

Volume

2

**AxesBrain**

---

Linguaggio di programmazione ISO



# Linguaggio di programmazione ISO

AxesBrain

---

## Linguaggio di programmazione AxesBrain ISO

AB&T Tecnologie Informatiche™  
Via dell'About, 2/A • 10015 Ivrea  
Fax +39 0125 234397  
[www.bausano.net](http://www.bausano.net)  
[info@bausano.net](mailto:info@bausano.net)

### Informazioni legali

Le informazioni con questo documento, incluse URL e gli altri riferimenti sul sito Internet, possono cambiare senza alcun avvertimento.

A meno di specifica annotazione, i riferimenti a compagnie, organizzazioni, prodotti, persone ed eventi sono fittizi e non associate con reali compagnie, organizzazioni, prodotti, persone ed eventi.

L'AB&T Tecnologie Informatiche™ può registrare, licenziare, richiedere il copyright o marchi e rivendicare la proprietà intellettuale a tutti gli argomenti trattati in questo documento.

Senza limitare i diritti sotto copyright, nessuna parte di questo documento può essere riprodotta, o modificata o trasmessa sotto ogni forma o mezzo ( elettronico, meccanico, per fotocopiare, per registrare, od altro), senza l'espressa autorizzazione dell' AB&T Tecnologie Informatiche™.

Eccetto che per accordi scritti con la AB&T Tecnologie Informatiche™ la fornitura di questo documento non autorizza nessuno alla registrazione, a dare licenze, a richiedere il copyright o marchi e a rivendicare la proprietà intellettuale agli argomenti trattati in questo documento.

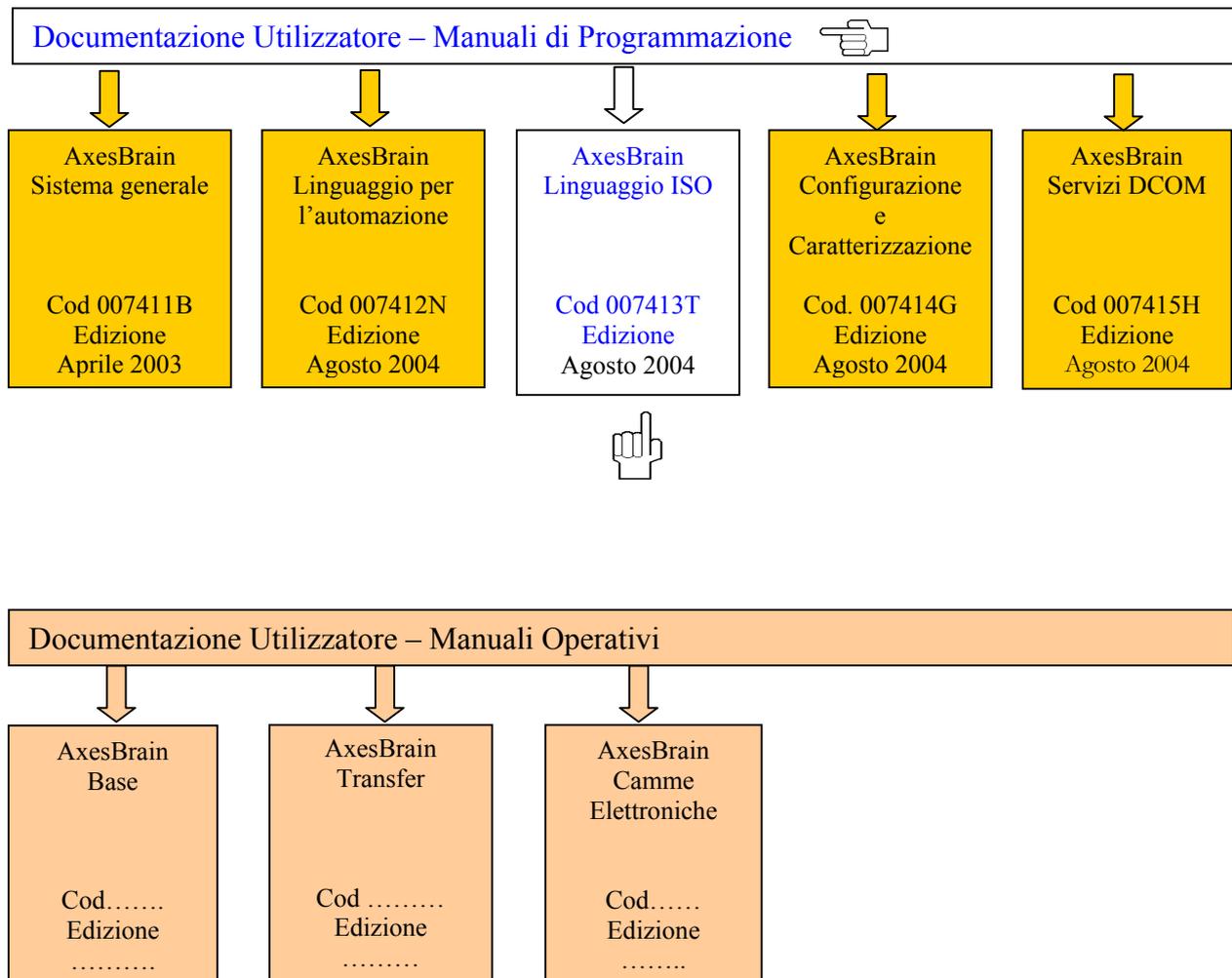
AxesBrain™, VisAlgo™, ScadaMERCURIO™ sono marchi registrati

©AxesBrain, ©VisAlgo sono tutelati da copyright

ActiveX, DirectX, JScript, Microsoft, Microsoft Press, MS-DOS, Visual Basic, Visual C++, Win32, Win32s, Windows, WDM, Windows NT, Windows 2000 e Windows Me sono o prodotti o marchi registrati dalla Microsoft Corporation negli Stati Uniti e/o negli altri Paesi.

I nomi di compagnie e prodotti citati in questo documento possono essere marchi registrati dai loro proprietari.

## Panoramica della Documentazione Utilizzatore AxesBrain Linguaggio ISO



## INDICE DEGLI ARGOMENTI

<b>1</b>	<b>INTRODUZIONE.....</b>	<b>1-1</b>
1.1	COME UTILIZZARE QUESTO MANUALE.....	1-1
1.2	SUPPORTO CLIENTI.....	1-1
1.3	LINGUAGGIO DI PROGRAMMAZIONE ISO ADATTO PER DESCRIVERE CICLI DI LAVORAZIONE.....	1-2
<b>2</b>	<b>ARCHITETTURA.....</b>	<b>2-1</b>
<b>3</b>	<b>LE BASI DELLA PROGRAMMAZIONE CN.....</b>	<b>3-1</b>
3.1	BLOCCO PROGRAMMA.....	3-1
3.2	PAROLA DI PROGRAMMA.....	3-2
<b>4</b>	<b>REGOLE DI PROGRAMMAZIONE.....</b>	<b>4-1</b>
<b>5</b>	<b>ABBINAMENTI RISORSE AI SINGOLI PROCESSI.....</b>	<b>5-1</b>
<b>6</b>	<b>COMANDI DI PREDEFINIZIONE NEL “SISTEMA.TXT” .....</b>	<b>6-1</b>
<b>7</b>	<b>ISTRUZIONI SPECIALI PER L’ABBINAMENTO RISORSE AL PROCESSO ISO .....</b>	<b>7-1</b>
<b>8</b>	<b>TABELLA ORIGINI.....</b>	<b>8-1</b>
8.1	DEFINIZIONI DEL FILE PER LA TABELLA ORIGINI.....	8-1
<b>9</b>	<b>TABELLA DEGLI UTENSILI.....</b>	<b>9-1</b>
9.1	DEFINIZIONI DEL FILE PER LA TABELLA TOOL .....	9-1
<b>10</b>	<b>TABELLA DEI CORRETTORI D PER GLI UTENSILI .....</b>	<b>10-4</b>
10.1	DEFINIZIONI DEL FILE PER LA TABELLA CORRETTORI .....	10-4
<b>11</b>	<b>TABELLA DEI PARAMETRI.....</b>	<b>11-5</b>
11.1	DEFINIZIONI DEL FILE PER LA TABELLA PARAMETRI .....	11-5
<b>12</b>	<b>TABELLA DELLE FUNZIONI M E G SPECIALI.....</b>	<b>12-1</b>
12.1	FUNZIONE G89.....	12-9
<b>13</b>	<b>INTERFACCIAMENTO CON LOGICA DI MACCHINA .....</b>	<b>13-1</b>
<b>14</b>	<b>FUNZIONI SPECIALI.....</b>	<b>14-1</b>
14.1	ISTRUZIONI DI I/O .....	14-1
14.2	ISTRUZIONE PERL’ AVVIAMENTO ALTRI PROCESSI ISO .....	14-2
14.3	ISTRUZIONE PER L’AVVIAMENTO PROCESSI PLC .....	14-3
14.4	ISTRUZIONE PER LA SINCRONIZZAZIONE DI PROCESSI ISO o PLC .....	14-4
14.5	RICHIAMO DI PROGRAMMI ISO ALL’ESTERNO DEL FILE. ....	14-5
<b>15</b>	<b>PIANO LAVORO.....</b>	<b>15-1</b>
15.1	PIANI DI LAVORO PRINCIPALI .....	15-1
<b>16</b>	<b>FUNZIONI DI MOVIMENTO.....</b>	<b>16-1</b>
<b>17</b>	<b>FUNZIONI G .....</b>	<b>17-1</b>

17.1	FUNZIONI PREPARATORIE ( G ).....	17-3
17.2	G00: POSIZIONAMENTO IN RAPIDO.....	17-3
17.3	G01: INTERPOLAZIONE LINEARE .....	17-5
17.4	G02-G03 : INTERPOLAZIONE CIRCOLARE.....	17-7
17.5	G04: SOSTA PROGRAMMATA.....	17-10
17.6	G08,G09: GUARDARE AVANTI OFF/ON .....	17-10
17.7	G17,G18,G19: SELEZIONE DEL PIANO DI LAVORO .....	17-13
17.8	G30: DECELERAZIONE FORZATA .....	17-13
17.9	G40,G41,G42: COMPENSAZIONE RAGGIO UTENSILE NEL PIANO .....	17-13
17.10	G49: DICHIARAZIONE VALORE CORRETTORE RAGGIO.....	17-16
17.11	G50: FINE ROTOTRASLAZIONE.....	17-17
17.12	G51: ROTOTRASLAZIONE .....	17-17
17.13	G60: FINE FATTORE DI SCALA.....	17-17
17.14	G61: FATTORE DI SCALA .....	17-17
17.15	G62:CENTRO DEL CERCHIO K1 ASSOLUTO K2 INCREMENTALE.....	17-18
17.16	G70,G71: COMMUTAZIONE METRICO/POLLICI .....	17-18
17.17	G78,G79 IMPOSTAZIONE TANGENZIALE AL PERCORSO 2D.....	17-19
17.18	G90,G91: PROGRAMMAZIONE ASSOLUTA E PROGRAMMAZIONE INCREMENTALE.....	17-23
17.19	G92: SHIFT DELLE ORIGINI ( G52).....	17-23
17.20	G201 PROGRAMMAZIONE CILINDRICA .....	17-25
17.21	G100-G1999: FUNZIONI PROGRAMMABILI.....	17-27
<b>18</b>	<b>CICLI FISSI DI FORATURA .....</b>	<b>18-28</b>
18.1	INTRODUZIONE .....	18-28
18.2	USO DEI CICLI FISSI.....	18-28
18.3	PARAMETRI DEI CICLI FISSI.....	18-29
18.4	CICLO FISSO G81 PER FORATURA .....	18-30
18.5	CICLO FISSO G82 PER FORATURA CON SOSTA.....	18-31
18.6	CICLO FISSO G83 PER FORATURA CON ROTTURA TRUCIOLO .....	18-32
18.7	CICLO FISSO G84 PER MASCHIATURA .....	18-33
18.8	CICLO FISSO G85 PER ALESATURA .....	18-35
18.9	CICLO FISSO G86 PER BARENATURA.....	18-36
18.10	CICLO FISSO G89.....	18-38
<b>19</b>	<b>FUNZIONI M.....</b>	<b>19-1</b>
19.1	FUNZIONI MISCELLANEE ( M ).....	19-2
19.2	M00: ARRESTO PROGRAMMATO .....	19-2
19.3	M03,M04,M07,M08,M13,M14: AVVIO MANDRINO E REFRIGERANTE.....	19-2
19.4	M05,M09: ARRESTO MANDRINO E REFRIGERANTE .....	19-3
19.5	M19: POSIZIONE ASSE MANDRINO.....	19-3
19.6	M10,M11: BLOCCAGGIO ASSI.....	19-3
19.7	M06: CAMBIO UTENSILE .....	19-4
19.8	M30: FINE PROGRAMMA .....	19-8
19.9	M31-M1999: FUNZIONI PROGRAMMABILI .....	19-8
<b>20</b>	<b>FUNZIONI T.....</b>	<b>20-1</b>
<b>21</b>	<b>FUNZIONI F, S E O .....</b>	<b>21-2</b>
21.1	F: AVANZAMENTO .....	21-2
21.2	S: VELOCITÀ DI ROTAZIONE MANDRINO .....	21-2
21.3	O: ORIGINI .....	21-3
<b>22</b>	<b>PARAMETRI.....</b>	<b>22-4</b>

<b>23</b>	<b>FUNZIONI DI CONTROLLO PROGRAMMA.....</b>	<b>23-1</b>
23.1	RIPETIZIONI DI PARTE DI PROGRAMMA .....	23-1
23.2	SOTTOPROGRAMMI INTERNI AL PROGRAMMA .....	23-1
<b>24</b>	<b>ALTRE FUNZIONI.....</b>	<b>24-1</b>
24.1	INTRODUZIONE COMMENTI.....	24-1
24.2	N: NUMERO DI SEQUENZA.....	24-1
24.3	“ “: STRINGA DI VISUALIZZAZIONE.....	24-1
24.4	? RICHIESTA VALORE DIRETTO PER PARAMETRO O GLOBALE .....	24-1



# CAPITOLO 1

## 1 INTRODUZIONE

### 1.1 Come utilizzare questo manuale

Questo manuale descrive come utilizzare le funzioni dedicate alla programmazione **ISO** nell'ambiente del sistema AxisBrain

Il suo scopo è provvedere ai programmatori un rapido aiuto e supporto specialmente per funzionalità che sono utilizzate infrequentemente e fornire offrire una rapida guida in merito.

Il manuale ha, tuttavia, brevi testi descrittivi facilmente comprensibili.

Familiarizzate prima con i simboli rappresentati in seguito al fine di interpretarne il significato nelle occasioni del loro utilizzo nei capitoli seguenti.

Se Lei ha alcune domande o suggerimenti di miglioramento o problemi di riunione che non sono coperto nell'AB&T-documentazione, per favore contatti la nostra assistenza tecnica:

### 1.2 Supporto clienti

Per domande o suggerimenti di miglioramento o problemi che non sono nella documentazione dell'AB&T, per favore contattare la nostra assistenza tecnica:

**AB&T** Tecnologie Informatiche™

**Via dell'About,2/A**

**10015 Ivrea ( TO)**

**Telephone: +39/0125 632826**

**Fax: +39/0125 234397**

[www.bausano.net](http://www.bausano.net)

[info@bausano.net](mailto:info@bausano.net)

### 1.3 Linguaggio di programmazione ISO adatto per descrivere cicli di lavorazione

*Il AxesBrainISO linguaggio di programmazione è indicato per la definizione dei cicli di lavoro per l'automazione .*

Per descrivere il ciclo di lavoro di una macchina automatica è necessario avere un linguaggio in grado di programmare il ciclo di lavoro. Esistono numerosi linguaggi di programmazione, il più diffuso è il linguaggio ISO o GCODE. Le caratteristiche del AxesBrainISO sono quelle di essere MultiProcesso in modo da poter soddisfare le esigenze delle macchine Transfer e delle macchine multitesta. Un'altra specifica del linguaggio è la possibilità di effettuare la medesima lavorazione su più teste parallele.

Il presente manuale intende fornire le nozioni essenziali per la caratterizzazione del prodotto **AxesBrainISO** .

In particolare esso è così organizzato:

Il **Capitolo 2 Architettura** descrive l'architettura del sottosistema di programmazione nel formato ISO e fornisce uno schema a blocchi dello stesso.

Il **Capitolo 3 Le basi della programmazione CN** fornisce i rudimenti inerenti la programmazione di partprogram di lavorazione

Il **Capitolo 4 Regole di programmazione** fornisce i rudimenti da utilizzare per la stesura di part program.

Il **Capitolo 5 Abbinamenti risorse ai singoli processi** introduce al concetto di abbinamento degli assi fisici o virtuali al sistema di coordinate cartesiane di ciascuna testa ( unità) di lavorazione.

Il **Capitolo 6 Comandi di predefinizione nel "SISTEMA.TXT"** descrive la possibilità di inserire dei comandi di abbinamento delle risorse al partprogram ISO

Il **Capitolo 7 Istruzioni speciali per l'abbinamento risorse al processo ISO** illustra le istruzioni speciali da utilizzare per l'abbinamento assi ISO/assi fisici .

Il **Capitolo 8 Tabella Origini** definisce il file da utilizzare per la programmazione delle tabelle e la loro attivazione o disattivazione.

Il **Capitolo 9 Tabella degli utensili** definisce il file da utilizzare per la programmazione delle tabelle utensili e la loro attivazione o disattivazione.

Il **Capitolo 10 Correttori D degli utensili** definisce il file da utilizzare per la programmazione delle tabelle dei correttori D utensili e la loro attivazione o disattivazione.

Il **Capitolo 11 Tabella dei Parametri** definisce il file da utilizzare per la programmazione delle tabelle parametri utilizzata per definire i parametri utilizzati nelle lavorazioni.

Il **Capitolo 12 Tabella delle Funzioni M e G speciali** definisce il file da utilizzare per la programmazione di funzioni di automazione speciali da utilizzare per la lavorazione ( p.pe. G89).

Il **Capitolo 13 Interfacciamento con LOGICA di MACCHINA** affronta il problema dell'interfacciamento degli automatismi con la logica di macchina, tramite 16-32 bit consecutivi per lo scambio dei sincronismi.

Il **Capitolo 14 Funzioni speciali** elenca le istruzioni speciali per la connessione diretta con i segnali IO e gli assi.

Il **Capitolo 15 Piano lavoro** definisce i piani di lavoro principali relativi agli assi lineari e/o rotanti.

Il **Capitolo 16 Funzioni di movimento** elenca e descrive le funzioni Gnn di movimento in relazione agli assi macchina di lavoro o posizionamento.

Il **Capitolo 17 Funzioni G** elenca e descrive tutte le funzioni preparatorie Gnn e fornisce i rudimenti essenziali sull'utilizzo e programmazione.

Il **Capitolo 18 Cicli fissi di foratura** introduce al concetto di ciclo fisso e descrive i cicli fissi di foratura disponibili ed utilizzabili nel sistema.

Il **Capitolo 19 Funzioni M** descrive ed elenca le funzioni varie o miscellanee Mnn disponibili ed utilizzabili nel sistema .

Il **Capitolo 20 Funzioni T** descrive le funzioni di cambio utensile Tnn disponibili ed utilizzabili nel sistema.

Il **Capitolo 21 Funzioni F, S e O** descrive le funzioni di avanzamentoF, rotazione mandrino S e origini O disponibili ed utilizzabili nel sistema.

Il **Capitolo 22 Parametri** illustra i concetti base inerenti alla programmazione parametrica con l'utilizzo dei parametri Pnnn disponibili.

Il **Capitolo 23 Funzioni di controllo programma** illustra i concetti base inerenti alla programmazione utilizzando le funzioni di ripetizione di parti di programma ( L ) e/o richiamo di sottoprogrammi.

Il **Capitolo 24 Altre Funzioni** descrive le funzioni di introduzione di commenti e numero di sequenza disponibili ed utilizzabili nel sistema.

## CAPITOLO 2

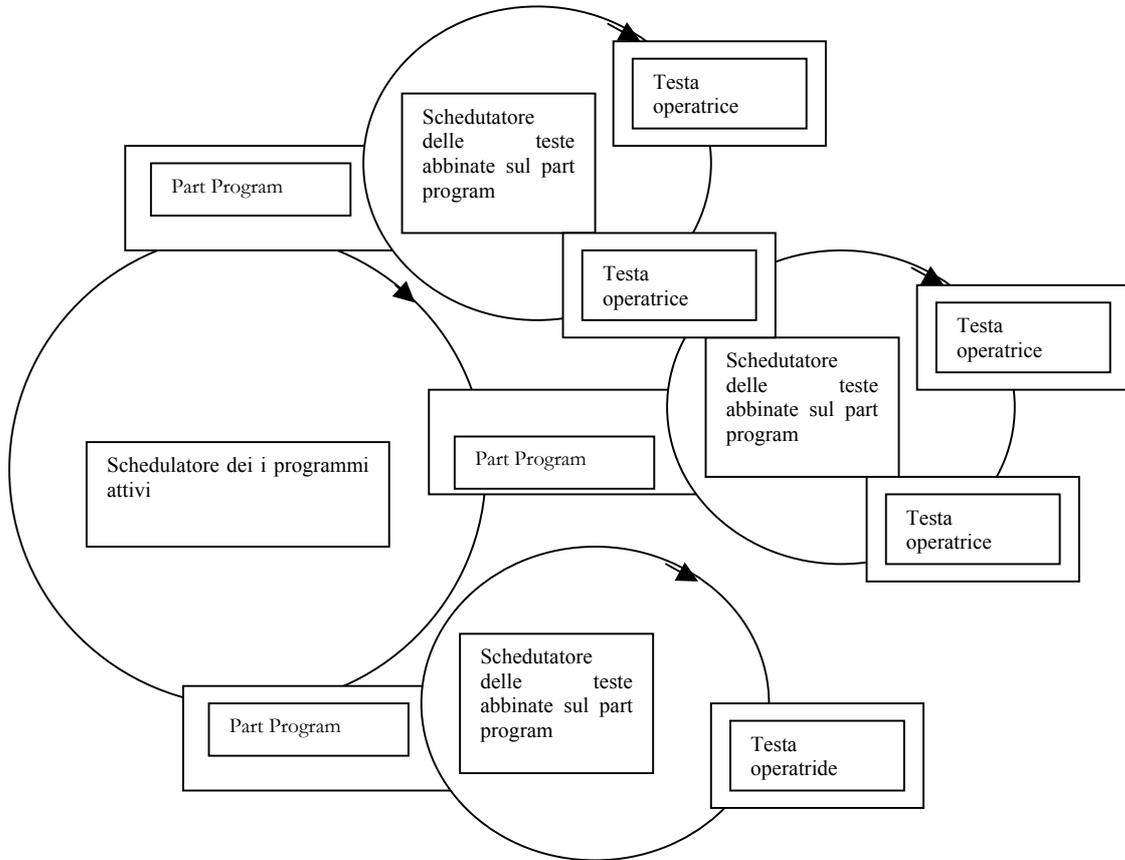
### 2 Architettura

*Il sottosistema AxesBrainISO è in grado di gestire una macchina utensili multi testa, sia con lavorazione parallela sia lavorazioni indipendenti.*

Ogni testa è vista come un abbinamento di assi fisici ad ogni sistema di coordinate rappresentante la testa operatrice o contesto CNC.

Il CNC è il contesto per l'esecuzione di part-program, procedure e blocchi di programmazione singoli.

Il sottosistema CNC schedula le attività richieste per ogni programma di lavorazione in formato ISO attivato a sua volta per ogni programma esiste un altro schedulatore che gestisce la realizzazione blocco per ogni testa abbinata al part program.



## CAPITOLO 3

### 3 Le basi della programmazione CN

Un **programma CN (part programm)** è una sequenza di fasi di lavorazione ed è diviso in blocchi programma. Contiene le informazioni che la macchina utensile richiede per eseguire il processo desiderato.

#### 3.1 Blocco programma

Le righe individuali di un programma CN si chiamano **blocchi programma**. Un blocco programma è di solito inteso come la più piccola fase di lavoro che si può intraprendere nella lavorazione di un pezzo in lavorazione.

È costituito da almeno un **numero blocco** e termina in un **carattere di fine blocco**. La lunghezza massima di un blocco programma è di 128 caratteri.

Un **numero blocco** viene posto all'inizio di qualsiasi blocco CN. È costituito dal **carattere indirizzo N** e di un massimo di quattro **cifre**. Gli zeri iniziali possono essere omessi. Per consentire la modifica di un programma, è generalmente necessaria una numerazione sequenziale dei blocchi CN con i numeri blocchi ascendenti. Utilizzando i numeri blocchi, è facile così accertare se un blocco CN desiderato si trova nel programma prima o dopo il blocco attualmente mostrato.

Per consentire l'inserimento nel programma in un secondo tempo di blocchi programma aggiuntivi senza troppo difficoltà, è consigliabile programmare i numeri blocchi in fasi di dieci.

#### Esempio:

```
N10 G90
N20 G1 X50 Y20 F3000 M5 S1000
N30 X15
N40 Y-20 M3
N50 G4 F1000
N60 M30
```

I blocchi programma vengono elaborati nella sequenza in cui sono stati memorizzati.

### 3.2 Parola di programma

L'informazione individuale in un programma si chiama una **parola programma**. Una parola programma contiene informazione di programma tecnica, geometrica o tecnologica ed è costituita da una **lettera indirizzo** ed una **sequenza di cifre** con o senza segno (formato indirizzo secondo DIN 66025, parte 1).

La sequenza delle parole programma in un blocco è arbitrario a parte il numero blocco, che deve essere sempre posizionato all'inizio del blocco.

La **lettera indirizzo** designa il tipo della parola programma. Ogni lettera indirizzo deve essere programmata solamente una volta per blocco CN.

La **sequenza delle cifre** di una parola è un numero intero o un numero, che consiste in un valore intero ed una frazione decimale, che potrebbe anche essere zero. Il decimale è separato dal numero intero da un punto; una virgola non è permessibile. I segni sono programmati tra la lettera indirizzo e la sequenza di cifre. I segni positivi, gli zeri iniziali e gli zeri non significativi dopo il punto decimale non necessitano la programmazione. Se il punto decimale non viene seguito da cifre significative, il punto decimale stesso può essere omesso.

#### Esempi:

G1	invece di	G01
M1	invece di	M01
X1234.5	invece di	X+1234.500
Y12	invece di	Y+12.00
Z-25.4	invece di	Z-0025.4

In generale, le parole programma possono essere con **istruzioni** o **condizioni aggiuntive**. Tramite l'**istruzione** (p. es. i codici G o M), un processo viene preparato o scattato nella macchina utensile o il controllo. Con le **condizioni aggiuntive**, le istruzioni sono descritte con più esattezza, p. es. specificando le coordinate di destinazione per un'istruzione di posizionamento.

Le parole programma si distinguono sia come **modali**, cioè, ritentive, sia **non modal**i. Le parole programma modali sono attive in tutti i seguenti blocchi programmi finché sono escluse o soprascritte da un'istruzione o una condizione aggiuntiva che le cancella. Le parole non modali sono attive soltanto nel blocco in cui sono programmate. Le istruzioni modali devono quindi essere programmate soltanto quando stanno cambiando o quando sono

necessarie aggiuntivamente. Solo le istruzioni non modali devono essere programmate in ogni blocco in cui sono necessarie.

Le istruzioni sono organizzate in **gruppi istruzioni**. In qualsiasi gruppo individuale sono sommate tutte quelle istruzioni, di cui soltanto una alla volta può essere effettiva.

Nelle pagine seguenti si trovano le tabelle dei codici G disponibili. Le tabelle comprendono degli appunti sulla divisione dei gruppi, sull'efficacia, un appunto che indica se l'istruzione rispettiva è attiva e se le istruzioni di posizionamento sono programmabili nello stesso blocco. È incluso anche un riassunto dei codici M che sono elaborati.



## CAPITOLO 4

### 4 Regole di programmazione

Un programma di lavorazione è costituito da un insieme di blocchi che vengono eseguiti in sequenza. Ciascun blocco corrisponde ad una sola linea ed è costituito da una o più funzioni, seguite dal loro valore.

Un blocco può contenere informazioni di commento al programma, che non hanno alcuna influenza sul programma stesso. In questo caso il testo deve essere preceduto dal carattere punto e virgola “;”.

Esempio

; inizio fresatura

N 023 X100.56 Y 41.9

Il carattere maggiore “>” all’inizio della riga indica un comando di controllo o di configurazione o di richiamo a procedure specifiche, come la configurazioni di assi, ripetizioni di blocchi, etc.

La programmazione è a formato variabile. Questo significa che il numero di caratteri necessari per esprimere un valore non è fisso, quindi non è richiesto nessun tipo di incolonnamento. E’ possibile introdurre degli spazi in ciascun punto del blocco, poiché vengono ignorati. Il punto decimale va introdotto solo per determinate funzioni. Se non viene indicato il segno, i valori sono considerati positivi, i valori negativi devono essere preceduti dal segno meno “-“. Indipendentemente dalla posizione delle funzioni all’interno del blocco l’esecuzione di ciascuna avviene nel seguente ordine:

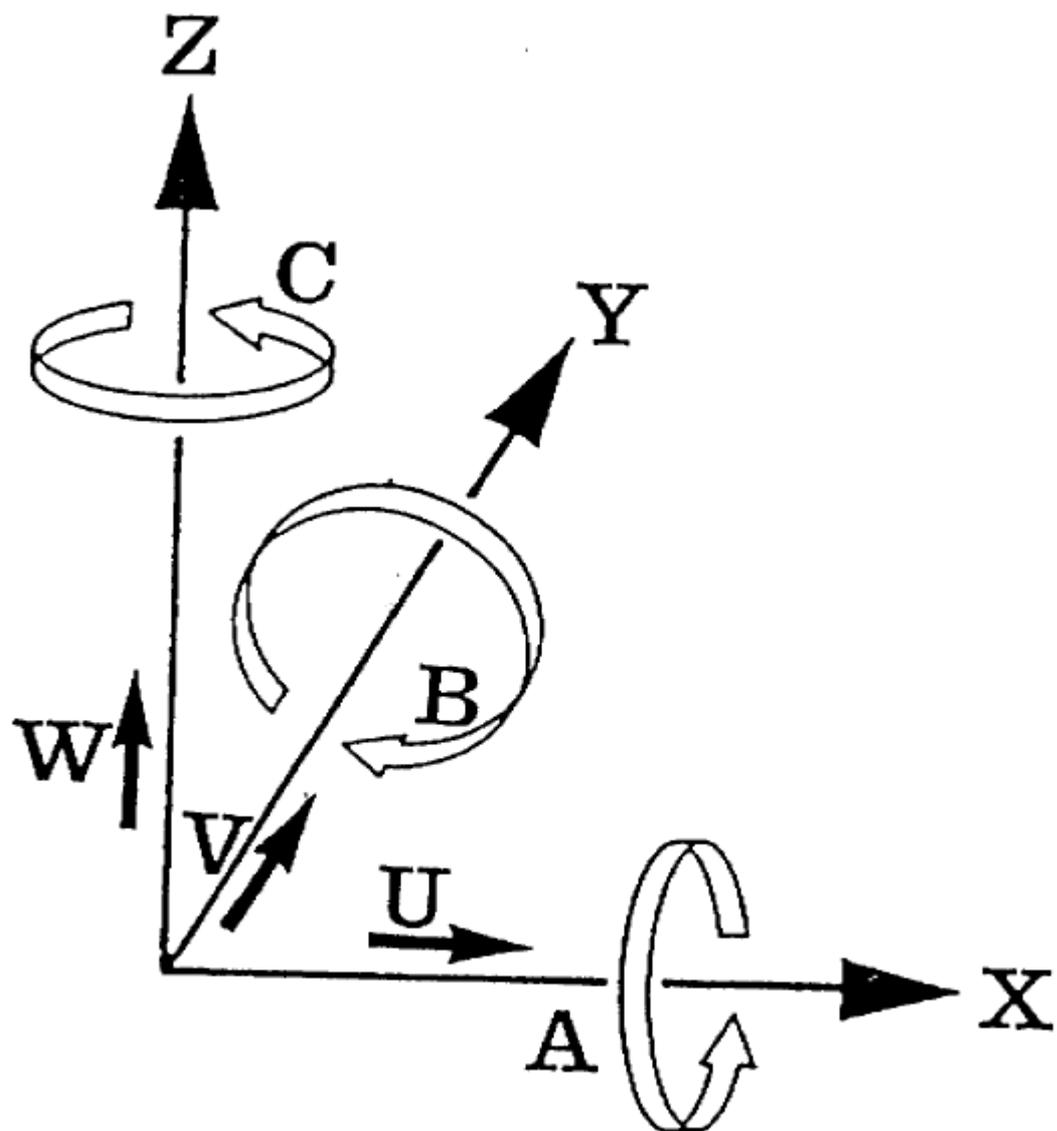
- funzione N
- funzione G
- funzioni M iniziali
- movimento assi
- funzioni M finali

All’interno di un blocco possono essere programmati più funzioni G e più funzioni M, ma solo una delle altre (X, Y, Z, T, F, ecc.)

Le funzioni G sono in diversi gruppi.

Non è possibile programmare in un blocco due o più funzioni G dello stesso gruppo.

Le funzioni M si dividono in funzioni iniziali, cioè eseguite prima del movimentarsi e in funzioni finali, cioè eseguite al termine del movimentarsi. Per ogni blocco di programmazione possono essere programmate due funzioni M iniziali e due funzioni M finali. Le funzioni M (miscellanee) servono generalmente per impartire alla macchina utensile dei comandi specifici riguardanti la predisposizione degli assi o del mandrino prima e dopo una certa operazione.



## CAPITOLO 5

### 5 Abbinamenti risorse ai singoli processi

Il sottosistema AxesBrainISO è in grado di gestire una macchina utensili multi testa, sia con lavorazione parallela sia lavorazioni indipendenti.

Per realizzare questa importante funzione si devono abbinare gli assi fisici o virtuali al sistema di coordinate cartesiane di ciascuna testa, questa operazione può essere fatta staticamente nella configurazione del sistema, oppure come set di comandi nel programma stesso.

Se ci sono più teste in lavorazione, è necessario definire le risorse in termini di ASSI,MANDRINO,IO per refrigerante e IO di controllo virtuali per l' interfacciamento con la Logica di Macchina, IO per i cicli fissi ed inoltre le tabelle delle Origini,Tool,Parametrica ed Funzioni d'abbinamento per G o M.

Se il sistema è monoprocesso le risorse possono essere abbinare tramite i comandi posti nel file "SISTEMA.TXT" sotto le voci [Abbinamento...], nel file sistema possono essere definite alcune altre predefinizioni, che se non cambiate vengono attivate al momento dell'esecuzione dei part program ISO

Se il sistema è multiprocesso possono essere le istruzioni speciali per l'abbinamento delle risorse al singolo processo ISO.

Per rendere meno visibile questi assegnamenti è possibile utilizzare un file con lo stesso nome con estensione **.DEF** dove vengono inserite le istruzioni eseguite per prima.



## CAPITOLO 6

### 6 Comandi di predefinitezza nel “SISTEMA.TXT”

Nel file “SISTEMA.TXT” possono essere inseriti dei comandi di abbinamento delle risorse ai “processi” ISO ed inoltre alcune predefinitezze.

I processi di lavorazione ISO ( G-CODE) sono massimo 32, vengono individuati da un ID ( identificativo numerico ) e da un Nome ( identificativo alfanumerico )

I parametri dei processi ISO vengono definiti il primo da “Base” e gli altri come “Testa” con il numero di processo – 1 fino al massimo di processi meno uno:

esempio 5 processi avremo

**Base\_xxxx**

**Testa1\_xxxxx**

**Testa2\_xxxxx**

**Testa3\_xxxxx**

**Testa4\_xxxxx**

Queste definizioni riguardano i valori di default, che vengono attivate all’inizio di ogni part program , le istruzioni non ISO di abbinamento risorse sostituiscono questi assegnamenti.

Le sessioni sono:

**a- Sessione identificativo nome del processo**

**[AbbinamentoNomeProcesso]**

Base\_NOME\_PROCESSO=[nome alfanumerico]

Testa1\_NOME\_PROCESSO=[nome alfanumerico]

..

Testa31\_NOME\_PROCESSO=[nome alfanumerico]

**Esempio**

[AbbinamentoNomeProcesso]

Base\_NOME\_PROCESSO=Unita011

Testa1\_NOME\_PROCESSO= Unita012

Testa2\_NOME\_PROCESSO= Unita021

**b- Sessione identificativo ID del processo**

**[AbbinamentoNumeroTesta]**

Base\_NumeroTesta=[numero identificativo ID del processo]

Testa1\_NumeroTesta=[ numero identificativo ID del processo]

..

Testa31\_NumeroTesta=[ numero identificativo ID del processo]

**Esempio**

[AbbinamentoNumeroTesta]

Base\_NumeroTesta=11

Testa1\_NumeroTesta=12

Testa2\_NumeroTesta=21

### c- Sessione abbinamento assi macchina ad assi ISO

I 9 assi canonici ISO sono X,Y,Z,A,B,C,U,V,W con questa sessione vengono abbinati agli assi fisici o virtuali del sistema.

Il valore -999 viene inserito al posto del nome asse fisico o virtuale, quando non esiste l'abbinamento.

#### [AbbinamentoAssiTeste]

Base\_X=[ nome asse fisico o virtuale ]  
Base\_Y=[ nome asse fisico o virtuale ]  
Base\_Z=[ nome asse fisico o virtuale ]  
Base\_A=[ nome asse fisico o virtuale ]  
Base\_B=[ nome asse fisico o virtuale ]  
Base\_C=[ nome asse fisico o virtuale ]  
Base\_U=[ nome asse fisico o virtuale ]  
Base\_V=[ nome asse fisico o virtuale ]  
Base\_W=[ nome asse fisico o virtuale ]  
Testa1\_X=[ nome asse fisico o virtuale ]  
Testa1\_Y=[ nome asse fisico o virtuale ]  
Testa1\_Z=[ nome asse fisico o virtuale ]  
Testa1\_A=[ nome asse fisico o virtuale ]  
Testa1\_B=[ nome asse fisico o virtuale ]  
Testa1\_C=[ nome asse fisico o virtuale ]  
Testa1\_U=[ nome asse fisico o virtuale ]  
Testa1\_V=[ nome asse fisico o virtuale ]  
Testa1\_W=[ nome asse fisico o virtuale ]  
...  
...  
Testa31\_X=[ nome asse fisico o virtuale ]  
Testa31\_Y=[ nome asse fisico o virtuale ]  
Testa31\_Z=[ nome asse fisico o virtuale ]  
Testa31\_A=[ nome asse fisico o virtuale ]  
Testa31\_B=[ nome asse fisico o virtuale ]  
Testa31\_C=[ nome asse fisico o virtuale ]  
Testa31\_U=[ nome asse fisico o virtuale ]  
Testa31\_V=[ nome asse fisico o virtuale ]  
Testa31\_W=[ nome asse fisico o virtuale ]

### Esempio

[AbbinamentoAssiTeste]

Base\_X=X011

Base\_Y=Y011

Base\_Z=Z011

Base\_A=-999

Base\_B=-999

Base\_C=C011

Base\_U=-999

Base\_V=-999

Base\_W=-999

Testa1\_X=X012

Testa1\_Y=Y012

Testa1\_Z=Z012

Testa1\_A=-999

Testa1\_B=-999

Testa1\_C=-999

Testa1\_U=-999

Testa1\_V=-999

Testa1\_W=-999

Testa2\_X=X021

Testa2\_Y=Y021

Testa2\_Z=Z021

Testa2\_A=-999

Testa2\_B=-999

Testa2\_C=-999

Testa2\_U=-999

Testa2\_V=-999

Testa2\_W=-999

#### **d- Sessione abbinamento mandrino S ISO a mandrino fisico**

Il mandrino S ISO con questa sessione viene abbinato ad un mandrino fisico del sistema. Il valore -999 viene inserito al posto nome mandrino, quando non esiste l'abbinamento.

##### **[AbbinamentoMandrinoTeste]**

```
Base_Mandrino=[ nome mandrino ]  
Testa1_Mandrino=[ nome mandrino ]  
..  
Testa31_Mandrino=[ nome mandrino ]
```

Esempio

```
[AbbinamentoMandrinoTeste]  
Base_Mandrino=S011  
Testa1_Mandrino=S012  
Testa2_Mandrino=S021
```

#### **e- Sessione abbinamento directory di lavoro part program ISO al processo**

Ogni processo ISO lavora su una directory separata, in modo da operare su un proprio archivio di programmi ISO (xxxxxx.prg).

##### **[AbbinamentoDirectoryPP]**

```
Base_Directory_PP=[ path directory\  
Testa1_Directory_PP=[ path directory\  
..  
Testa31_Directory_PP=[ path directory\  
]
```

Esempio

```
[AbbinamentoDirectoryPP]  
Base_Directory_PP=C:\Programmi\AB&T\AxisBrainStudio\CNC_Unita011\  
Testa1_Directory_PP=C:\Programmi\AB&T\AxisBrainStudio\CNC_Unita012\  
Testa2_Directory_PP=C:\Programmi\AB&T\AxisBrainStudio\CNC_Unita021\  
]
```

## f- Sessione abbinamento file tabella origini al processo ISO

Ogni processo ISO lavora su una tabella propria di **origini**, tramite questa sessione è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

### [AbbinamentoORIGINI]

```
Base_FILE_ORIGINE=[percorso e nome file]
Testa1_FILE_ORIGINE=[percorso e nome file]
...
Testa31_FILE_ORIGINE=[percorso e nome file]
```

Esempio

```
[AbbinamentoORIGINI]
Base_FILE_ORIGINE=C:\Transfer\Staz01\Unita011_MS\Origin.dat
Testa1_FILE_ORIGINE=C:\Transfer\Staz02\Unita012_MS\Origin.dat
Testa2_FILE_ORIGINE=C:\Transfer\Staz03\Unita021_MS\Origin.dat
```

## g- Sessione abbinamento file tabella TOOL al processo ISO

Ogni processo ISO lavora su una tabella propria di **Tool**, tramite questa sessione è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

### [AbbinamentoTOOL]

```
Base_FILE_TOOL =[percorso e nome file]
Testa1_FILE_TOOL =[percorso e nome file]
...
Testa31_FILE_TOOL =[percorso e nome file]
```

Esempio

```
[AbbinamentoTOOL]
Base_FILE_TOOL=C:\Transfer\Staz01\Unita011_MS\Tool.dat
Testa1_FILE_TOOL=C:\Transfer\Staz02\Unita012_MS\Tool.dat
Testa2_FILE_TOOL=C:\Transfer\Staz03\Unita021_MS\Tool.dat
```

## h- Sessione abbinamento file tabella parametri P al processo ISO

Ogni processo ISO lavora su una tabella propria di **parametri P**, tramite questa sessione è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

### [AbbinamentoPARAMETRI]

```
Base_FILE_PARAMETERS =[percorso e nome file]
Testa1_FILE_PARAMETERS =[percorso e nome file]
...
Testa31_FILE_PARAMETERS =[percorso e nome file]
```

Esempio

```
[AbbinamentoPARAMETRI]
Base_FILE_PARAMETERS=C:\Transfer\Staz01\Unita011_MS\SlotParam.dat
Testa1_FILE_PARAMETERS=C:\Transfer\Staz02\Unita012_MS\SlotParam.dat
Testa2_FILE_PARAMETERS=C:\Transfer\Staz03\Unita021_MS\SlotParam.dat
```

## i- Sessione abbinamento file tabella Funzioni M e G al processo ISO

Ogni processo ISO lavora su una tabella propria di **Funzioni M e G**, tramite questa sessione è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

### [AbbinamentoFUNZIONI\_MG]

```
Base_FILE_FUNZIONI_MG =[percorso e nome file]
Testa1_FILE_FUNZIONI_MG =[percorso e nome file]
...
Testa31_FILE_FUNZIONI_MG =[percorso e nome file]
```

Esempio

```
[AbbinamentoFUNZIONI_MG]
Base_FILE_FUNZIONI_MG=C:\Transfer\Staz01\Unita011_MS\SlotGM.dat
Testa1_FILE_FUNZIONI_MG=C:\Transfer\Staz02\Unita012_MS\SlotGM.dat
Testa2_FILE_FUNZIONI_MG=C:\Transfer\Staz03\Unita021_MS\SlotGM.dat
```

## **l- Sessione abbinamento file tabella Correttori Utensili al processo ISO**

Ogni processo ISO lavora su una tabella propria di **Correttori Utensili**, tramite questa sessione è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

### **[AbbinamentoCORREZ]**

```
Base_FILE CORREZIONI =[percorso e nome file]
Testa1_FILE_CORREZIONI =[percorso e nome file]
...
Testa31_FILE_CORREZIONI =[percorso e nome file]
```

Esempio

```
[AbbinamentoCORREZ]
Base_FILE_CORREZIONI= C:\Transfer\Staz01\Unita011_MS\Correz.dat
Testa1_FILE_CORREZIONI= C:\Transfer\Staz01\Unita011_MS\Correz.dat
Testa2_FILE_CORREZIONI= C:\Transfer\Staz01\Unita011_MS\Correz.dat
```

## **m- Sessione abbinamento gestione della coppia tramite una globale G caricata dalla logica di macchina per il processo ISO**

Se la logica di macchina acquisisce il valore di coppia e lo fornisce come valore percentuale in una variabile globale G, quest'ultima può venire indicata al processo ISO abbinato. Il valore -999 viene inserito al posto del numero della Globale , quando non esiste l'abbinamento.

### **[AbbinamentoCoppia]**

```
Base_Globale_Coppia =[numero Globale]
Testa1_Globale_Coppia =[ numero Globale]
...
Testa31_Globale_Coppia =[ numero Globale]
```

Esempio

```
[AbbinamentoCoppia]
Base_Globale_Coppia=11011
Testa1_Globale_Coppia=11012
Testa2_Globale_Coppia=-999
```

### **n- Sessione abbinamento abilitazione e disabilitazione del processo ISO tramite lo stato contenuto una globale G specifica**

Attraverso lo stato 1 = abilitato e 0=disabilitato contenuto in una G specifica per ogni processo ISO, è possibile gestire la funzionalità di attivazione e disattivazione del processo ISO.

Il valore -999 viene inserito al posto del numero della Globale , quando non esiste l'abbinamento.

#### **[AbbinamentoG\_Disabilitazione]**

Base\_Globale\_Disabilitazione =[numero Globale]

Testa1\_Globale\_Disabilitazione =[ numero Globale]

...

Testa31\_Globale\_Disabilitazione =[ numero Globale]

Esempio

[AbbinamentoG\_Disabilitazione]

Base\_Globale\_Disabilitazione=-999

Testa1\_Globale\_Disabilitazione=-999

Testa2\_Globale\_Disabilitazione=-999

### **o- Sessione abbinamento default sistema misura in mm o pollici**

Il processo ISO a un proprio sistema di misura metrico o pollici, se non specificato attraverso questa voce è metrico, qualora lo si voglia in pollici bisogna definire al valore numerico 1 il parametro del processo ISO desiderato.

Il valore -999 indica sistema metrico, come del resto anche 0.

#### **[AbbinamentoInch]**

Base\_INCH=[flag 1 per sistema in pollici]

Testa1\_INCH=[flag 1 per sistema in pollici]

..

Testa31\_INCH=[flag 1 per sistema in pollici]

Esempio

[AbbinamentoInch]

Base\_INCH=-999

Testa1\_INCH=-999

Testa2\_INCH=-999

**p- Sessione abbinamento definizione nelle G02 e G03 del modo di interpretare i valori del centro del raggio (I-J-K) in assoluto rispetto all'origine pezzo attivo od in incrementale rispetto alla posizione corrente.**

In ogni processo ISO è possibile definire i valori del centro del raggio contenuti nelle G02 e G03 in assoluto rispetto al sistema di riferimento in corso od in incrementale rispetto alla posizione corrente. Se non dichiarato in modo esplicito in questa sessione il sistema li interpreta in modo assoluto.

Il valore -999 indica sistema assoluto, come del resto anche 0.

**[AbbinaIncrementaleG0302]**

Base\_INCREMENTALEG0302=[flag 1 per sistema in incrementale]  
Testa1\_INCREMENTALEG0302=[flag 1 per sistema in incrementale]  
..  
Testa31\_INCREMENTALEG0302=[flag 1 per sistema in incrementale]

Esempio

[AbbinaIncrementaleG0302]  
Base\_INCREMENTALEG0302=-999  
Testa1\_INCREMENTALEG0302=-999  
Testa2\_INCREMENTALEG0302=-999

**q- Sessione abbinamento modalità di esecuzione G01**

In ogni processo ISO è possibile definire se l'esecuzione dei blocchi G01-G02-G03 sono in modalità continua ( default) oppure con arresto degli assi ad ogni blocco.

Il valore -999 indica modalità continua, come del resto anche 0.

**[AbbinaNonContinuo]**

Base\_NONCONTINUO =[flag 1 per modalità con arresto assi]  
Testa1\_NONCONTINUO =[ flag 1 per modalità con arresto assi]  
..  
Testa31\_NONCONTINUO =[ flag 1 per modalità con arresto assi]

Esempio

[AbbinaNonContinuo]  
Base\_NONCONTINUO=-999  
Testa1\_NONCONTINUO=-999  
Testa2\_NONCONTINUO=-999

## r- Sessione abbinamento output digitale per M08 refrigerante primario

In ogni processo ISO è possibile definire l'output digitale per a funzione di refrigerante primario M08

### [AbbinamentoRefrigerantePrincipale]

Base\_RefrigerantePrimario =[ nome output digitale]

Testa1\_RefrigerantePrimario =[ nome output digitale]

..

Testa31\_RefrigerantePrimario =[ nome output digitale]

Esempio

[AbbinamentoRefrigerantePrincipale]

Base\_RefrigerantePrimario=EV\_RP011

Testa1\_RefrigerantePrimario EV\_RP012

Testa2\_RefrigerantePrimario EV\_RP021

## s- Sessione abbinamento output digitale per M07 refrigerante secondario

In ogni processo ISO è possibile definire l'output digitale per a funzione di refrigerante secondario M07

### [AbbinamentoRefrigeranteSecondario]

Base\_RefrigeranteSecondario =[ nome output digitale]

Testa1\_RefrigeranteSecondario =[ nome output digitale]

..

Testa31\_RefrigeranteSecondario =[ nome output digitale]

Esempio

[AbbinamentoRefrigeranteSecondario]

Base\_RefrigeranteSecondario=EV\_RS011

Testa1\_RefrigeranteSecondario= EV\_RS012

Testa2\_RefrigeranteSecondario=EV\_RS021

## t- Sessione abbinamento Utensile preparatorio

In ogni processo ISO è possibile definire il T senza M6 ( Cambio utensile ) nell'istruzione viene inteso come funzione preparatoria , per default non viene attivata come funzione preparatoria

### [ToolPreparatorio]

Base\_ ToolPreparatorio = [ 1 o -999 ]

Testa1\_ ToolPreparatorio =[ 1 o -999]

..

Testa31\_ ToolPreparatorio =[ 1 o -999]

Esempio

[ToolPreparatorio]

Base\_ ToolPreparatorio=-999

Testa1\_ ToolPreparatorio=-999

Testa2\_ ToolPreparatorio=-999

## u- Sessione abbinamento Visualizzazione assi su interfaccia

Perogni processo ISO è possibile definire gli assi visualizzati sull'interfaccia operatore

### [VisualizzaAssiTeste]

VisuaBase\_X\_NomeVisualizzato= [ nome visualizzato o-999 ]  
VisuaBase\_X\_AsseFisico= [ nome asse fisico o-999 ]  
VisuaBase\_Y\_NomeVisualizzato= [ nome visualizzato o-999 ]  
VisuaBase\_Y\_AsseFisico= [ nome asse fisico o-999 ]  
VisuaBase\_Z\_NomeVisualizzato= [ nome visualizzato o-999 ]  
VisuaBase\_Z\_AsseFisico= [ nome asse fisico o-999 ]  
VisuaBase\_A\_NomeVisualizzato= [ nome visualizzato o-999 ]  
VisuaBase\_A\_AsseFisico= [ nome asse fisico o-999 ]  
VisuaBase\_B\_NomeVisualizzato= [ nome visualizzato o-999 ]  
VisuaBase\_B\_AsseFisico= [ nome asse fisico o-999 ]  
VisuaBase\_C\_NomeVisualizzato= [ nome visualizzato o-999 ]  
VisuaBase\_C\_AsseFisico= [ nome asse fisico o-999 ]  
VisuaBase\_U\_NomeVisualizzato= [ nome visualizzato o-999 ]  
VisuaBase\_U\_AsseFisico= [ nome asse fisico o-999 ]  
VisuaBase\_V\_NomeVisualizzato= [ nome visualizzato o-999 ]  
VisuaBase\_V\_AsseFisico= [ nome asse fisico o-999 ]  
VisuaBase\_W\_NomeVisualizzato= [ nome visualizzato o-999 ]  
VisuaBase\_W\_AsseFisico= [ nome asse fisico o-999 ]

Visua1\_X\_NomeVisualizzato= [ nome visualizzato o-999 ]  
Visua1\_X\_AsseFisico= [ nome asse fisico o-999 ]  
Visua1\_Y\_NomeVisualizzato= [ nome visualizzato o-999 ]  
Visua1\_Y\_AsseFisico= [ nome asse fisico o-999 ]  
Visua1\_Z\_NomeVisualizzato= [ nome visualizzato o-999 ]  
Visua1\_Z\_AsseFisico= [ nome asse fisico o-999 ]  
Visua1\_A\_NomeVisualizzato= [ nome visualizzato o-999 ]  
Visua1\_A\_AsseFisico= [ nome asse fisico o-999 ]  
Visua1\_B\_NomeVisualizzato= [ nome visualizzato o-999 ]  
Visua1\_B\_AsseFisico= [ nome asse fisico o-999 ]  
Visua1\_C\_NomeVisualizzato= [ nome visualizzato o-999 ]  
Visua1\_C\_AsseFisico= [ nome asse fisico o-999 ]  
Visua1\_U\_NomeVisualizzato= [ nome visualizzato o-999 ]  
Visua1\_U\_AsseFisico= [ nome asse fisico o-999 ]  
Visua1\_V\_NomeVisualizzato= [ nome visualizzato o-999 ]  
Visua1\_V\_AsseFisico= [ nome asse fisico o-999 ]  
Visua1\_W\_NomeVisualizzato= [ nome visualizzato o-999 ]  
Visua1\_W\_AsseFisico= [ nome asse fisico o-999 ]

## Esempio

```
[VisualizzaAssiTeste]
VisuaBase_X_NomeVisualizzato=-999
VisuaBase_X_AsseFisico=-999
VisuaBase_Y_NomeVisualizzato=-999
VisuaBase_Y_AsseFisico=-999
VisuaBase_Z_NomeVisualizzato=-999
VisuaBase_Z_AsseFisico=-999
VisuaBase_A_NomeVisualizzato=-999
VisuaBase_A_AsseFisico=Z011
VisuaBase_B_NomeVisualizzato=-999
VisuaBase_B_AsseFisico=Z021
VisuaBase_C_NomeVisualizzato=-999
VisuaBase_C_AsseFisico=Z011
VisuaBase_U_NomeVisualizzato=-999
VisuaBase_U_AsseFisico=Z021
VisuaBase_V_NomeVisualizzato=-999
VisuaBase_V_AsseFisico=Z011
VisuaBase_W_NomeVisualizzato=-999
VisuaBase_W_AsseFisico=Z021
Visua1_X_NomeVisualizzato=-999
Visua1_X_AsseFisico=-999
Visua1_Y_NomeVisualizzato=-999
Visua1_Y_AsseFisico=-999
Visua1_Z_NomeVisualizzato=-999
Visua1_Z_AsseFisico=-999
Visua1_A_NomeVisualizzato=-999
Visua1_A_AsseFisico=-999
Visua1_B_NomeVisualizzato=-999
Visua1_B_AsseFisico=-999
Visua1_C_NomeVisualizzato=-999
Visua1_C_AsseFisico=-999
Visua1_U_NomeVisualizzato=-999
Visua1_U_AsseFisico=-999
Visua1_V_NomeVisualizzato=-999
Visua1_V_AsseFisico=-999
Visua1_W_NomeVisualizzato=-999
Visua1_W_AsseFisico=-999
```

#### v- Sessione abbinamento fattore di scala visualizzata

Perogni processo ISO è possibile definire la scala dei grafico percorso utensile sull'interfaccia operatore

```
[AbbinamentoScalaVisualizzata]
Base_ScalaVisualizzata=-999
Testa1_ScalaVisualizzata=-999
Testa2_ScalaVisualizzata=-999
Testa3_ScalaVisualizzata=-999
Testa5_ScalaVisualizzata=-999
Testa6_ScalaVisualizzata=-999
Testa7_ScalaVisualizzata=-999
Testa8_ScalaVisualizzata=-999
Testa9_ScalaVisualizzata=-999
Testa10_ScalaVisualizzata=-999
Testa11_ScalaVisualizzata=-999
Testa12_ScalaVisualizzata=-999
```

**Funzionalità specifiche definite automaticamente, per la sincronizzazione con la logica di macchina.**

Sono previsti degli automatismi per l'interconnessione con la LOGICA di MACCHINA, attraverso 16 o 32 bit consecutivi per lo scambio di stati e richieste.

Ed in particolare:

<b>Logica Macchina</b>	<b>CNC-ISO</b>
Bit 0 Logica OK = 1	Bit 0 Part Program Start = 1
Bit 1 Rich. Stop	Bit 1 Part Program Stop = 1
Bit 2 Rich. PassoPasso	Bit 2 Part Program in PassoPasso = 1
Bit 3 Rich. Kill Part program	Bit 3 Part Program Killed
Bit 4 Allarme = 1	Bit 4 Part Program Allarme acquisito e Part Program in Stop
Bit 5 Warning = 1	Bit 5 Part Program Warning acquisito
Bit 6 Info = 1	Bit 6 Part Program Info acquisito
Bit 7 Lock = 1	Bit 6 Part Program Lock acquisito
Bit 8 Next Step	Bit 8
Bit 9 Manuale	Bit 9
Bit 10	Bit 10
Bit 11	Bit 11
Bit 12	Bit 12
Bit 13	Bit 13
Bit 14	Bit 14
Bit 15	Bit 15

Bit 16 Bit 0 Msg Allarme	Bit 16
Bit 17 Bit 1 Msg Allarme	Bit 17
Bit 18 Bit 2 Msg Allarme	Bit 18
Bit 19 Bit 3 Msg Allarme	Bit 19
Bit 20 Bit 0 Msg Warning	Bit 20
Bit 21 Bit 1 Msg Warning	Bit 21
Bit 22 Bit 2 Msg Warning	Bit 22
Bit 23 Bit 3 Msg Warning	Bit 23
Bit 24 Bit 0 Msg Info	Bit 24
Bit 25 Bit 1 Msg Info	Bit 25
Bit 26 Bit 2 Msg Info	Bit 26
Bit 27 Bit 3 Msg Info	Bit 27
Bit 28 Bit 0 Msg Lock	Bit 28
Bit 29 Bit 1 Msg Lock	Bit 29
Bit 30 Bit 2 Msg Lock	Bit 30
Bit 31 Bit 3 Msg Lock	Bit 31

### Vengono scritti dal processo ISO

#### [AbbinamentoOutputDigitaleControllo]

Base\_OutputDigitaleControllo=[nome input od output digitale bit 0 ]  
 Testa1\_OutputDigitaleControllo=[nome input od output digitale bit 0 ]  
 ..  
 Testa31\_OutputDigitaleControllo=[nome input od output digitale bit 0 ]

### Vengono letti dal processo ISO

#### [AbbinamentoInputDigitaleControllo]

Base\_InputDigitaleControllo=[nome input od output digitale bit 0 ]  
 Testa1\_InputDigitaleControllo=[nome input od output digitale bit 0 ]  
 ..  
 Testa31\_InputDigitaleControllo=[nome input od output digitale bit 0 ]

### Numero di bit 16 o 32

#### [AbbinamentoNumeroBitControllo]

Base\_NumeroBitControllo=[NumeroBit 16 0 32]  
 Testa1\_NumeroBitControllo=[NumeroBit 16 0 32]  
 ..  
 Testa31\_NumeroBitControllo=[NumeroBit 16 0 32]

**Funzionalità future definite automaticamente, ma non interpretate dal sistema.**

**[AbbinamentoOutputDigitaleCicloFisso]**

Base\_OutputDigitaleCicloFisso=[nome input od output digitale]

Testa1\_OutputDigitaleCicloFisso=[nome input od output digitale]

..

Testa31\_OutputDigitaleCicloFisso=[nome input od output digitale]

**[AbbinamentoInputDigitaleCicloFisso]**

Base\_InputDigitaleCicloFisso=[nome input od output digitale]

Testa1\_InputDigitaleCicloFisso=[nome input od output digitale]

..

Testa31\_InputDigitaleCicloFisso=[nome input od output digitale]

**[AbbinamentoOutputAnalogicoCicloFisso]**

Base\_OutputAnalogicoCicloFisso=[nome output analogico]

Testa1\_InputDigitaleCicloFisso=[nome input od output digitale]

..

Testa31\_InputDigitaleCicloFisso=[nome input od output digitale]

**[AbbinamentoInputAnalogicoCicloFisso]**

Base\_InputAnalogicoCicloFisso=[nome input analogico]

Testa1\_InputAnalogicoCicloFisso=[nome input analogico]

..

Testa31\_InputAnalogicoCicloFisso=[nome input analogico]

**[AbbinamentoChiaveInputControllo]**

Base\_ChiaveInputControllo=[nome input od output digitale]

Testa1\_ChiaveInputControllo=[nome input od output digitale]

..

Testa31\_ChiaveInputControllo=[nome input od output digitale]

## Parametri globali a tutti i processi ISO

### [ParametriCNC]

**Rapido**=Valore predefinito di rapido se si desidera un valore minore di quello impostato in caratterizzazione

**Feed**=Feed di default impostata all'inizio del programma ISO, verrà sostituito dalla prima F trovata nel part program

**Speed**= Speed di default impostata all'inizio del programma ISO, verrà sostituito dalla prima S trovata nel part program

**X\_offsetTavola**=offset asse X TAVOLA , verrà sostituito dalla prima O trovata nel part program

**Y\_offsetTavola**= offset asse Y TAVOLA , verrà sostituito dalla prima O trovata nel part program

**Z\_offsetTavola**= offset asse Z TAVOLA , verrà sostituito dalla prima O trovata nel part program

**Multiprocesso**=Se zero viene attivata la funzionalità multi testa mono processo ( Default 1)

**GlobaleStatoMacchina**=Globale per l'interscambio con la Logica di macchina

BLOCCA_STATI	-1 se letto dal sistema a -1 blocca gli stati macchina
NO_STATI	0
AUTOMATICO	1
MANUALE	3

**GlobaleComandiMacchina**=Globale per l'interscambio con la Logica di macchina

BLOCCA_COMANDI	-1 se letto dal sistema a -1 blocca i comandi macchina
NO_CICLO	0
VIA_CICLO	3
BREAK	10
HOLD	11

**GlobaleServizi**=Globale per l'interscambio con la Logica di macchina

BLOCCA_SERVIZI	-1 se letto dal sistema a -1 blocca l'attivazione dei servizi
NO_SERVIZI_ATTIVI	0
MDI	1
COMM	2
FUNZIONI_MANUALI	3
IMPOSTA_Pezzo	4
MODALITA_CICLO	5

**GlobaleProtezioni**=NON utilizzata dal sistema, futuri sviluppi

**GlobaleStatoPLC**=Indicando un numero di globale, quest'ultima viene utilizzata dal sistema per scrivere sull'interfaccia la scritta indicata nei due parametri successivi 1 o 0.

**ScrittaAttivaStatoPLC**=STRINGA scritta se **GlobaleStatoPLC=1**

**ScrittaNonAttivaStatoPLC**= STRINGA scritta se **GlobaleStatoPLC=0**

**TipoIOFeedHold**=0 o 1 =input o 2= output fa le veci della **GlobaleStatoPLC**

**NomeIOFeedHold**= Nome IO

**GlobaleComandoPLC**= Indicando un numero di globale, quest'ultima viene utilizzata dal sistema per scrivere sull'interfaccia la scritta indicata nei due parametri successivi 1 o 0.

**ScrittaAttivaComandoPLC**=STRINGA scritta se **GlobaleComandoPLC = 1**

**ScrittaNonAttivaComandoPLC**= STRINGA scritta se **GlobaleComandoPLC = 0**

**TipoIOHold**=0 o 1 =input o 2= output fa le veci della **GlobaleComandoPLC**

**NomeIOHold**=Nome IO

**Break**= Messo al valore di 1 sostituisce il pulsante di “**Selezione Processo**” (F10) con il pulsante di “**BREAK**”, sull'interfaccia CNC.

**GlobaleUnitaSelezionata** = Indicando un numero di globale, quest'ultima viene caricata dal sistema con l'ID del processo selezionato sull'interfaccia.

**GlobaleUnitaSelezionabileDaPLC**= Indicando un numero di globale, quest'ultima viene utilizzata **quando varia** ( quindi solo al cambiamento ) per selezionare il processo ISO sull'interfaccia con l'ID del processo indicata.

**Attenzione!!!** Solo il cambiamento di valore della **GlobaleUnitaSelezionabileDaPLC**, effettua il cambiamento di selezione del processo ISO.

**FlagM00**= se 1 arresta la funzionalità di lavoro e la sospende fino ad avere una un nuovo avvio ( rprea ciclo) altrimenti lascia la gestione alla logica di macchina

**MoltiplicatoreG4**= Moltiplica il parametro espresso in G4 unità di tempo in millisecondi

**AvviamentoInG1**= se 1 attiva per default l modalità G1 all'inizio del programma

## Esempio

```
[ParametriCNC]
Rapido=10000
Feed=5000
Speed=1000
X_offsetTavola=0
Y_offsetTavola=0
Z_offsetTavola=0
Multiprocesso=1
GlobaleStatoMacchina=0
GlobaleProtezioni=0
GlobaleComandiMacchina=0
GlobaleServizi=0
Break=0
GlobaleStatoPLC=0
ScrittaAttivaStatoPLC=Attivo
ScrittaNonAttivaStatoPLC=Disattivo
GlobaleComandoPLC=0
ScrittaAttivaComandoPLC=HOLD
ScrittaNonAttivaComandoPLC=HOLD
TipoIOFeedHold=0
NomeIOFeedHold=
TipoIOHold=0
NomeIOHold=
GlobaleUnitaSelezionata=0
GlobaleUnitaSelezionabileDaPLC=0
FlagM00=0
MoltiplicatoreG4=1
AvviamentoInG1=1
```



## CAPITOLO 7

### 7 Istruzioni speciali per l'abbinamento risorse al processo ISO

Per poter muovere degli assi indipendenti nella multiprogrammazione è necessario abbinare ad ogni asse ISO : X,Y,Z,A,B,C,U,V,W il nome dell'asse fisico definito nel file "SISTEMA.TXT", per poter effettuare queste operazioni si possono usare le istruzioni speciali seguenti.

#### > Configura teste CNC

**predispone il sistema ad accettare la configurazione vera e propria degli assi teorici che sono: X Y Z A B C U V W , il mandrino Mandrino e le risorse IO**

> Base\_[ nome asse teorico tra:X Y Z A B C U V W ] = [ nome asse fisico o virtuale ]

quindi avremo comandi di assegnamento per le teste aggiuntive:

> Testa\_[numero della testa aggiuntiva]\_[ nome asse teorico tra:X Y Z A B C U V W ] = [ nome asse fisico o virtuale ]

**quindi avremo comandi di assegnamento per le risorse di servizio:**

>Base\_Mandrino = [ nome mandrino ]

>Base\_RefrigerantePrimario = [ nome output digitale ]

>Base\_RefrigeranteSecondario = [ nome output digitale ]

>Base\_InputDigitaleCicloFisso = [ nome input digitale ]

>Base\_OutputDigitaleControllo = [ nome output digitale ]

>Base\_InputAnalogicoCicloFisso = [ nome input analogico ]

>Base\_OutputAnalogicoCicloFisso = [ nome output analogico ]

>Base\_NumeroTesta = [ numero della testa da indicare nella istruzione SEC della LM ]

>Base\_ChiaveInputControllo = [ stringa da indicare nella istruzione SEC della LM ]

>Base\_InputDigitaleControllo = [ nome input od output digitale ]

>Base\_OutputDigitaleControllo = [ nome input od output digitale ]

>Base\_NumeroBitControllo = [ numero di bit dell'IO del controllo ]

**Alla fine della fase di configurazione dovrà esserci il comando:**

>Fine\_configura\_teste\_CNC

Le istruzioni successive permettono di definire i file contenenti le tabelle ORIGINI, TOOL, Parametri e abbinamento G o M speciali per il part program in lavorazione.

Normalmente il sistema le cerca nella directory `..\DAT`, ovviamente nel multiprocesso possono nascere delle esigenze di avere delle tabella differenziate per ogni testa indipendente.

```
>FILE_ORIGINE=C:\AxesBrainStudio\Dat\Origin.dat
>FILE_TOOL=C:\AxesBrainStudio\Dat\Tool.dat
>FILE_CORRETTORI=C:\AxesBrainStudio\Dat\Correz.dat
>FILE_PARAMETRI=C:\AxesBrainStudio\Dat\SlotParam.dat
>FILE_FUNZIONI_MG=C:\AxesBrainStudio\Dat\SlotGM.dat
```

Esempio:

```
> Configura_teste_CNC
> Base_X=A1
> Base_Y=B1
> Base_Z=C1
> Base_Mandrino=M1
> Testa_2_X=A2
> Testa_2_Y=B2
> Testa_2_Z=C2
> Testa_2_Mandrino =M2
>Base_RefrigerantePrimario=CAN_O1
>Base_RefrigeranteSecondario=CAN_O2
>Base_InputDigitaleCicloFisso=CAN_I1
>Base_OutputDigitaleControllo=CAN_O1
>Base_InputAnalogicoCicloFisso=SIX_I_1
>Base_OutputAnalogicoCicloFisso=DACMS1T1
>Base_NumeroTesta=1
>Base_ChiaveInputControllo=LM_Controllo
>Base_NumeroBitControllo=32
>Base_InputDigitaleControllo=Q_S1T1.1
>Base_OutputDigitaleControllo=I_S1T1.1
>Fine_configura_teste_CNC
```

## CAPITOLO 8

### 8 Tabella Origini

La tabella delle ORIGINI viene utilizzata per attivare o disattivare (O0) le origini durante la lavorazione, ed è collocata nel file delle origini.

Ogni processo ISO lavora su una tabella propria di **Origini**, tramite la sessione **[AbbinamentoOrigini]** del file “sistema.txt” è possibile specificare l’intero percorso ed il nome del file contenente la tabella.

#### [AbbinamentoORIGINI]

Base\_FILE\_ORIGINE=[percorso e nome file]

Testa1\_FILE\_ORIGINE=[percorso e nome file]

...

Testa31\_FILE\_ORIGINE=[percorso e nome file]

L’istruzione speciale : **>FILE\_ORIGINE= [path e nome file delle origini ]** può essere usata per cambiare il file nel arco del part program in lavorazione sulla testa, vedi “Istruzioni speciali per l’abbinamento risorse al processo ISO”

#### 8.1 Definizioni del file per la tabella origini

[ParametriGeneralOrigini]

NumeroOrigini=[numero origini]

[O1]

Descrizione=

FlagMMInch=0

X=0

Y=0

Z=0

A=0

B=0

C=0

U=0

V=0

W=0



## CAPITOLO 9

### 9 Tabella degli utensili

La tabella degli utensili ( TOOL ) viene utilizzata per attivare o disattivare gli utensili durante la lavorazione, ed è collocata nel file dei Tool.

Ogni processo ISO lavora su una tabella propria di **Utensili (TOOL)**, tramite la sessione **[AbbinamentoTOOL]** del file “sistema.txt” è possibile specificare l’intero percorso ed il nome del file contenente la tabella.

#### [AbbinamentoTOOL]

```
Base_FILE_TOOL =[percorso e nome file]
Testa1_FILE_TOOL =[percorso e nome file]
...
Testa31_FILE_TOOL =[percorso e nome file]
```

L’istruzione speciale : **>FILE\_TOOL= [path e nome file dei TOOL ]** può essere usata per cambiare il file nel arco del part program in lavorazione sulla testa ,vedi “Istruzioni speciali per l’abbinamento risorse al processo ISO”

### 9.1 Definizioni del file per la tabella TOOL

```
[ParametriGeneraliTool]
NumeroTool=32
```

```
[T1]
```

Descrizione	= descrizione dell’utensile
PartProgramGAL	= part program AxesBrain logica di macchina ( M06 )
Diametro	= diametro dell’utensile
Passo	= Passo vite ( maschiatura)
FlagMMInchPasso	= Flag 0 se mm oppure 1 se INCH
RT	= correzione raggio utensile
LC	= correzione lunghezza
NM	= numero magazzino
PM	= posizione magazzino X

PD	=	posizione intermedia Y
Spec	=	no,si 0,1
Size	=	no defined,small,medium,large,extra 0,1,2,3,4
MaxLife	=	Vita massima minuti
MinLife	=	Vita min. minuti
RealLife	=	Vita residua minuti
CP	=	Capostipite Numero Tool capostipite
SC	=	Scaduto no,si 0,1
Free1	=	Free1 ( passato nella funzione M06 alla logica di macchina )
Free2	=	Free2 ( passato nella funzione M06 alla logica di macchina )
Free3	=	Free3 ( passato nella funzione M06 alla logica di macchina )
Free4	=	Free4 ( passato nella funzione M06 alla logica di macchina )

**I parametri passati dalla funzione M06 o dalla funzione T nel cambio utensile verso il part program AxesBrain ( Logica di macchina ) sono: (il numero delle Locali sono dimensionate almeno a 64 )**

- L1 = Numero asse Z rispetto al Piano di lavoro
- L2 = Numero mandrino S
- L3 = Numero AsseMandrino AS oppure 0
- L4 = Numero della testa definita in "Base\_NumeroTesta" oppure 0
- L5 = Speed mandrino ( valore di speed attivo blocco compreso )
  
- L6 = Tipo di funzione chiamante la logica di macchina  
3 = M06    4 = T (Tool) 5= T (Tool) preparatorio ( configurazione in sistema .txt )
  
- L7 = IDToolPrec
- L8 = RT raggio utensile
- L9 = LC correzione lunghezza
- L10 = NM numero magazzino
- L11 = PM posizione magazzino X
- L12 = PD posizione intermedia Y
- L13 = Spec no,si 0,1
- L14 = Size no defined,small,medium,large,extra 0,1,2,3,4
- L15 = MaxLife Vita massima minuti
- L16 = MinLife Vita min. minuti
- L17 = RealLife Vita residua minuti
- L18 = CP Capostipite Numero Tool capostipite
- L19 = SC Scaduto no,si 0,1
- L20 = Free1
- L21 = Free2
- L22 = Free3
- L23 = Free4

- L26 = IDToolSuccessivo
- L27 = RT raggio utensile
- L28 = LC correzione lunghezza
- L29 = NM numero magazzino
- L30 = PM posizione magazzino X
- L31 = PD posizione intermedia Y
- L32 = Spec no,si 0,1
- L33 = Size no defined,small,medium,large,extra 0,1,2,3,4
- L34 = MaxLife Vita massima minuti
- L35 = MinLife Vita min. minuti
- L36 = RealLife Vita residua minuti
- L37 = CP Capostipite Numero Tool capostipite
- L38 = SC Scaduto no,si 0,1
- L39 = Free1
- L40 = Free2
- L41 = Free3
- L42 = Free4
  
- L51 = 1 su richiesta da MDI

Al ritorno le medesimi Locali sono ricaricate nei registri del Processo chiamante

## CAPITOLO 10

### 10 Tabella dei Correttori D per gli utensili

La tabella dei Correttori viene utilizzata per attivare o disattivare la correzione utensili durante la lavorazione, ed è collocata nel file dei Correttori.

Ogni processo ISO lavora su una tabella propria di **Utensili (TOOL)**, tramite la sessione **[AbbinamentoCORREZ]** del file “sistema.txt” è possibile specificare l’intero percorso ed il nome del file contenente la tabella.

#### **[AbbinamentoCORREZ]**

Base\_FILE CORREZIONI =[percorso e nome file]

Testa1\_FILE\_CORREZIONI =[percorso e nome file]

...

Testa31\_FILE\_CORREZIONI =[percorso e nome file]

. L’istruzione speciale : **>FILE\_CORRETTORI= [path e nome file dei CORRETTORI ]** può essere usata per cambiare il file nel arco del part program in lavorazione sulla testa o processo, vedi

“Istruzioni speciali per l’abbinamento risorse al processo ISO”

#### 10.1 Definizioni del file per la tabella Correttori

[ParametriGeneraliCorrettori]

NumeroCorrettori=99

[C1]

Descrizione=

RT=0

LC= 0.0000

Free1=0

Free2=0

Free3=0

Free4=0

## CAPITOLO 11

### 11 Tabella dei Parametri

La tabella dei Parametri viene utilizzata per predefinire dei Parametri da utilizzare durante la lavorazione, ed è collocata nel file dei Parametri.

Ogni processo ISO lavora su una tabella propria di **Utensili (TOOL)**, tramite la sessione **[AbbinamentoPARAMETRI]** del file “sistema.txt” è possibile specificare l’intero percorso ed il nome del file contenente la tabella.

```
[AbbinamentoPARAMETRI]
Base_FILE_PARAMETRI =[percorso e nome file]
Testa1_FILE_PARAMETRI =[percorso e nome file]
...
Testa31_FILE_PARAMETRI =[percorso e nome file]
```

L’istruzione speciale : **>FILE\_PARAMETRI= [path e nome file dei PARAMETRI ]** può essere usata per cambiare il file nel arco del part program in lavorazione sulla testa, vedi

“Istruzioni speciali per l’abbinamento risorse al processo ISO”

#### 11.1 Definizioni del file per la tabella Parametri

```
[ParametriGenerali]
NumeroParametri=4
Slot_1=10,123 ; P10=123
Slot_2=7,345 ; P7=345
Slot_3=4,167 ; P4=167
Slot_4=50,312 ; P50=312
```



## CAPITOLO 12

### 12 Tabella delle Funzioni M e G speciali

La tabella delle Funzioni per M e G speciali viene utilizzata per definire dei Funzioni di automazione da utilizzare durante la lavorazione, ed è collocata nel file dei Funzioni Me G speciali.

Ogni processo ISO lavora su una tabella propria di **Funzioni M e G**, tramite la sessione **[AbbinamentoFUNZIONI\_MG]** del file “sistema.txt” è possibile specificare l’intero percorso ed il nome del file contenente la tabella.

#### **[AbbinamentoFUNZIONI\_MG]**

Base\_FILE\_FUNZIONI\_MG =[percorso e nome file]

Testa1\_FILE\_FUNZIONI\_MG =[percorso e nome file]

...

Testa31\_FILE\_FUNZIONI\_MG =[percorso e nome file]

L’istruzione speciale : **>FILE\_FUNZIONI\_MOG= [path e nome file delle Funzioni M e G speciali ]** può essere usata per cambiare il file nel arco del part program in lavorazione sulla testa, vedi

“Istruzioni speciali per l’abbinamento risorse al processo ISO

Automaticamente vengono caricate le prime 7 Locali del part program chiamato nel seguente modo:

L1 = Numero asse Z rispetto al Piano di lavoro

L2 = Numero mandrino S

L3 = Numero AsseMandrino AS oppure 0

L4 = Numero della testa definita in “Base\_NumeroTesta” oppure 0

L5 = Speed mandrino ( valore di speed attivo blocco compreso )

L6 = 1 se M, 2 se G

L7 = Numero della G o M chiamante

L8 = 1 se il comando in MDI

Automaticamente vengono caricate le Locali L11-L15 del part program chiamato nel seguente modo:

L11 = K

L12 = Q

L13 = J

L14 = I

L15 = H

L16 = Numero Origine

L17 = Numero Tool

L18 = Numero Correttore

**Esempio di file di interfacciamento diretto:**

/\*

L1 = Numero asse Z rispetto al Piano di lavoro

L2 = Numero mandrino S

L3 = Numero AsseMandrino AS oppure 0

L4 = Numero della testa definita in "Base\_NumeroTesta" oppure 0

L5 = Speed mandrino ( valore di speed attivo blocco compreso )

L6 = 1 se M, 2 se G

L7 = Numero della G o M chiamante

L11 = K

L12 = Q

L13 = J

L14 = I

L15 = H

L16 = Numero Origine

L17 = Numero Tool

L18 = Numero Correttore

\*/

-JMP/L7

-DIS/1:Funzione sconosciuta M,L7,3,0

-DIS/-2:Funzione sconosciuta M,L7,3,0

-RET

```
;  
_____  
0- ; M0  
; Attesa consenso operatore  
;
```

```
-TMM/100  
-SEC/SET_MESSAGE,4  
-SEC/CLR_MESSAGE,5
```

```
Attesa-TMM/20
```

```
-NET/1  
-LD/M1.4  
-EQU/O3  
-IDI/I7,0,Attesa  
-SEC/CLR_MESSAGE,4  
-SEC/SET_MESSAGE,5  
-SDO/O3,1  
-RET
```

```
;  
_____
```

```
;  
_____  
30- ; M30  
; Fine lavorazione  
;  
-RET
```

```
;  
_____
```

```
;  
_____  
40- ; M40  
; Rump ON  
;  
-SDO/O6,1  
-TMM/G1006  
-RET
```

```
;  
_____
```

```
;  
_____  
41- ; M41  
; Rump OFF  
;  
-SDO/O6,0  
-TMM/G1007  
-RET
```

```
;  
_____
```

```
;  
;  
-----  
100-                ; M100  
                   ; Gate ON  
                   ;  
                   ;  
-SDO/O10,1  
-TMM/G1004  
-RET  
;  
;  
-----  
101-                ; M101  
                   ; Gate OFF  
                   ;  
                   ;  
-SDO/O10,0  
-TMM/G1005  
-RET  
;  
;  
-----  
110-                ; M110  
                   ; Apertura Gas Inerte  
                   ;  
                   ;  
-SDO/O1,1  
-TMM/G1002  
-RET  
;  
;  
-----  
111-                ; M111  
                   ; Chiusura Gas Inerte  
                   ;  
                   ;  
-SDO/O1,0  
-TMM/G1003  
-RET  
;  
;  
-----  
126-                ; M126  
                   ; Apertura Shutter  
                   ;  
                   ;  
-NET/2  
-LD/M1.3  
-A/I8  
-PEX/-SDO/O8,1  
-PEX/-TMM/G1000  
-RET  
;  
-----
```

```
;  
_____  
127-                ; M127  
                    ; Chiusura Shutter  
                    ;  
    -SDO/O8,0  
    -TMM/G1001  
    -RET  
;  
_____  
;  
_____  
128-                ; M128  
                    ; Aspiratore ON  
                    ;  
    -SDO/O5,1  
    -TMM/G1008  
    -RET  
;  
_____  
;  
_____  
129-                ; M129  
                    ; Aspiratore OFF  
                    ;  
    -SDO/O5,0  
    -TMM/G1009  
    -RET  
;  
_____  
;  
_____  
140-                ; M140  
                    ; Zero pezzo  
                    ;  
    -CAP/9:X,0  
    -CAP/9:Y,0  
    -RAV/1:L12,X    ;Leggo X reale Assoluto  
    -OTC/1:O:1:X,L12  
    -RAV/1:L12,Y    ;Leggo Y reale Assoluto  
    -OTC/1:O:1:Y,L12  
    -MDI/1:O1  
    -RET  
;  
_____
```

**Il File della Tabella delle Funzioni M e G speciali è così composta:**

[ParametriGeneraliMoG]

**NumeroMoG**=[Numero complessivo delle funzioni G più le funzioni M ]

**Esecuzione\_M\_maggiore\_o\_uguale\_di**=[Numero della funzione M ]

**PartProgramGAL\_M**=[Nome file part program di automazione abbinato alle funzioni M]

**Esecuzione\_G\_maggiore\_o\_uguale\_di**=[Numero della funzione G ]

**PartProgramGAL\_G**=[ Nome file part program di automazione abbinato alle funzioni G]

[FunXXXX]

Kind=[ Kind 1=M 2=G ]

iD=[Numero della funzione G o M]

Descrizione=[Descrizione della funzione]

PartProgramGAL= [ Nome file del part program di automazione abbinato alla funzione]

NumeroLocali=[YYYY]

SlotInput\_1=[Numero della Locale],[stringa indicante il **contenuto** da caricare ]

SlotInput\_2=[Numero della Locale],[stringa indicante il **contenuto** da caricare ]

.....

SlotInput\_YYYY=[Numero della Locale L] ,[stringa indicante il **contenuto** da caricare ]

NumeroParametri=[ZZZZ]

SlotOutput\_1=[Numero del Parametro P], [espressione matematica ( con le variabili L )]

SlotOutput\_2=[Numero del Parametro P], [espressione matematica ( con le variabili L )]

..

SlotOutput\_ZZZZ=[Numero del Parametro P], [espressione matematica ( con le variabili L )]

## Contenuti caricabili nelle LOCALI:

- 1-Numero Asse **Axis** con relativa indicazione **X,Y,Z,A,B,C,U,V,W**
- 2-Numero Asse **WorkPlaneAxis** con relativa indicazione **X,Y,Z**
- 3-Numero Mandrino **Spindle**
- 4-Numero Asse mandrino **AxisSpindle**
- 5-Numero della testa definita in “Base\_NumeroTesta” **Unit**
- 6-Valore dell’asse teorico al **XT,YT,ZT,AT,BT,CT,UT,VT,WT**
- 7-Valore Speed **Speed**
- 8-Valore Feed **Feed**
- 9-Valore **J** quota rapido entrata      PosZ\_Inizio
- 10-Valore **Z** quota rapido entrata      PosZ\_Foratura
- 11-Valore **Q** quota rapido uscita      PosZ\_Ritorno
- 12-Valore **F** feed lavoro      VelZ\_Foratura o Passo
- 13-Valore **K** ritardo      Tempo in Millisec
- 14-Valore **I** incremento      Incremento Successivo
- 15-Valore **H** safety      Incremento Sicurezza
- 16-Valore indicato nell’istruzione o 0 di **X,Y,Z,A,B,C,U,V,W**
- 17-Valore del parametro **P con relativo numero**

**Esempio:**

```
[ParametriGeneraliMoG]
NumeroMoG=2
Esecuzione_M_maggiore_o_uguale_di=35
PartProgramGAL_M=M_Tutti.pp
Esecuzione_G_maggiore_o_uguale_di=0
PartProgramGAL_G=

; Kind 1=M 2=G
[Fun1]
Kind=1
iD=56
Descrizione=Funzione M56
PartProgramGAL=M56.pp

NumeroLocali=4
SlotInput_1=10,123+P1      L10=123+P1
SlotInput_2=17,AxisX      ; L17=Numero asse X
SlotInput_3=14,Unit       ; L14=Numero testa del processo
SlotInput_4=50,Spindle    ; L23=Numero mandrino S

NumeroParametri=2
SlotOutput_1=21,123+L1    ; P21=123+L1
SlotOutput_2=3,345+G1    ; P3=345+G1

[Fun2]
Kind=2
iD=45
Descrizione=Funzione G45

PartProgramGAL=G45.pp

NumeroLocali=4
SlotInput_1=10,123+P1    ; L10=123+P1
SlotInput_2=7,345       ; L7=345
SlotInput_3=4,167       ; L4=167
SlotInput_4=50,312      ; L23=312

NumeroParametri=2
SlotOutput_1=21,123+L1   ; P21=123+L1
SlotOutput_2=3,345+G1   ; P3=345+G1
```

## 12.1 Funzione G89

La funzione Ciclo Fisso G89 se dichiarata nel file “sistema.txt” nella voce PartProgramG89 della sezione ParametriGenerali.

### [ParametriGenerali]

**PartProgramG89= [ nome part program di automazione]**

Ad ogni movimento se segue la Funzione G89, viene richiamato il part program di automazione definito nel file “sistema.txt”, con il passaggio di parametri:

- L1 = Numero asse Z rispetto al Piano di lavoro
- L2 = Numero mandrino S
- L3 = Numero AsseMandrino AS oppure 0
- L4 = Numero della testa definita in “Base\_NumeroTesta” oppure 0
- L5 = Speed mandrino ( valore di speed attivo blocco compreso )

- L6= PosZ\_Inizio
- L7 = PosZ\_Foratura
- L8 = PosZ\_Ritorno
- L9 = VelZ\_Foratura o Passo
- L10 = Tempo in Millisec oppure Percentuale stiramento
- L11 = Primo incremento da PosZ\_Inizio
- L12 = Incremento successivo (L11 – 10%di L11 in ISO)
- L13 = Incremento Sicurezza

I parametri trasferiti sono quelli indicati rispettivamente nei corrispettivi indicatori ISO:

- |   |                      |   |
|---|----------------------|---|
| J | quota rapido entrata | PosZ_Inizio   |
| Z | quota lavoro finale  | PosZ_Foratura   |
| Q | quota rapido uscita  | PosZ_Ritorno  |
| F | feed lavoro          | VelZ_Foratura o Passo   |
| K | ritardo              | Tempo in Millisec oppure perc. stiramento o zero m. rigida  |
| I | incremento           | Primo incremento da PosZ_Inizio ( gli altri incrementi sono dopo il primo pari a I , di I –10%sempre di I ) |
| H | safety               | Incremento Sicurezza  |



## CAPITOLO 13

### 13 Interfacciamento con LOGICA di MACCHINA

Sono previsti degli automatismi per l'interconnessione con la LOGICA di MACCHINA, attraverso 16-32 bit consecutivi per lo scambio di stati e richieste. Ed in particolare:

<b>Logica Macchina</b>	<b>CNC-ISO</b>
Bit 0 Logica OK = 1	Bit 0 Part Program Start = 1
Bit 1 Rich. Stop	Bit 1 Part Program Stop = 1
Bit 2 Rich. PassoPasso	Bit 2 Part Program in PassoPasso = 1
Bit 3 Rich. Kill Part program	Bit 3 Part Program Killed
Bit 4 Allarme = 1	Bit 4 Part Program Allarme acquisito e Part Program in Stop
Bit 5 Warning = 1	Bit 5 Part Program Warning acquisito
Bit 6 Info = 1	Bit 6 Part Program Info acquisito
Bit 7 Lock = 1	Bit 6 Part Program Lock acquisito
Bit 8 Next Step	Bit 8
Bit 9 Manuale	Bit 9
Bit 10	Bit 10
Bit 11	Bit 11
Bit 12	Bit 12
Bit 13	Bit 13
Bit 14	Bit 14
Bit 15	Bit 15
Bit 16 Bit 0 Msg Allarme	Bit 16
Bit 17 Bit 1 Msg Allarme	Bit 17
Bit 18 Bit 2 Msg Allarme	Bit 18
Bit 19 Bit 3 Msg Allarme	Bit 19
Bit 20 Bit 0 Msg Warning	Bit 20
Bit 21 Bit 1 Msg Warning	Bit 21
Bit 22 Bit 2 Msg Warning	Bit 22
Bit 23 Bit 3 Msg Warning	Bit 23

Bit 24 Bit 0 Msg Info	Bit 24
Bit 25 Bit 1 Msg Info	Bit 25
Bit 26 Bit 2 Msg Info	Bit 26
Bit 27 Bit 3 Msg Info	Bit 27

Bit 28 Bit 0 Msg Lock	Bit 28
Bit 29 Bit 1 Msg Lock	Bit 29
Bit 30 Bit 2 Msg Lock	Bit 30
Bit 31 Bit 3 Msg Lock	Bit 31

La definizione delle modalità di automatismo per l'interconnessione con la LOGICA di MACCHINA, avviene tramite i comandi di abbinamento collocati nel file "sistema.txt" ed in particolare:

[AbbinamentoOutputDigitaleControllo]

Base\_OutputDigitaleControllo=[nome input od output digitale]  
Testa1\_OutputDigitaleControllo=[nome input od output digitale]

[AbbinamentoInputDigitaleControllo]

Base\_InputDigitaleControllo=[nome input od output digitale]  
Testa1\_InputDigitaleControllo=[nome input od output digitale]

[AbbinamentoNumeroTesta]

Base\_NumeroTesta=[nome input od output digitale]  
Testa1\_NumeroTesta=[nome input od output digitale]

[AbbinamentoChiaveInputControllo]

Base\_ChiaveInputControllo=[nome input od output digitale]  
Testa1\_ChiaveInputControllo=[nome input od output digitale]

[AbbinamentoNumeroBitControllo]

Base\_NumeroBitControllo=[NumeroBit]  
Testa1\_NumeroBitControllo=[NumeroBit]

Inoltre le istruzioni speciali nel part program sostituiscono gli abbinamenti predefiniti

> Configura\_teste\_CNC

predispone il sistema ad accettare la configurazione vera e propria degli assi teorici che sono: X Y Z A B C U V W , il mandrino Mandrino e le risorse IO

>Base\_NumeroTesta = [ numero della testa da indicare nella istruzione SEC della LM ]

>Base\_ChiaveInputControllo = [ stringa da indicare nella istruzione SEC della LM]

>Base\_InputDigitaleControllo = [ nome input od output digitale ]

>Base\_OutputDigitaleControllo = [ nome input od output digitale ]

>Base\_NumeroBitControllo = [ numero di bit dell'IO del controllo ]

Alla fine della fase di configurazione dovrà esserci il comando:

>Fine\_configura\_teste\_CNC

>

Nella logica di macchina viene richiamato il part program definito nella dichiarazione degli IO digitali.

**Esempio :**

**Nel Part Program ISO**

```
>Configura_teste_CNC
.....
>Base_NumeroTesta=1
>Base_ChiaveInputControllo=LM_Controllo
>Base_InputDigitaleControllo=Q_S1T1.1
>Base_OutputDigitaleControllo=I_S1T1.1
>Base_NumeroBitControllo=32
>Fine_configura_teste_CNC
>
```

**Nel file SISTEMA.TXT**

```
////////////////////////////////////
//          input Logica di Macchina con ISO          //
////////////////////////////////////

i=IO_NAME(501 , "I_S1T1" )
b={
Description="LM Input Staz. 1 Testa 1"
Data_type="Word"           // "Bit" "Nibble" "Byte" "Word" "Dword"
TaskChangeFront=1         // 1=change front 0=always
TaskChangeValue=0        // delta change value for analog signal
TaskModeRun=1            // 0=always 1=No if running 2=Error if running
TaskOnePP="LM_S1T1.pp"
TaskOneName="InOne"
TaskZeroPP="LM_S1T1.pp"
TaskZeroName="InZero"
Estension="1"
Index_system=0
FalseTrue_logic=0
Kind_sensor="DIGITAL_INPUT"
Kind_board=0
Rack_number=0
Address_port=0
Physical_number=1
e=}

////////////////////////////////////
//          Output INFO Logica di Macchina          //
////////////////////////////////////

i=IO_NAME( 501, "Q_S1T1" )
b={
Description="LM Output Staz. 1 Testa 1"
Data_type="Word"           // "Bit" "Nibble" "Byte" "Word" "Dword"
TaskChangeFront=1         // 1=change front 0=always
TaskChangeValue=0        // delta change value for analog signal
TaskModeRun=1            // 0=always 1=No if running 2=Error if running
TaskOnePP="LM_S1T1.pp"
TaskOneName="OutOne"
TaskZeroPP="LM_S1T1.pp"
```

```
TaskZeroName="OutZero"  
Estension="1"  
Index_system=0  
FalseTrue_logic=0  
Kind_sensor="DIGITAL_OUTPUT"  
Kind_board=0  
Rack_number=0  
Address_port=0  
Physical_number=512+1  
e=}
```

**Nel PART PROGRAM di automazione LM\_S1T1.PP**

```
*****  
;* Stazione 1 Testa 1 *  
*****  
;  
;  
*****  
;* CNC->Automation variazione a livello uno *  
*****  
PartProgram[InOne]  
  
-RET
```

```
*****  
;* CNC->Automation variazione a livello zero *  
*****  
PartProgram[InZero]  
  
-RET
```

```
*****  
;* Automation->CNC variazione a livello uno *  
*****  
PartProgram[OutOne]  
  
-SEC/LM_Controllo,1  
-RET
```

```
*****  
;* Automation->CNC variazione a livello zero*  
*****  
PartProgram[OutZero]  
  
-SEC/LM_Controllo,1  
-RET
```

## CAPITOLO 14

### 14 Funzioni speciali

#### 14.1 Istruzioni di I/O

Per la connessione diretta con i segnali di Input ed Output Digitali sono previste due istruzioni speciali :

1. OUTPUT(nome oppure numero del IO digitale)= [1 o 0]
2. [Parametro] = INPUT(nome oppure numero del IO digitale)

#### Esempi:

```
OUTPUT(101)=1  
OUTPUT(CAN_O1)=0  
P1= INPUT(101)  
P1= INPUT(CAN_I1)  
L=loop  
P3=INPUT(CAN_O1)  
G4 500
```

## 14.2 Istruzione per l' avviamento altri processi ISO

Per eseguire l'avviamento di un altro processo ISO si può utilizzare l'istruzione :

### **>ISO=Nome programma%Nome del processo**

il file o programma si deve trovare nella directory del processo

### **>ISO=ISOCNC%Nome del processo**

Verrà eseguito l'ultimo programma selezionato dall'operatore

### **>ISO=Nome programma%NumeroDelProcesso**

il file o programma si deve trovare nella directory di sistema CNC

#### **Esempi:**

1) >ISO=A.prg%scara

Il file A.prg è un file con istruzioni ISO : il file si deve trovare nella directory del processo

X..  
x..  
G2X...

Verrà eseguito sul processo di nome : scar

2) >ISO=ISOCNC%scara

Verrà eseguito l'ultimo programma selezionato dall'operatore sul processo di nome : scar

3) >ISO=Fresa1.prg%3

Il file Fresa1.prg è un file con istruzioni ISO : il file si deve trovare nella directory del processo

X..  
x..  
G2X...

Verrà eseguito sul processo di numero : 3

### 14.3 Istruzione per l'avviamento processi PLC

Per eseguire l'avviamento di un processo PLC si può utilizzare l'istruzione :

**>PLC=Nome programma%WAIT**

**>PLC=Nome programma**

La voce WAIT se presente fa sì che il programma ISO che attiva il processo PLC attende la fine del processo PLC, prima di proseguire nell'esecuzione dell'istruzione successiva

Nelle locali del programma PLC sono caricate le seguenti voci:

L1 = Numero asse Z rispetto al Piano di lavoro

L2 = Numero mandrino S

L3 = Numero AsseMandrino AS oppure 0

L4 = Numero della testa definita in "Base\_NumeroTesta" oppure 0

L5 = Speed mandrino ( valore di speed attivo blocco compreso )

L6 = Tipo di funzione chiamante la logica di macchina 6=Con Wait 7= Senza wait

#### **Esempio:**

>PLC=Alimenta%WAIT

Il file Alimenta.pp è un file con istruzioni PLC

-LET/G1,G+L1

.

-MOV/Z,3

-END

#### 14.4 Istruzione per la sincronizzazione di processi ISO o PLC

Per attendere il completamento di un processo ISO o di processo PLC si utilizza l'istruzione :

```
>WAIT=PLC  
>WAIT=%Nome del processo ISO  
>WAIT=Nome del processo ISO  
>WAIT= PLC %Nome del processo 1 ISO%Nome del processo 2 ISO
```

**Esempi:**

1)

```
>ISO=A.prg%scara  
>PLC=Alimenta  
..  
..  
..  
>WAIT= PLC%Scara
```

Attende il completamento del processo PLC e del programma di lavorazione ISO di nome A.prg attivato sul processo scara

2)

```
>ISO=A.prg%scara  
..  
..  
..  
>WAIT= %Scara
```

Attende il completamento del programma di lavorazione ISO di nome A.prg attivato sul processo scara

3)

```
>ISO=A.prg%scara  
>ISO=B.prg%FresaA  
>PLC=Alimenta  
..  
..  
..  
>WAIT= PLC%Scara  
>WAIT= %FresaA
```

Attende il completamento del processo PLC, del programma di lavorazione ISO di nome A.prg attivato sul processo scara e del programma di lavorazione ISO di nome B.prg attivato sul processo FresaA

## 14.5 Richiamo di programmi ISO all'esterno del file.

Per eseguire il richiamo di programmi su un altro file si utilizza l'istruzione :

**L Nome file ISO , sottoprogramma nel file ISO**  
**L Nome file ISO**

si può richiamare un sottoprogramma ( Funzione1 ) in un altro file ( Libreria )

L Libreria, Funzione1

### **Esempio:**

**L Biella, Interno**

**L Asola**

Il file Biella può essere così composto

x..  
y..

G32

L=Interno

..  
G32

Il file Asola può essere così composto

x..  
y..



## CAPITOLO 15

### 15 Piano lavoro

I nomi degli assi macchina possono essere scelti tra i seguenti: X,Y,Z,A,B,C,U,V,W

Due qualsiasi di questi determinano l'ascissa e l'ordinata del piano di lavoro, su cui si applica la correzione raggio utensile nel piano, mentre un terzo assume la funzione di asse perpendicolare al piano di lavoro, asse a cui viene applicata la correzione lunghezza.

Gli eventuali altri assi sono correlati a questi, ma non sono interessati dai correttori raggio e lunghezza.

#### 15.1 Piani di lavoro principali

Le funzioni G17,G18 e G19 specificano i tre piani principali predisposti dal sistema.

G17 Piano di lavoro XY, asse perpendicolare Z, l'ascissa del piano cartesiano è X, l'ordinata è Y

G18 Piano di lavoro ZX, asse perpendicolare Y, l'ascissa del piano cartesiano è Z, l'ordinata è X

G19 Piano di lavoro YZ, asse perpendicolare X, l'ascissa del piano cartesiano è Y, l'ordinata è Z

La programmazione dei pezzi in G18 ed in G19 va fatta ruotando il disegno in modo che l'asse dell'ascissa (Z in G18,Y in G19) diventi l'asse orizzontale con i valori positivi rivolti verso destra. Con questo accorgimento la programmazione risulta coerente con quella fatta in G17.

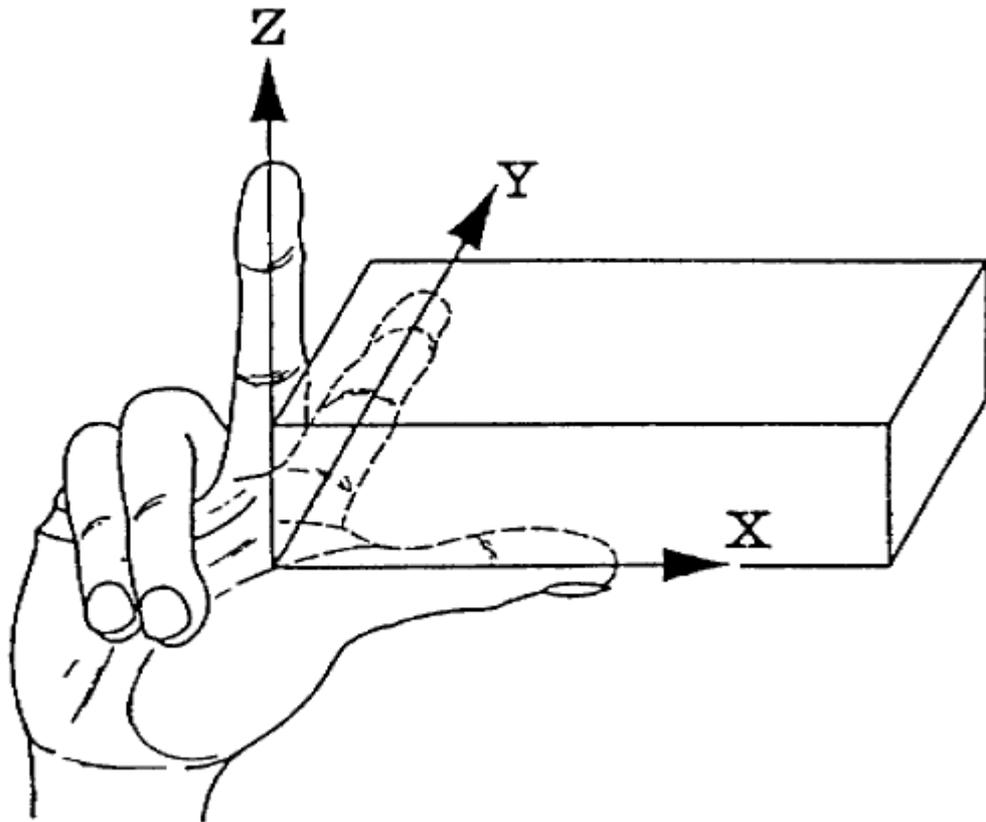
I valori delle ascisse a destra dello zero sono positivi, quelli a sinistra negativi. I valori delle ordinate sopra lo zero sono positivi, quelli sotto negativi.

Volendo definire piani di lavoro diversi da XY,ZX e YZ, si deve utilizzare la funzione G!/ seguita dai nomi degli assi che nell'ordine saranno: ascissa, ordinata ed asse perpendicolare.

Ad esempio la frase G17 UVC assegnerà all'asse U il ruolo di ascissa, all'asse V il ruolo di ordinata ed all'asse C il ruolo di asse perpendicolare.

Le funzioni G17,G18,G19 vanno programmate in un blocco da sole.

Ci sono due tipi di asse, **assi di alimentazione (assi lineari)** e **assi rotatori**. Le loro disposizione e direzione sono normalizzati in DIN 66217. Ci sono tre assi di alimentazione di base. Questi sono designati X, Y e Z. La loro posizione relativa l'uno all'altro può essere determinata con l'aiuto della regola della mano destra:



Determinazione della posizione dei tre assi di alimentazione di base utilizzando la regola della mano destra. Secondo DIN 66217 l'asse Z per una macchina utensile corrisponde a, o corre in modo parallelo all'asse dell'alberino di lavoro.

L'asse principale nel piano di posizionamento viene designato l'asse X. Esso corre in modo parallelo al tavolo di lavorazione pezzi, e preferibilmente in modo orizzontale.

## CAPITOLO 16

### 16 Funzioni di movimento

Il movimento degli assi viene realizzato programmando il nome seguito dal valore della coordinata da raggiungere.

I nomi degli assi macchina possono essere scelti tra i seguenti: X, Y, Z, A, B, C, U, V, W.

Assi di gruppi meccanici diversi possono avere lo stesso nome.

Il valore della coordinata può essere positivo o negativo (il segno + può non essere programmato); usare il punto decimale e non la virgola per separare la parte intera dai decimali.

Il movimento compiuto dagli assi lineari per raggiungere la posizione programmata dipende dal tipo di funzione preparatoria (G) usata. Se il blocco non contiene funzioni G, il movimento alla posizione programmata è eseguito in interpolazione lineare, cioè seguendo la linea retta che unisce il punto di partenza con il punto d'arrivo programmato. Le funzioni X, Y e Z sono modali e quindi non devono essere programmate nuovamente se la posizione dei rispettivi assi non cambia.

U, V, W sono assi aggiuntivi lineari e paralleli. Definiscono le quote alle quali dovranno posizionare gli assi aggiuntivi. Per convenzione le funzioni U, V, W indicano gli assi aggiuntivi lineari e gli assi paralleli a X, Y e Z. Per la programmazione delle funzioni U, V e W rimane valido ciò che è stato detto riguardo alle funzioni X, Y e Z.

A, B, C sono assi aggiuntivi rotativi. Definiscono le quote alle quali si dovranno posizionare gli assi aggiuntivi identificati con i nomi A, B, e C. Per convenzione le funzioni A, B e C indicano gli assi aggiuntivi rotanti rispettivamente attorno ad X, Y e Z. Le funzioni A, B e C sono modali e poiché esprimono quote angolari, vanno programmati in gradi.

Le funzioni di movimento sono: G00, G01, G02, G03



## CAPITOLO 17

### 17 Funzioni G

Sono funzioni preparatorie che servono a predisporre il sistema o la macchina utensile alle operazioni successive. Esse sono composte dalla lettera G seguita da 2 (in questo ambiente di programmazione) o 3 cifre e possono essere valide nel blocco in cui sono programmate o fino a quando non sono cancellate da un'altra funzione. Quelle utilizzate in questo ambiente di programmazione sono elencate nel seguito:

G00	posizionamento rapido degli assi
G01	interpolazione lineare
G02	interpolazione circolare o elicoidale senso orario
G03	interpolazione circolare o elicoidale senso antiorario
G04	pausa temporizzata, tempo di pausa programmato.
G08	decelerazione alla fine del blocco che la contiene
G09	attivazione modalità "Guardare Avanti"
G17	specifica XY come piano lavoro e Z asse perpendicolare
G18	specifica ZX come piano lavoro e Y asse perpendicolare
G19	specifica YZ come piano lavoro e X asse perpendicolare
G30	decelerazione alla fine del blocco che la contiene e ripartenza in continuo
G32	fine subroutine
G40	annulla G41 e G42
G41	attivazione correzione raggio, utensile a sinistra del profilo
G42	attivazione correzione raggio, utensile a destra del profilo
G49	dichiarazione valore raggio
G50	fine rototraslazione
G51	rototraslazione
G52	shift delle origini uguale G92
G54	mirror X
G55	mirror Y
G56	mirror Z
G57	mirror X e Y
G58	mirror Z e X
G59	mirror Y e Z
G60	fine fattore di scala
G61	fattore di scala
G62	centro del cerchio K1 assoluto K2 Incrementale
G70	programmazione in pollici

G71	programmazione in millimetri
G75	programmazione cartesiana
G76	programmazione polare
G78	impostazione tangenziale al percorso 2D
G79	fine impostazione tangenziale al percorso 2D
G80	annullamento cicli fissi
G81	ciclo fisso per foratura
G82	ciclo fisso per lamatura
G83	ciclo fisso per foratura profonda
G84	ciclo fisso per maschiatura
G85	ciclo fisso per alesatura
G86	ciclo fisso per barenatura
G89	ciclo fisso con richiamo di un part program AxesBrain
G90	programmazione assoluta
G91	programmazione incrementale
G92	shift delle origini
G100 - G1999	funzioni AxesBrain

## **17.1 Funzioni preparatorie ( G )**

### **17.2 G00: Posizionamento in rapido**

Causa il movimento degli assi in interpolazione lineare, alla massima velocità consentita dalla macchina eventualmente variata dal comando di “Override Feed”.

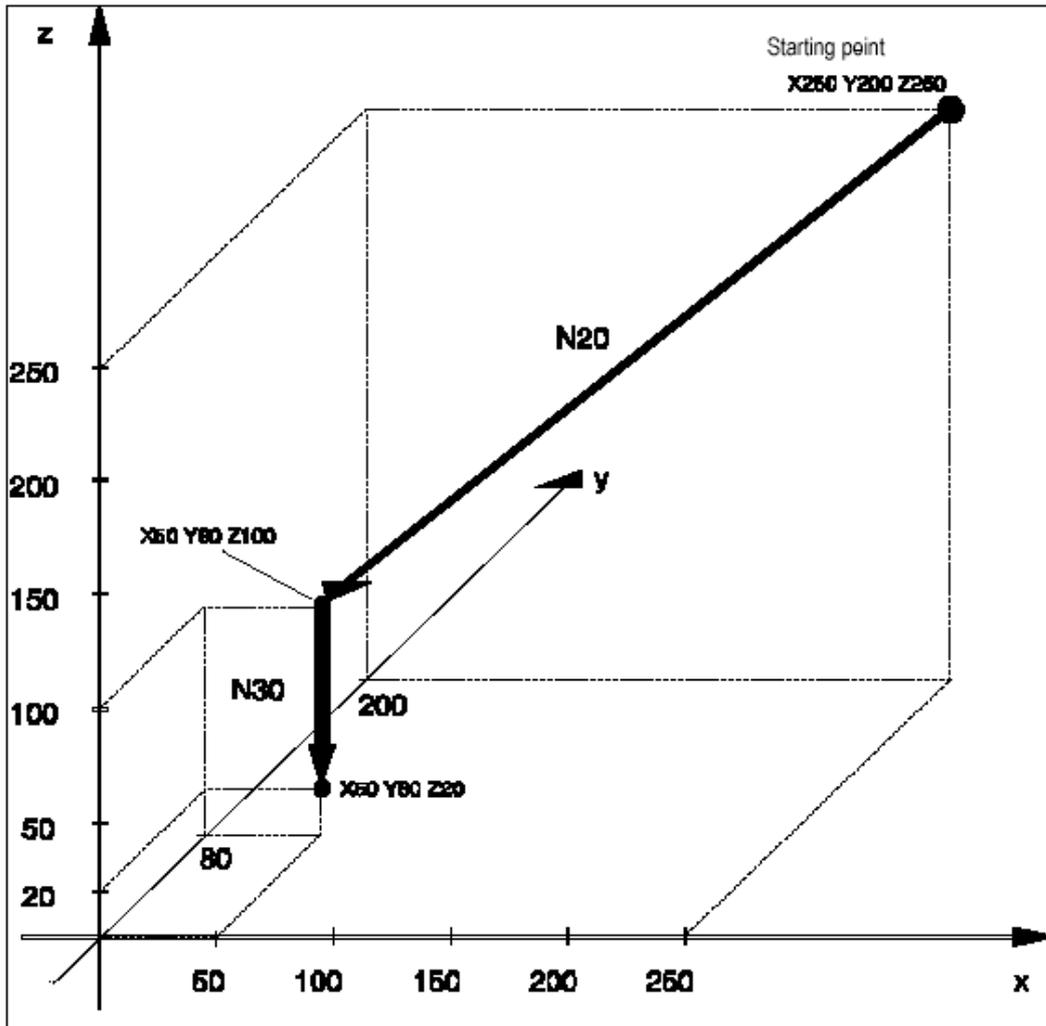
Esempio:

(Posizione di partenza: X = 250, Y = 200, Z = 250;)

N10 G90

N20 G0 X50 Y80 Z100      Spostarsi al punto X50 Y80 Z100 in traversa rapida

N30 Z20                      poi spostarsi al Z20 in traversa rapida.



### 17.3 G01: Interpolazione lineare

Causa il movimento contemporaneo degli assi programmati in un blocco, in linea retta, dal punto in cui essi si trovano fino alla posizione programmata. Tale movimento avverrà alla velocità di lavoro programmata con la funzione F. Si noti che, qualora nel blocco non compaiano funzioni G, il movimento verso la posizione programmata sarà eseguito in interpolazione lineare che devono muoversi. L'interpolazione lineare è consentita su uno qualunque dei piani coordinati (XY, ZX, YZ) e anche nello spazio, cioè anche nello spazio, cioè con movimento contemporaneo dei tre assi. In presenza di assi aggiuntivi continui e programmabili si può anche avere l'interpolazione lineare su più di tre assi (9 al massimo).

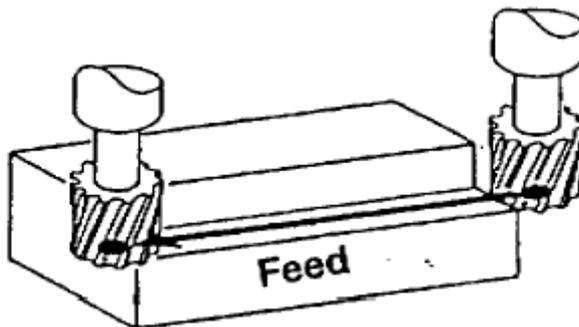
#### Sintassi

G1X... Y... F... ..

L'istruzione **interpolazione lineare (interpolazione in linea retta)** nella velocità dell'avanzamento viene selezionata utilizzando la parola di programma G01. Le seguenti sono possibili o necessari come condizioni aggiuntive:

- le coordinate di destinazione
- la velocità di avanzamento
- la velocità della rotazione o del taglio

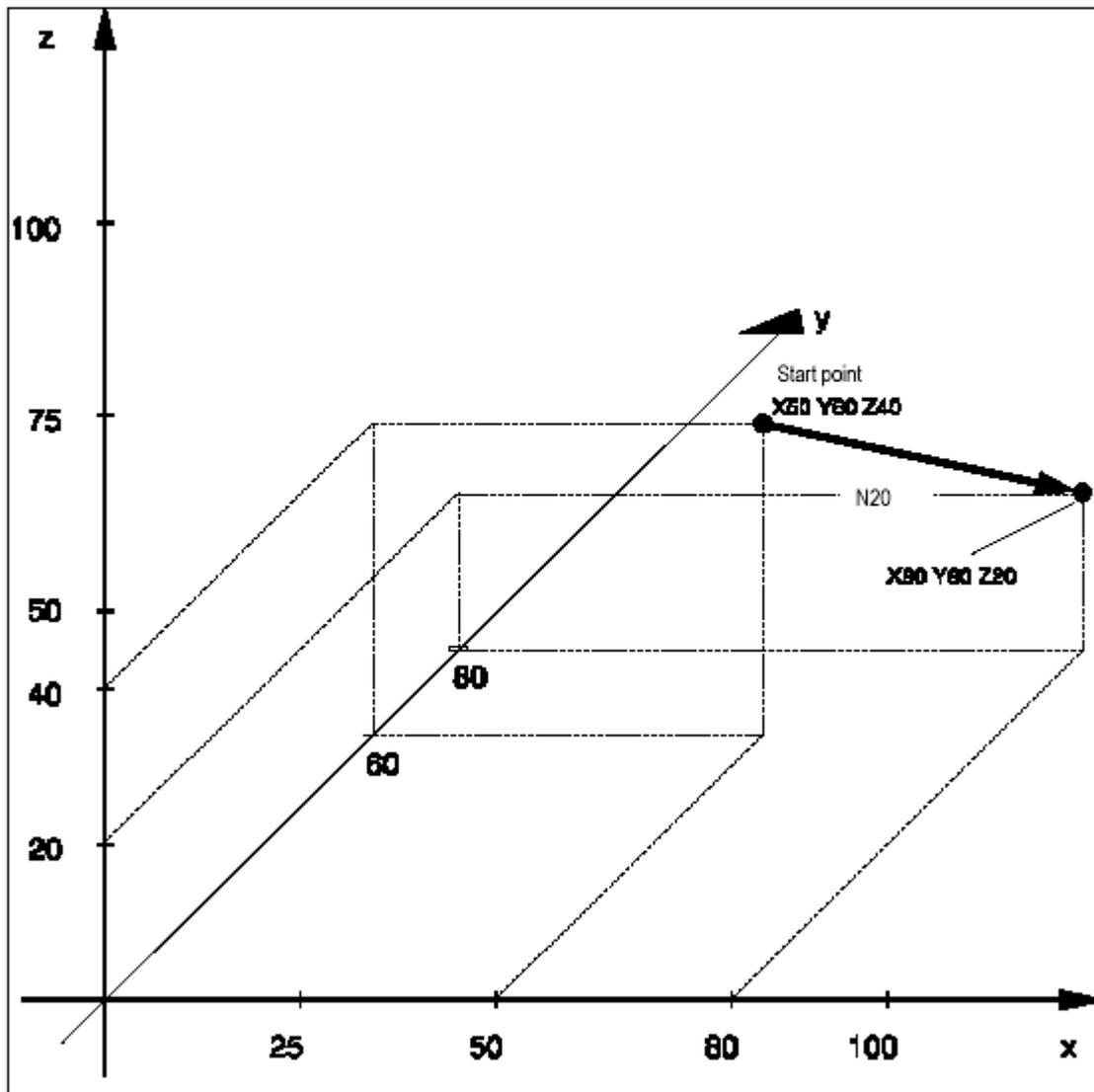
L'istruzione G01 fa in modo che l'attrezzo venga posizionato in un linea retta rispetto al punto di destinazione indicato con la velocità d'avanzamento che è stata specificata come una condizione aggiuntiva o che era già stata programmata. (Velocità di avanzamento, velocità di rotazione e velocità di taglio sono tutte effettive in modo modale). Tutti gli assi programmati nel blocco sono posizionati simultaneamente. Il percorso di movimento dell'attrezzo può essere o una linea retta parallela all'asse oppure una linea retta non parallela all'asse.



**Esempio:**

(Posizione di partenza: X = 50, Y = 60, Z = 40)

N10 G90			
N20 G1	X80 Y80 Z20	F40	S100
	punto di destinazione	velocita di avanzamento 40 mm/min	velocita di rotazione 100 g/min



## 17.4 G02-G03 : Interpolazione circolare

Le funzioni G02 e G03 causano il movimento lungo un arco di cerchio, fino al raggiungimento della posizione programmata nel medesimo blocco. Come punto di inizio dell'arco, il controllo assume il precedente punto programmato.

Programmazione punto finale e centro. Si utilizzano le funzioni:

G02 o G03 per definire il senso orario (G02) o antiorario (G03)

X Y Z per definire il punto di arrivo

I J K per definire il centro dell'arco di cerchio, rispettivamente lungo gli assi X, Y e Z

### Sintassi:

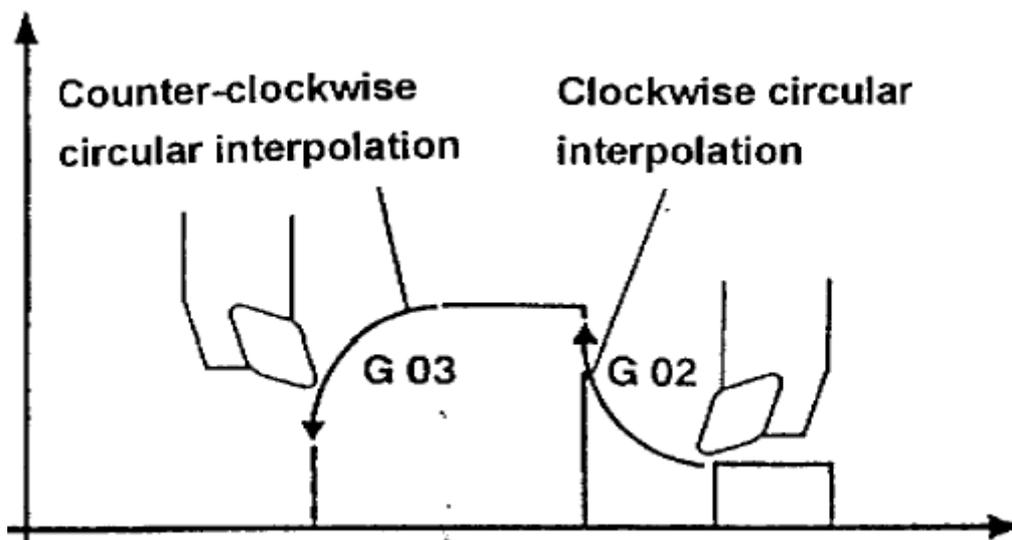
G2/G3 X... Y... I... J... (G17 attivo)

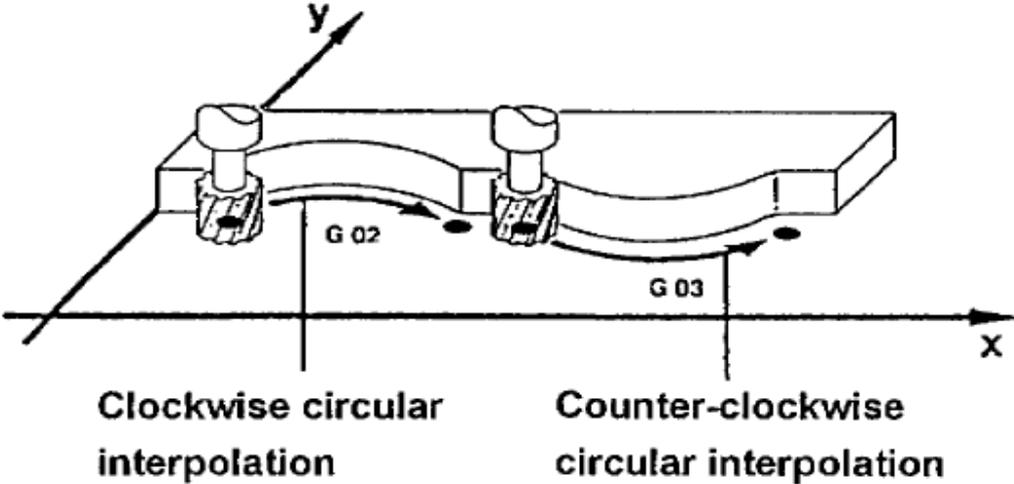
G2/G3 Z... X... K... I... (G18 attivo)

G2/G3 Y... Z... J... K... (G19 attivo)

L'istruzione di posizionamento per interpolazione circolare con punto centrale specificato in direzione oraria viene selezionata con la parola programma G02. L'istruzione di posizionamento per interpolazione circolare con punto centrale specificato in direzione antioraria viene selezionata con la parola programma G03.

Queste istruzioni sono utilizzate per la programmazione dei contorni curvati del pezzo in lavorazione. La curva deve essere posta nel piano identificato dalle istruzioni G17 a G20.

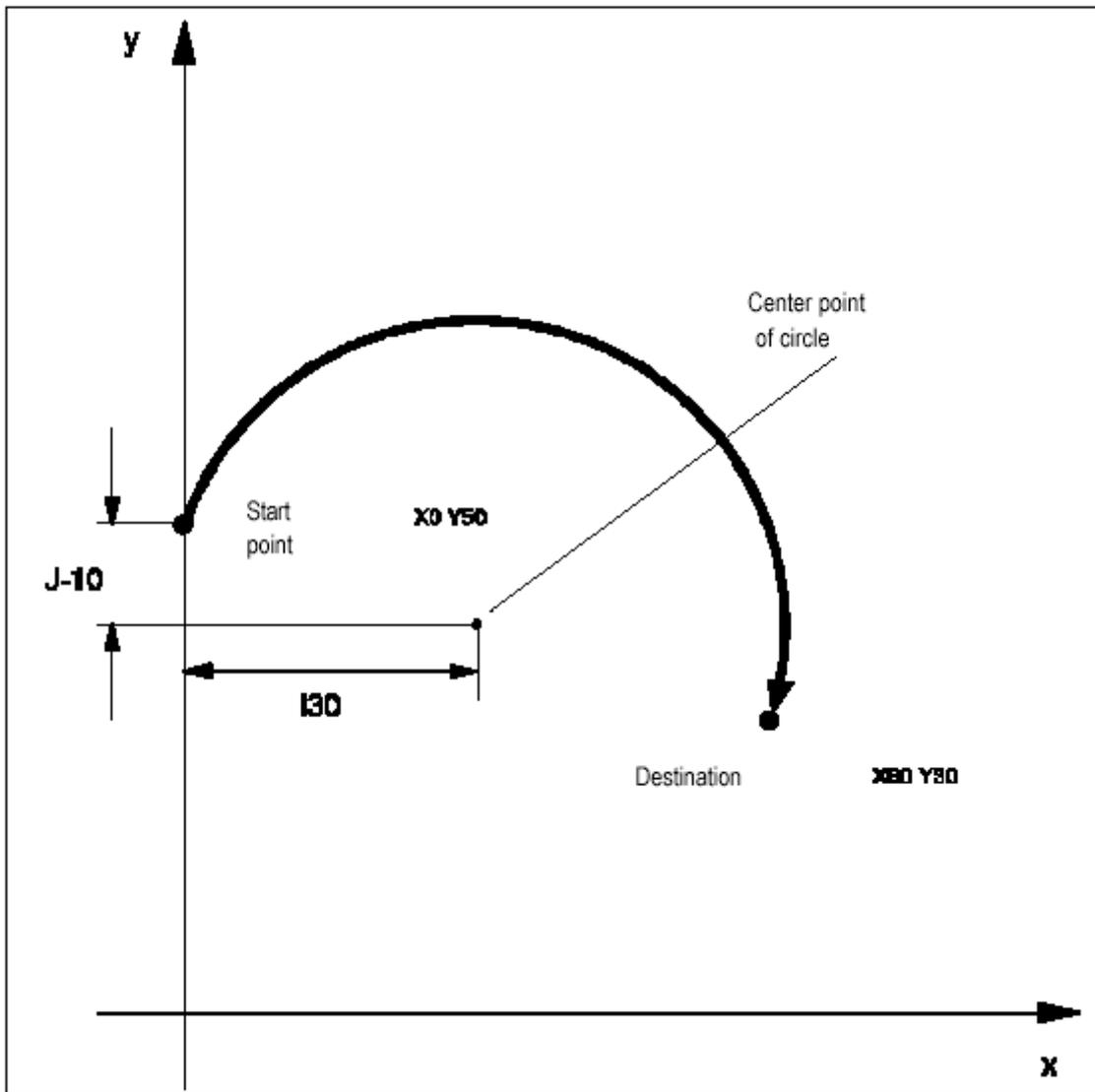




**Esempio:**

(Posizione di partenza: X = 0, Y = 50)

N30 G2	X60 Y30	130 J-10	F200
	destinazione	dimensione incrementale del centro cerchio relativa alla posizione di partenza	



### 17.5 G04: Sosta programmata

Causa l'arresto del movimento degli assi macchina per il tempo definito in millesimi di secondo

**Sintassi:**

G04            pausa temporizzata, tempo di pausa programmato.

### 17.6 G08,G09: Guardare avanti OFF/ON

La funzione **Guardare Avanti** viene attivata usando l'istruzione **G09** o dalle funzioni **G01 G02** e **G03** e disattivata usando l'istruzione **G08**.

**Sintassi:**

G08            decelerazione alla fine del blocco che la contiene

G09            attivazione modalità "Guardare Avanti"

## Metodo di operazione della funzione “Guardare Avanti”

Come norma, cioè con G08 attivo, i blocchi di moto CN vengono lavorati come segue:

All’inizio del blocco di moto CN, l’accelerazione avviene da 0 alla velocità di avanzamento. Alla fine avviene la frenatura del blocco CN, in modo che la velocità di avanzamento sia zero quando il punto di destinazione del blocco di movimento viene raggiunto e la corsa si arresta esattamente a quel punto.

La funzione “Guardare Avanti” comunque, ha il seguente effetto.

Quando “Guardare Avanti” è attiva il sistema riconosce, qualche blocco CN in anticipo, in quali posizioni gli assi devono essere accelerati o frenati. La velocità di avanzamento viene automaticamente regolata tramite accelerazione o frenatura. La regolazione viene fatta prendendo in considerazione i seguenti fattori:

la velocità di avanzamento programmato nei blocchi CN individuali

la curva del percorso e gli angoli, prendendo in considerazione i valori massimi ammissibili di accelerazione degli assi

le velocità d’asse massime ammissibili

Così un avanzamento uniforme è garantito in anticipo per due o più blocchi CN . Questo porta ad una lavorazione più uniforme e più veloce (in qualche caso, molto più veloce), la quale, a sua volta, risulta in una qualità più elevata di superficie e in una produttività aumentata . Per garantire questo, il controllo deve considerare non solo i blocchi CN attuali, ma deve anche “guardare avanti” e prendere in considerazione il corso dei blocchi CN che seguono. Per avere una velocità costante di avanzamento su più blocchi in anticipo, quando “Guardare Avanti” è attivo, il moto non si ferma alle destinazioni di blocco programmate, ma continua con la velocità di avanzamento raggiunta alla fine del blocco .

Se la velocità di avanzamento deve essere ridotta a 0 alla fine del blocco, per es. perché G09 è stato disattivato, il moto si ferma esattamente alla destinazione dell’ultimo blocco prima della disattivazione della funzione “Guardare Avanti”.

### Metodo di operazione:

Con il controllo un elevato numero di blocchi CN può essere “guardato” in anticipo con la funzione G09 attiva. Il numero di blocchi CN che possono essere Guardati in anticipo dal sistema dipende dallo spazio di memoria disponibile nel finecorsa blocchi dinamico: questo è almeno 4 blocchi.

Quando si lavorano più blocchi CN con la funzione “Guardare Avanti” attiva, la velocità di avanzamento è limitata in modo che una riduzione a 0 della velocità di avanzamento sia possibile fino all’ultimo blocco da lavorare con G09 attivo e che ci sia almeno un punto di interpolazione in ogni blocco.

Se un blocco senza istruzioni di posizionamento appare dentro una sequenza di blocchi CN che sono da lavorare con G09 attivo, la velocità di avanzamento viene ridotta a 0 alla fine del blocco si moto precedente.

Se, quando G09 è attivo, i tempi minimi di esecuzione blocco non sono troppo corti o i tempi massimi di preparazione blocco non sono troppo lunghi, è sempre garantito che un nuovo blocco nella preparazione di geometria sarà finito in tempo e che la procedura di interpolazione è disponibile per la lavorazione in tempo.

Così è quindi possibile, se necessario, accelerare o frenare da una distanza di parecchi blocchi.

### Esempio:

```

N30   G9
N40   G4   F500   o N40 G11
N50   G1   X20   Y30
...
N200  M30
    
```

### Nota:

Quando “Guardare Avanti” è attiva, può essere redditizio limitare l’accelerazione usando l’istruzione “accelerazione programmabile” . Questo causa un’impostazione piana dell’accelerazione quando “Guardare Avanti” è attiva.

### 17.7 G17,G18,G19: Selezione del piano di lavoro

Definiscono il piano di lavoro nel quale si possono programmare l'interpolazione circolare, la compensazione del raggio utensili e i cicli fissi di foratura.

Le tre funzioni G17, G18, G19 selezionano rispettivamente i piani XY( asse utensile Z), ZX ( asse utensile Y) e YZ( asse utensile X).

Il piano di lavoro è, per definizione, il piano perpendicolare all'asse utensile.

#### Sintassi:

G17 ... Selezione del piano X/Y

G18 ... Selezione del piano Z/X

G19 ... Selezione del piano Y/Z

### 17.8 G30: Decelerazione forzata

Causa la decelerazione forzata con la ripartenza in movimentazione continua

#### Sintassi:

G30 decelerazione alla fine del blocco che la contiene e ripartenza in continuo

### 17.9 G40,G41,G42: Compensazione raggio utensile nel piano

La compensazione del raggio utensile nel piano consente di programmare una con tornitura facendo riferimento direttamente alle quote del disegno e cioè del profilo da realizzare, senza considerare l'ingombro dell'utensile. Tale ingombro dovrà essere specificato o nella tabella utensile oppure direttamente con la G49 . Durante l'esecuzione del programma il CNC calcola, in base al valore del raggio utensile impostato, la traiettoria che deve essere seguita

dal centro utensile per ottenere il profilo programmato. Per attivare la compensazione raggio sono previste le funzioni G41( utensile a sinistra del pezzo) e G42 ( utensile a destra del pezzo). La posizione dell'utensile rispetto al pezzo dipende non solo dalla funzione programmata, ma anche dal segno del valore assegnato al valore del correttore; se il segno è negativo, l'effetto delle funzioni G41 e G42 non è più quello descritto sopra ma è invertito( ad esempio, la funzione G41 attiverà la compensazione raggio a destra invece che a sinistra). Il concetto di utensile a sinistra o a destra del pezzo è dato percorrendo il profilo nella direzione programmata.

La compensazione è attiva nel piano di lavoro definito tramite G17, G18 o G19. Qualora il movimento dell'utensile sia programmato contemporaneamente su tre assi, la compensazione del raggio è comunque eseguita nel piano di lavoro selezionato.

Il controllo gestisce una logica che permette di riconoscere ed eliminare eventuali interferenze tra percorso compensato e pezzo.

Tale logica utilizza una "LOOK-AHEAD " di 128 enti .

Per "LOOK-AHEAD " si intende la capacità che ha il controllo di leggere e quindi di conoscere il percorso utensile con un certo anticipo prima di eseguirlo.

Il principio usato per il calcolo del percorso compensato è il seguente:

prima di partire con l'esecuzione del programma il controllo legge una certa quantità di segmenti programmati pari a 128. Il controllo trasla poi ogni segmento a destra ( G42) o a sinistra ( G41) di una quantità pari al raggio impostato .

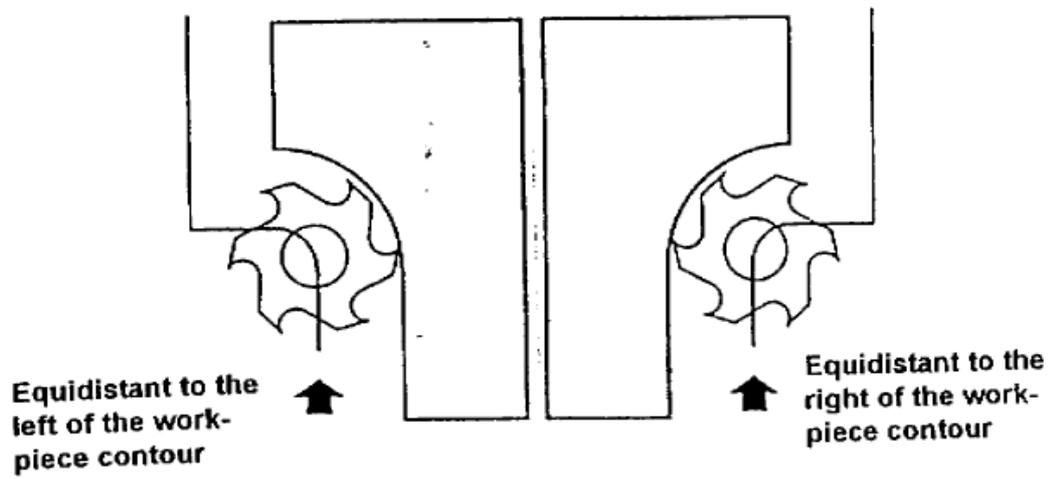
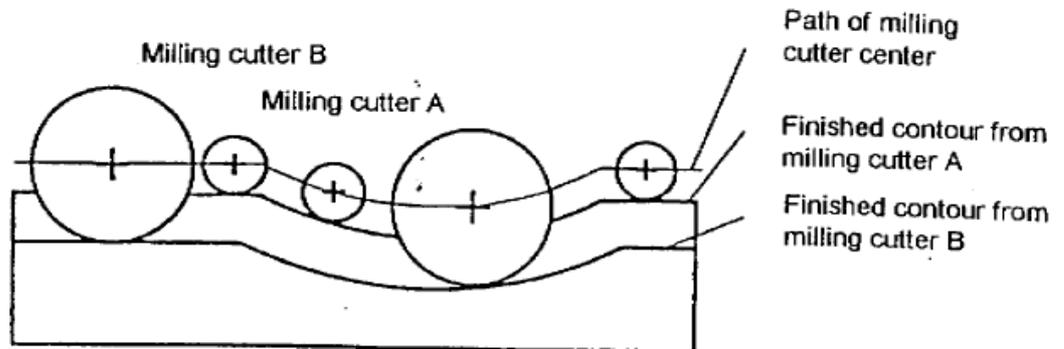
per ciascun segmento traslato viene calcolato il punto d'intersezione con il segmento successivo (traslato anch'esso). Il punto d'intersezione è considerato come punto finale di quel segmento e punto iniziale del successivo

la compensazione esterna di spigoli avviene in interpolazione circolare.

A questo punto il CNC prende in esame il primo segmento compensato e lo confronta con tutti i segmenti successivi traslati:

se il segmento in questione non interseca nessuno dei segmenti successivi( ad eccezione di quello adiacente ) significa che non esistono interferenze e quindi il segmento viene eseguito. Questa operazione di confronto viene ripetuta per il secondo segmento traslato e poi per ciascuno dei segmenti successivi, fino ad esaurimento;

se il segmento interseca anche solo uno dei successivi segmenti, questo significa che viene a crearsi un anello chiuso e quindi un'interferenza. In tal caso il controllo elimina automaticamente tutti i segmenti dell'anello chiuso e manda in esecuzione solamente i punti che non si trovano all'interno dell'anello.



## 17.10 G49: dichiarazione valore correttore raggio

Definisce il valore del correttore raggio

### Sintassi:

G49            dichiarazione valore raggio

Esempio:

F2000

G0X0Y0

G49 I4

G41

G62 K2

G1X15Y5

X10Y15

Y45

G2X30Y65I20

G1X85

G2X90Y60R-5

G1X95

Y15

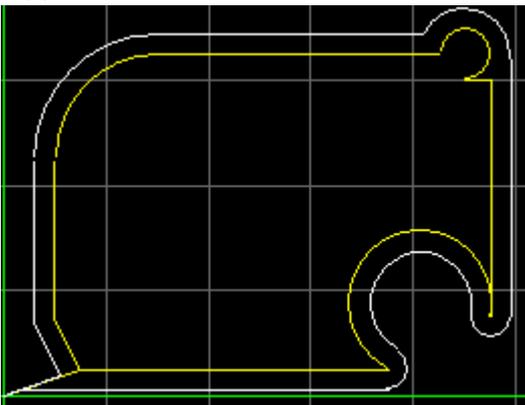
G3X75Y5R-14

G1X15

G1X0Y0

G40

M30



### 17.11 G50: Fine Rototraslazione

Annulla la funzione G51

### 17.12 G51: Rototraslazione

Tramite la rototraslazione parziale, è possibile rototraslare un programma intero o una parte di un programma dentro un piano attivo.

#### Sintassi:

G51                    rototraslazione

Il valore in X traslazione lungo la coordinata ascissa  
Il valore in Y traslazione lungo la coordinata ordinata  
Il valore in I angolo in gradi di rotazione.

### 17.13 G60: Fine Fattore di scala

Annulla la funzione G61

### 17.14 G61: Fattore di scala

La funzione G61 attiva il fattore di scala definito come parametro

#### Sintassi:

G61...fattore di scala                    le coordinate sono moltiplicate per il fattore di scala

G62                    centro del cerchio K1 assoluto K2 Incrementale

### **17.15 G62:centro del cerchio K1 assoluto K2 Incrementale**

La funzione G62 serve per definire la modalità di programmazione delle quote del centro del cerchio ( La quale è definita per default nel sistema.txt ) .

**Sintassi:**

G62 K1          Programmazione assoluta delle quote del centro del cerchio

G62 K2          Programmazione incrementale delle quote del centro del cerchio

### **17.16 G70,G71: Commutazione metrico/pollici**

Le funzioni G70 e G71 selezionano rispettivamente il sistema di misura pollici e il sistema di misura metrico.

**Sintassi:**

G70...          Programmazione nel formato pollici

G71...          Programmazione nel formato metrico

## 17.17 G78,G79 Impostazione tangenziale al percorso 2D

### Sintassi:

G78 C... Impostazione tangenziale al percorso 2D ON

G79 Impostazione tangenziale al percorso 2D OFF

La funzione **Impostazione tangenziale al percorso 2D** permette l'orientamento di un'asse rotatorio durante un movimento di corsa in un piano in modo che un angolo impostato con la tangente sia sempre ottenuto al punto raggiunto ogni volta.



### Esempi di applicazioni:

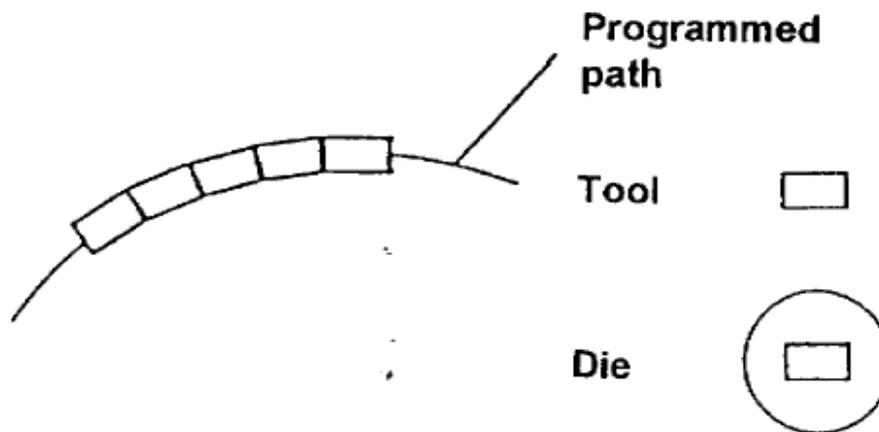
- **Segare**  
Per ottenere il contorno mentre si sega, la sega deve essere girata durante il movimento della corsa in modo che la lama della sega sia ogni volta posizionata in modo tangenziale al contorno.
- **Saldare al laser**  
Durante la saldatura al laser, l'avanzamento del materiale deve essere eseguito ad un certo angolo rispetto al raggio laser. Il materiale deve essere sempre convogliato nella direzione della lavorazione davanti al raggio laser.

- Tornitura

Se durante la tornitura, il materiale è sempre da rimuovere con la punta dell'attrezzo di taglio, la punta deve sempre essere guidata in modo tangenziale lungo il contorno del pezzo in lavorazione. Se invece il materiale deve essere rimosso dal pezzo in lavorazione tramite posizione sull'attrezzo di taglio, l'attrezzo di taglio deve sempre essere condotto ad un certo angolo inclinato lungo il contorno del pezzo in lavorazione.

- Punzonatura/erosione

Se il contorno deve essere ottenuto tramite punzonatura o erosione, l'attrezzo o lo stampo deve essere sempre orientato secondo il contorno desiderato del pezzo in lavorazione.



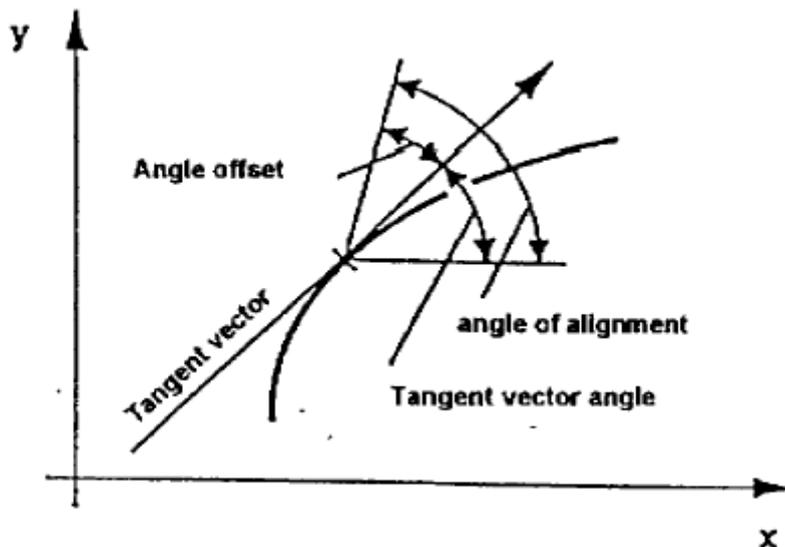
Il **vettore tangenziale** è un vettore unità che indica nella direzione istantanea del moto nel piano attivo ad ogni punto del percorso di moto. L'angolo di vettore tangenziale è l'angolo formato tra il vettore tangenziale e l'asse principale del sistema di coordinate. L'angolo di allineamento viene calcolato dalla somma dell'angolo di vettore tangenziale e qualsiasi sfalsamento di angolo che potrebbe essere stato programmato.

Programmazione:

La funzione **impostazione tangenziale al percorso 2D** viene attivata dal comando modalmente effettivo **G78**. Questa funzione è effettiva a partire dal blocco successivo che contiene G78. Nel blocco C78, il nome dell'asse rotatorio **deve essere programmato con il valore dello sfalsamento dell'angolo tangenziale**. Per programmare un imbocco con un certo angolo relativo alla rispettiva tangente al percorso di moto (sfalsamento d'angolo), il valore dell'asse rotatorio deve essere specificato con il valore di sfalsamento d'angolo desiderato insieme a G78.

La funzione **impostazione tangenziale al percorso 2D** viene **disattivata** utilizzando il comando G79. Quando la funzione **impostazione tangenziale al percorso 2D** viene attivata, l'asse rotatorio segue il percorso più corto (gira in  $<180^\circ$ ) verso l'angolo di allineamento all'inizio dell'elaborazione usando questa funzione. La funzione **impostazione tangenziale al percorso 2D** è già attiva quando il blocco G78 viene elaborato.

I valori dello sfalsamento d'angolo programmato insieme con l'indirizzo d'asse dell'asse rotatorio sono limitati da  $-360^\circ$  a  $+360^\circ$ .



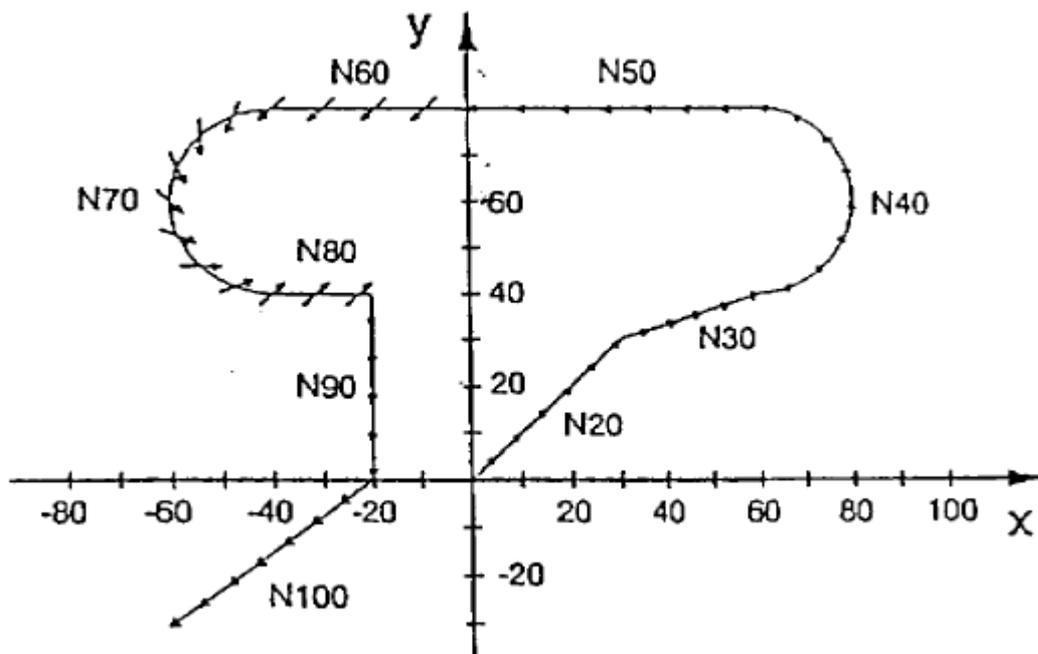
Nei blocchi G78 nessuna programmazione dell'asse rotatorio stesso è possibile. Soltanto lo sfalsamento d'angolo per l'asse rotatorio può essere specificato.

Con G78 modalmente effettivo, invece, l'asse rotatorio, insieme anche con altri assi, può essere programmato come al solito. L'imbocco viene allora disattivato nel blocco in cui l'asse rotatorio viene programmato con G78 modalmente effettivo.

Esempio di programma:

N10	G1	X0	Y0	C0	F3000	
N20	G78	C0				
N30	G1	X30	Y30			Angolo di allineamento 45°
N40	G1	X60	Y40			Angolo di allineamento circa 16,5°
N50	G3	Y80	J20			Imbocco tangenziale ad arco circolare
N60	G1	X0				Angolo di allineamento 180°
N70	G78	C45				
N80	G1	X-40	C45			Angolo di allineamento 225°
N90	G3	Y40	J-20			Angolo di allineamento 45° + angolo vettore tangenziale
N100	G1	X-20				Angolo di allineamento 45°
N110	G78	C0				
N120	G1	Y0				Angolo di allineamento 270°
N130	G1	X-60	Y-30	M30		Angolo di allineamento circa 217°

Per esempio, alla transizione di blocco N120/N130, l'asse rotatorio ruota da 270° a circa 217° utilizzando il percorso più corto, cioè, esso ruota circa 53° in direzione oraria.



### 17.18 G90,G91: Programmazione assoluta e programmazione incrementale

Con queste funzioni si predispone il controllo a considerare le quote programmate come quote incrementali. La programmazione incrementale stabilisce che tutte le coordinate di movimento degli assi ( X,Y e Z) e le coordinate del centro dei cerchi ( I,J e K) vengano riferite alla posizione del precedente punto programmato. In programmazione assoluta tutte le quote sono riferite allo zero pezzo. Nello stesso programma , secondo la convenienza, si possono alternare cicli di lavoro con programmazione assoluta e cicli con programmazione incrementale. La funzione G91 abilita la programmazione in incrementale mentre la funzione G90 attiva la programmazione assoluta.

Si ricorda che, all'accensione è attiva la funzione G90. Sia la funzione G90 che la G91 sono già attive nel blocco in cui vengono programmate

#### Sintassi:

G90... Programmazione di dimensione assoluta

G91... Programmazione di dimensione incrementale

Con le istruzioni G90 e G91 viene fatta una conversione tra la programmazione di dimensione assoluta (immissione di dimensione di riferimento, G90) e la programmazione di dimensione incrementale (immissione di dimensione incrementale, G91).

### 17.19 G92: Shift delle origini ( G52)

La funzione G92 effettua uno shift delle origini del valore indicato

#### Sintassi:

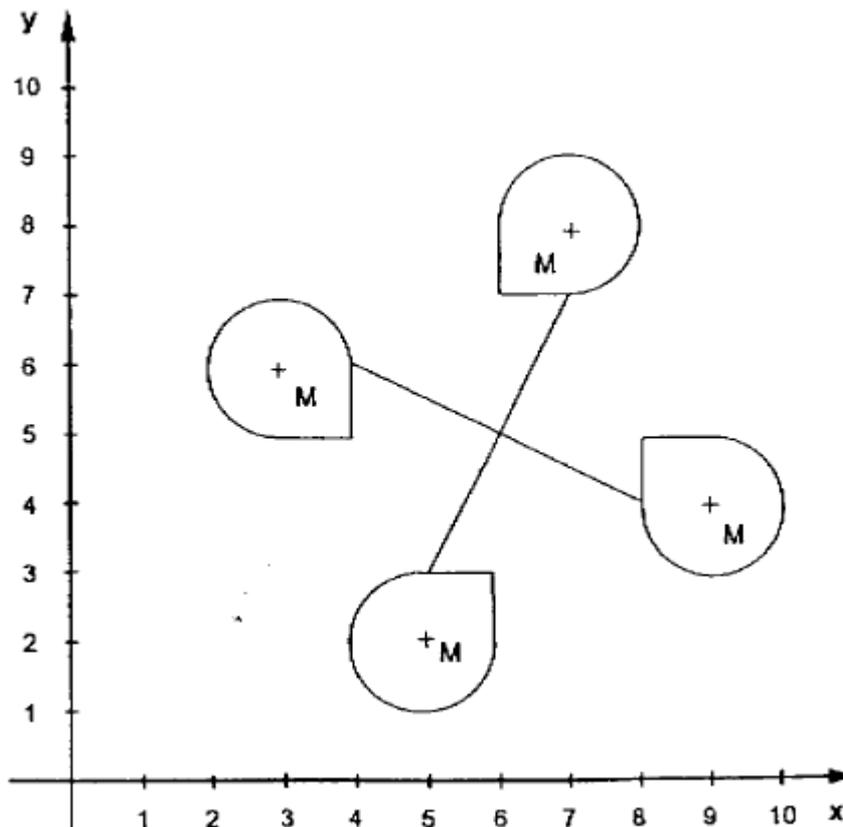
G92 X...Y... effettua uno shift delle origini del valore indicato

Esempio di programma G92, G51, G90, G91:

```
N10 G62 K2  
N20 X6 Y5 F1000  
N30 G92 X6 Y5 ;Setting of zero point  
N40 G90  
N50 LFIGURA K4
```

```
N60 G92  
N70 G90  
N80 X0Y0R  
N90 M30
```

```
L=FIGURA  
N10 G90  
N20 G1 X2 Y-1  
N30 G3 X3 Y0 I1  
N40 G1 X2  
N50 Y-1  
N60 X0 Y0  
N70 G91  
N80 G51 I90  
N90 G32
```



## 17.20 G201 Programmazione cilindrica

La funzione G201 permette la lavorazione di profili qualunque disposti su un cilindro, muovendo un asse rotativo ed un asse lineare coincidente con l'asse del cilindro.

Il profilo disposto sul manto del cilindro va linearizzato, cioè disposto su un piano cartesiano dove l'ascissa è rappresentata dall'asse virtuale (rotativo linearizzato) e l'ordinata dall'asse lineare.

Il profilo può essere programmato con le funzioni G1,G2,G3.

Può essere programmata la correzione del raggio utensile.

I piani di lavorazione possibili si programmano con la funzione G17 seguita dal nome di tre assi, che rappresentano rispettivamente l'ascissa e l'ordinata del piano cartesiano (l'asse virtuale e l'asse lineare interessati alla lavorazione) e l'asse su cui applicare la correzione lunghezza.

Esempio: G17 VYZ

Il formato di programmazione è:

G201 VBJ17.5  
V Asse virtuale  
B Asse rotativo  
Jxx Raggio del cilindro in millimetri

La programmazione cilindrica è chiusa dalla funzione G200 programmata da solo nel blocco.

La velocità di avanzamento viene programmata in millimetri al minuto con la funzione F.

Il sistema determina le velocità da assegnare ai due assi in lavoro, rotativo e lineare, in modo da mantenere costante sul profilo la velocità di avanzamento.

A fine programma ripristinare il piano di lavorazione normale con la funzione G17

## Esempio di programmazione cilindrica

G49 I0.2  
G41  
G17 VYZ

G201 VBJ17.6  
G0 V27.296 Y-112.289  
G1 V27.496 Y-112.285  
G1 V28.426 Y-112.265

..

..

..

G1 V104.962  
G40  
G17  
G200

## 17.21 G100-G1999: Funzioni programmabili

Il costruttore della macchina può predisporre alcune di queste funzioni attraverso l'abbinamento di un part program di automazione.

La tabella delle Funzioni per M e G speciali viene utilizzata per definire dei Funzioni di automazione da utilizzare durante la lavorazione, ed è collocata nel file dei Funzioni Me G speciali.

Ogni processo ISO lavora su una tabella propria di **Funzioni M e G**, tramite la sessione **[AbbinamentoFUNZIONI\_MG]** del file "sistema.txt" è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

### **[AbbinamentoFUNZIONI\_MG]**

Base\_FILE\_FUNZIONI\_MG =[percorso e nome file]

Testa1\_FILE\_FUNZIONI\_MG =[percorso e nome file]

...

Testa31\_FILE\_FUNZIONI\_MG =[percorso e nome file]

L'istruzione speciale : **>FILE\_FUNZIONI\_MOG= [path e nome file delle Funzioni M e G speciali ]** può essere usata per cambiare il file nel arco del part program in lavorazione sulla testa, vedi

“Istruzioni speciali per l'abbinamento risorse al processo ISO

Automaticamente vengono caricate le prime 7 Locali del part program chiamato nel seguente modo:

L1 = Numero asse Z rispetto al Piano di lavoro

L2 = Numero mandrino S

L3 = Numero AsseMandrino AS oppure 0

L4 = Numero della testa definita in "Base\_NumeroTesta" oppure 0

L5 = Speed mandrino ( valore di speed attivo blocco compreso )

L6 = 1 se M, 2 se G

L7 = Numero della G o M chiamante

## CAPITOLO 18

### 18 Cicli fissi di foratura

#### 18.1 Introduzione

La programmazione delle procedure di foratura può essere semplificata con i cicli di foratura. La selezione disponibile dei cicli di foratura copre i casi standard più importanti. Il programmatore deve solo definire qualche parametro, per adattare i cicli di foratura alla sua applicazione particolare.

I cicli di foratura descritti in seguito possono essere implementati come livelli di espansione. Per fare ciò, è necessario che la programmazione di ciclo del livello di ciclo 1 viene implementata anch'essa.

Il produttore della macchina utensile può cambiare la sequenza del programma nei cicli individuali di trapanatura, se richiesto. Se questo è stato fatto nel Vostro caso, siete pregati di riferirvi alla documentazione del produttore della macchina.

La chiamata e la messa a punto dei cicli di foratura sono modellate secondo DIN 66025.

#### 18.2 Uso dei cicli fissi

Un ciclo fisso di foratura in un programma CN parziale è sempre programmato nelle seguenti fasi:

Assegnare i parametri

Selezionare il ciclo fisso desiderato

Spostarsi alla posizione di lavoro in X e Y (una volta o ripetutamente)

Chiamare automaticamente e eseguire il ciclo fisso selezionato dopo aver raggiunto la posizione di lavoro

Deselezionare il ciclo fisso

Queste fasi sono spiegate individualmente nel testo seguente:

Nei cicli fissi elencati nel seguente testo, vengono usati i termini **piano di riferimento, piano di ritiro e profondità finale del foro**.

Il piano di riferimento si trova alla distanza di sicurezza sopra la superficie del pezzo in lavorazione; questo significa che sopra questo piano è possibile spostarsi verticalmente nella traversa rapida. Sotto questo piano, la traversa rapida è possibile solo nella direzione Z+, cioè allontanandosi dal pezzo in lavorazione. I movimenti dell'avanzamento incominciano dal piano di riferimento.

Il piano di ritiro è il piano al quale il mandrino si sposta alla fine del ciclo fisso. Il mandrino è quindi alla posizione di movimento libero.

La profondità finale del foro è ottenuta dal punto di misurazione dell'attrezzo. Questo è la punto di lavoro per una punta elicoidale, per esempio, oppure un punto arbitrario sulla superficie superiore per una macchina alesatrice.

### 18.3 Parametri dei cicli fissi

I cicli fissi prevedono una serie di parametri che sono comuni a buona parte di essi e che sono elencati sotto. Ulteriori dettagli su detti parametri si possono trovare nella descrizione dei singoli cicli.

I	incremento in profondità nel ciclo
J	quota di inizio foro
Q	piano di svincolo superiore ( se non programmato Q=J)
K	tempo di attesa in millesimi di secondo

La funzione G80 annulla il ciclo fisso selezionato.

## 18.4 Ciclo fisso G81 per foratura

La funzione G81 definisce il ciclo di foratura o di lamatura .

Il ciclo di foratura o lamatura si svolge nelle seguenti fasi:

1. Movimento rapido nel piano alla posizione X... Y...
2. Movimento rapido dell'asse Z al piano J
3. Lavoro con una velocità di avanzamento di F.. mm/min fino a raggiungere la quota Z...
4. Ritorno rapido, con mandrino in moto, al piano J.. o Q se specificato, che è il piano sul quale avverranno tutti gli spostamenti tra i cicli di foratura.

Esempio:

G81 J17.5 Z7 Q17.5 F100

; J quota rapido entrata

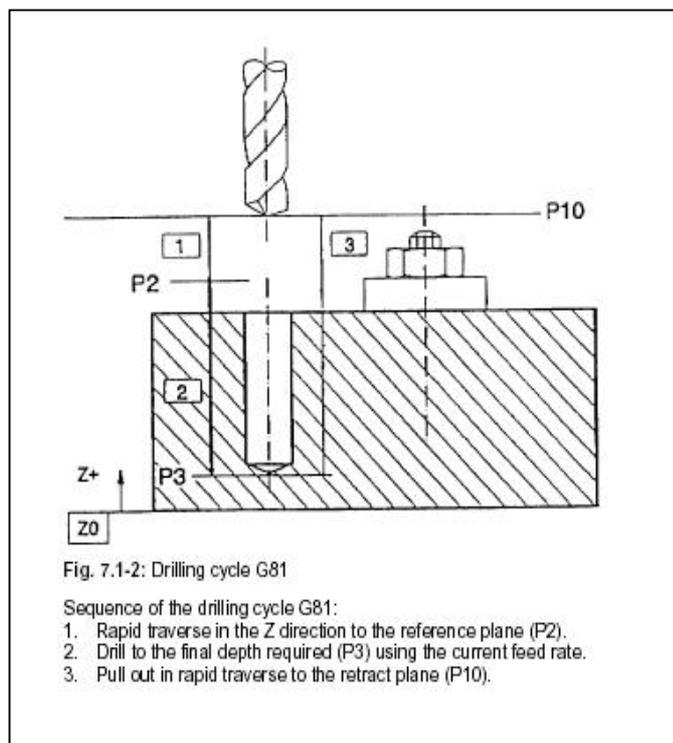
; Z quota lavoro finale

; Q quota rapido uscita

; F feed lavoro

### 1.1 G81 Ciclo fisso per foratura

J quota rapido entrata 1  
 Z quota lavoro finale 2  
 Q quota rapido uscita 3  
 F feed lavoro



## 18.5 Ciclo fisso G82 per foratura con sosta

La funzione G82 definisce il ciclo di foratura o di lamatura .

Il ciclo di foratura o lamatura si svolge nelle seguenti fasi:

- 1 Movimento rapido nel piano alla posizione X... Y...
- 2 Movimento rapido dell'asse Z al piano J
- 3 Lavoro con una velocità di avanzamento di F.. mm/min fino a raggiungere la quota Z...
- 4 Sosta per il tempo K in millesimi di secondo
- 5 Ritorno rapido, con mandrino in moto, al piano J.. o Q se specificato, che è il piano sul quale avverranno tutti gli spostamenti tra i cicli di foratura.

Esempio:

G82 J17.5 Z7 Q17.5 F100 K500

; J quota rapido entrata

; Z quota lavoro finale

; Q quota rapido uscita

; F feed lavoro

; K sosta in millisecondi al fondo della foratura

### 1.2 G82 Ciclo fisso per foratura con sosta

J	quota rapido entrata	1
Z	quota lavoro finale	2
Q	quota rapido uscita	4
F	feed lavoro	
K	ritardo	3

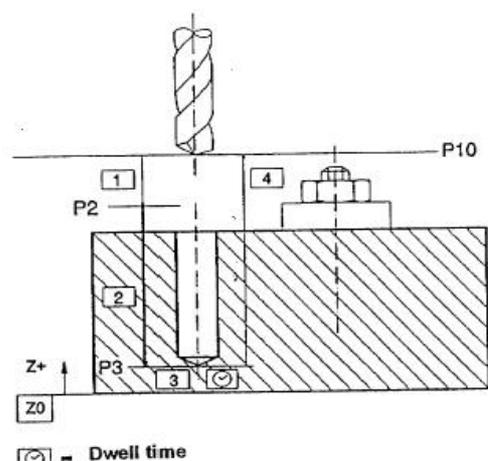


Fig. 7.1-3: Drilling cycle G82

Sequence of the drilling cycle G82:

1. Rapid traverse in the Z direction to the reference plane (P2).
2. Drill to the final hole depth (P3) using the current feed rate.
3. Wait for the dwell time (P4) to elapse before breaking contact with the workpiece.
4. Pull out in rapid traverse to the retract plane (P10).

## 18.6 Ciclo fisso G83 per foratura con rottura truciolo

La funzione G83 definisce il ciclo di foratura con ritorno dell'utensile fuori pezzo per lo scarico del truciolo .

Il ciclo di foratura o lamatura si svolge nelle seguenti fasi:

- 1 Movimento rapido nel piano alla posizione X... Y...
- 2 Movimento rapido dell'asse Z al piano J...
- 3 Definizione di n=0
- 4 Lavoro con una velocità di avanzamento di F.. per una profondità I...+ (I...-10%)\*n
- 5 Ritorno rapido al piano J...con mandrino in moto
- 6 Attesa per un tempo K in millesimi di secondo
- 7 Rapido a quota J...+ I...+ (I...-10%)\*n –incremento di sicurezza H...
- 8 Ripetizione al punto 4 fino al raggiungimento della quota Z....con n=n+1
- 9 Ritorno rapido, con mandrino in moto, al piano J.. o Q se specificato, che è il piano sul quale avverranno tutti gli spostamenti tra i cicli di foratura.

### 1.3 G83 Ciclo fisso per foratura con rottura truciolo

J	quota rapido entrata	1
Z	quota lavoro finale	10
Q	quota rapido uscita	11
F	feed lavoro	
K	ritardo	
I	incremento	
H	safety	

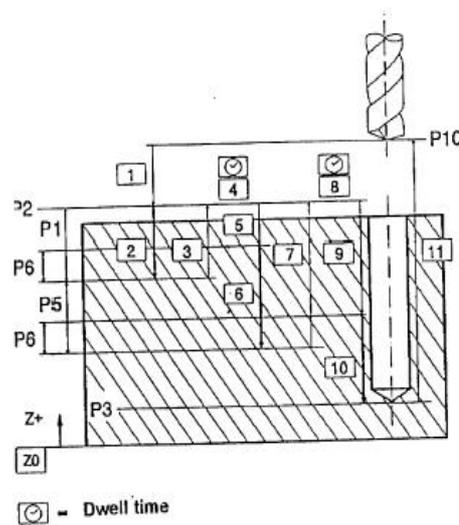


Fig. 7.1-4: Drilling cycle G83

Sequence of the drilling cycle G83:

1. Rapid traverse in the Z direction to the reference plane (P2).
2. Drill using the current feed rate with the first feed-in value (P1) to the depth 1.
3. Pull out in rapid traverse to the reference plane (P2).
4. To allow the drill bit to cool, the Z axis remains on the reference plane (P2) during the dwell time (P4).
5. Move in rapid traverse to P1-P6 (first feed-in minus safety clearance) in the hole.
6. Drill to the depth 2: P6+P5 (safety clearance plus feed-in) using the current feed rate.

## 18.7 Ciclo fisso G84 per maschiatura

La funzione G84 definisce il ciclo di maschiatura .

Il valore di K assume il seguente valore:

- 1 Stiramento del compensatore a fondo foro ( in millesimi di millimetro) per motori mandrino in corrente continua senza trasduttore. La F di avanzamento viene ridotta in percentuale.
- 2 Non deve essere programmato per motori mandrino in corrente continua con trasduttore ( maschiatura rigida).

Il ciclo di maschiatura si svolge nelle seguenti fasi:

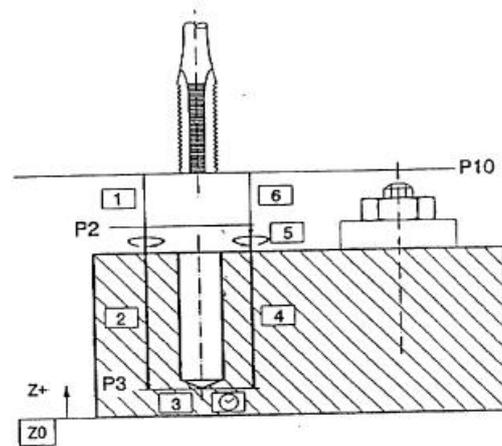
- 1 Movimento rapido nel piano alla posizione X... Y...
- 2 Movimento rapido dell'asse Z al piano J
- 3 Lavoro con una velocità di avanzamento di F... La velocità di avanzamento si specifica programmando F... in millesimi per giro mandrino(passo filetto).
- 4 Inversione rotazione mandrino, e quindi ritorno alla velocità di lavoro J...( o Q se specificato), che è il piano sul quale avverranno tutti gli spostamenti tra i cicli di maschiatura
- 5 Ripristino del senso di rotazione mandrino iniziale ( M03 per filettature destre, M04 per filettature sinistre)

Esempio ciclo maschiatura

```
G84 J-10 Z-35 Q0 F4.4 S500 K0  
; J quota entrata rapido  
; Z quota lavoro finale  
; Q quota rapido d'uscita  
; F passo  
; K=0  
; S=speed
```

## 1.4 G84 Ciclo fisso per maschiatura

J quota rapido entrata 1  
 Z quota lavoro finale 3  
 Q quota rapido uscita 6  
 F passo  
 K perc. stiramento o zero m. rigida



 - Dwell time

Fig. 7.1-5: Drilling cycle G84

Sequence of the drilling cycle G84:

1. Rapid traverse in the Z direction to the reference plane (P2).
2. Drill using the current feed rate and clockwise rotating spindle (M03) to the final hole depth (P3).
3. Reverse spindle, i.e. the direction of rotation changes; a pause is made for the dwell time (P4).
4. Pull out using the current feed rate to the reference plane (P2).
5. Reverse spindle, i.e. spindle's direction of rotation is again clockwise.
6. Move in rapid traverse to the retract plane (P10).

## 18.8 Ciclo fisso G85 per alesatura

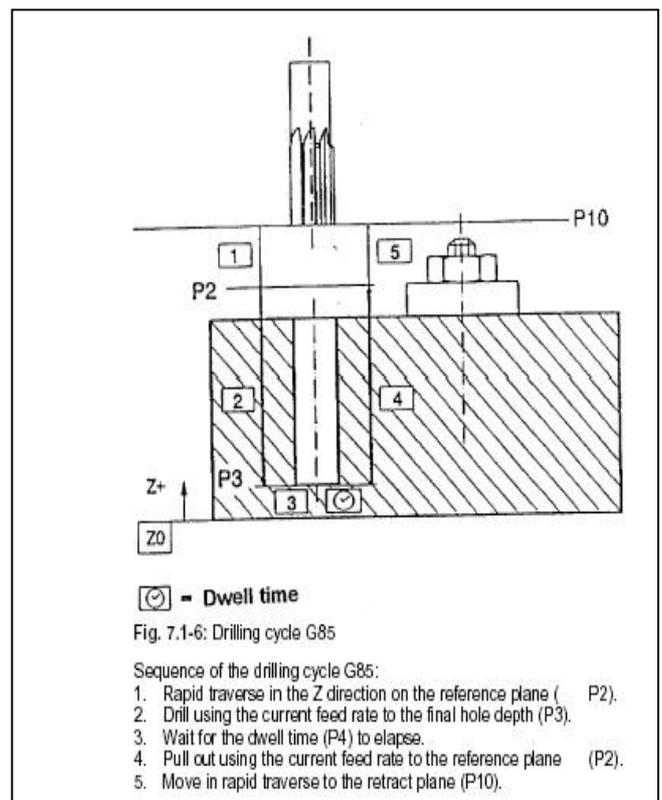
La funzione G85 definisce il ciclo di alesatura .

Il ciclo di alesatura si svolge nelle seguenti fasi:

- 1 Movimento rapido nel piano alla posizione X... Y...
- 2 Movimento rapido dell'asse Z al piano J
- 3 Lavoro con una velocità di avanzamento di F.. mm/min fino a raggiungere la quota Z...
- 4 Sosta per il tempo K in millesimi di secondo
- 5 Ritorno con una velocità di avanzamento di F.. mm/min, con mandrino in moto, al piano J
- 6 Ritorno rapido, con mandrino in moto, al piano J.. o Q se specificato, che è il piano sul quale avverranno tutti gli spostamenti tra i cicli di alesatura.

### 1.5 G85 Ciclo fisso per alesatura

J	quota rapido entrata	1
Z	quota lavoro finale	3
Q	quota rapido uscita	5
F	feed lavoro	2
K	ritardo	



## 18.9 Ciclo fisso G86 per barenatura

La funzione G86 definisce il ciclo di barenatura .

Il ciclo di barenatura si svolge nelle seguenti fasi:

- 1 Movimento rapido nel piano alla posizione X... Y...
- 2 Movimento rapido dell'asse Z al piano J
- 3 Lavoro con una velocità di avanzamento di F.. mm/min fino a raggiungere la quota Z...
- 4 Sosta per il tempo K in millesimi di secondo
- 5 Arresto mandrino alla quota 0 di M19 ( asse mandrino)
- 6 Sosta per il tempo I in millesimi di secondo
- 7 Movimento in incrementale di DX e DY
- 8 Ritorno rapido, con mandrino fermo, al piano J.. o Q se specificato, che è il piano sul quale avverranno tutti gli spostamenti tra i cicli di foratura.
- 9 Ritorno dell'
- 10 Ripristino rotazione mandrino

Esempio

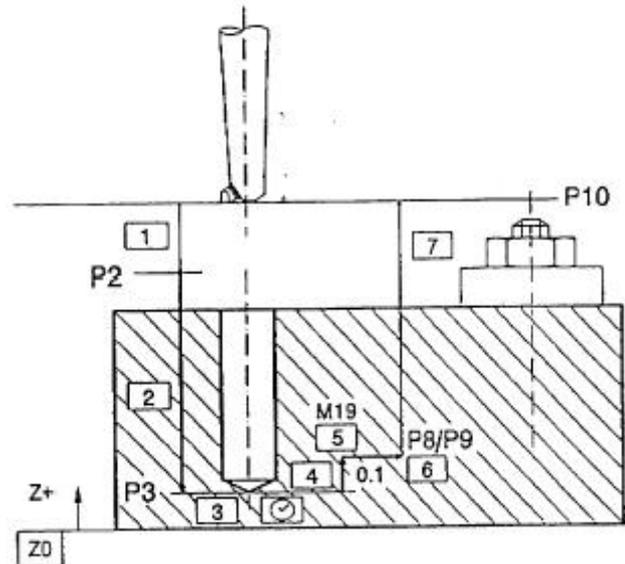
```
G86 J40 Z30 Q40 F100 H123 DY-0.5 K2000 I2000  
; J quota entrata rapido  
; Z quota lavoro finale  
; Q quota rapido d'uscita  
; F velocita foratura  
; H posizione in gradi come M19  
; K tempo in milli sec fine di Z prima di fermare il mandrino  
; I tempo in milli sec fine movimento DX DY  
; DX movimento delta in X  
; DY movimento delta in Y
```

```
N09 Y-30 ; blocco dove avviene prima barenatura  
N10 Y-29 ; blocco dove avviene seconda barenatura  
N11 Y-28 ; blocco dove avviene terza barenatura
```

```
N12 G80
```

## 1.6 G86 Ciclo fisso per barenatura

J quota rapido entrata  
 Z quota lavoro finale  
 Q quota rapido uscita  
 F feed lavoro  
 K ritardo  
 I ritardo  
 H rotazione in gradi mandrino  
 DX spostamento in increm. lungo X  
 DY spostamento in increm. lungo Y



 = Dwell time

Fig. 7.1-7: Drilling cycle G86

Sequence of the drilling cycle G86:

1. Rapid traverse in the Z direction to the reference plane (P2).
2. Bore to the final hole depth (P3) using the current feed rate.
3. Wait for the dwell time (P4) to elapse.
4. Move away 0.1 mm using the current feed rate.
5. Spindle is orientated to 0 degrees (M19).
6. Spindle is moved in the X or Y axis by the lift distance (P8 or P9).
7. Pull out to the retract plane (P10) in rapid traverse.

## 18.10 Ciclo fisso G89

La funzione Ciclo Fisso G89 se dichiarata nel file “sistema.txt” nella voce PartProgramG89 della sezione ParametriGenerali, attiva il part program di automazione.

### [ParametriGenerali]

**PartProgramG89= [ nome part program di automazione]**

Ad ogni movimento se seguono la Funzione G89, viene richiamato il part program di automazione definito nel file “sistema.txt”, con il passaggio di parametri:

- L1 = Numero asse Z rispetto al Piano di lavoro
- L2 = Numero mandrino S
- L3 = Numero AsseMandrino AS oppure 0
- L4 = Numero della testa definita in “Base\_NumeroTesta” oppure 0
- L5 = Speed mandrino ( valore di speed attivo blocco compreso )

- L6= PosZ\_Inizio
- L7 = PosZ\_Foratura
- L8 = PosZ\_Ritorno
- L9 = VelZ\_Foratura o Passo
- L10 = Tempo in Millisec oppure Percentuale stiramento
- L11 = Primo incremento da PosZ\_Inizio
- L12 = Incremento successivo (L11 – 10%di L11 in ISO)
- L13 = Incremento Sicurezza

I parametri trasferiti sono quelli indicati rispettivamente nei corrispettivi indicatori ISO:

- |   |                      |   |
|---|----------------------|---|
| J | quota rapido entrata | PosZ_Inizio   |
| Z | quota lavoro finale  | PosZ_Foratura   |
| Q | quota rapido uscita  | PosZ_Ritorno  |
| F | feed lavoro          | VelZ_Foratura o Passo   |
| K | ritardo              | Tempo in Millisec oppure perc. stiramento o zero m. rigida  |
| I | incremento           | Primo incremento da PosZ_Inizio ( gli altri incrementi sono dopo il primo pari a I , di I –10%sempre di I ) |
| H | safety               | Incremento Sicurezza  |

## CAPITOLO 19

### 19 Funzioni M

Sono le cosiddette funzioni varie o miscellanee. Sono programmabili con la lettera M seguita da un numero di cifre compreso tra 2 e 4 (da M00 a M9999).

M00	arresto programmato
M03	rotazione oraria del mandrino
M04	rotazione antioraria del mandrino
M05	arresto del mandrino
M06	cambio utensile
M07	attivazione refrigerante secondario
M08	attivazione refrigerante primario
M09	disattivazione refrigerante
M10	attivazione bloccaggio assi
M11	disattivazione bloccaggio assi
M13	rotazione oraria mandrino e attivazione refrigerante
M14	rotazione antioraria mandrino e attivazione refrigerante
M19	orientamento mandrino
M30	fine programma, azzera le funzioni ausiliarie attive
M31 - M1999	funzioni AxesBrain

## 19.1 Funzioni miscelanee ( M )

### 19.2 M00: Arresto programmato

Causa l'arresto degli assi e non del mandrino, l'esecuzione del programma riprende con la funzione di Start da Logica di macchina.

**Sintassi:**

M00

Se non sono state fatte altre disposizioni nel programma PLC, l'istruzione M00 abilita l'interruzione di un programma CN allo scopo di eseguire una misurazione, o simile.

Dopo la lavorazione di un blocco CN in cui l'istruzione M00 stata programmata il controllore interrompe l'esecuzione del programma. Tutti i valori modali vengono conservati.

### 19.3 M03,M04,M07,M08,M13,M14: Avvio mandrino e refrigerante

Causano l'avvio della rotazione del mandrino e/o l'erogazione del refrigerante:

M03	rotazione oraria del mandrino
M04	rotazione antioraria del mandrino
M07	attivazione refrigerante secondario

M08	attivazione refrigerante primario
M13	rotazione oraria mandrino e attivazione refrigerante
M14	rotazione antioraria mandrino e attivazione refrigerante

#### **19.4 M05,M09: Arresto mandrino e refrigerante**

Causano l'arresto della rotazione del mandrino e/o l'erogazione del refrigerante:

M05	arresto del mandrino
M09	disattivazione refrigerante

#### **19.5 M19: Posizione asse mandrino**

Causa il posizionamento del mandrino con motori in corrente continua con trasduttore di posizione alla posizione indicata

#### **19.6 M10,M11: Bloccaggio assi**

Comandano rispettivamente il bloccaggio e lo sbloccaggio degli assi per macchine utensili equipaggiate con apposito dispositivo.

M10	attivazione bloccaggio assi
M11	disattivazione bloccaggio assi

All'accensione del controllo o dopo un passaggio manuale/automatico gli assi sono bloccati ed è attiva la funzione M10. Impostando il modo di funzionamento automatico l'asse o gli assi da muovere vengono sbloccati appena il controllo legge il blocco contenente il movimento di tali assi. L'asse o gli assi che non devono eseguire movimenti rimangono bloccati. Anche con la funzione M10 ( bloccaggio assi) attiva vengono sbloccati solamente l'asse o gli assi che devono eseguire movimenti; la differenza è che, una volta arrivati sul punto richiesto, tali assi sono nuovamente bloccati. Nel modo di esecuzione .

## 19.7 M06: Cambio utensile

Consentono l'esecuzione di un cambio utensile e/o l'attivazione dei valori raggio e lunghezza relativi all'utensile programmato con la funzione T.

M06 arresto del programma ed attivazione dei valori di raggio e lunghezza

Esempio di chiamata con M6 Txxxx

/\*

L1 = Numero asse Z rispetto al Piano di lavoro

L2 = Numero mandrino S

L3 = Numero AsseMandrino AS oppure 0

L4 = Numero della testa definita in "Base\_NumeroTesta" oppure 0

L5 = Speed mandrino ( valore di speed attivo blocco compreso )

L6 = Tipo di funzione chiamante la logica di macchina

3=M06 4= T(Tool)

L7 = IDToolPrec

L8 = RT raggio utensile

L9 = LC correzione lunghezza

L10 = NM numero magazzino

L11 = PM posizione magazzino X

L12 = PD posizione intermedia Y

L13 = Spec no,si 0,1

L14 = Size no defined,small,medium,large,extra 0,1,2,3,4

L15 = MaxLife Vita massima minuti

L16 = MinLife Vita min. minuti

L17 = RealLife Vita residua minuti

L18 = CP Capostipite Numero Tool capostipite

L19 = SC Scaduto no,si 0,1

L20 = Free1

L21 = Free2

L22 = Free3

L23 = Free4

L26 = IDToolSuccessivo

L27 = RT raggio utensile

L28 = LC correzione lunghezza

L29 = NM numero magazzino

L30 = PM posizione magazzino X

L31 = PD posizione intermedia Y

L32 = Spec no,si 0,1

L33 = Size no defined,small,medium,large,extra 0,1,2,3,4

L34 = MaxLife Vita massima minuti  
 L35 = MinLife Vita min. minuti  
 L36 = RealLife Vita residua minuti  
 L37 = CP Capostipite Numero Tool capostipite  
 L38 = SC Scaduto no,si 0,1  
 L39 = Free1  
 L40 = Free2  
 L41 = Free3  
 L42 = Free4

Al ritorno le medesimi Locali sono ricaricate nei registri del Processo chiamante

\*/

-DIS/4:Cosa =,L6,3,0

-JEQ/L6,3,PresenteM3

-RET

PresenteM3-

-DIS/5:Numero Testa =,L4,3,0

-DIS/6:Numero ToolPrima =,L7,3,0

-DIS/7:Numero ToolDopo =,L26,3,0

-DIS/8:Numero Mandrino =,L2,3,0

-JNE/L26,0,Cambia

-RET

Cambia-

-JEQ/L4,11,Testa1

-JEQ/L4,41,Testa4

-JEQ/L4,61,Testa6

-RET

;

; Testa unita 1

;

;

Testa1-

-DIM/64

-SPD/S011,0

-RAV/9:G19211,X011

-RAV/9:G19011,Z011

-CAP/9:X011,0

-CAP/9:Z011,0

-MOV/X011,G16211

; -MOV/X011,-40

-MOV/Z011,G16011  
-CAP/9:X011,G19211  
-CAP/9:Z011,G19011

-LET/L51,1,L52,L26  
-CAL/,ToolRev1:L51,L52  
-JNE/L51,1,ErrTesta1  
-RET

ErrTesta1-

-KYB/ErroreTesta1  
-TKM/0:11:0:L1  
-SPD/S011,0  
-LET/G14011,and(G14011,not(4)) ; Unità No Running  
-RET

;

; Testa unita 4

;

Testa4-

-DIM/64  
-SPD/S041,0  
-RAV/9:G19241,X041  
-RAV/9:G19041,Z041  
-CAP/9:X041,0  
-CAP/9:Z041,0  
-MOV/Z041,G16041  
-MOV/X041,G16241  
-CAP/9:X041,G19241  
-CAP/9:Z041,G19041

-LET/L51,1,L52,L26  
-CAL/,ToolRev4:L51,L52  
-JNE/L51,1,ErrTesta4  
-RET

ErrTesta4-

-KYB/ErroreTesta4  
-TKM/0:41:0:L1  
-SPD/S041,0  
-LET/G14041,and(G14041,not(4)) ; Unità No Running  
-RET

;

; Testa unita 6

;

Testa6-

-DIM/64

-SPD/S061,0  
-RAV/9:G19061,Z061  
-RAV/9:G19261,X061  
-CAP/9:Z061,0  
-CAP/9:X061,0  
-MOV/Z061,G17061  
;  
-MOV/X061,G17261  
-MOV/X061,16  
-CAP/9:Z061,G19061  
-CAP/9:X061,G19261  
  
-LET/L51,1,L52,L26  
-CAL/,ToolRev6:L51,L52  
-JNE/L51,1,ErrTesta6  
-RET  
ErrTesta6-  
-KYB/ErroreTesta6  
-TKM/0:61:0:L1  
-SPD/S061,0  
-LET/G14061,and(G14061,not(4)) ; Unità No Running  
-RET

## 19.8 M30: Fine programma

Causa il fine programma, azzera le funzioni ausiliarie attive

## 19.9 M31-M1999: Funzioni programmabili

Il costruttore della macchina può predisporre alcune di queste funzioni oppure, attraverso l'abbinamento di un part program di automazione.

La tabella delle Funzioni per M e G speciali viene utilizzata per definire dei Funzioni di automazione da utilizzare durante la lavorazione, ed è collocata nel file dei Funzioni Me G speciali.

Ogni processo ISO lavora su una tabella propria di **Funzioni M e G**, tramite la sessione **[AbbinamentoFUNZIONI\_MG]** del file "sistema.txt" è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

### **[AbbinamentoFUNZIONI\_MG]**

Base\_FILE\_FUNZIONI\_MG =[percorso e nome file]

Testa1\_FILE\_FUNZIONI\_MG =[percorso e nome file]

...

Testa31\_FILE\_FUNZIONI\_MG =[percorso e nome file]

L'istruzione speciale : **>FILE\_FUNZIONI\_MOG= [path e nome file delle Funzioni M e G speciali ]** può essere usata per cambiare il file nel arco del part program in lavorazione sulla testa, vedi

“Istruzioni speciali per l'abbinamento risorse al processo ISO

Automaticamente vengono caricate le prime 7 Locali del part program chiamato nel seguente modo:

L1 = Numero asse Z rispetto al Piano di lavoro

L2 = Numero mandrino S

L3 = Numero AsseMandrino AS oppure 0

L4 = Numero della testa definita in "Base\_NumeroTesta" oppure 0

L5 = Speed mandrino ( valore di speed attivo blocco compreso )

L6 = 1 se M, 2 se G

L7 = Numero della G o M chiamante

[ParametriGeneraliMoG]

NumeroMoG=2

; Kind 1=M 2=G

[Fun1]

Kind=1

iD=56

Descrizione=

PartProgramGAL=M56.pp

NumeroLocali=4

SlotInput\_1=10,123+P1 ; L10=123+P1

SlotInput\_2=7,345 ; L7=345

SlotInput\_3=4,167 ; L4=167

SlotInput\_4=50,312 ; L23=312

NumeroParametri=2

SlotOutput\_1=21,123+L1 ; P21=123+L1

SlotOutput\_2=3,345+G1 ; P3=345+G1



## CAPITOLO 20

### 20 Funzioni T

La funzione T serve per il cambio utensile, manuale o automatico. Le cifre che seguono la funzione T definiscono il numero dell'utensile da richiamare.

Oltre all'utensile la funzione T richiama tutti i parametri memorizzati nella tabella dei dati utensili (correttori lunghezza,raggio,ecc.)

Il modo di programmare il cambio utensile (manuale, automatico sequenziale o no, con o senza braccio scambiatore) dipende da come è stato realizzato dall'utente e da come è stata La tabella degli utensili ( TOOL ) viene utilizzata per attivare o disattivare gli utensili durante la lavorazione, ed è collocata nel file dei Tool.

Ogni processo ISO lavora su una tabella propria di **Utensili (TOOL)**, tramite la sessione **[AbbinamentoTOOL]** del file "sistema.txt" è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

#### **[AbbinamentoTOOL]**

Base\_FILE\_TOOL =[percorso e nome file]

Testa1\_FILE\_TOOL =[percorso e nome file]

...

Testa31\_FILE\_TOOL =[percorso e nome file]

.L'istruzione speciale : **>FILE\_TOOL= [path e nome file dei TOOL ]** può essere usata per cambiare il file nel arco del part program in lavorazione sulla testa ,vedi

"Istruzioni speciali per l'abbinamento risorse al processo ISO"

## CAPITOLO 21

### 21 Funzioni F, S e O

La funzione F definisce la velocità di avanzamento durante la lavorazione ed è programmata con la lettera F seguita da 3 cifre che esprime la velocità stessa. La funzione F è modale

La funzione S definisce la velocità di rotazione del mandrino. Si programma con la lettera S seguita da sei cifre che esprimono la velocità espressa direttamente in giri/min.

La funzione O richiama le origini in fase di azzeramento, da O1 a O99.

La funzione O0 disabilita le origini e le coordinate sono riferite allo zero macchina. La funzione O-1 richiama, dopo la programmazione di O0, l'ultima origine programmata.

#### 21.1 F: Avanzamento

La velocità di avanzamento deve essere programmata tramite la funzione F e il valore deve essere espresso nell'unità di misura selezionato. Eventualmente variata tramite potenziometro "override fed". La funzione F è modale.

**Esempio:**

F1500 X45

Se è selezionato il sistema metrico, lo spostamento alla quota X 45 mm avviene alla velocità di 1500 mm/min.

#### 21.2 S: Velocità di rotazione mandrino

Specifica la velocità di rotazione per mandrini in corrente continua; il valore deve essere espresso in giri al minuto. La funzione S è modale.

### 21.3 O: Origini

La funzione O0 disabilita le origini e le coordinate sono riferite allo zero macchina. La funzione O-1 richiama, dopo la programmazione di O0, l'ultima origine programmata.

Specifica il numero della origine della tabella delle ORIGINI i cui valori vengono utilizzati per attivare o disattivare (O0) le origini pezzo durante la lavorazione, ed è collocata nel file delle origini.

Ogni processo ISO lavora su una tabella propria di **Origini**, tramite la sessione **[AbbinamentoOrigini]** del file "sistema.txt" è possibile specificare l'intero percorso ed il nome del file contenente la tabella.

#### **[AbbinamentoORIGINI]**

Base\_FILE\_ORIGINE=[percorso e nome file]

Testa1\_FILE\_ORIGINE=[percorso e nome file]

...

Testa31\_FILE\_ORIGINE=[percorso e nome file]

L'istruzione speciale : **>FILE\_ORIGINE= [path e nome file delle origini ]** può essere usata per cambiare il file nel arco del part program in lavorazione sulla testa, vedi

"Istruzioni speciali per l'abbinamento risorse al processo ISO"

## CAPITOLO 22

### 22 Parametri

Il sistema permette per ogni istanza di part program ISO 256 parametri definiti dalla lettera P.

I parametri P possono essere usati al posto delle dichiarazioni numeriche di posizione od altro come esempio :

```
P34=10  
P35=500  
XP34Y0  
G4 P35
```

Inoltre possono essere usato per controllare dei rami differenti di lavoro con la seguente sintassi

```
{Pm=Pn} Li  
{Pm>Pn} Li  
{Pm<Pn} Li  
{Pm<=Pn} Li  
{Pm>=Pn} Li  
{Pm<>Pn} Li
```

#### **Esempio:**

```
L=loop  
P3=P3+1  
{P3<10}loop
```

I Parametri possono essere assegnati con espressioni matematiche con i seguenti operatori.

**Le espressioni matematiche hanno i seguenti operatori :**

abs	Assoluto di un numero
acos	Arco coseno
and	And tra due numeri
asin	Arco seno
atan	Arco tangente
atanw	Arco tangente di Y,X
ceil	Arrotondamento verso l'alto di un numero decimale in un numero intero
cos	Coseno
cosh	Coseno iperbolico
deg	Trasformazione in gradi di un angolo espresso in radianti
exp	Esponenziale
floor	Arrotondamento verso il basso di un numero decimale in un numero intero
logd	Logaritmo decimale
logn	Logaritmo naturale
lshift	Shift verso sinistra di un numero
max	Massimo fra di enne espressioni
min	Minimo fra enne espressioni
mod	Modulo fra due numeri
not	Negazione booleana di un numero
or	Or fra due numeri
pi	P greco
rad	Trasformazione in radianti di un angolo espresso in gradi
rshift	Shift verso destra di un numero
sin	Seno
sinh	Seno iperbolico
sqr	Radice quadrata
tan	Tangente
tanh	Tangente iperbolica
xor	Or esclusivo fra due numeri
tim	restituisce il valore in secondi dal 1 gennaio 1970

Gli operatori in una espressione sono:

- + somma
- sottrazione
- / divisione
- \* moltiplicazione
- ^ elevato
- ( parentesi aperta

) parentesi chiusa

### **Esempio di espressione matematica**

P5=12

P6= 10

P1= MAX(SIN(RAD(P5+12),COS(RAD(P5+12\* P6/56)))

; nella parametro P1 viene caricato il risultato della espressione:

; MAX(SIN(RAD(P5+12),COS(RAD(P5+12\* P6/56)))





## CAPITOLO 23

### 23 Funzioni di controllo programma

#### 23.1 Ripetizioni di parte di programma

Utilizzando i codici L è possibile ripetere n volte un programma o parte di esso. Il numero massimo di ripetizioni è 32767.

La parte di programma che si vuole ripetere è racchiusa fra una definizione di riferimento “label” e l’istruzione di salto alla label seguita dal numero di ripetizioni.

Il numero di ripetizioni può essere un numero o un parametro.

##### **Esempio**

```
.....  
.....  
L=12          definizione della label  
.....  
.....  
L12 K8       salta alla label 12 per 8 volte
```

#### 23.2 Sottoprogrammi interni al programma

Si intende per sottoprogramma una sequenza di blocchi che possono essere richiamati da punti diversi del programma principale (ad esempio la successione dei vari punti su cui applicare i diversi cicli fissi, foratura, carenatura, alesatura, ecc.) o un profilo da richiamare più volte in punti diversi o con correttori raggio diversi.

Il sottoprogramma è richiamato programmando la funzione L seguita dal numero del sottoprogramma.

I sottoprogrammi interni al programma principale vanno programmati alla fine dello stesso, dopo la funzione M30.

Devono iniziare con la funzione L = numero sottoprogramma e terminare con la funzione di ritorno G32.

Un programma può chiamare un altro e così via fino ad massimo di 8 livelli di annodamento.

.

si può richiamare un sottoprogramma ( Funzione1 ) in un altro file ( Libreria )

L Libreria, Funzione1

### Esempio:

```
%  
N5M3S2000  
N15P1=-90  
G50  
O0  
G0ZP15  
G0X0  
G0Y0  
M11  
N25G0CP1  
N30M10  
N35O1  
N40LRUOTA  
N45G00X70.696Y-77  
N50G0Z5  
N55G1Z-2F99  
N60G01X-610Y-77F1000  
N65G0Z20  
N70P1=-90  
N75LRUOTA  
N80G00X70.696Y-2750  
N85G0Z5  
N90G1Z-2F99  
N95G01X-610Y-2750F1000  
N100G0Z20  
N105M30
```

; Routine

L=RUOTA

P3=P11-P13

P4=P12-P14

P5=P3\*cos(rad(P1))-P4\*sin(rad(P1))

P6=P3\*sin(rad(P1))+P4\*cos(rad(P1))

P7=P3-P5

P8=P4-P6

G51 XP7 YP8

G32

; P11 = Distanza X zero macchina centro tavola

; P12 = Distanza Y zero macchina centro tavola

; P13 = Quota origine X attuale

; P14 = Quota origine Y attuale

; P1 = Angolo di rotazione tavola ( incrementale )

L=PARK

O0

G0ZP15

G0X0

G0Y0

G32



## CAPITOLO 23

### 24 Altre Funzioni

#### 24.1 Introduzione commenti

All'interno del programma di lavorazione possono essere introdotte delle frasi di commento: tali frasi devono essere precedute dal carattere ; ( punto e virgola).

#### 24.2 N: Numero di sequenza

Specifica il numero di blocco. Non è richiesto che sia sequenziale e può eventualmente essere omessa. Convien comunque che sia sequenziale per facilitare l'arresto e la ripresa della lavorazione da un blocco prefissato. La funzione N non è modale.

#### 24.3 “ “: Stringa di visualizzazione

La stringa contenuta tra “ ” è visualizzata nella finestra apposita

Se termina con ‘:’ viene memorizzata e la prossima stringa accodata

Se il comando successivo una richiesta di introduzione valore diretto esempio P12=?

La scritta precedente terminata da ‘:’ viene caricata nel dialogo

#### 24.4 ? Richiesta valore diretto per parametro o globale

Se si trova a sinistra accoda il valore nella stringa esempio ?=P12, se si trova a destra richiede una introduzione di valore al P o alla G specificata, esempio G2=? Oppure P45=?.

I valori sono richiesti tramite un quadro dialogo su cui si può evidenziare una scritta tramite “ terminata con ‘:’, vedi stringa di visualizzazione esempio “ dammi valore:” P12=?

## Esempio per stringa e inserimento valore

L=Inizio

P12=1 P13=89 ; caricamento default

"dammi valore:" P12=?

"dammi valore:" P13=?

"valori inseriti :"

"P12=:" ?=P12 " P13=:" ?=P13

{P12>12}Oltre

F456

G0X0Y0R

X100Y0

X300Y578

X0Y0

L=Oltre

LInizio

M30

