANALISI SINTATTICA DI FUNZIONI

L'analisi sintattica di ogni singolo blocco di funzione viene compiuta direttamente in fase di inserzione dei blocchi, perciò in operatività di "function", nei termini precisati di seguito.

In relazione alla parte mnemonica viene verificato se il comando inserito è fra quelli riconosciuti in operatività di function.

In relazione alla parte parametrica si possono presentare due casi :

- 1) in presenza di parametri simbolici viene verificato solo il numero di parametri inseriti
- 2) in presenza di parametri numerici è verificata anche la correttezza del formato, nella parte inerente il numero di cifre digitate.

È peraltro possibile inserire comandi parzialmente parametrizzati; in tal caso su ogni parametro viene applicata l'analisi sintattica corrispondente.

È inoltre sempre necessario suddividere la parte mnemonica dei comandi dalla parte parametrica, con l'inserimento di almeno uno spazio, e questo anche con comandi che normalmente sono accettati consecutivi, in operatività di "EDIT". Esempi tipici, a tale proposito sono i comandi di quote (X,Y, Z) e di tempo monostabile.

Se un blocco non è sintatticamente corretto, viene visualizzato il messaggio di errore *SYNTAX ERROR* con rifiuto del blocco.

L'analisi logica di ogni blocco viene compiuta in fase di compilazione (fase di traduzione di un programma-utente), in riferimento ad un comando "F". Se il blocco non è logicamente corretto, è visualizzato un messaggio di errore, con modalità che rientrano nel caso di analisi logica di un blocco di programma-utente.

In fase di compilazione, un comando "F" di impostazione di funzione viene tradotto con l'inserimento nel programma principale, in sostituzione del comando stesso, dell'insieme di istruzioni corrispondente. Tale operazione di sostituzione comporta anche l'impostazione di tutti i parametri numerici.

Ad illustrazione di quanto detto viene riportato un esempio.

```
P : P01                intestazione del programma in
-                       operativita' di "EDIT"
-
-
F 10 : A = 2   B = 05   C = 100   (Blocco A)
-
-
-
F : F10 :             intestazione di funzione in
ASS                       operativita' di "FUNCTION"
SET      AB
DEL      2
X        C
RES      AB
#
```

Blocco A) E' un blocco di richiamo di una funzione in operativita' di "EDIT". La funzione F10 richiede di specificare 3 parametri indicati in modo parametrico come : A, B, C.

In compilazione il blocco A) viene sostituito con le istruzioni seguenti :

```
ASS
SET      205
DEL      2
X        100
RES      205
#
```

MESSAGGI DI DIAGNOSTICA

A seguito di una programmazione scorretta, che pur rispetti la sintassi formale del linguaggio, in traduzione del programma il controllo genera una segnalazione di errore. La tabella riportata di seguito associa, ad ogni segnalazione di errore, le cause di diagnostica.

- ERROR 1 da impostazione di un numero maggiore di 65536. Origine di errore : comando "TIME".
- ERROR 1A da impostazione di un numero maggiore di 255. Origini di errore : comandi "OUT" e "T".
- ERROR 2 da impostazione di una cifra di identificazione di linea, su porto input/output, maggiore di 7. Origini di errore : comandi "SET", "RES", "WIZ" e "WNZ".
Da impostazione del numero di identificazione di un porto di input/output maggiore di 03 (sui porti di input) o di 05 (sui porti di output).
Origini di errore : comandi di "WIZ", "WNZ", "SET", "RES" e "OUT".
- ERROR 3 da richiesta di velocita' con numero di identificazione maggiore di 7; di ritardo con numero maggiore di 7.
Origini di errore : comandi di "VELI" e "DEL".
- ERROR 3A da impostazione di velocita', ritardi.
Origini di errore : comando "TIME".
- ERROR 4 da istruzioni di "repeat" in struttura di nesting maggiore di 3 o di mancanza di corrispondenza numerica fra comandi "REP" e "ENDR".
Origini di errore : comandi "REP", "ENDR", "RET" e "HALT".
- ERROR 5 da comandi tipo "SUB XX" e "CALL XX" con "XX" di valore numerico non ammesso.
- ERROR 5A da comando di "CALL" a sottoprogramma non impostato; da comando "F" di impostazione di function non impostata.

230. F BLOK
OF BARR

- ERROR 6 da comando di salto a blocco PXXXX con etichetta NXXXX non trovata; da istruzione di salto condizionato non seguita da istruzione di salto incondizionato (tipo "BRA").
- ERROR 7 da overflow di occupazione di memoria in traduzione di programma principale o di sottoprogrammi.
Da overflow di occupazione di memoria destinata all'organizzazione dei programmi-sorgenti richiamati in main-program.
- ERROR 7A errore da eccedenza nel numero di identificatori di riga. Origine di errore : comando NXXXX.
- ERROR 7B errore da eccedenza nel numero di istruzioni di salto incondizionato. Origine di errore : comando "BRA".
- ERROR 7C errore da eccedenza sul numero di istruzioni di CALL ammesse.
- ERROR 8 errore per mancanza dell'istruzione di *HALT* (al termine di un programma principale) o di un'istruzione di *RET* (a chiusura di un sottoprogramma).
- ERROR 9A da comandi con richiesta di specificazione del codice di asse, con codice non riconosciuto valido.
Origini di errore : comandi "VELi", "OFFSi" e "ASSEi".
- ERROR FF da traduzione di comando "F nn" di impostazione di funzione, in relazione alla parte di specificazione dei parametri.
Da traduzione di comandi non abilitati, in procedura di sottoprogramma.