

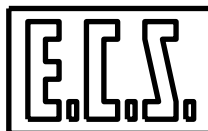
**MANUALE DI  
APPLICAZIONE  
CNC Serie WIN**

**Release SW 4.00**

**COD.: 720P397**

**EDIZIONE OTTOBRE 2006**

**REV. 6**







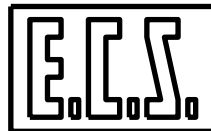
**MANUALE DI  
APPLICAZIONE  
CNC Serie WIN**

**Release SW 4.00**

**COD.: 720P397**

**EDIZIONE OTTOBRE 2006**

**REV. 6**



**E.C.S.** Electronic Control Systems - FIRENZE  
Via Garibaldi, 84/86 - 50041 CALENZANO - (FI) ITALIA

---

---



### CLAUSOLE GENERALI

Il presente manuale è stato realizzato con la massima cura ed attenzione, ma è possibile che qualche dato (o caratteristica tecnica) sia incompleto od errato; pertanto potranno imporsi dei cambiamenti di specifiche o di dati rispetto a quelle qui riportate senza possibilità di darne preavviso, ma che daranno luogo ad aggiornamenti del manuale.

In caso si evidenziassero da parte dell'Utente problemi od inesattezze si prega di contattare il seguente Ufficio:

PRODUCTS ENGINEERING – ECS

E-mail: [g.cocimano@ecsitaly.it](mailto:g.cocimano@ecsitaly.it)

Previa richiesta di PW, è possibile fare il download del presente manuale aggiornato, sul Sito ECS all'indirizzo: [www.ecsitaly.com/manuals.php](http://www.ecsitaly.com/manuals.php)

Il presente manuale non può essere riprodotto, tutto o in parte, con alcun mezzo senza il consenso scritto della ECS S.p.A.



### ELENCO DEI CAPITOLI AGGIORNATI

ARGOMENTO	EDIZIONE IN VIGORE REV. 6		N° DI PAGINA
	Situazione	Data	
Copertina	Rev. 6	Ottobre 2006	
Frontespizio	Rev. 6	Ottobre 2006	I
Elenco dei cap. aggiornati	Rev. 6	Ottobre 2006	III
Premessa	Rev. 4	Gennaio 2005	IV ÷ V
Tabella delle revisioni	Rev. 6	Ottobre 2006	VI ÷ X
Indice	Rev. 4	Gennaio 2005	XI ÷ XXII
Capitolo 1	Rev. 5	Settembre 2005	Pag. 1-1 ÷ 1-24
Capitolo 2	Rev. 4	Gennaio 2005	Pag. 2-1 ÷ 2-44
Capitolo 3	Rev. 6	Ottobre 2006	Pag. 3-1 ÷ 3-70
Capitolo 4	Rev. 3	Settembre 2004	Pag. 4-1 ÷ 4-16
Capitolo 5	Rev. 4	Gennaio 2005	Pag. 5-1 ÷ 5-30
Capitolo 6	Rev. 6	Ottobre 2006	Pag. 6-1 ÷ 6-40
Capitolo 7	Rev. 4	Gennaio 2005	Pag. 7-1 ÷ 7-16
Capitolo 8	Rev. 2	Maggio 2004	Pag. 8-1 ÷ 8-8
Capitolo 9	Rev. 6	Ottobre 2006	Pag. 9-1 ÷ 9-60
Indice Analitico	Rev. 4	Gennaio 2005	Pag. A ÷ J



---

---

## PREMESSA

### SCOPO

Questo manuale si propone di fornire tutte le informazioni necessarie ai progettisti che debbono applicare i Controlli Numerici (CNC) ECS della serie WIN, 1801 / 2801 / 4801 (per foratrici, fresatrici, alesatrici, machining center, genericamente indicati come **Frese**) e 1802 / 4802 (per **Torni**), sulle loro Macchine Utensili.

Sono qui raccolte, infatti, tutte le informazioni necessarie per la realizzazione della LOGICA DI MACCHINA realizzata, in linguaggio C, tramite il PLC integrato nel CNC.

### RIFERIMENTI

Il progettista che deve applicare un CNC ECS della serie WIN, potrà aver bisogno di consultare anche gli altri manuali che sono via via citati in questo documento:

* CONNESSIONI HARDWARE	720P406
* AMBIENTE e PROCEDURE di SVILUPPO e DEBUG PLC	720P394
* MANUALE TARATURE	720P385
* NORME D'USO PER FRESA	720P392
* NORME D'USO PER TORNIO	720P409
* NORME DI PROGRAMMAZIONE PER FRESA	720P395
* NORME DI PROGRAMMAZIONE PER TORNIO	720P389
* PROGRAMMAZIONE AVANZATA	720P396
* RESOURCE BUILDER	720P398

### NOTE PER LA CONSULTAZIONE

Come accennato sopra, questo manuale si riferisce ad ambedue i tipi (o categorie) di macchina in cui si può pensare di dividere il mondo della Macchina Utensile:

- \* foratrici, fresatrici, alesatrici, machining center
- \* torni orizzontali e verticali

La maggior parte delle informazioni sono valide sia per l'una che per l'altra categoria; quando sarà necessario fare delle distinzioni, verranno segnalate rispettivamente con:

{F}      Per FRESA

{T}      Per TORNIO

Quando larghe parti del testo sono relative ad una sola categoria, saranno segnalate anche da una barra sul margine sinistro come nell'esempio:

{F} I CNC ECS 1801 / 2801 / 4801 controllano gli **utensili** della Macchina Utensile (M.U.) attraverso uno specifico Software, sempre presente, denominato **TCI (Tool Controller Interno)** che permette di gestire fino a 150 utensili con un massimo di 255 "taglianti" (ovvero correttori).

{T} Sul tornio può essere presente anche un Mandrino Secondario con il funzionamento tipico dei mandrini di fresatura.



## NOTE PER LA SERIE WIN

I CNC della serie WIN derivano dai precedenti e ben noti CNC della serie D e pertanto le informazioni contenute nel presente manuale sono, per la maggior parte, uguali a quelle già presenti nel relativo MANUALE DI APPLICAZIONE 720P367.

Il progettista che ha già applicato i CNC ECS della serie xxxxD, potrà trovare le novità e differenze indicate con:

**{W}**      Per serie WIN

Quando larghe parti del testo sono relative a novità e differenze della serie WIN rispetto la serie D, saranno segnalate anche da una barra sul margine sinistro come nell'esempio:

**{W}** I CNC ECS 1401WIN / 2401WIN / 2701WIN controllano gli **utensili** della Macchina Utensile (M.U.) attraverso uno specifico Software, sempre presente, denominato **TCI (Tool Controller Interno)** che permette di gestire fino a 150 utensili con un massimo di 255 "taglienti" (ovvero correttori).

Dalla release di Software V3.0, i CNC per Fresa possono avere, in opzione, alcune funzioni tipiche dei Torni ed i CNC per Torni, viceversa, possono "commutare" in modo Fresa per eseguire le lavorazioni caratteristiche della fresatura.



<b>TABELLA DELLE REVISIONI</b>		
<b>Rev. N.</b>	<b>Descrizione</b>	<b>Data edizione</b>
<b>1</b>	<p>Par. 1.1.1 Sinonimi di stato del CNC: pon, MODE_...; OTWIST; IBLAXMND; OSYS...;</p> <p>Par. 1.2.5 Sinonimi utilizzati per le F.A.: OCOD...; NEWSAT ecc. per selezione Fresa/Tornio; NEWSMN ecc. per selezione Mandrino</p> <p>Par. 1.5.1 G84ENT</p> <p>Par. 1.6.3 Tutte le variabili float diventano di tipo double (doppia precisione)</p> <p>Par. 1.7 SELEZIONE APPARATO FRESA/TORNIO</p> <p>Par. 2.2.1 Nuovi sinonimi IDRNON#, IBLAXMND, IFEEHLD, IAXOFF, OAXOFF, ONBAGM#. Eliminato sinonimo: ONEWP</p> <p>Par. 2.5.1 Nuovo: Assi incrementali Sercos in Gantry</p> <p>Par. 2.5.2 Nuovo: Assi Assoluti</p> <p>Par. 2.7 Nuovo: Asse OFF</p> <p>Par. 2.9 3 volantini anche per Tornio; 4 per DRF</p> <p>Par. 2.9.2 Eliminati sinonimi IVJ, IVJL, IVJH; sostituiti da IVJ1, IVJ2</p> <p>Par. 2.9.3.1 Modifiche per quarto volante DRF</p> <p>Par. 2.9.5.1 Routine aggvol(); eliminati TIMPV1 e 2, Aggiunti APVOL4, SVOL4, RICVFED</p> <p>Par. 2.9.5.2 Sostituita ini_drf() con iniz_drf() per quarto volante DRF</p> <p>Par. 2.10 Override: aggiornato per nuove tastiere BASIC</p> <p>Par. 2.10.2.1 Aggiunta gestovr3() per tre overrides a encoder</p> <p>Par. 2.10.2.2 Modifiche a gestovr() per mandrino secondario</p> <p>Par. 2.10.2.3 Aggiunta gestovrc()</p> <p>Par. 2.11 Aggiunta gestione assi Sercos</p> <p>Par. 2.11.1 Tolti sinonimi: DGTURN. Aggiunti: DGVERS, DGSETP, IDGTAR, OCURR</p> <p>Par. 2.11.2.1 Aggiunto MECS11 per EDD</p> <p>Par. 3.2 Eliminate G600 e varorg()</p> <p>Par. 3.2.1 Unificata qwax() per Fresa e Tornio</p> <p>Par. 3.5.1 COPIA: aggiunte nuove definizioni di sinonimi</p> <p>Par. 3.8.2.4 Aggiunta AGC e tolta DGA</p> <p>Par. 3.8.3 Nuovo sinonimo OBGSET</p> <p>Par. 3.8.7 Nuovo: Assi Assoluti Gantry</p> <p>Par. 3.8.8 Nuovo: Homing assi Sercos incrementali in Gantry</p> <p>Par. 3.8.9 Nuovo: Homing Assi Gantry SERCOS con righe incrementali assolute</p> <p>Par. 3.9 Block retrace: aggiornato tutto il capitolo; aggiunta assirtrc().</p> <p>Par. 3.10 Nelle varie routines i parametri float diventano double</p> <p>Par. 3.11 Aggiunte funzioni per Mandrino Secondario</p> <p>Par. 3.13 Nuovo: modifica soglie assi</p> <p>Par. 3.14 Nuovo: modifica dei finecorsa assi</p> <p>Par. 3.15 Nuovo: eventi esterni e Probe ON/OFF</p> <p>Par. 3.16 Nuovo: Assi Virtuali</p> <p>Par. 4.5 Tolti i sinonimi I_VOL#, O_VOL#, VOLFRE#</p> <p>Par. 4.5.3 Nuovo: note su OCPLC</p>	<b>20/07/03</b>





**TABELLA DELLE REVISIONI**

<b>Rev. N.</b>	<b>Descrizione</b>	<b>Data edizione</b>
	<p>Par. 5.1 Aggiunto secondo e terzo mandrino, note su fresa/tornio e selezione mandrino</p> <p>Par. 5.1 Nuovi sinonimi; nota su IMNDATT</p> <p>Par. 5.2.2.2 Aggiunta routine SetMovMnd()</p> <p>Par. 5.2.2.3 Aggiunta routine TstMovMnd()</p> <p>Par. 5.2.3 Cambio scala mandrino: unificate le varie routines per fresa e tornio</p> <p>Par. 5.2.4 Cambio gamma: tolte le differenze tra fresa e tornio</p> <p>Par. 5.7 Eliminata gestione mandrino in modo slave</p> <p>Par. 5.10 Eliminata visualizzazione Assi remoti</p> <p>Par. 5.11 Aggiunta tabella funzioni per modifica Speed</p> <p>Par. 6.1 Aggiunti sinonimi TAUCNC, TAUPLC, IOPLCSTS, IOPLCRET.</p> <p>Par. 6.2 Eliminata funzione pwoff()</p> <p>Par. 6.3 Aggiunto sinonimo ACTALAT</p> <p>Par. 6.3.3 Nuovo: Gestione Storico Allarmi ed Eventi del CNC</p> <p>Par. 6.5.1 Modifiche alla funzione sel_pp()</p> <p>Par. 6.5.2 ...6 Nuove funzioni per il Part-Program</p> <p>Par. 6.7.3 Nuova funzione IOPLCReadPLCLanguage()</p> <p>Par. 6.7.4 Nuova funzione IOPLCShutDown()</p> <p>Par. 6.9.1 Nuovi sinonimi per Opzioni</p> <p>Par. 6.13 Eliminate funzioni per lettura/scrittura I/O del PC</p> <p>Par. 6.13 Aggiunta funzione IOPLCRunFileExe()</p> <p>Par. 6.15 Comunicazioni seriali da PLC: paragrafo eliminato (era vuoto)</p> <p>Par. 6.15 Nuove funzioni di scrittura/lettura files</p> <p>Par. 6.16 Nuove funzioni miscellanee</p> <p>Par. 7.1.1 Aggiunto sinonimo TASBASIC</p> <p>Par. 7.1.11 CODTAS diventa di tipo "word"</p> <p>Par. 7.4 Nuovo: Commutazione di due Tastiere</p> <p>Par. 7.5 Nuovo: TASTIERA PORTATILE</p> <p>Par. 8.5 Nuovo: Gestione formato attuale del CNC</p> <p>Par. 9.5.1 IOPLC_nread_tt() e IOPLCGet_nread_tt() sostituiscono nread_tt()</p> <p>Par. 9.5.1.1 Nuova struttura TT.BIN; nuovo esempio lettura tabella</p> <p>Par. 9.7 Nuovi sinonimi</p> <p>Par. 9.8.2.1 Nuovi comandi WRCUST, RDCUST</p> <p>Par. 9.13.1 Tolta nread_tt() obsoleta</p>	
<b>2</b>	<p>Par. 1.1.1 Sinonimi di stato del CNC: OG37, ORIPOK;</p> <p>Par. 1.7 Test apparato e opzioni da Part-Program</p> <p>Par. 1.8 Nuovo: Compilazione condizionale</p> <p>Par. 2.3.3 Uso di COSTUSER anziché COST</p> <p>Par. 2.4 Uso di COSTUSER anziché COST</p> <p>Par. 2.8 Uso di COSTUSER anziché COST</p> <p>Par. 2.9.5.1 Routine aggvol(); aggiunti CODITAS, CODIIKR, CODICR</p> <p>Par. 2.10 Override: tolto riferimento a ovrrel(), non più presente</p> <p>Par. 3.5.2 Uso di COSTUSER anziché COST</p> <p>Par. 3.7.2 Uso di COSTUSER anziché COST</p>	<b>15/06/04</b>

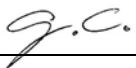
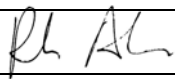


**TABELLA DELLE REVISIONI**

Rev. N.	Descrizione	Data edizione
	Par. 3.9.1.3 Block retrace: aggiunta bit 4 su stato di fasertrc(). Par. 3.17 Nuovo: Riposizionamento contemporaneo assi Par. 3.18 Nuovo: Funzione antislittamento per piegatubi Par. 3.19 Nuovo: Compensazione dinamica usura utensile Par. 4.5.1 Uso di COSTUSER anziché COST Par. 4.6.3.3 Aggiunto TypeMov 888, 951, 952, 953. 954 per Gantry Sercos Par. 5.2.4.2 Uso di COSTUSER anziché COST Par. 5.2.6.1 Uso di COSTUSER anziché COST Par. 6.5.1 Modifiche alla funzione sel_pp() Par. 6.5.2 GetPPSel: lunghezza nome file Par. 6.16.2 Nuovo: s_man() salva modo Manuale Par. 6.16.3 Nuovo: r_man() ripristina modo Manuale Par. 7 Aggiunto nota su MacroMenuDev Par. 8.3 Corretto: itoa() anziché sprintf() Par. 9.5.2 Aggiunti valori del campo <b>com</b> di TT.BIN Par. 9.8 Nuovi sinonimi TIPTM, TIPTRIC, TIPTMEM Par. 9.16 Nuovo: routine <b>movmag()</b>	
<b>3</b>	Par. 1.1.1.1 Feed massima in modo FAST (dry-run) Cap. 2 Il numero max di assi per CNC 4801/2 passa da 12 a 16 Il segnale IDRNON# agisce su tutti gli assi Digitali Aggiunta gestione assi Mechatrolink Par. 2.2.1 Aggiunto sinonimo AXESNUM: numero assi configurati Par. 2.3.4.1 Routine aggax(): temporizzatori per assi 12...15 Par. 2.4.1 Nuova: ATTENZIONE... Par. 2.11.2.1 Nota per allarmi su azionamenti con IDGOFF Cap. 3 Il numero max di assi per CNC 4801/2 passa da 12 a 16 Par. 3.6 Aggiunta nota per asse Step opzionale Par. 3.11 Aggiunte funzioni ValueFeedEx(), ValueDefModeCNC(), ValueModeCNC(), ValueModeExCNC(), DisableRqAxes(). Par. 3.11.1.13 Specificato che ValueFeed() legge la feed interpretata Par. 3.11.1.14 Nuovo: ValueFeedEx() per feed in esecuzione Par. 3.11.1.15 Nuovo: ValueDefModeCNC() per modo G94_97 di default Par. 3.11.1.16 Nuovo: ValueModeCNC() per modo G94_97 interpretato Par. 3.11.1.17 Nuovo: ValueModeCNCEx() per modo G94_97 in esecuz. Par. 3.11.1.18 Nuovo: DisableRqAxes() per disabilitare richieste da PLC Par. 3.11.2.4 Funzione SetSpeed(): nuovo esito=3 Par. 3.15.2 Cicli di misura: aggiunte note per assi Mechatrolink Par. 3.20 Nuovo: correzione tabellare errori assi Cap. 4 Il numero max di assi per CNC 4801/2 passa da 12 a 16 Cap. 5 Il numero max di assi per CNC 4801/2 passa da 12 a 16 Par. 5.10 Nuovo: M05 E ATTESA MANDRINO FERMO Par. 6.9.1 OP_AX#: gestione sino a 16 assi Par. 9.16.1.1 Il numero max di asse per ORD MAG passa da 11 a 15	<b>01/09/04</b>
<b>4</b>	Par. 1.1.1 Nuovo segnale <b>G669</b> Sellatura Flangie Par. 1.1.1.1 Aggiornata la descrizione del funzionamento CNC in modo PROVA PROGRAMMA [FAST] Par. 1.2.6.1 Nuova funzione set_G()	<b>31/01/05</b>



TABELLA DELLE REVISIONI

Rev. N.	Descrizione	Data edizione
	Par. 2.2.1 ILS#: nota per libera servo e IAXOFF non contemporanei OAXOFF: precisazione su asse OFF se OBAG/OBVV=0 Par. 2.5 Collegamento Micro di zero Par. 2.7 L'asse è in stato di OFF se <b>O BAG#=0 e OAXOFF#=1</b> Par. 2.9.5.1 Nuova selezione volantini su IKR-CAN Par. 2.11.1 Nota per valore di OCURR# Par. 3.8..... Note per Gantry Multislave Par. 3.9.1.1 plcrtcrf(): nuovo codice di esito -6 Par. 3.9.1.2 plcrtcrf(): aggiunto codice di esito 1 e -1 Par. 3.11 Corretto G94/95 (errato) in G94/96 Eliminata ValueFeedEX(); ValueFeed() dà il valore attuale Eliminata ValueModeExCNC(); ValueModeCNC(): val. att. ValueFeed() dà il valore attuale dell'interpolatore Par. 3.11.1.13 Par. 3.11.1.14 Eliminata ValueFeedEX() Par. 3.11.1.15 Corretto G94_97 (errato) in G94_96; Mode è unsigned Par. 3.11.1.16 ValueModeCNC() legge il valore attuale dell'interpolatore Par. 3.11.1.17 Eliminata ValueModeCNCEX() Par. 3.15.2 Contatto Touch-Probe collegato sugli ingressi <b>EXT2</b> Par. 3.20.1.3 Corretto get_CorrNumber (errato) con get_CorrFileCode Par. 3.21 Nuovo: Gestione RETRACT ASSI in EMERGENZA Par. 5.2.4.2 Aggiunta nota per valore di #294 Par. 5.10.1.3 Corretta nota 2 Par. 6.3.1 Modificato esempio con uso di ACTALAT Par. 6.3.2 Riscritto esempio con uso di ACTALAT Par. 6.3.4 Nuovo: IOPLCSendSMS: invia messaggio tramite SMS Par. 6.5.7 Nuovo: PARTPSTS: cambio o modifica part-program Par. 7.1.1 Eliminato AT_DISAB e nota relativa (non esiste più)	
<b>5</b>	Par. 1.6 Specificato il numero di Timers realmente utilizzabili	<b>15/09/05</b>
<b>6</b>	Par. 3.22 Aggiunto assi "inseguitori" Par. 6.6 Aggiunto descrizione seganli OKEYIM OKEYSN OKEYAS Par. 9.5.1 Lunghezza TT.BIN passa da 256 a 1024 record Par. 9.5.2 Cambiato unità di misura lxoﬀ roﬀ lzoﬀ da micro a mm Par. 9.7 Aggiunto precisazione segnali TM TIPTM TRIC TIPTRIC..... Par. 9.8.2.1 Esempio uso WRCUST RDCUST numero da 256 a 1024	<b>10/10/06</b>
Emesso dal PE		(firma) 
Approvato dal Progetto		(firma) 



---

---

**Note :**



## INDICE

<b>1</b>	<b>SEGNALI DI STATO E FUNZIONI AUSILIARIE .....</b>	<b>1-1</b>
1.1	<b>SEGNALI DI STATO DEL CNC .....</b>	<b>1-1</b>
1.1.1	Sinonimi di stato del CNC.....	1-1
1.1.1.1	Note varie sui segnali di stato del CNC.....	1-5
1.2	<b>FUNZIONI AUSILIARIE.....</b>	<b>1-8</b>
1.2.1	Parametri relativi alle F.A. ....	1-8
1.2.2	INDIRIZZI F.A. ESTERNE.....	1-8
1.2.3	CLASSI DI F.A. E ORDINE DI EMISSIONE .....	1-9
1.2.3.1	Classe anticipata .....	1-9
1.2.3.2	Classe intermedia.....	1-9
1.2.3.3	Classe posticipata .....	1-9
1.2.3.4	Regola di programmazione delle F.A. ....	1-9
1.2.3.5	Tabella ordine di emissione F.A. ....	1-10
1.2.4	Codici F.A. "M" già assegnati da E.C.S. ....	1-11
1.2.5	Sinonimi utilizzati per le F.A. ....	1-11
1.2.6	Codici del Parametro Generico .....	1-14
1.2.6.1	set_G: Emissione delle funzioni G .....	1-14
1.2.7	SINCRONIZZAZIONE DELLE F.A. ....	1-15
1.2.8	Riposizionamento delle F.A. ....	1-15
1.3	<b>F.A. SUBROUTINE .....</b>	<b>1-16</b>
1.4	<b>SECONDA E TERZA F.A. MISCELLANEA.....</b>	<b>1-16</b>
1.4.1	Assi abbinati alla 2 <sup>a</sup> e/o ALLA 3 <sup>a</sup> F.A. ....	1-16
1.5	<b>FUNZIONI SET .....</b>	<b>1-17</b>
1.5.1	G84 (Macro di Maschiatura) .....	1-18
1.5.2	G85 (Macro di alesatura con ritorno in lavoro) .....	1-18
1.5.3	G86 (Macro di alesatura con arresto del mandrino).....	1-19
1.5.4	G87 ({F} Macro di alesatura con ritorno a mandrino orientato).....	1-19
1.5.5	G184 (Macro di maschiatura rigida) .....	1-19
1.5.6	G201÷211, G614 (Cicli di misura con probe ON-OFF) .....	1-19
1.5.7	OVER, CERC, ROMBO (Subroutine per prove assi).....	1-19
1.5.8	CGAMMA (Subroutine per cambio gamma).....	1-20
1.5.9	M40 ÷ M45 (Subroutine per cambio gamma) .....	1-20
1.5.10	WOOD ({F} macchine per il legno) .....	1-20
1.6	<b>TEMPORIZZATORI.....</b>	<b>1-20</b>
1.6.1	Sinonimi per i temporizzatori .....	1-21
1.6.2	lotime: funzione per definire il tempo di un ritardo.....	1-21
1.7	<b>SELEZIONE APPARATO FRESA/TORNIO.....</b>	<b>1-22</b>
1.8	<b>COMPILAZIONE CONDIZIONALE.....</b>	<b>1-24</b>
<b>2</b>	<b>ASSI.....</b>	<b>2-1</b>
2.1	<b>GENERALITÀ.....</b>	<b>2-1</b>
2.1.1	Parametri di taratura degli assi.....	2-1



---

<b>2.2</b>	<b>ASSI INTERPOLANTI (CNC)</b> .....	<b>2-2</b>
2.2.1	Sinonimi utilizzati per gli ASSI .....	2-2
<b>2.3</b>	<b>AGGANCIAMENTO ASSI E GESTIONE ORIGINI</b> .....	<b>2-8</b>
2.3.1	Modo JOG .....	2-8
2.3.2	Modo SET .....	2-9
2.3.3	Modi MDI, AUTOMATICO, RIPOSIZIONAMENTO .....	2-9
2.3.4	Routines a disposizione del PLC .....	2-10
2.3.4.1	Routine "AGGAX" .....	2-10
2.3.4.2	Routine "MODIJOG" .....	2-12
<b>2.4</b>	<b>BLOCCAGGIO ASSI</b> .....	<b>2-13</b>
2.4.1	Bloccaggio/Sbloccaggio assi: funzionamento e temporizzazioni .....	2-14
2.4.1.1	Esempi .....	2-14
2.4.1.2	Sottoprogrammi .....	2-16
<b>2.5</b>	<b>SET ASSI</b> .....	<b>2-17</b>
2.5.1	ASSI INCREMENTALI SERCOS IN GANTRY {W} .....	2-18
2.5.2	ASSI ASSOLUTI {W} .....	2-18
<b>2.6</b>	<b>LIBERA SERVO</b> .....	<b>2-19</b>
2.6.1	Segnali ILS# e OLS# per LIBERA SERVO .....	2-19
2.6.2	Note generali .....	2-20
<b>2.7</b>	<b>ASSE OFF</b> .....	<b>2-21</b>
<b>2.8</b>	<b>FINECORSA SOFTWARE E HARDWARE</b> .....	<b>2-21</b>
<b>2.9</b>	<b>VOLANTINI</b> .....	<b>2-22</b>
2.9.1	Generalità .....	2-22
2.9.2	Sinonimi utilizzati per i VOLANTINI .....	2-22
2.9.3	Gestione Volantini .....	2-24
2.9.3.1	Inserimento Volantini .....	2-24
2.9.3.2	Disinserimento Volantini .....	2-26
2.9.4	Volantini gestiti in FEED .....	2-27
2.9.4.1	Gestione tastatore continuo come volantini Feed {F} .....	2-28
2.9.5	Routines a disposizione del PLC .....	2-29
2.9.5.1	Routine "AGGVOL" .....	2-29
2.9.5.2	Routine "INIZ_DRF" .....	2-32
<b>2.10</b>	<b>VERRIDE</b> .....	<b>2-33</b>
2.10.1	Sinonimi utilizzati per gli OVERRIDE .....	2-33
2.10.2	Routines a disposizione del PLC .....	2-34
2.10.2.1	Routine "GESTOVR3" .....	2-34
2.10.2.2	Routine "GESTOVR" .....	2-37
2.10.2.3	Routine "GESTOVR3" .....	2-39
<b>2.11</b>	<b>Gestione per assi con azionamenti digitali ECSLINK, SERCOS o Mechatrolink</b> .....	<b>2-40</b>
2.11.1	Sinonimi utilizzati per gli assi digitali .....	2-40
2.11.2	Routines a disposizione del PLC .....	2-43
2.11.2.1	Routine "IDDDRIVE" .....	2-43
<b>3</b>	<b>ASSI: FUNZIONI SPECIALI</b> .....	<b>3-1</b>
3.1	<b>COMPENSAZIONE TERMICA</b> .....	<b>3-1</b>

---



---

3.1.1	Sinonimi utilizzati per la compensazione da PLC.....	3-1
3.1.2	Funzioni per Attivare/Disattivare la compensazione Asse da PLC .....	3-2
3.1.2.1	start_corr_ext: Attiva la compensazione Asse da PLC .....	3-2
3.1.2.2	stop_corr_ext: Disattiva la compensazione Asse da PLC .....	3-3
3.1.2.3	is_corr_ext: Testa se è attiva la compensazione Asse da PLC .....	3-3
3.1.2.4	is_cmd_corr: Testa se c'è un comando comp. asse da PLC in corso .....	3-4
3.1.3	Esempio di compensazione termica assi da PLC .....	3-4
<b>3.2</b>	<b>FUNZIONI PER QUOTE ASSI .....</b>	<b>3-4</b>
3.2.1	qvax: Legge la quota visualizzata di un asse .....	3-4
3.2.2	get_qr_pintl: Legge la quota qr ed il parametro pintl .....	3-5
<b>3.3</b>	<b>RTCP (Rotational Tool Center Point) {F} .....</b>	<b>3-6</b>
3.3.1	tartwt: Legge/scrive i parametri RTCP.....	3-6
<b>3.4</b>	<b>ASSE PENDOLANTE {F} .....</b>	<b>3-7</b>
3.4.1	Programmazione di un asse pendolante .....	3-7
3.4.1.1	Comando per il pendolamento tra due quote.....	3-7
3.4.1.2	Comando per il pendolamento attorno alla quota attuale .....	3-8
3.4.1.3	Comando per disattivare il pendolamento.....	3-8
3.4.1.4	Note e precisazioni.....	3-8
<b>3.5</b>	<b>COPIA {F}.....</b>	<b>3-9</b>
3.5.1	Sinonimi utilizzati per la funzione COPIA .....	3-9
3.5.2	Tarature sul file COST .....	3-10
3.5.3	Routines a disposizione del PLC.....	3-11
3.5.3.1	Routine "COPIA" .....	3-11
<b>3.6</b>	<b>ASSE STEP (Opzionale).....</b>	<b>3-12</b>
<b>3.7</b>	<b>ASSE DUMMY {F}.....</b>	<b>3-12</b>
3.7.1	Caratteristiche dell'asse definito Dummy .....	3-12
3.7.2	Funzionamento asse Dummy .....	3-13
3.7.3	Parametri dell'asse Dummy.....	3-14
3.7.4	Esempio di configurazione asse Dummy .....	3-15
<b>3.8</b>	<b>ASSI GANTRY .....</b>	<b>3-17</b>
3.8.1	Funzione degli assi Gantry .....	3-17
3.8.2	Realizzazione di assi Gantry sui CNC ECS serie WIN .....	3-17
3.8.2.1	Aggancio assi in GANTRY .....	3-17
3.8.2.2	Sgancio assi in GANTRY .....	3-18
3.8.2.3	Compensa assi in GANTRY.....	3-18
3.8.2.4	Compensazione automatica assi in GANTRY .....	3-19
3.8.3	Sinonimi utilizzati per gli Assi Gantry.....	3-19
3.8.4	Funzionamento degli assi Gantry .....	3-20
3.8.4.1	Allarmi.....	3-21
3.8.4.2	Fine Corsa Software.....	3-21
3.8.4.3	Rapidi ed Accelerazioni.....	3-21
3.8.4.4	Bloccaggi.....	3-22
3.8.5	Assi in Gantry in ciclo di digitalizzazione .....	3-22
3.8.6	Presa di zero.....	3-22
3.8.7	Assi assoluti Gantry .....	3-25
3.8.8	Homing assi Sercos incrementali in Gantry .....	3-25
3.8.8.1	SET0 - Esegue il SET degli assi Sercos in GANTRY .....	3-26
3.8.9	Homing Assi Gantry SERCOS con righe incrementali assolute .....	3-27
3.8.10	Gestione riaggancio degli assi in Gantry .....	3-28

---



---

<b>3.9</b>	<b>BLOCK RETRACE {F}</b> .....	<b>3-29</b>
3.9.1	Funzioni a disposizione del PLC.....	3-29
3.9.1.1	Funzione " plctrtrc " .....	3-29
3.9.1.2	Funzione " plctrtrcf " .....	3-30
3.9.1.3	Funzione " fasertrc " .....	3-30
3.9.1.4	Funzione " assirtrc " .....	3-30
3.9.1.5	Funzione " cutrtrc " .....	3-31
3.9.2	Note di funzionamento .....	3-31
3.9.3	Esempio di gestione Block-retrace da PLC .....	3-32
<b>3.10</b>	<b>MODIFICA ACCELERAZIONE ASSI</b> .....	<b>3-33</b>
3.10.1	Gestione da Part-Program .....	3-33
3.10.1.1	Modifica Accelerazione.....	3-33
3.10.1.2	Ripristina Accelerazione di taratura.....	3-33
3.10.2	Gestione da PLC.....	3-34
3.10.2.1	AxAcc: Modifica Accelerazione Asse .....	3-34
3.10.2.2	SavAxAccTar: Salva Parametri di Taratura Accelerazione Asse.....	3-34
<b>3.11</b>	<b>MODIFICA DELLA FEED E SPEED</b> .....	<b>3-35</b>
3.11.1	Feed Assi .....	3-36
3.11.1.1	SetAxDefFeed: Imposta la velocità (Feed) di Default Asse Indipendente .....	3-36
3.11.1.2	SetDefFeed: Imposta la velocità (Feed) di Default Assi CNC .....	3-36
3.11.1.3	SetAxLimFeed: Limita la velocità massima Asse .....	3-36
3.11.1.4	SetAxTarFeed: Ripristina la velocità Asse di taratura .....	3-36
3.11.1.5	SetLimFeedOn: Limita la velocità di percorso della punta utensile.....	3-37
3.11.1.6	SetLimFeedOff: Annulla il limite di velocità di percorso della punta utensile .....	3-37
3.11.1.7	SetFeed: Setta la velocità (Feed) degli assi CNC (Contouring).....	3-37
3.11.1.8	ValueAxDefFeed: Legge il valore della Feed di Default Asse Indipendente.....	3-37
3.11.1.9	ValueDefFeed: Legge il valore della Feed di Default Assi CNC .....	3-38
3.11.1.10	ValueAxLimFeed: Legge il valore limite attuale per la velocità Asse.....	3-38
3.11.1.11	ValueAxTarFeed: Legge il valore di taratura per la velocità Asse .....	3-38
3.11.1.12	ValueLimFeed: Legge il valore limite attuale per la velocità di percorso .....	3-38
3.11.1.13	ValueFeed: Legge il valore della Feed degli assi CNC.....	3-39
3.11.1.14	ValueFeedEx: Legge il valore della Feed in esecuzione degli assi CNC .....	3-39
3.11.1.15	ValueDefModeCNC: Ritorna la modalità G94_96 di Default del CNC.....	3-39
3.11.1.16	ValueModeCNC: Ritorna la modalità G94_96 del CNC.....	3-39
3.11.1.17	ValueModeCNCEX: Ritorna la modalità G94_96 eseguita del CNC.....	3-39
3.11.1.18	DisableRqAxes: Disabilita alcune richieste del PLC .....	3-40
3.11.2	Speed Mandrino.....	3-40
3.11.2.1	SetDefSpeed: Imposta la velocità di rotazione (Speed) di Default Mandrino .....	3-40
3.11.2.2	SetLimSpeed: Limita la velocità massima Mandrino .....	3-40
3.11.2.3	SetTarSpeed: Ripristina la velocità di taratura del Mandrino .....	3-40
3.11.2.4	SetSpeed: Setta il valore di velocità di rotazione (Speed) del Mandrino .....	3-41
3.11.2.5	ValueDefSpeed: Legge il valore della Speed di default .....	3-41
3.11.2.6	ValueLimSpeed: Legge il valore limite attuale per la Speed .....	3-41
3.11.2.7	ValueTarSpeed: Legge il valore di taratura per la Speed .....	3-42
3.11.2.8	ValueSpeed: Legge il valore di velocità di rotazione (Speed) del Mandrino .....	3-42
3.11.3	Note sulla gestione della modifica Feed Assi e Speed Mandrino.....	3-42
3.11.4	Esempio di uso delle funzioni di limitazione nel PLC .....	3-43
<b>3.12</b>	<b>SINCRONIZZAZIONE ASSI</b> .....	<b>3-44</b>
3.12.1	Predisposizioni.....	3-44
3.12.2	Sinonimi utilizzati per la funzione SINCRONIZZAZIONE ASSI .....	3-45
3.12.3	Funzioni a disposizione del PLC.....	3-45
3.12.3.1	AxsFnzAxmOn: attiva il sincronismo dell'asse Slave .....	3-45
3.12.3.2	AxsFnzAxmOff: disattiva il sincronismo dell'asse Slave.....	3-46

---





---

3.12.3.3	AxSynch: comanda un movimento sincronizzato dell'asse Slave.....	3-46
3.12.4	Note di funzionamento.....	3-47
3.12.5	Esempio di sincronizzazione .....	3-47
<b>3.13</b>	<b>MODIFICA SOGLIE DI POSIZIONAMENTO E DI FUGA DEGLI ASSI .....</b>	<b>3-49</b>
3.13.1	Gestione da Part-Program.....	3-49
3.13.1.1	Modifica valori ATTUALI delle soglie di POSIZIONAMENTO e di FUGA.....	3-49
3.13.1.2	Ripristina le soglie di POSIZIONAMENTO e di FUGA.....	3-49
3.13.2	Gestione da PLC .....	3-50
3.13.2.1	ValueAxTarThresold: Legge i valori Attuali di Taratura delle soglie di Posizionamento e di Fuga.....	3-50
3.13.2.2	SetAxThresold: Modifica i valori di Taratura ed Attuali delle soglie di Posizionamento e di Fuga .....	3-50
<b>3.14</b>	<b>MODIFICA DEI FINE CORSA ASSI.....</b>	<b>3-51</b>
3.14.1	Funzioni a disposizione del PLC .....	3-51
3.14.1.1	ValueAxTarTravelLimit: Legge il valore dei FINE CORSA di Taratura .....	3-51
3.14.1.2	ValueAxTravelLimit: Legge il valore dei FINE CORSA (Travel Limit) attuali .....	3-52
3.14.1.3	ExcludeAxTravelLimit: Esclude i FINE CORSA (Travel Limit) .....	3-52
3.14.1.4	SetAxTravelLimitOn: Imposta i FINE CORSA (Travel Limit) da PLC .....	3-52
3.14.1.5	SetAxTravelLimitOff: Ripristina i FINE CORSA (Travel Limit) di TARATURA.....	3-53
<b>3.15</b>	<b>EVENTI ESTERNI E PROBE ON/OFF.....</b>	<b>3-53</b>
3.15.1	Sinonimi utilizzati dal PLC per eventi esterni .....	3-53
3.15.2	Cicli di misura .....	3-54
3.15.2.1	Presenza di assi Sercos, connessi ad azionamenti Indramat.....	3-54
3.15.2.2	Presenza di schede assi analogiche Touch-Probe (senza assi SERCOS).....	3-55
3.15.2.3	Assenza di schede assi analogiche Touch-Probe e di assi Sercos o Mechatrolink.....	3-56
3.15.2.4	Note di applicazione .....	3-56
<b>3.16</b>	<b>ASSI VIRTUALI {F} .....</b>	<b>3-57</b>
3.16.1	Funzioni per Attivare/Disattivare gli Assi Virtuali da PLC .....	3-57
<b>3.17</b>	<b>RIPOSIZIONAMENTO CONTEMPORANEO DI TUTTI GLI ASSI.....</b>	<b>3-58</b>
3.17.1	Funzioni per il riposizionamento contemporaneo di tutti gli assi .....	3-58
3.17.1.1	SetAllAxs: Attiva/disattiva il riposizionamento contemporaneo.....	3-58
3.17.1.2	FeedAllAxs: Definizione della Feed per il riposizionamento contemporaneo .....	3-58
<b>3.18</b>	<b>FUNZIONE ANTISLITTAMENTO PER PIEGATUBI {F}.....</b>	<b>3-59</b>
3.18.1	File AXS.TAR per "Antislittamento" .....	3-59
3.18.2	init_T_Axis: inizializza i parametri per asse Piegatubi.....	3-61
3.18.3	get_T_Axis_Override: legge l'override dell'asse T .....	3-62
<b>3.19</b>	<b>COMPENSAZIONE DINAMICA DELL'USURA UTENSILE .....</b>	<b>3-62</b>
3.19.1	Funzioni per la compensazione dinamica da PLC .....	3-62
3.19.1.1	plcrusr: legge/scrive il coefficiente di usura radiale da PLC.....	3-62
3.19.1.2	plclusr: legge/scrive il coefficiente di usura longitudinale da PLC.....	3-63
3.19.1.3	scarusr: legge/scrive il valore aggiuntivo di usura radiale da PLC.....	3-63
3.19.1.4	scalusr: legge/scrive il valore aggiuntivo di usura longitudinale da PLC.....	3-63
<b>3.20</b>	<b>CORREZIONE TABELLARE ERRORE ASSI.....</b>	<b>3-64</b>
3.20.1	Funzioni per gestire le compensazioni tabellare Assi da PLC .....	3-64
3.20.1.1	open_CorrFile: Selezione il set di compensazioni Assi.....	3-64
3.20.1.2	get_CorrFileNumber: Legge il set di compensazioni Assi attivo.....	3-64
3.20.1.3	get_CorrFileCode: Verifica l'esito del comando di selezione .....	3-65

---



<b>3.21</b>	<b>Gestione RETRACT ASSI in EMERGENZA .....</b>	<b>3-65</b>
3.21.1	Funzionamento .....	3-65
3.21.2	Sinonimi utilizzati per il Retract assi in Emergenza .....	3-66
3.21.3	Note Importanti .....	3-67
3.21.4	Norme di Programmazione .....	3-67
3.21.4.1	Esempio di Part-Program con gestione standard ECS prevista dal PLC .....	3-68
<b>3.22</b>	<b>ASSI "INSEGUITORI" .....</b>	<b>3-68</b>
3.22.1	Norme di programmazione .....	3-69
<b>4</b>	<b>ASSI INDIPENDENTI (FRE).....</b>	<b>4-1</b>
<b>4.1</b>	<b>Asse dichiarato INDIPENDENTE in AXS.TAR .....</b>	<b>4-1</b>
<b>4.2</b>	<b>Asse dichiarato indipendente da PART PROGRAM (Istruzione &lt;FRE:;&gt;).....</b>	<b>4-1</b>
<b>4.3</b>	<b>Gestione assi Indipendenti da Part Program .....</b>	<b>4-2</b>
<b>4.4</b>	<b>Gestione assi Indipendenti da PLC.....</b>	<b>4-2</b>
<b>4.5</b>	<b>Sinonimi utilizzati per gli assi indipendenti .....</b>	<b>4-3</b>
4.5.1	Note su IAXPLC#.....	4-5
4.5.2	Note su IJOGP# e IJOGN#.....	4-6
4.5.3	Note su OCPLC .....	4-6
<b>4.6</b>	<b>ROUTINES E FUNZIONI DEL PLC.....</b>	<b>4-7</b>
4.6.1	Routine "HLDFRE".....	4-7
4.6.2	Funzioni di movimentazione Assoluta/Incrementale di un Asse PLC .....	4-8
4.6.2.1	AxMovAbsRap: movimentazione di asse su quote assolute in rapido.....	4-8
4.6.2.2	AxMovAbsWork: movimentazione di asse su quote assolute in lavoro .....	4-8
4.6.2.3	AxMovIncRap: movimentazione di asse su quote incrementali in rapido .....	4-8
4.6.2.4	AxMovIncWork: movimentazioni di asse su quote Incrementali in lavoro.....	4-8
4.6.2.5	AxMovRoundInc: movimentazione di asse tondo su pos. angolare assoluta in lavoro	4-9
4.6.2.6	AxMovRoundDec: movimentazione di asse tondo su pos. angolare assoluta in lavoro	4-9
4.6.2.6.1	AxMovRoundNoZero: movimentazione di asse tondo su pos. angolare assoluta in lavoro	4-9
4.6.2.7	AxMovRound: movimentazione "perpetua" di asse tondo.....	4-9
4.6.2.8	Parametri in ingresso ed in uscita .....	4-9
4.6.3	Funzioni di movimentazione generica G800-G900 .....	4-10
4.6.3.1	AxMov: movimentazione G800-G900 su quote assolute .....	4-10
4.6.3.2	AxMovInc: movimentazione G800-G900 su quote incrementali .....	4-11
4.6.3.3	Parametri in ingresso ed in uscita .....	4-11
4.6.4	Funzioni per la movimentazione di un Asse PLC in JOG.....	4-13
4.6.4.1	AxJogPVelRap: movimento in JOG in senso positivo alla velocità di rapido .....	4-13
4.6.4.2	AxJogNVelRap: movimento in JOG in senso negativo alla velocità di rapido .....	4-13
4.6.4.3	AxJogPVelWork: movimento in JOG in senso positivo alla velocità di lavoro .....	4-13
4.6.4.4	AxJogNVelWork: movimento in JOG in senso negativo alla velocità di lavoro.....	4-13
4.6.4.5	AxJogSpRap: movimento in JOG di spazio alla velocità di rapido.....	4-13
4.6.4.6	AxJogSpWork: movimento in JOG di spazio alla velocità di lavoro. ....	4-13
4.6.4.7	Descrizione parametri in ingresso ed in uscita.....	4-13
4.6.5	AxFeeD: modifica della Feed di un Asse PLC in movimento.....	4-14
4.6.6	Funzioni di predisposizione valide per Assi PLC.....	4-14
4.6.6.1	AxPlcModeSetOn: predisposizione a mantenere gestito da PLC un asse in caso di SET da fare o perso.....	4-15



---

4.6.6.2	AxPlcModeSetOff: predisposizione per far ritornare gestito da CNC un asse in caso di SET da fare o perso.....	4-15
4.6.6.3	IsAxPlcModeSet: fornisce informazioni sulla predisposizione attiva.....	4-15
4.6.7	AxPlcAn: annullamento di movimenti e comandi in corso su un asse PLC .....	4-15

## **5 MANDRINO.....5-1**

### **5.1 GENERALITÀ..... 5-1**

5.1.1	Parametri di taratura del mandrino .....	5-3
5.1.2	Sinonimi utilizzati per il mandrino .....	5-4

### **5.2 MANDRINO ASSE DEL CNC ..... 5-8**

5.2.1	Aggancio mandrino nelle varie modalità CNC.....	5-8
5.2.2	ROUTINES A DISPOSIZIONE DEL PLC .....	5-9
5.2.2.1	Routine "AGGMND" .....	5-9
5.2.2.2	Routine "SetMovMnd" .....	5-11
5.2.2.3	Routine "TstMovMnd" .....	5-11
5.2.3	Gestione cambio scala mandrino .....	5-12
5.2.3.1	on_refmnd : attiva la funzione di cambio scala riferimento .....	5-12
5.2.3.2	off_refmnd : disattiva la funzione di cambio scala riferimento .....	5-13
5.2.3.3	tst_refmnd : testa la velocità mandrino attuale.....	5-13
5.2.4	Gestione gamme mandrino .....	5-14
5.2.4.1	Mandrino Monogamma .....	5-15
5.2.4.2	Tarature contenute nel file COST.....	5-15
5.2.4.3	Sottoprogramma SPEED .....	5-16
5.2.4.4	Sottoprogramma CGAMMA .....	5-16
5.2.5	Esempio di gestione di un cambio gamma.....	5-17
5.2.5.1	Tarature in AXS.TAR.....	5-17
5.2.5.2	Tarature in COSTUSER .....	5-17
5.2.5.3	Equazioni nel PLC .....	5-17
5.2.6	Mandrino in velocità.....	5-17
5.2.6.1	Rotazione .....	5-18
5.2.7	SET automatico di asse-mandrino con trasduttore. ....	5-18
5.2.8	Asse-mandrino senza trasduttore.....	5-19
5.2.9	Abilitazione azionamento mandrino.....	5-19
5.2.10	Bloccaggio asse mandrino .....	5-21
5.2.11	Mandrino pilotato in spazio.....	5-21
5.2.11.1	Caratteristiche necessarie alla M.U.....	5-21
5.2.11.2	Orientamento mandrino tramite M19.....	5-22
5.2.11.3	Orientamento mandrino libero.....	5-22
5.2.11.4	Velocità di orientamento.....	5-22
5.2.12	Mandrino come asse tondo .....	5-23

### **5.3 MANDRINO GESTITO DAL PLC..... 5-23**

### **5.4 FILETTATURA ..... 5-24**

5.4.1	KEY durante la filettatura (G33, G34, G35).....	5-24
-------	---	------

### **5.5 STOP MANDRINO ..... 5-24**

### **5.6 ALLARMI MANDRINO ..... 5-25**

### **5.7 AGGANCIAMENTO MANDRINO FRV / CNC ..... 5-25**

5.7.1	Gestione tramite Funzioni da PLC.....	5-26
5.7.1.1	AggMndFrv: comando di aggancio Free/velocità.....	5-26
5.7.1.2	AggMndCnc: comando di aggancio CNC .....	5-26



5.7.1.3	TstFrvCnc: test comando di aggancio .....	5-27
5.7.1.4	TstStsMnd: test stato di aggancio .....	5-27
<b>5.8</b>	<b>MANDRINO ABBINATO A MATRICI DINAMICHE .....</b>	<b>5-27</b>
5.8.1	Funzionamento G345 .....	5-28
<b>5.9</b>	<b>MODIFICA DELLA SPEED .....</b>	<b>5-28</b>
<b>5.10</b>	<b>M05 E ATTESA MANDRINO FERMO .....</b>	<b>5-29</b>
5.10.1	Gestione M05 tramite Funzioni da PLC.....	5-29
5.10.1.1	OnMfrM05: Predisporre emissione M05 con attesa mandrino fermo.....	5-29
5.10.1.2	OffMfrM05: Predisporre emissione M05 senza attesa mandrino fermo.....	5-29
5.10.1.3	TstMfrM05: test predisposizione attuale.....	5-30
<b>6</b>	<b>MISCELLANEA .....</b>	<b>6-1</b>
6.1	Sinonimi miscellanei.....	6-1
6.2	POWER-ON .....	6-2
6.3	MESSAGGI PARAMETRIZZATI .....	6-2
6.3.1	Esempio di utilizzo con stringa : .....	6-5
6.3.2	Esempio di utilizzo con numero : .....	6-5
6.3.3	Gestione Storico Allarmi ed Eventi del CNC.....	6-5
6.3.3.1	Eventi registrati al BOOT, ALT, EXE del CNC .....	6-6
6.3.3.2	Eventi registrati alla pressione di un tasto della tastiera CNC:.....	6-6
6.3.3.3	Eventi registrati alla richiesta di cambio modo del CNC.....	6-6
6.3.3.4	Eventi registrati al cambio modo del CNC.....	6-6
6.3.3.5	Eventi registrati alla variazione dei segnali OHLD, OESE, IFHD, FATTOM .....	6-7
6.3.3.6	Eventi registrati allo START e FINE Part-Program .....	6-7
6.3.3.7	Allarmi registrati .....	6-7
6.3.4	Gestione Invio Messaggi tramite SMS.....	6-8
6.3.4.1	IOPLCSendSMS: invia messaggio tramite SMS .....	6-8
6.4	GESTIONE CONVERTITORI A/D E D/A .....	6-9
6.4.1	Convertitori A/D.....	6-9
6.4.2	Convertitori D/A.....	6-11
6.4.3	Bufferizzazione Convertitori A/D.....	6-12
6.4.4	Funzioni per bufferizzazione convertitori A/D .....	6-13
6.4.4.1	start_buf_ad: attiva la bufferizzazione .....	6-13
6.4.4.2	stop_buf_ad: disattiva la bufferizzazione.....	6-13
6.4.4.3	read_buf_ad: legge i valori bufferizzati dal CNC .....	6-13
6.5	Funzioni per il Part-Program.....	6-14
6.5.1	sel_pp: selezione Part-Program .....	6-14
6.5.2	GetPPSel: leggi il nome del Part-Program selezionato .....	6-15
6.5.3	GetExecutionTime: legge il tempo di esecuzione.....	6-15
6.5.4	IOPLCBlockPPSel: blocco/sblocco del Part-Program .....	6-15
6.5.5	IOPLCSelectGraphPP: selezione del Part-Program di Grafica .....	6-16
6.5.6	IsMtXOn: Test matrice dinamica.....	6-16
6.5.7	Cambio o modifica Part-Program .....	6-17
6.6	KEY .....	6-17
6.6.1	Regole per la scrittura del sottoprogramma KEY: .....	6-18
6.6.2	KEY Immediati .....	6-18
6.6.3	KEY Sincroni.....	6-19



---

6.6.4	KEY Asincroni.....	6-20
6.6.5	Note importanti .....	6-20
6.6.6	Precisazioni sull'uso di KEY .....	6-21
<b>6.7</b>	<b>CONTROLLO DELLO STATO DEL CNC.....</b>	<b>6-21</b>
6.7.1	is_cnc_work: testa se il CNC sta lavorando .....	6-21
6.7.2	cnc_all: mette il CNC in Allarme .....	6-21
6.7.3	IOPLCReadCNCLanguage: legge la Lingua attuale del CNC .....	6-22
6.7.4	IOPLCShutDown: "spegne" il CNC.....	6-22
<b>6.8</b>	<b>SINCRONIZZAZIONI CNC↔PLC senza arresto sul blocco.....</b>	<b>6-23</b>
6.8.1	tst_out: testa i bit delle parole di sincronizzazione CNC↔PLC.....	6-23
6.8.2	zero_out: azzerà i bit delle parole di sincronizzazione CNC↔PLC .....	6-23
<b>6.9</b>	<b>OPZIONI SOFTWARE.....</b>	<b>6-24</b>
6.9.1	rd_opz: legge le word OPZIONI presenti sul CNC .....	6-24
<b>6.10</b>	<b>LETTURA/SCRITTURA DI VARIABILI .....</b>	<b>6-25</b>
6.10.1	read_var: lettura di una variabile di sistema.....	6-25
6.10.2	write_var: scrittura di una variabile di sistema.....	6-25
6.10.3	rw_var: lettura o scrittura di una variabile di sistema .....	6-25
6.10.4	Parametri in ingresso ed in uscita .....	6-25
<b>6.11</b>	<b>ACCESSO AD AREE U.....</b>	<b>6-27</b>
6.11.1	Funzioni per accesso all'area U.....	6-27
<b>6.12</b>	<b>COMANDI MANUALI DA PLC .....</b>	<b>6-28</b>
6.12.1	qalcmd: invio comandi MDI al CNC.....	6-28
<b>6.13</b>	<b>ESEGUIBILE ESTERNO AL CNC .....</b>	<b>6-29</b>
6.13.1	IOPLCRunFileExe: lancio di eseguibile esterno al CNC .....	6-29
<b>6.14</b>	<b>AUTOAPPRENDIMENTO {F} .....</b>	<b>6-29</b>
6.14.1	Sinonimi utilizzati per l'autoapprendimento .....	6-30
6.14.2	set_marker: definisce un marker .....	6-30
6.14.3	set_text: definisce un testo .....	6-30
6.14.4	StoreAxisValue: registra le quote assi.....	6-31
6.14.5	StoAxValMark: registra le quote assi con Marker .....	6-31
6.14.6	StoreText: registra un testo .....	6-31
6.14.7	wTestApp: stato della richiesta di memorizzazione attuale.....	6-31
<b>6.15</b>	<b>FUNZIONI DI SCRITTURA/LETTURA FILES.....</b>	<b>6-32</b>
6.15.1	IOPLCWriteFile: scrittura file .....	6-32
6.15.1.1	Esempio di scrittura File .....	6-33
6.15.2	IOPLCReadFile: lettura file.....	6-37
6.15.3	IOPLCGetRecord: lettura del record .....	6-38
<b>6.16</b>	<b>FUNZIONI MISCELLANEE.....</b>	<b>6-39</b>
6.16.1	GetSerialNumber: lettura numero di commessa .....	6-39
6.16.2	s_man: salva modo manuale.....	6-39
6.16.3	r_man: ripristina modo manuale .....	6-39
<b>7</b>	<b>TASTIERA .....</b>	<b>7-1</b>
7.1	Gestione della tastiera.....	7-2

---



---

7.1.1	Sinonimi utilizzati per la gestione tastiera.....	7-2
7.1.2	Codici dei tasti operativi del CNC .....	7-4
7.1.3	Routines a disposizione del PLC .....	7-4
7.1.3.1	Routine "TASTCNC" .....	7-4
7.1.3.2	Routine "TASFUECS" .....	7-5
7.1.4	Tasti funzione CUSTOM .....	7-6
7.1.4.1	Esempio di gestione Tasti funzione CUSTOM .....	7-7
<b>7.2</b>	<b>I/O su tastiera .....</b>	<b>7-9</b>
<b>7.3</b>	<b>Modulo IKR (Interfaccia Keyboard Remoto).....</b>	<b>7-10</b>
7.3.1	Sinonimi utilizzati per la gestione IKR.....	7-10
<b>7.4</b>	<b>Commutazione di due Tastiere .....</b>	<b>7-12</b>
<b>7.5</b>	<b>TASTIERA PORTATILE.....</b>	<b>7-12</b>
<b>8</b>	<b>FINESTRE VIDEO PERSONALIZZATE.....</b>	<b>8-1</b>
8.1	Sinonimi utilizzati per la gestione FINESTRE VIDEO .....	8-1
<b>8.2</b>	<b>Routines a disposizione del PLC.....</b>	<b>8-2</b>
8.2.1	Routine "WINPLC" .....	8-2
8.2.2	Routine "GESTWPLC" .....	8-2
8.2.3	Routine "WWRITEWINDOW" .....	8-2
<b>8.3</b>	<b>Esempi di gestione WINDOWS PLC .....</b>	<b>8-3</b>
<b>8.4</b>	<b>Logo personalizzato .....</b>	<b>8-6</b>
<b>8.5</b>	<b>Gestione formato attuale del CNC.....</b>	<b>8-6</b>
8.5.1	IOPLCWinvisActualForm: gestisce formato attuale CNC .....	8-7
<b>9</b>	<b>CAMBIO UTENSILI .....</b>	<b>9-1</b>
<b>9.1</b>	<b>GENERALITÀ .....</b>	<b>9-1</b>
9.1.1	Parametri di taratura del Cambio Utensili .....	9-2
<b>9.2</b>	<b>GESTIONE DEL MAGAZZINO UTENSILI .....</b>	<b>9-3</b>
9.2.1	Gestione a POSTI DEDICATI.....	9-3
9.2.2	Gestione RANDOM.....	9-3
9.2.3	Gestione RANDOM FISSO.....	9-4
<b>9.3</b>	<b>DIMENSIONE DELL'UTENSILE .....</b>	<b>9-4</b>
<b>9.4</b>	<b>TT.BIN (file della tabella utensili: Tool Table) .....</b>	<b>9-4</b>
<b>9.5</b>	<b>Routines a disposizione del PLC.....</b>	<b>9-5</b>
9.5.1	IOPLC_nread_tt: lettura dei dati di un'utensile .....	9-5
9.5.2	IOPLCGet_nread_tt: lettura dei dati di un utensile .....	9-6
<b>9.6</b>	<b>CLASSIFICAZIONE DEI CAMBI UTENSILI AUTOMATICI .....</b>	<b>9-14</b>
<b>9.7</b>	<b>SEGNALI LOGICI.....</b>	<b>9-15</b>

---



---

<b>9.8</b>	<b>DIALOGO TCI ↔ PLC .....</b>	<b>9-27</b>
9.8.1	Comandi inviati da TCI a PLC. ....	9-27
9.8.2	Comandi inviati da PLC a TCI. ....	9-28
9.8.2.1	WRCUST e RDCUST: scrittura e lettura dei parametri Custom. ....	9-29
<b>9.9</b>	<b>PROCEDURA CARICO/SCARICO UTENSILI DA MAGAZZINO .....</b>	<b>9-31</b>
9.9.1	PROCEDURA MANUALE .....	9-31
9.9.2	PROCEDURA AUTOMATICA .....	9-31
9.9.2.1	Carico utensile .....	9-32
9.9.2.2	Scarico utensile .....	9-33
<b>9.10</b>	<b>COMPORAMENTO TCI IN FASE DI RESET.....</b>	<b>9-34</b>
<b>9.11</b>	<b>VARI TIPI DI CAMBI UTENSILI AUTOMATICI.....</b>	<b>9-35</b>
9.11.1	C.Ut. a TORRETTA .....	9-36
9.11.1.1	TORRETTA(TYP='T') - Posti dedicati(TGM=0) - SENZA Ut. Intermedi (UTI=0) ..	9-36
9.11.2	C.Ut. a CATENA ACICLICO.....	9-38
9.11.2.1	ACICLICO (TYP=' ') - RANDOM (TGM=1) - SENZA Ut. Intermedio (UTI=0) .....	9-38
9.11.2.2	ACICLICO (TYP=' ') - RANDOM (TGM=1) - CON Ut. Intermedio (UTI=1 o 2) ...	9-40
9.11.2.3	ACICLICO (TYP=' ') - Posti DEDICATI(TGM=0) - CON Ut. Intermed. (UTI=1 o 2) .	9-42
9.11.2.4	ACICLICO (TYP=' ') - Posti DEDICATI (TGM=0) - SENZA Ut. Intermedio (UTI=0)	9-44
9.11.3	C.Ut. a CATENA CICLICO .....	9-46
9.11.3.1	CICLICO (TYP='C') - RANDOM (TGM=1) - CON Ut. Intermedio (UTI=1 o 2) .....	9-46
<b>9.12</b>	<b>CAMBIO UTENSILI NEL TORNIO {T}.....</b>	<b>9-49</b>
9.12.1	Parametri di taratura del Cambio Utensili.....	9-50
9.12.2	GESTIONE DEGLI UTENSILI .....	9-50
9.12.2.1	Gestione a POSTI DEDICATI .....	9-50
<b>9.13</b>	<b>TT.BIN (file della tabella utensili: Tool Table) {T} .....</b>	<b>9-51</b>
<b>9.14</b>	<b>CLASSIFICAZIONE DEI CAMBI UTENSILI AUTOMATICI {T}.....</b>	<b>9-51</b>
9.14.1	SEGNALI LOGICI.....	9-52
9.14.2	DIALOGO TCI ↔ PLC.....	9-52
<b>9.15</b>	<b>VARI TIPI DI CAMBI UTENSILI AUTOMATICI {T} .....</b>	<b>9-52</b>
9.15.1	C.Ut. a REVOLVER (TYP='R') .....	9-54
9.15.2	C.Ut. a TORRETTA (TYP='T').....	9-55
9.15.3	C.Ut. a CATENA ACICLICO (TYP=' ') - SENZA Ut. Intermedio (UTI=0) .....	9-56
9.15.4	C.Ut. a CATENA ACICLICO (TYP=' ') - CON Ut. Intermedio (UTI=1 o 2) .....	9-58
<b>9.16</b>	<b>RICERCA UTENSILE NEL MAGAZZINO .....</b>	<b>9-60</b>
9.16.1	Routine "MOV MAG" .....	9-60

---



---

---

**Note:**





## CAPITOLO 1

### 1 SEGNALI DI STATO E FUNZIONI AUSILIARIE

#### 1.1 SEGNALI DI STATO DEL CNC

##### 1.1.1 Sinonimi di stato del CNC

Per la gestione del funzionamento "base" del CNC tramite il PLC, sono presenti alcuni segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario, è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

**{W}** Nella prima colonna, il sinonimo scritto in **GRASSETTO** è quello da preferire; l'eventuale altro sinonimo è comunque valido ed è stato mantenuto per compatibilità con la serie D.

Quando si parla del Mandrino ci si riferisce sia al mandrino principale sia secondario; quando ci siano differenze, queste saranno indicate esplicitamente.

<b>pon</b>	<b>CNC → PLC</b>	<b>power on</b> <b>pon=1</b> solo durante il primo "giro" del PLC dopo l'accensione del CNC o dopo un passaggio ALT → EXE. Per maggiori dettagli vedi l'apposito paragrafo al Capitolo 6.
<b>MODE_SET</b> <b>OSET</b>	<b>CNC → PLC</b>	<b>CNC in modo SET ASSI.</b> Possono essere attivate le procedure (sottoprogrammi SETnn e SETAL) per la ricerca dello zero degli assi. Tutti gli assi sono agganciati in punto a punto ad eccezione dell'asse mandrino che si comporta allo stesso modo di JOG.
<b>MODE_JOG</b> <b>OJOG</b>	<b>CNC → PLC</b>	<b>CNC in modo JOG.</b> Si possono muovere gli assi con gli appositi comandi o pulsanti oppure con i volantini. Gli assi (sia CNC sia FRE) sono agganciati in contouring. <b>Attenzione:</b> se l'asse mandrino è in rotazione, rimane in rotazione ed agganciato in punto a punto e velocità. Se è orientato viene agganciato in contouring. Se il mandrino non ha trasduttore rimane indipendente e non si può muovere in JOG.
<b>OJPS</b>	<b>CNC → PLC</b>	<b>JOG POSITIVO IN CORSO.</b> Un comando di JOG in direzione positiva è in corso.
<b>OJNG</b>	<b>CNC → PLC</b>	<b>JOG NEGATIVO IN CORSO.</b> Un comando di JOG in direzione negativa è in corso.



<b>MODE_MDI</b> OMAN	<b>CNC → PLC</b>	<p><b>CNC in modo MANUALE ( MDI : Manual Data Input ).</b> Il CNC accetta comandi introdotti dall'operatore nell'apposita finestra di input. Gli assi sono agganciati CNC o FRE come imposto in AXS.TAR e/o eventualmente modificato dal file COST. Durante l'esecuzione o lettura di programmi la situazione di aggancio assi (in punto a punto o contouring) varia a seguito delle istruzioni &lt;CNC:...&gt; o &lt;FRE:...&gt;. Con RESET viene ripristinata la situazione definita in AXS.TAR.</p>
<b>MODE_AUT</b> OAUTO	<b>CNC → PLC</b>	<p><b>CNC in modo AUTOMATICO.</b> Il CNC può eseguire il Part-Program di lavoro selezionato. Per gli assi, valgono le stesse considerazioni fatte per MODE_MDI.</p>
<b>MODE_FST</b> OFAST	<b>CNC → PLC</b>	<p><b>CNC in modo PROVA PROGRAMMA [FAST]</b> Gli spostamenti dichiarati a FEED di lavoro nel Part-Program, vengono eseguiti a velocità rapida, regolata, come nella modalità G00, dall'override feed. Vedi precisazioni a fine tabella (par. 1.1.1.1)</p>
<b>MODE_FND</b> OFIND	<b>CNC → PLC</b>	<p><b>CNC in modo RICERCA [FIND]</b> Sono accettati i comandi per ricercare una un'operazione desiderata nel Part-Program.</p>
<b>MODE_RPZ</b> ORPZ	<b>CNC → PLC</b>	<p><b>CNC in modo RIPOSIZIONAMENTO.</b> Possono essere dati i comandi per riportare gli assi sull'ultima quota programmata o voluta dal CNC dopo una ricerca o lettura o dopo un'uscita dal modo AUTOMATICO; possono essere "riposizionate" anche le F.A. dichiarate tali in FA.TAB. Per gli assi valgono le stesse considerazioni fatte per MODE_MDI.</p>
<b>OCOK</b>	<b>CNC → PLC</b>	<p><b>Condizioni iniziali OK.</b> Vedi nota a fine tabella.</p>
<b>OAOK</b>	<b>CNC → PLC</b>	<p><b>Part-Program sospeso in AUTOMATICO</b> Ora il CNC <u>non è in modo automatico</u> ma è sospeso un Part-Program iniziato in modo [AUTO]; quando il CNC verrà rimesso in [AUTO], OAOK verrà messo a zero e verrà resettato OCOK = 1. Se il CNC ha OAOK=0 e CNOK=0 significa che il CNC non ha in sospeso la lettura/esecuzione di nessun Part-Program.</p>
<b>OESE</b>	<b>CNC → PLC</b>	<p><b>OESE = 1</b> se l'interpolatore oppure una F.A. è in <b>esecuzione</b>. <b>ESECUZIONE CONTINUA:</b> il segnale è sempre VERO fino a che non viene trovato nel Programma M02, oppure viene premuto il pulsante [RESET]. <b>ESECUZIONE SINGOLA:</b> il segnale è VERO fino a movimento o "azione" F.A. terminata. <b>LETTURA:</b> sia in SINGOLA sia in CONTINUA si ha lo stesso comportamento del modo esecuzione. <b>MDI (Manual Data Input):</b> Il segnale è VERO fino ad esecuzione del comando terminata. <b>JOG:</b> il segnale è VERO fino a termine movimento. <b>JOG con VOLANTINI attivi:</b> il segnale è sempre VERO. In qualsiasi modalità sia il CNC, premendo <b>[RESET]</b> diventa <b>OESE=0</b>.</p>



<b>MODE_EXE</b> OEXE	<b>CNC → PLC</b>	<b>CNC in modo ESECUZIONE.</b> MODE_EXE = 0 se il CNC è in modo <b>LETTURA</b> .
<b>MODE_CNT</b> OCNT	<b>CNC → PLC</b>	<b>CNC in modo ESECUZIONE o LETTURA CONTINUA.</b> MODE_CNT = 0 se il CNC è in modo Esecuzione/Lettura <b>SINGOLA</b> .
<b>OTWIST</b> OTWT {F}	<b>CNC → PLC</b>	<b>TWIST attivo.</b> È attiva la modalità RTCP per teste rotative (TWIST).
<b>OM01</b>	<b>CNC → PLC</b>	<b>M01 attivo.</b> CNC predisposto per eseguire gli stop programmati opzionali.
<b>OSKP</b>	<b>CNC → PLC</b>	<b>SKIP attivo.</b> CNC predisposto per eseguire i blocchi opzionali /N.
<b>OHLAX</b>	<b>CNC → PLC</b>	<b>ASSI in HOLD (BLOCCO).</b> Il CNC passa in questo stato per richiesta dell'operatore (a seguito della pressione del pulsante AXES H.) o per richiesta del PLC (IBLAX) o per il passaggio 0→1 di OHLMND. Il PLC può fermare le "azioni" in svolgimento sulla Macchina Utensile (M.U.), tenendo presente che il CNC provvede automaticamente a portare a zero la velocità di tutti gli assi CNC, mantenendoli fermi in ciclo di spazio; NON viene fermato il mandrino. All'atto dello sblocco dato dal pulsante [REL], il CNC riavvierà gli assi e le "azioni" interrotte potranno essere riprese. Lo stato di OHLAX: <ul style="list-style-type: none"><li>- Viene azzerato al passaggio AUTO→MDI.</li><li>- Viene azzerato al passaggio MDI→AUTO con Part-Program NON in esecuzione (condizioni iniziali OCOK = 0).</li><li>- Riassume la condizione precedente l'uscita dal modo AUTO, al passaggio MDI→AUTO, con Part-Program in esecuzione (condizioni iniziali OCOK = 1).</li></ul> OHLAX = 0 all'accensione o dopo RESET o dopo il comando di sblocco dato dal pulsante [REL] o dopo cambio modo, purché l'ingresso IBLAX sia = 0.
<b>OHLMND</b>	<b>CNC → PLC</b>	<b>MANDRINO in HOLD (BLOCCO).</b> Il CNC passa in questo stato su richiesta dell'operatore (a seguito della pressione del pulsante HOLD) o su richiesta del PLC (IBLAXMND) o per altre situazioni rilevate dal CNC stesso; contemporaneamente viene forzato a 1 anche OHLAX. Il PLC deve fermare le "azioni" in svolgimento sulla Macchina Utensile (M.U.) tenendo presente che il CNC provvede automaticamente a portare a zero la velocità del mandrino (e di tutti gli assi CNC, mantenendoli fermi in ciclo di spazio). All'atto dello sblocco dato dal pulsante [REL], il CNC riavvierà il mandrino e le "azioni" interrotte dovranno essere riprese. Lo stato di OHLMND <u>NON</u> cambia con il passaggio AUTO→MDI o MDI→AUTO. OHLMND = 0 all'accensione o dopo RESET o dopo il comando di sblocco dato dal pulsante [REL] solo se l'ingresso IBLAXMND è = 0.



<b>OHLD</b>	<b>CNC → PLC</b>	<p><b>CNC in HOLD (BLOCCO).</b> Il CNC passa in questo stato quando entrambi i segnali <u>OHLMND e OHLAXS</u> sono = 1 su richiesta dell'operatore (a seguito della pressione del pulsante HOLD) o su richiesta del PLC (IBLAXMND) o per altre situazioni rilevate dal CNC stesso. OHLD = 0 all'accensione o dopo RESET o dopo il comando di sblocco dato dal pulsante [REL] o dopo cambio modo, purché l'ingresso IBLAXMND sia = 0.</p>
<b>OALL</b>	<b>CNC → PLC</b>	<p><b>CNC in ALLARME.</b> E' una situazione di emergenza. Il PLC deve bloccare i motori, fermare il mandrino, ecc. ... Con OALL=1 viene messo a 1 il segnale OHLMND e di conseguenza i segnali OHLAX ed OHLD. OALL verrà rimesso a 0 con [RESET] se la situazione che lo ha provocato non permane più, altrimenti, terminato il ciclo di RESET, OALL ritorna = 1.</p>
<b>ORST</b>	<b>CNC → PLC</b>	<p><b>RESET in corso.</b> <b>ORST = 1</b> (per un tempo non inferiore a 500 msec) se viene premuto il pulsante [RESET] sulla tastiera del CNC, all'accensione del CNC e ad ogni passaggio ALT/EXE. Il PLC deve usare ORST per ripristinare "a zero" la situazione della M.U. . Vedere, a fine tabella, la sequenza delle operazioni fatte dal CNC in fase di RESET.</p>
<b>IBLAXMND</b> IBL	<b>PLC → CNC</b>	<p><b>Richiesta di BLOCCO CNC.</b> Posto a 1 impone velocità 0 a tutti gli assi CNC ed al mandrino ed impedisce il proseguimento del Part Program. Posto poi a 0, permette la ripartenza di assi, mandrino e Part Program, che avverrà dopo la pressione del tasto [REL] di annullo (RELEASE) del BLOCCO CNC. Mettere IBLAXMND=1 transitoriamente se si vuole bloccare sia assi CNC che il mandrino; esegue le stesse funzioni del pulsante HOLD della tastiera. Mettere IBLAXMND=1 fisso se si vuole rendere inattivo il pulsante [REL].</p>
<b>IBLAX</b>	<b>PLC → CNC</b>	<p><b>Richiesta di BLOCCO ASSI.</b> Posto a 1 impone velocità 0 a tutti gli assi CNC ma non al mandrino ed impedisce il proseguimento del Part Program. Posto poi a 0, permette la ripartenza di assi e Part Program, che avverrà dopo la pressione del tasto REL.</p>
<b>IRES</b>	<b>PLC → CNC</b>	<p><b>Consenso al RESET CNC.</b> Vedi nota a fine tabella.</p>
<b>NORESET</b>	<b>PLC → CNC</b>	<p><b>Blocca la prosecuzione della fase di RESET</b> (MAX 15 sec) dalla premuta del tasto RESET. Esso é da utilizzare quando é necessario comunque terminare una certa attività indipendentemente dalla premuta del tasto di RESET. Se si supera il tempo max (15 secondi) il CNC da allarme.</p>



<b>CK1SEC</b>	<b>CNC → PLC</b>	<b>CLOCK 1 SECONDO.</b> E' un segnale di durata un giro di PLC e di frequenza 1 secondo; può essere utile per pilotare contatori, ad esempio per la durata e/o l'intervallo di lubrificazione.
<b>OSYSMILL</b>	<b>CNC → PLC</b>	<b>Sistema FRESA installato</b> Vedi nota a fine tabella.
<b>OSYSTURN</b>	<b>CNC → PLC</b>	<b>Sistema TORNIO installato</b> Vedi nota a fine tabella.
<b>OG37</b>	<b>CNC → PLC</b>	<b>Funzionalità TRANSMIT attiva</b> Settato da G37; resettato da G36, RESET, %.
<b>ORIPOK</b>	<b>CNC → PLC</b>	<b>Tutti gli organi programmati sono riposizionati</b> Valido solo in modo Riposizionamento
<b>G669</b>	<b>CNC → PLC</b>	<b>Macro G669 Sellatura Flangie in corso sul CNC</b>

#### 1.1.1.1 Note varie sui segnali di stato del CNC

##### **CNC in modo PROVA PROGRAMMA [FAST]**

Dalla release Software V3.02, è possibile definire in COSTUSER (vedi #157 in COST) la Feed massima in dry-run; se = 0 viene usata la massima velocità rapida definita in AXS.TAR.

L'operatore può comunque stabilire una diversa velocità massima; premendo infatti il tasto DRY RUN del menu verticale, la Window di "richiesta conferma attivazione DRY RUN" propone, in un campo di data-entry, la feed attualmente valida; l'operatore può semplicemente confermare quanto predisposto premendo una seconda volta il tasto stesso o scrivere il nuovo valore desiderato, memorizzarlo con [ENTER] ed infine confermare con il solito tasto.

La routine standard di libreria tasfuecs(), descritta nel capitolo 7, realizza il funzionamento suddetto utilizzando la funzione SetLimFeedOn( ) descritta nel capitolo 3. Uscendo dal modo FAST, la tasfuecs() toglie la limitazione tramite la funzione SetLimFeedOff(). La variabile ALIMFEED (ALIMDRYR da V4.00) segnala, se = 1, che la limitazione è stata attivata e nella variabile double FMDRYRUN è presente il valore imposto. Definendo, ad esempio al pon e prima della chiamata di tasfuecs(), un valore diverso da 0 per FMDRYRUN, questo viene usato come valore proposto alla prima attivazione del modo Fast, "sovrascrivendo" quanto eventualmente definito su #157 o la massima rapida di AXS.TAR . Il PLC può controllare il valore attuale di FMDRYRUN introdotto dall'operatore (WDRYRUN segnala che è aperta la finestra di conferma) ed eventualmente limitarlo ad un valore massimo ammesso. Quando il CNC è in modo PROVA PROGRAMMA (MODE\_FST=1) eventuali modifiche a FMDRYRUN non hanno effetto; per modificare la velocità massima è comunque possibile usare la funzione SetLimFeedOn() con un diverso valore.

Dalla release Software V4.00, il funzionamento suddetto è valido solo se viene definito <%#112=1> in COSTUSER (vedi descrizione in COST); per default è definito <%#112=0> in COST e la Window di "richiesta conferma attivazione DRY RUN" non prevede la possibilità di definire una feed e quindi il DRY RUN viene eseguito alla velocità rapida.

Con <%#112=1> in COSTUSER, la feed proposta all'accensione del CNC (ed al passaggio ALT->EXE) dipende da se e come è definito #157 in COSTUSER:

- non definito (default): l'ultima feed definita dall'operatore prima dello spegnimento.
- #157=0 : la velocità rapida
- #157>0 : la velocità definita in COSTUSER

Se il PLC usa le funzioni SetLimFeedOn() e SetLimFeedOff() con una sua logica, è preferibile non permettere all'operatore la modifica della feed, lasciando <%#112=0> come definito in COST.

C'è da notare che la routine tasfuecs() verifica se la limitazione è già attiva quando viene richiesto il modo DRY RUN, salva il valore attuale (in VALLIMFD) e permette di definirne uno inferiore; all'uscita dal modo DRY RUN la tasfuecs() ripristina il valore precedente se nel frattempo non è stato modificato



dal PLC (se il valore attuale è diverso da quello salvato o da quello introdotto dall'operatore, la limitazione non viene modificata né disattivata).

Il PLC non deve chiamare la SetLimFeedOff() durante il DRY RUN (MODE\_FST=1) ma eventualmente forzare un valore più basso di quello attuale e ripristinarlo in base alle condizioni di macchina.

**N.B.** Il funzionamento sopra descritto è disponibile se si usano i tasti funzione del menu verticale nella configurazione standard ECS definita in BASMENU.MEN a partire dalla release software V3.02; volendo attivare questa funzionalità con menu custom precedenti, occorre cambiare il codice dal tasto DRY RUN (normalmente P6 del menu 5) da 159 a 280.

### Sistema Installato FRESA oppure TORNIO

I CNC ECS serie WIN (dalla release Software V3.00) possono gestire **due tipi di macchine** (Fresa e Tornio) con la possibilità di attivare dinamicamente parte delle funzionalità dell'altro tipo di macchina. La configurazione "di default" della macchina è comunicata al PLC tramite i due segnali OSYSMILL (Fresa) ed OSYSTURN (Tornio). Questi segnali non sono normalmente usati dal PLC, mentre possono essere importanti quelli che segnalano il cambio di funzionalità e lo stato attualmente attivo. Per il cambio di funzionalità sono previste alcune funzioni Gxxx specifiche (G630, G631, G632) che appaiono, al PLC, come Funzioni Ausiliarie e che quindi sono dettagliatamente spiegate al Paragrafo 1.2 "Funzioni Ausiliarie".

### Condizioni iniziali OK.

Il segnale **OCOK** è settato dal CNC quando viene premuto il pulsante START ed il CNC sa quale programma deve iniziare ad eseguire.

OCOK rimane settato fino a che non viene eseguito M02 (e ricevuto FATTO dal PLC) o fino a RESET. Se il CNC è in modo MANUALE, o JOG o SET, OCOK=1 fino a che non è terminato il comando dato. L'esecuzione di un KEY (qualunque tipo) non altera lo stato di OCOK.

Riassumendo, il segnale di condizioni iniziali OK è VERO quando (considerando il modo attivo) il CNC sa quale programma eseguire anche se in ESECUZIONE/LETTURA SINGOLA.

Uscendo dal modo AUTO, OCOK viene salvato; è questo segnale che dice al CNC se si deve fare riposizionamento oppure no ritornando in AUTO.

### RESET in corso.

Sequenza delle operazioni fatte dal CNC in fase di RESET:

1. Ferma tutti gli assi (CNC, FRE e MANDRINO). Ad assi fermi rende VERO il segnale OHLAX. A mandrino fermo rende VERO il segnale OHLD.
2. Rende VERO il segnale **ORST** e attende che il PLC dia consenso alla prosecuzione della fase reset attraverso il segnale **IRES=1**. Se entro circa 20 sec. il PLC non rende VERO il segnale IRES, il CNC esce dalla fase di reset con allarme specifico.
3. Attende che il PLC ed il Tool Controller rendano il "fatto" delle F.A. in corso. Se entro circa 20 sec. questo non avviene, il CNC esce dalla fase di reset con allarme specifico. Il tempo di 20 sec. è unico per la fase 2 e 3.
4. Per i soli assi che hanno IEME=1 (se non già fatto precedentemente) mette OBOK=0. Per i soli assi che hanno ISGA=1 e OBOK=1, effettua lo sgancio asse.
5. Eventuali volantini inseriti vengono tolti: 0⇒OVL1, OVL2, OVL3. Vengono tolti anche eventuali consensi ad emissione di KEY 0⇒OKEYIM, OKEYSN, OKEYAS. Viene tolto funzionamento copia e 0⇒OMDGCPY e OSTOCPY
6. Viene eseguito il programma RESET forzando modo AUTO e EC. Se nel programma RESET vi sono errori si esce dalla fase di reset con allarme specifico dell'errore di sintassi del Programma.
7. Si rimette modo e sottomodo di partenza e si riinizializzano gli assi che hanno il segnale OBOK=0.
8. Si agganciano tutti gli assi in funzione del modo in cui è il CNC. Se vi sono errori sulle schede assi nella fase 7 e/o 8 si esce dalla fase di RESET con allarme specifico sull'asse interessato.



9. Fase di reset finita OK: 0⇒ORST, OHLD, OHLAX.

Quando il CNC esce con allarme specifico nelle varie fasi appena viste, per alcuni tau viene messo 0⇒OHLAX, OHLD, OALL, poi questi segnali ritornano = 1. Inoltre viene messo 0⇒ORST e 1⇒OALL. Al termine della fase di RESET, il CNC ritorna nello stesso modo e sottomodo in cui era quando è stato comandato il reset.

#### **Consenso al RESET CNC.**

Con IRES=1, il PLC dà il consenso alla prosecuzione della fase di RESET sul CNC.

Il PLC deve mettere IRES=1 dopo che ha inizializzato i suoi parametri interni ed asserito, se necessario, ISGA=1.

Il PLC deve mantenere IRES=1 fintanto che ORST=1.

La normale gestione del segnale IRES è già realizzata in una routine standard del PLC, eseguita automaticamente ad ogni loop. Solo se il PLC desidera inibire per un certo tempo (max. 20 sec. circa) la prosecuzione del RESET, deve mettere a 1 il segnale **NORESET** (vedi anche capitolo MISCELLANEA).



## 1.2 FUNZIONI AUSILIARIE

Per funzioni ausiliarie, d'ora in avanti chiamate F.A., s'intendono i codici utilizzati sia dal CNC sia quelli disponibili per pilotare accessori e devices esterni.

Queste si dividono in due gruppi: **F.A. interne** e **F.A. esterne**. Questa monografia prende in considerazione le **F.A. esterne**; per quelle interne al CNC (funzioni G) rimandiamo alla lettura delle "Norme di Programmazione".

I CNC ECS serie WIN dividono le F.A. esterne in:

- M** ⇒ PRIMA FUNZIONE MISCELLANEA
- S** ⇒ FUNZIONE SPEED
- T** ⇒ FUNZIONE TOOL
- H** ⇒ SECONDA FUNZIONE MISCELLANEA
- E** ⇒ TERZA FUNZIONE MISCELLANEA
- D** ⇒ FUNZIONE CORRETTORE UTENSILE **{F}**

### 1.2.1 Parametri relativi alle F.A.

Le F.A. esterne di normale uso sulle Macchine Utensili sono già predisposte da ECS per un funzionamento corretto, ma per alcune di esse è possibile variare il modo di agire. Queste F.A. sono definite negli appositi files di taratura (vedi MANUALE TARATURE 720P385).

Qui di seguito, sono riassunti i files che contengono tarature riguardanti le F.A..

COST	Parametri vari
FA.TAB	F.A. subroutine
SECFA, THIFA	Seconda e Terza F.A.

### 1.2.2 INDIRIZZI F.A. ESTERNE

M00÷M199	MISCELLANEE. Servono principalmente per impartire comandi esecutivi.
...± 32768	SECONDA E TERZA F.A. MISCELLANEA Sono tipicamente utilizzate per la gestione di assi indexati (ad esempio tavole o testine hirt).
S0÷S4294967295	SPEED. Utilizzate per programmare la velocità del mandrino.
T00÷T4294967295	TOOL. Identificano l'utensile ed il correttore ( <b>{F}</b> lunghezza e raggio; <b>{T}</b> lunghezza su X e Z, raggio) da abbinare all'utensile al momento del cambio utensile.
D0÷D4294967295 <b>{F}</b>	DCOR. Correttore dimensioni geometriche utensile da abbinare al Tool montato.





### 1.2.3 CLASSI DI F.A. E ORDINE DI EMISSIONE

Le F.A. sono funzionalmente divise in tre classi:

**ANTICIPATE** (F.A. SUBROUTINE)

**INTERMEDIE** (IMMEDIATE VERSO PLC)

**POSTICIPATE** (F.A. SUBROUTINE)

e sono caratterizzate all'interno del file di taratura FA.TAB, in cui ogni record definisce una F.A.. Le F.A. definite in FA.TAB come ANTICIPATE o POSTICIPATE sono F.A. SUBROUTINE e saranno discusse più avanti in questo capitolo.

Si definiscono F.A. SUBROUTINE quelle funzioni ausiliarie esterne che fanno riferimento ad un sottoprogramma (scritto secondo le normali regole di programmazione ISO e LIP) e lo mettono in esecuzione. Tutte le F.A. anticipate o posticipate sono F.A. subroutine; d'ora in avanti saranno chiamate anche FASUB.

#### 1.2.3.1 Classe anticipata

Le F.A. appartenenti a questa classe saranno emesse dal CNC verso il PLC, con una certa priorità fra di loro, ma sempre ANTICIPATE rispetto alle F.A. INTERMEDIE, a movimenti degli assi e alle F.A. POSTICIPATE (vedi Tabelle Ordine di Emissione F.A.)

#### 1.2.3.2 Classe intermedia

Queste F.A. vengono emesse tutte insieme, dopo tutte le F.A. ANTICIPATE ed assieme al movimento degli assi, ma sicuramente prima delle F.A. POSTICIPATE

#### 1.2.3.3 Classe posticipata

Queste F.A. vengono emesse con un certo ordine di priorità fra di loro, ma sempre dopo le F.A. ANTICIPATE, dopo le F.A. INTERMEDIE e dopo che il movimento degli assi è stato effettuato.

#### 1.2.3.4 Regola di programmazione delle F.A.

Nella stessa operazione N... del Part-Program possono essere scritte le F.A. M, S, D, E, H, T con la seguente regola:

- a) UNA **M** INTERMEDIA
- b) UNA **M** ANTICIPATA + UNA **M** INTERMEDIA
- c) UNA **M** ANTICIPATA + UNA **M** POSTICIPATA
- d) UNA **M** INTERMEDIA + UNA **M** POSTICIPATA

ed inoltre una F.A. **S**, una F.A. **T**, una F.A. **D {F}**.

In uno stesso record è possibile programmare come intermedie:

UNA F.A. **H** + UNA F.A. **E** + UNA F.A. **M** + UNA F.A. **S** oppure **T** oppure **D {F}**

In base alle dichiarazioni effettuate nel file di taratura FA.TAB, si possono presentare più casi, di cui riportiamo alcuni esempi nelle TABELLE A, B, C di questo paragrafo.

---

---

### 1.2.3.5 Tabelle ordine di emissione F.A.

**TABELLA A**

F.A. SUB Anticipate				INTERMEDIE	FUNZIONI posticipate	CICLI	SOTTO progr.	F.A. SUB Posticipate	SALTI	
M40	S	M03	T	Mov. Assi + 1 Funz. M + 1 Funz. H + 1 Funz. E + G800 ... G999	MACRO	G81	L<. .>	M00	JMP	
...		M04			G100	...	G600	M01		
...		M11	D		...	G89	G799	M02		IFF
M45		M13	{F}		G299	...	M05			
		M14					M06			
	M19				M10	M30				

**TABELLA B**

F.A. SUB Anticipate	INTERMEDIE	FUNZIONI posticipate	CICLI	SOTTO pro.	F.A. SUB Posticipate			SALTI	
M03	Mov. assi + 1 Funz. M + 1 Funz. H + G800 ... G999	MACRO	G81	L<. .>	E	S	T	M00	JMP
M04		G100	...	G600			D	M01	IFF
M11		...	G89	...			{F}	M02	
M13		G299	...	G799				M05	
M14								M06	
M19								M10	

**TABELLA C**

F.A. SUB Anticipate	INTERMEDIE	FUNZIONI posticipate	CICLI	SOTTO Prog.	F.A. SUB Posticipate	SALTI
M03	Mov. Assi + 1 Funz. M + 1 Funz. H + 1 Funz. E + 1 Funz. S o T o D + G800 ... G999	MACRO	G81	L<. .>	M00	JMP
M04		G100	...	G600	M01	IFF
M11		...	G89	...	M02	
M13		G299	...	G799	M05	
M14					M06	
M19					M10	

#### COMMENTI

Le tabelle A, B, C, esemplificano tre diversi modi di gestire le funzioni ausiliarie:

La **TAB. A** è valida nel caso sia usato il file di taratura FA.TAB fornito da ECS, dove, in particolare, vengono definite le M40...44, le S..., le T... e le D... come F.A. anticipate.

La **TAB. B** è valida nel caso che le S..., le T... e le D... siano state definite posticipate; anche la E..., in quest'esempio, è definita posticipata.

La **TAB. C** è valida nel caso che le S... e le T... siano definite intermedie.

Salvo dove indicato con il segno + (più), le funzioni raccolte in una stessa colonna sono "alternative" e non possono, quindi, essere programmate nello stesso blocco di Part-Program.



#### 1.2.4 Codici F.A. "M" già assegnati da E.C.S.

NOTA

(M) = MODALE: Sono quelle istruzioni il cui effetto permane fino a che non venga cambiato da un'altra istruzione della stessa categoria.

(A) = AUTOCANCELLANTE: Sono quelle il cui effetto permane solo nell'operazione in cui sono programmate.

<b>M00 ÷ M01</b>	(A)	Stop programmati
<b>M02</b>	(A)	Fine programma
<b>M03</b>	(M)	Mandrino in moto rotazione oraria (vedi Cap. "MANDRINO")
<b>M04</b>	(M)	Mandrino in moto rotazione antioraria (vedi Cap. "MANDRINO")
<b>M05</b>	(M)	Arresto rotazione mandrino (implicita in M19 e {F}M06)
<b>M06</b>	(A)	Cambio utensile {F} con arresto mandrino (vedi Cap. "CAMBIO UTENSILE")
<b>M10</b>	(M)	Bloccaggio assi
<b>M11</b>	(M)	Sbloccaggio assi
<b>M13/M14</b>	(M)	Come M03/M04 ma con refrigerante erogato
<b>M15</b>	(M)	Abilitazione cambio gamma mandrino automatico (programmare in un blocco a se stante).{T} Solo per mandrino secondario.
<b>M17</b>	(A)	Fine sottoprogramma
<b>M19</b>	(A)	Arresto mandrino con orientamento
<b>M30</b>	(A)	Come M02
<b>M40÷M45</b>	(M)	Cambio gamme di velocità mandrino

Tutte queste F.A., fatta eccezione per M17 e M00-M01, sono definite FASUB e sono presenti nel software di base del CNC. Nel caso vengano modificate, il costruttore della M.U. deve ben precisarne il funzionamento nella documentazione che accompagna la Macchina.

#### 1.2.5 Sinonimi utilizzati per le F.A.

Per la gestione delle F.A. tramite il PLC, sono presenti alcuni segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC). Le modalità di sincronizzazione **CNC ← → PLC** sono descritte più avanti in questo capitolo.

<b>OCODM1</b> OM (unsigned char)	<b>CNC → PLC</b>	Valore della F.A. M (0...199). Il valore è valido con NEWM = 1.
<b>NEWM</b>	<b>CNC → PLC</b>	Nuova F.A. M.
<b>FATTOM</b>	<b>PLC → CNC</b>	Fatto della Nuova F.A. M. Posto a 1 comunica al CNC che la M è stata acquisita. Il CNC può proseguire con la prossima istruzione.
<b>OCODFA2</b> OFA2 (short)	<b>CNC → PLC</b>	Valore della seconda F.A. Miscellanea (± 32768). Il valore è valido con NEWFA2 = 1.
<b>NEWFA2</b>	<b>CNC → PLC</b>	Nuova seconda F.A. Miscellanea



<b>FATFA2</b>	<b>PLC → CNC</b>	Fatto della Nuova seconda F.A. Miscellanea Posto a 1 comunica al CNC che la 2 <sup>a</sup> F.A. è stata acquisita. Il CNC può proseguire con la prossima istruzione.
<b>OCODFA3</b> OFA3 (short)	<b>CNC → PLC</b>	Valore della terza F.A. Miscellanea (± 32768). Il valore è valido con NEWFA3 = 1.
<b>NEWFA3</b>	<b>CNC → PLC</b>	Nuova terza F.A. Miscellanea
<b>FATFA3</b>	<b>PLC → CNC</b>	Fatto della Nuova terza F.A. Miscellanea Posto a 1 comunica al CNC che la 3 <sup>a</sup> F.A. è stata acquisita. Il CNC può proseguire con la prossima istruzione.
<b>OTDS</b> (long) <b>OTDSL</b> (short) <b>OTDSH</b> (short)	<b>CNC → PLC</b>	Valore della F.A. T o D o S (0 ÷ 4294967295). Con NEWT = 1, il valore riguarda la T (Tool) Con NEWD = 1, il valore riguarda la D (correttore tool) Con NEWS = 1, il valore riguarda la S (Speed) OTDSL è la word leggera di OTDS OTDSH è la word pesante di OTDS
<b>NEWT</b>	<b>CNC → PLC</b>	Nuova F.A. T. (Tool)
<b>NEWD {F}</b>	<b>CNC → PLC</b>	Nuova F.A. D. (correttore tool)
<b>NEWS</b>	<b>CNC → PLC</b>	Nuova F.A. S. (Speed)
<b>FATTOT</b>	<b>PLC → CNC</b>	Fatto della Nuova F.A. T. Posto a 1 comunica al CNC che la T è stata acquisita. Il CNC può proseguire con la prossima istruzione.
<b>FATTOD {F}</b>	<b>PLC → CNC</b>	Fatto della Nuova F.A. D. Posto a 1 comunica al CNC che la D è stata acquisita. Il CNC può proseguire con la prossima istruzione.
<b>FATTOS</b>	<b>PLC → CNC</b>	Fatto della Nuova F.A. S. Posto a 1 comunica al CNC che la S è stata acquisita. Il CNC può proseguire con la prossima istruzione.
<b>NEWSAT</b>	<b>CNC → PLC</b>	Nuova istruzione <SAT: . . > (Selezione apparato) Per maggiori dettagli vedere il paragrafo "Selezione apparato"
<b>OCODSAT</b> (unsigned char)	<b>CNC → PLC</b>	Valore dell'ultima istruzione <SAT:..> programmata per la selezione dell'apparato. 0 = <SAT:OFF> (G630) Apparato FRESA 1 = <SAT:P> (G631) Apparato TORNIO Assi Principali 2 = <SAT:S> (G632) Apparato TORNIO Assi Secondari Il valore è aggiornato con NEWSAT = 1.
<b>FATSAT</b>	<b>PLC → CNC</b>	Fatto della Nuova istruzione <SAT: . . >. Posto a 1 comunica al CNC che l'istruzione è stata eseguita. Il CNC può proseguire con la prossima istruzione.



<b>IAPPATT</b> (unsigned char)	<b>PLC → CNC</b>	Apparato attualmente attivo. 0 = Apparato FRESA 1 = Apparato TORNIO Assi Principali 2 = Apparato TORNIO Assi Secondari
<b>NEWSMN</b>	<b>CNC → PLC</b>	Nuova istruzione <SMN: . . > (Selezione Mandrino) <b>Per maggiori dettagli vedere il Capitolo 5 “Mandrino”</b>
<b>OCODSMN</b> (unsigned char)	<b>CNC → PLC</b>	Valore dell'ultima istruzione <SMN:..> programmata per la selezione del Mandrino. 0 = <SMN:P> (G633) Mandrino Principale 1 = <SMN:S> (G634) Mandrino Secondario 2 = <SMN:U> (G635) Utensile Rotante Il valore è aggiornato con NEWSMN = 1.
<b>FATSMN</b>	<b>PLC → CNC</b>	Fatto della Nuova istruzione <SMN: . . >. Posto a 1 comunica al CNC che l'istruzione è stata eseguita. Il CNC può proseguire con la prossima istruzione.
<b>IMNDATT</b> (unsigned char)	<b>PLC → CNC</b>	Mandrino attualmente attivo. 0 = Mandrino Principale 1 = Mandrino Secondario 2 = Utensile Rotante
<b>NEWGEN</b>	<b>CNC → PLC</b>	Nuovo parametro generico.
<b>OCGEN</b> (unsigned char)	<b>CNC → PLC</b>	Codice del parametro generico (0..255). Il valore è valido con NEWGEN = 1. Il valore del parametro è presente su OVGEN Per il significato del codice vedere l'apposita tabella più avanti.
<b>OVGENF</b> (double) <b>OVGEN</b> (long) <b>OVGENL</b> (short) <b>OVGENH</b> (short)	<b>CNC → PLC</b>	Valore del parametro generico Su OCGEN c'è il codice che specifica il tipo di dato. OVGENF è il valore float in doppia precisione OVGEN è il valore intero (0 ÷ 4294967295). OVGENL è la word leggera di OVGEN OVGENH è la word pesante di OVGEN



## 1.2.6 Codici del Parametro Generico

Il valore presente su OCGEN quando è NEWGEN = 1, definisce il tipo di dato presente su OVGEN secondo la seguente tabella:

1	Allarme per tasto su ICTS rifiutato
3	Allarme KEY totale o del singolo record, stack del PATT pieno; KEY insequibile
4	Allarme KEY totale o del singolo record, allarme generico
16	Allarme riferito ad asse di ordine 0
÷	Per gli altri assi sommare numero asse a questo codice (asse n 11 = codice 27)
27	Su OVGENL c'è il codice attenzione Su OVGENH c'è il codice allarme.
99	Contiene il valore impostato con istruzione <SET:nnn> I valori da 0 a 29999 sono usati per sincronizzazioni tra programmi LIP e PLC. I valori da 0 a 1000 sono riservati a programmi di fornitura ECS. I valori da 30000 a 32767 sono riservati per colloqui particolari fra programmi LIP ed altri task
100	Contiene il dato dell'istruzione send <SND:cod;td;dato> Da 100 a 255 identifica vari tipi di SND Da 100 a 199 richiesti dal PLC. Da 200 a 255 emessi da programmi LIP
200	Codice della send inviata con istruzione <LNT:..> o <DLN:...> o {F}<DRL:...> o {T}<DLU:...> o {T}<DRG:...> per valore del Tool interessato
201	Come 200 per valore lunghezza utensile ({T} su asse X)
205	{T} Come 200 per valore lunghezza utensile su asse Z
202	Come 200 per valore raggio utensile
203	Codice fisso della send inviata da CNC a TCI con istruzione <ZTL>.
204	Emessa F.A. G. Vedi paragrafo 1.2.6.1 a fine tabella Il valore della G è su OVGENL ; OVGENH = 0

### 1.2.6.1 set\_G: Emissione delle funzioni G

Dalla release software V4.00 è possibile definire se e quali funzioni G debbono essere "emesse" verso il PLC. Per default, il CNC non emette le funzioni G per non rallentare l'esecuzione del Part Program; il PLC può però usare la funzione set\_G() per modificare questo funzionamento.

Chiamata: **esito = set\_G (numG)**

Parametro di ingresso:

**numG** = (short) Numero della funzione G che deve essere emessa. Sono accettati valori compresi tra 0 e 999. Se **numG = -1**, vengono emesse tutte le G da G14 a G999 esclusa la G58 (come su release software precedenti).

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- 0** Comando accettato.
- 1** Comando non accettato: valore non valido.

Volendo attivare l'emissione di alcune G, il PLC deve chiamare, ad esempio al pon, la funzione set\_G() più volte con i valori desiderati. La selezione rimane attiva fino ad OFF/ON del CNC o al prossimo passaggio ALT->EXE.



### 1.2.7 SINCRONIZZAZIONE DELLE F.A.

Descriviamo qui di seguito la modalità di emissione e sincronizzazione tra il CNC ed il PLC per quanto riguarda le F.A. . Il CNC emette una F.A. quando questa è NUOVA e cioè quando questa è letta nel programma e quindi non necessariamente diversa da quella emessa precedentemente.

Quando il CNC vuole emettere una nuova F.A., ne scrive il codice all'indirizzo relativo sulle parole di uscita viste al paragrafo 1.2.5

Contemporaneamente setta il NEWxx corrispondente, per segnalare che la F.A. è NUOVA.

Il PLC può, a questo punto, porre a zero il segnale di "fatto" della stessa F.A. (FATTOxx) costringendo il CNC ad attendere l'esecuzione di quanto richiesto dalla funzione ausiliaria. Da questo momento il CNC non emette più nessuna F.A. e termina l'eventuale blocco di programma in corso (se questo non prevede emissioni di F.A.)

Ad esecuzione conclusa, il PLC rimetterà a 1 il segnale di "fatto" ed il CNC potrà riprendere ad elaborare ed eseguire i blocchi di programma successivi o emettere altre F.A. appartenenti allo stesso record.

I "CODICI" ed i segnali di "NUOVA FA" vengono emessi anche nel modo lettura. E' a cura del PLC decidere se/quando ignorare le nuove F.A., se eseguirle o solo memorizzarle; il PLC deve dare in ogni caso il consenso di proseguire al CNC, tramite i segnali di "fatto" FATTOxx.

### 1.2.8 Riposizionamento delle F.A.

Come più ampiamente spiegato nel MANUALE TARATURE 720P385 (file FA.TAB), il CNC garantisce il riposizionamento delle funzioni ausiliarie dichiarate riposizionabili, se emesse in lettura/ricerca. Ovviamente ciò è valido solo per funzioni aventi codici mutuamente esclusivi.

Il riposizionamento è possibile solo se l'organo fisico da riposizionare è gestito da almeno due funzioni ausiliarie.

Esempio classico può essere una tavola indexata a 4 posizioni tramite 4 F.A.; la posizione ultima letta viene riposizionata solo se è diversa dall'ultima eseguita.

Funzioni ausiliarie che devono essere riposizionate ma che sono comandate con un singolo codice, vanno gestite dal PLC memorizzandone l'emissione in lettura o ricerca tramite flag e attivando l'azione fisica quando il CNC ripassa in esecuzione.

Esempio classico può essere il refrigerante. Resettando il CNC, normalmente il PLC ne annulla l'uscita; incontrando poi la funzione ausiliaria in lettura o ricerca, non ne viene richiesto il riposizionamento in quanto l'ultima funzione eseguita era uguale a quella letta.

È possibile [da rel. SW V2.00] forzare il riposizionamento anche quando l'ultima funzione eseguita era uguale a quella letta per ultima; è sufficiente porre **RIP='A'** (Always), anziché **RIP='Y'**, nelle righe corrispondenti del file FA.TAB .



---

### 1.3 F.A. SUBROUTINE

Le Funzioni Ausiliarie (M, T, S, D) definite nel file FA.TAB come ANTICIPATE o POSTICIPATE vengono interpretate dal CNC come F.A. SUBROUTINE (FASUB), attivando la chiamata ai relativi file (Mxx, TOOL, SPEED, DCOR).

Le F.A. definite come INTERMEDIE non richiamano i file sopra elencati, ma vengono inoltrate immediatamente al PLC per la loro esecuzione.

Una F.A. definita anticipata o posticipata e quindi FASUB, se programmata con il carattere "\$" prima della F.A. stessa, diventa intermedia e questo consente di emettere immediatamente il codice della funzione verso il PLC, senza richiamare il file SUBROUTINE.

Come scritto nel paragrafo precedente, una F.A. che deve essere riposizionata deve emettere il suo codice verso il PLC anche in lettura o ricerca. Per questo, l'istruzione contenente \$... (codice della F.A.) contenuta in una FASUB riposizionabile deve essere sempre eseguita. Per saltare eventualmente le altre istruzioni non eseguibili in lettura o ricerca, occorre testare nella subroutine la variabile di sistema dove è riportato lo stato del CNC (esecuzione o lettura/ricerca); ad es. l'istruzione:

```
N260 <IFF<%V1201.6>:N270;N1000;N270> { in lettura?:NO;SI;NO }
```

provoca il salto alla N1000 se il CNC è in lettura o ricerca, altrimenti si passa alla N270

Si ricorda che tutte le variabili di sistema accessibili sono elencate nel file VARCNC.DEF consultabile attraverso gli appositi tasti funzione presenti nel menu del CNC.

### 1.4 SECONDA E TERZA F.A. MISCELLANEA

La SECONDA F.A. "H" e la TERZA F.A. "E" sono utilizzabili normalmente come F.A. con la possibilità di renderle ANTICIPATE e/o POSTICIPATE attraverso la dichiarazione nel file FA.TAB. Il loro nome non deve corrispondere alla sigla nessun asse del CNC se nel campo NAM=...del file AXS.TAR non è stato aggiunto il valore opzionale ";2" oppure ";3".

Nel caso siano dichiarate ANTICIPATE e/o POSTICIPATE in FA.TAB la SECONDA F.A. richiamerà il programma SECFA e la terza F.A. il programma THIFA. All'interno dei programmi SECFA e/o THIFA il valore della F.A. stessa è accessibile sulla variabile %%V7005.

Per altri chiarimenti sull'uso della SECONDA e TERZA F.A. usate come sigle assi consultare il MANUALE TARATURE 720P385, files AXS.TAR, GEN.TAR, SECFA e THIFA.

#### 1.4.1 Assi abbinati alla 2<sup>a</sup> e/o ALLA 3<sup>a</sup> F.A.

Questi assi (normalmente testine e/o tavole con dentatura HIRT) sono assi del CNC che devono essere posizionati solo su quote multiple di un dato valore detto anche passo (in genere 0,5 gradi, oppure 1, 2, 5 gradi, ecc..).

Per associare un asse alla 2<sup>a</sup> o 3<sup>a</sup> funzione ausiliaria si deve aggiungere il numero identificativo di seguito alla sigla dell'asse interessato, nel parametro NAM contenuto nel file AXS.TAR come nel seguente esempio:

**GEN: ORD=5 NAM=B;2**

dove:

**B** : nome asse





**2** : indica ASSE con dentatura HIRT abbinato alla SECONDA F.A. (3 per TERZA F.A.)

Lo stesso nome-asse **B** deve essere definito nel parametro **FA23** contenuto nel file GEN.TAR, sostituendolo al/ai caratteri di default che sono H ed E come segue:

**FA23=B;E**

La SECONDA F.A. ha quindi nome uguale all'asse con dentatura Hirt.

Occorre poi definire, nel file FA.TAB, la F.A. **B** (sigla usata nell'esempio), come F.A. anticipata o posticipata; così facendo il CNC richiamerà il sottoprogramma SECFA in seguito alla programmazione di un movimento dell'asse B.

In SECFA si deve fare il controllo della quota programmata (il valore della quota è scritto nella variabile V7005) e successivamente, se OK, eseguire il record contenente il comando di movimento alla quota programmata:

**N.. \$B<V7005>** oppure **N..<MVM:B; <V7005>**

Questa istruzione fa eseguire al CNC il movimento dell'asse B.

E' compito della routine SECFA dare un eventuale allarme se il comando è dato con CNC in modo JOG VELOCITÀ o arrotondare la quota ad un multiplo del passo della dentatura Hirt se il comando è dato con il CNC in JOG DI SPAZIO.

E' parimenti compito della routine SECFA, in MDI e/o in AUTOMATICO, arrotondare eventualmente la quota al valore ammesso più prossimo oppure dare all'operatore una segnalazione di errata programmazione.

Se l'asse è in libera servo (OLS#=1) e si tenta di muovere in JOG un asse definito FASUB, il CNC dà la diagnostica di "codice tasto rifiutato" sul parametro generico.

Quando la 2<sup>a</sup> o 3<sup>a</sup> funzione ausiliaria è associata ad un asse, il codice relativo non viene emesso verso il PLC (OCODFA2 e/o OCODFA3 e relativi NEWFA2 e/o NEWFA3) e non è perciò usabile come Funzione Ausiliaria. Il movimento dell'asse FASUB non avviene contemporaneamente al movimento di eventuali altri assi scritti nella stessa N... ma avviene prima (F.A. dichiarata anticipata in FA.TAB) o dopo (F.A. dichiarata posticipata in FA.TAB).

Quanto detto per la seconda F.A. è valido anche per la terza F.A. che chiamerà la routine THIFA. Il CNC può gestire, infatti, fino a 2 assi associati ad F.A. miscellanee.

I sottoprogrammi generici SECFA e THIFA contenuti nel CNC sono configurati per l'utilizzo della 2<sup>a</sup> e 3<sup>a</sup> F.A. non associate ad un asse ed indirizzate con le lettere standard H ed E; contengono comunque indicazioni utili per l'uso come asse.

Per l'uso come asse, sostituire eventualmente la sigla H o E con quella desiderata, inserire le istruzioni per effettuare le verifiche necessarie, abilitare l'emissione del comando di posizionamento (la variabile V7005 contiene la quota programmata) e cancellare l'istruzione di emissione F.A. ( \$x<LTI(%%V7005)> ).

La verifica della quota programmata può essere fatta leggendo la variabile V7005.

## 1.5 FUNZIONI SET

Le istruzioni <SET:nnn> sono usate tipicamente nei sottoprogrammi per comunicare informazioni ad altri enti. In particolare quelle con valori da 0 a 29999 sono riservate al dialogo con il PLC; in questo caso viene settato il segnale:

<b>PSET</b>	<b>CNC → PLC</b>	Nuova istruzione SET con valore 0 ÷ 29999. La temporizzazione è uguale a NEWGEN ed il valore è presente su OVGEN.
-------------	------------------	--

Alcune funzioni di corredo su ogni CNC ECS serie WIN sono sviluppate come MACRO, ossia come sottoprogrammi scritti in chiaro usando le istruzioni del linguaggio ISO standard e del linguaggio specifico ECS LIP (per la programmazione indiretta). Per il dialogo con il PLC sono utilizzate alcune istruzioni di tipo <SET:nnn> (con nnn compreso nel range 0 ÷ 999, valori riservati ad ECS) ed alcuni segnali specifici descritti nei paragrafi seguenti.

Dovendo sviluppare MACRO specifiche per una particolare macchina o per un utilizzatore e volendo usare istruzioni SET per la sincronizzazione, il progettista dovrà usare valori compresi nel range 1000 ÷ 29999.

### 1.5.1 G84 (Macro di Maschiatura)

L'inizio e la fine della macro G84 sono segnalati dalle istruzioni:

**INIZIO G84**     <SET:84>

**FINE G84**       <SET:184>

Per il PLC è disponibile il segnale **G84**

**G84=1** : Macro G84 in corso

Con tale segnale le routines standard di libreria PLC **tastcnc( )** per la gestione della tastiera e **gestovr( )** per gli overrides:

- Forzano 100% su override SPEED
- Disabilitano il pulsante HOLD della pulsantiera CNC.

La routine **tastcnc( )**, con G84 in corso, sospende la richiesta di passaggio in esecuzione singola e la rende attiva a fine macro; inoltre accetta il segnale RETG84 per il retract di emergenza previsto dalla Macro G84.

L'inizio e la fine della fase di entrata in G84 sono segnalati dalle istruzioni:

**INIZIO FASE DI ENTRATA**     <SET:211>

**FINE FASE DI ENTRATA**       <SET:210>

Per il PLC è disponibile il segnale **G84ENT**

**G84ENT=1** : Fase di entrata della Macro G84 in corso

Con tale segnale la routine standard di libreria PLC **tastcnc( )** permette l'interruzione della maschiatura in corso tramite il comando **RETG84** (per il retract di emergenza previsto dalla Macro G84) solo durante la fase di entrata.

### 1.5.2 G85 (Macro di alesatura con ritorno in lavoro)

L'inizio e la fine della macro G85 sono segnalati dalle istruzioni:

**INIZIO G85**     <SET:85>

**FINE G85**       <SET:185>

Se il PLC deve rilevare l'esecuzione di questa Macro, può farlo con un'equazione del tipo:

```
G85= PSET && (OVGEN == 85) || G85 && !(PSET && (OVGEN == 185)) && !ORST;  
/* G85 IN CORSO SU CN */
```



### 1.5.3 G86 (Macro di alesatura con arresto del mandrino)

L'inizio e la fine della macro G86 sono segnalati dalle istruzioni:

**INIZIO G86**     <SET:86>  
**FINE G86**       <SET:186>

Il CNC, con questa macro attiva, quando ferma il mandrino non toglie l'emissione del refrigerante, il PLC se lo reputa necessario, può farlo di sua iniziativa.

### 1.5.4 G87 ({F} Macro di alesatura con ritorno a mandrino orientato)

L'inizio e la fine della macro G87 sono segnalati dalle istruzioni:

**INIZIO G87**     <SET:87>  
**FINE G87**       <SET:187>

La macro G87 comanda l'orientamento mandrino tramite il sottoprogramma M19; il PLC non deve impedire la corretta abilitazione del mandrino data dalla M19 tramite la routine di libreria **aggmnd( )**.

### 1.5.5 G184 (Macro di maschiatura rigida)

L'inizio e la fine della macro G184 sono segnalati dalle istruzioni:

**INIZIO G184**   <SET:284>  
**FINE G184**     <SET:280>

Per il PLC è disponibile il segnale **G184**

**G184=1** : Macro G184 in corso

La macro G184 controlla il mandrino come asse CNC, quindi, se necessario, si deve abilitare correttamente l'azionamento mandrino.

### 1.5.6 G201÷211, G614 (Cicli di misura con probe ON-OFF)

L'inizio e la fine dei cicli di misura sono segnalati dalle istruzioni:

**INIZIO**           <SET:200>  
**FINE**             <SET:201>

Per il PLC è disponibile il segnale **TESTON**

**TESTON=1** : Ciclo di misura in corso

Con TESTON=1, il PLC deve permettere movimenti assi in lavoro pur con mandrino fermo ed abilitare il funzionamento del probe.

### 1.5.7 OVER, CERC, ROMBO (Subroutine per prove assi)

L'inizio e la fine di queste subroutine sono segnalati dalle istruzioni:

**INIZIO**           <SET:202>  
**FINE**             <SET:203>

Per il PLC è disponibile il segnale **ABLAVMF**

**ABLAVMF=1** : Prova assi in corso

Con ABLAVMF=1, il PLC deve permettere movimenti assi in lavoro pur con mandrino fermo. Anche la routine **TESTON** (inizializzazioni cicli misura) emette <SET:202> e setta ABLAVMF. La routine **TESTOFF** (chiusura cicli misura) emette <SET:203> e resetta ABLAVMF.



---

### 1.5.8 CGAMMA (Subroutine per cambio gamma)

L'inizio e la fine del cambio gamma sono segnalati dalle istruzioni:

**INIZIO**            <SET:40>  
**FINE**                <SET:140>

La routine standard di libreria PLC **aggmnd( )** per la gestione del mandrino della Fresa o dei mandrini principale e secondario del Torno, usa questi SET per comandare l'abilitazione dell'azionamento relativo.

### 1.5.9 M40 ÷ M45 (Subroutine per cambio gamma)

L'esecuzione di queste subroutines viene segnalata al PLC tramite l'emissione di SET di codice:

**440 + numero gamma**            (ad es. <SET:442> per M42)

Il PLC può utilizzare questi SET per segnalare all'operatore che il cambio gamma automatico eseguito in base alla Speed programmata non è più attivo e/o per predisporre l'azionamento in maniera opportuna. Per i particolari dei cambi gamma vedere il capitolo relativo al MANDRINO.

### 1.5.10 WOOD ({F} macchine per il legno)

Le varie subroutines sviluppate da ECS per la lavorazione guidata di pannelli in legno o materiali simili, usano una serie d'istruzioni <SET:nnn> nel range **300 ÷ 399** per segnalare al PLC quale ciclo di lavoro o fase si sta eseguendo. Tali SET sono letti ed interpretati dalla routine standard di libreria PLC **wood\_plc( )** e quindi non debbono essere usati per altri fini.

## 1.6 TEMPORIZZATORI

Il PLC integrato nei CNC ECS della serie WIN mette a disposizione del progettista la possibilità di associare dei temporizzatori a qualsiasi segnale (ingressi e uscite da/verso il CNC o la Macchina Utensile, segnali interni, "contatti" di temporizzati, ecc.)

Ogni temporizzatore è dotato di un segnale pilota (tipo la bobina di un relè) e di più uscite "funzione del tempo" (tipo contatti di relè). Inoltre ogni temporizzatore è dotato di una funzione di "settaggio" del tempo di ritardo.

La transizione **da 0 a 1** o **da 1 a 0** del segnale pilota attiva il ritardo impostato e provoca la commutazione dei contatti associati con le regole che sono date più avanti.

I temporizzatori disponibili sono 128 di cui alcuni riservati alle routines standard fornite di base da ECS; il progettista del PLC può usare i temporizzatori da 12 a 89 e da 96 a 127 (dunque in totale 109 timers).



### 1.6.1 Sinonimi per i temporizzatori

Per la gestione di segnali temporizzati nel PLC, sono presenti i sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C.

<b>TIME_nn</b>	<b>Segnale pilota</b> del temporizzatore <b>nn</b> . nn è valido nel campo 00 ÷ 127 .
<b>RITP_nn</b>	<b>Ritardo POSITIVO.</b> Contatto ritardato all'eccitazione del TIME_nn corrispondente. <b>RITP_nn = 1</b> dopo che il segnale pilota rimane a 1 per un tempo uguale a quello impostato. <b>RITP_nn = 0</b> nel giro di PLC seguente a quello in cui il segnale pilota ritorna allo stato 0 (nessun ritardo alla diseccitazione).
<b>RITN_nn</b>	<b>Ritardo NEGATIVO.</b> Contatto ritardato alla diseccitazione del TIME_nn <b>RITN_nn = 1</b> nel giro di PLC seguente a quello in cui il segnale pilota passa allo stato 1 (nessun ritardo all'eccitazione). <b>RITN_nn = 0</b> dopo che il segnale pilota rimane a 0 per un tempo uguale a quello impostato.
<b>DERP_nn</b>	<b>DERIVATA POSITIVA</b> Contatto "derivato" sul fronte positivo di TIME_nn. Il contatto rimane a 1, <u>per un giro del PLC</u> , al passaggio <b>0⇒1</b> del segnale pilota corrispondente
<b>DERN_nn</b>	<b>DERIVATA NEGATIVA</b> Contatto "derivato" sul fronte negativo di TIME_nn. Il contatto rimane a 1, <u>per un giro del PLC</u> , al passaggio <b>1⇒0</b> del segnale pilota corrispondente

### 1.6.2 lotime: funzione per definire il tempo di un ritardo

Chiamata: **lotime (ntmp,time)**

Parametri di ingresso:

**ntmp** = (unsigned short) Numero d'ordine del temporizzatore; sono validi i valori compresi tra 0 e 127.

**time** = (double) Tempo, in secondi, da caricare nel temporizzatore **ntmp**.

Un temporizzatore è normalmente settato una sola volta al **pon (power-on)**; nulla vieta però che il tempo sia modificato dinamicamente richiamando la funzione con valori diversi secondo le necessità della logica di macchina. In questo secondo caso, il nuovo tempo sarà acquisito alla prossima variazione **0⇒1** oppure **1⇒0** del segnale pilota.

Nel caso che di un temporizzatore siano usati solo i contatti derivati, non occorre chiamare la funzione di **lotime(...)** anche se è preferibile farlo comunque con un tempo di 0.0 secondi per renderne esplicito l'uso.

```
/*-----*/
/* ESEMPIO D'USO DI TEMPORIZZATORI */
/*-----*/
if(pon)
{
  lotime (0,0.3);
  /* Setta tempo di caduta ABIL0 in M10 (0.3 SEC) */
  lotime(13,1);
  /* Segnale pilota = MUON : MACCHINA ACCESA */
  lotime(14,5);
  /* Segnale pilota = AUXON : AUSILIARI INSERITI */
  lotime(15,8);
  /* Segnale pilota = AUXON : AUSILIARI INSERITI */
  lotime(18,0.5);
  /* Segnale pilota = INTERM : INTERMITTENZA */
  lotime(50,0.0);
  /* Segnale pilota = ABMAN : ABILITAZIONE MANUALE */
}

TIME_13= MUON;
TIME_14= MUON;
TIME_15= MUON;
  /* Vari contatti ritardati all'accensione M.U. */

TIME_18= !INTERM ;
INTERM= !INTERM && RITP_18 || INTERM && RITN_18;
  /* Bistabile per lampade a intermittenza */

TIME_50= ABMAN ;
  /* Per derivata su caduta ABMAN */

if(DERN_50)
{
  RICVOL= ARICVOL= 0;
  /* Sgancia Volantini */
  COMAXS= 0 ;
  /* Forza asse X */
}
```

## 1.7 SELEZIONE APPARATO FRESA/TORNIO

Come già detto più sopra, i CNC ECS serie WIN (dalla release Software V3.00) possono gestire **due tipi di macchine** (Fresa e Tornio) con la possibilità di attivare dinamicamente parte delle funzionalità dell'altro tipo di macchina. La configurazione "di default" della macchina è comunicata al PLC tramite i due segnali OSYSMILL (Fresa) ed OSYSTURN (Tornio).

Sul Tornio, è anche possibile definire **due coppie di assi di tornitura** da usare in alternativa; per le necessarie configurazioni e tarature, vedere quanto scritto per il campo ATN del file AXS.TAR sul MANUALE TARATURE 720P385.

La selezione delle funzionalità (**apparato e assi attivi**) è fatta tramite le istruzioni G630, G631, G632.

**G630** Richiesta attivazione apparato Fresa.



Comunicato al PLC con **OCODSAT=0**

**G631** Richiesta attivazione apparato Torno con assi Principali attivi.

Comunicato al PLC con **OCODSAT=1**

**G632** Richiesta attivazione apparato Torno con assi Secondari attivi.

Comunicato al PLC con **OCODSAT=2**

In effetti le funzioni G63x suddette sono delle macro (scritte con le normali regole di programmazione dei Part-Program) e possono essere quindi modificate dal costruttore della macchina secondo le proprie esigenze, mantenendo però le istruzioni già presenti. Al loro interno, queste funzioni richiamano le istruzioni elementari LIP <SAT:...> che eseguono il cambio apparato attivo:

**<SAT:OFF>** Attiva l'apparato Fresa.  
**<SAT:P>** Attiva l'apparato Torno con assi principali di tornitura.  
**<SAT:S>** Attiva l'apparato Torno con assi secondari di tornitura.

e che vengono comunicate al PLC tramite lo strobe **NEWSAT** e con i codici CODSAT visti più sopra.

Il PLC, dopo aver eseguito eventuali predisposizioni necessarie per il nuovo apparato da rendere attivo, deve scrivere in **IAPPATT** il valore opportuno e settare **FATSAT**.

Il PLC può NON modificare le sue predisposizioni e lo stato di IAPPATT in modo Lettura, mettendo comunque FATSAT=1; il CNC riemetterà il comando di selezione apparato al primo passaggio in Esecuzione (se IAPPATT diverso da OCODSAT).

Se non ci sono azioni da eseguire sulla macchina per il cambio di apparato, è sufficiente che il PLC ricopi su IAPPATT il valore presente su OCODSAT:

```
if(NEWSAT)
{
  IAPPATT = OCODSAT;
}
/* Gestione selezione APPARATO. */
```

#### **N.B.:**

1. Le istruzioni G630 e G631 di cambio apparato sono ammesse solo se presente la relativa opzione (OPZ\_FRETOR per la Fresa o OPZ\_TORFRE per il Torno).
2. L'istruzione G632 per la selezione degli assi secondari di tornitura è possibile solo se l'apparato "di default" è un Torno (OSYSTURN=1)
3. Le istruzioni G630, G631, G632, hanno effetto anche con CNC in modo "lettura"
4. Il carattere "%" come pure ":" ripristinano l'apparato attivo sulla M.U. (IAPPATT).
5. All'accensione il CNC attiva l'apparato "di default". Qualora sia richiesto che il CNC si accenda con lo stesso apparato selezionato allo spegnimento, è necessario che il PLC scriva l'apposito codice nella variabile IAPPATT.
6. Un Sottoprogramma può testare l'apparato con:  
N230 <IFF<%V1912 & 'OCT02000'>:N235;N250;N235>  
{ CNC TORNIO ? Si;No;Si }
7. Un Sottoprogramma può testare l'opzione FRETOR o TORFRE con:  
N235 <IFF<%V1912 & 'OCT04000'>:N240;N250;N240>  
{ OPZ\_TORFRE ? Si;No;Si }



## 1.8 COMPILAZIONE CONDIZIONALE

Dalla release di SW 3.02, è possibile eseguire delle compilazioni condizionate dalla release del CNC utilizzando il define CNREL.

<b>CNREL</b> (word)	Definisce la <b>release</b> SW del CNC. Simbolo utilizzabile per compilazione condizionale.
------------------------	--

La variabile CNREL identifica la versione attuale del SW CNC con la seguente logica:

-	CNREL	=	3020	identifica	la	Versione	V3.02
-	CNREL	=	3021	identifica	la	Versione	V3.02A
-	CNREL	=	3022	identifica	la	Versione	V3.02B
-	CNREL	=	4000	identifica	la	Versione	V4.00

E' quindi possibile diversificare il codice sorgente del PLC in base alla Release applicando la compilazione condizionale:

```
#if CNREL >= 4000
.
. Codice per Release uguali o superiori alla V4.00
.
#else
.
. Codice per Release inferiori alla V4.00
.
#endif

#if CNREL == 3020
.
. Codice per Release uguale alla V3.02
.
#endif

#ifndef CNREL
.
. Codice per Release inferiore V3.02
. (non essendo definita la variabile CNREL)
.
#endif
```

Si ricorda che CNREL non è un vero e proprio sinonimo del PLC, pertanto volendo visualizzarne il valore occorre scrivere:

```
CNREL_AP= CNREL ;
```

con CNREL\_AP definito come word sull'area A.

E' comunque lecito scrivere:

```
if (CNREL==3020)
{
....
....
....
}
```





---

---

## CAPITOLO 2

### 2 ASSI

In questo capitolo vengono trattati i segnali e le funzioni che riguardano la "normale" gestione assi realizzata sui CNC ECS della serie WIN.

Alcune funzioni "speciali" o trattamenti particolari non sono qui descritti oppure sono solo accennati; per il loro approfondimento si dovrà fare riferimento ai capitoli successivi.

#### 2.1 GENERALITÀ

I CNC ECS serie WIN possono gestire (dalla release software V4.00) fino a 16 assi. Ogni asse, oltre ad una lettera che lo distingue, è caratterizzato da un numero d'ordine; tale numero, secondo il tipo di CNC, è compreso tra:

Numero	Fresa	Tornio
<b>0 ... 2</b>	<b>1801</b>	<b>1802</b>
<b>0 ... 5</b>	<b>2801</b>	
<b>0 ... 15</b>	<b>4801</b>	<b>4802</b>

Il sistema di misura di ciascun asse è di tipo ottico (detto anche encoder) e può essere incrementale o assolutizzato. Sono ammessi anche encoder simulati, generati dall'azionamento dell'asse elaborando opportunamente il segnale del trasduttore sul motore.

Uno degli assi può essere il Mandrino Principale, che può essere dotato di trasduttore di misura oppure esserne sprovvisto. Può essere presente anche un Mandrino Secondario, con tutte le caratteristiche del Mandrino Principale.

Sono disponibili 12 convertitori Digitali/Analogici (D/A) che sono condivisi con il PLC; tramite opportune tarature definite nel file SWCNF.TAR (vedi MANUALE TARATURE 720P385), si può decidere l'appartenenza del singolo convertitore al CNC od al PLC. Se utilizzati dal PLC, sono indirizzati come uscite VDA0, VDA1, . . . ,VDA11 .

Ogni convertitore può fornire una tensione continua tra -10 e +10 Volt in corrispondenza dei valori numerici compresi tra -32768 e +32767.

##### 2.1.1 Parametri di taratura degli assi

Le caratteristiche di ciascun asse sono definite negli appositi files di taratura ed in particolare in AXS.TAR (vedi MANUALE TARATURE 720P385). Il numero d'ordine contenuto in questo file (abbinato a quanto scritto in SWCNF.TAR) definisce la corrispondenza tra tarature software e circuiteria hardware.

Qui di seguito, sono riassunti i vari files che contengono tarature riguardanti gli assi.

SWCNF.TAR	Numero assi, configurazione Hardware.
AXS.TAR	Parametri assi
GEN.TAR	Tarature generali
COST	Parametri vari
COSTUSER	Personalizzazione dei parametri di COST
SETxx, SETAL, RIGABS, ENCABSL, ENCABST	Procedure di SET (presa di zero)
M10, M11	Bloccaggio/bloccaggio assi.



SECFA, THIFA	Per assi gestiti come Funzioni Ausiliarie.
COR.TAR	Correzione errore assi
VIDAX.TAR	Assi da visualizzare in ambiente grafico

La funzione di questi files ed il relativo contenuto sono dettagliatamente spiegati nel MANUALE TARATURE 720P385.

## 2.2 ASSI INTERPOLANTI (CNC)

Si definisce "Asse Interpolante" un asse il cui movimento ed il raggiungimento della quota programmata avvengono in maniera coordinata e sincrona con gli altri assi. Un asse interpolante è talvolta definito **controllato in Contouring** o **Asse CNC**.

Un asse è configurato "Interpolante" in taratura (attraverso il file AXS.TAR) o reso tale da programma (attraverso l'istruzione <CNC: nome\_asse>).

Un asse definito "interpolante" può essere reso "indipendente" da programma (attraverso l'istruzione <FRE: nome\_asse>); per la relativa gestione, si rimanda a quanto scritto nel capitolo ASSI INDIPENDENTI (FRE).

### 2.2.1 Sinonimi utilizzati per gli ASSI

Per la gestione di ciascun asse tramite il PLC, sono presenti vari segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Sono presenti anche altri segnali, con i relativi sinonimi, per gestioni particolari (assi indipendenti, gantry, pendolanti, ecc.); saranno visti nei capitoli successivi. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

**{W}** Nella prima colonna, il sinonimo scritto in **GRASSETTO** è quello da preferire; l'eventuale altro sinonimo è comunque valido ed è stato mantenuto per compatibilità con la serie D.

Quando si parla del Mandrino ci si riferisce sia al mandrino principale sia secondario; quando ci siano differenze, queste saranno indicate esplicitamente.

Nei sinonimi dove compare il carattere "#", occorre sostituire ad esso il numero d'ordine dell'asse al quale il segnale è riferito. Tale numero, secondo il tipo di CNC, sarà compreso tra 0 e 2 (1801, 1802), tra 0 e 5 (2801) o tra 0 e **15** (4801, 4802).

Esempi:

- IEME0** richiesta di emergenza relativa all'asse di ordine 0.
- IEME1** richiesta di emergenza relativa all'asse di ordine 1.
- IEME5** richiesta di emergenza relativa all'asse di ordine 5.

<b>AXESNUM</b> (short)	<b>CNC → PLC</b>	Numero di assi configurati (vedi SWCNF.TAR)
<b>IDRVON#</b>	<b>PLC → CNC</b>	Drive ON. Posto a 1 segnala al CNC che l'azionamento dell'asse # è acceso, attivo e pronto a comandare il relativo motore. Se l'asse è Digitale, l'azionamento relativo viene abilitato. Posto a 0 impedisce che l'asse relativo venga mosso. Se l'asse è Digitale, l'azionamento relativo viene disabilitato



<b>IBLAXMND</b> IBL	<b>PLC → CNC</b>	Blocco CNC. Posto a 1 impone velocità 0 a tutti gli assi CNC ed al mandrino ed impedisce il proseguimento del Part Program. Posto poi a 0, permette la ripartenza di assi, mandrino e Part Program, che avverrà dopo la pressione del tasto REL.
<b>IBLAX</b>	<b>PLC → CNC</b>	Blocco assi. Posto a 1 impone velocità 0 a tutti gli assi CNC ma non al mandrino ed impedisce il proseguimento del Part Program. Posto poi a 0, permette la ripartenza di assi e Part Program, che avverrà dopo la pressione del tasto REL.
<b>IFEEDHLD</b> IFHD	<b>PLC → CNC</b>	Feed-hold assi CNC. Posto a 1 impone velocità 0 a tutti gli assi CNC; non interviene su assi FRE. Gli assi si fermano senza uscire dalla traiettoria programmata. E' possibile fermare gli assi (IFEEDHLD=1) in qualsiasi momento. Posto poi a 0, permette l'immediata ripartenza degli assi.
<b>IRAP</b>	<b>PLC → CNC</b>	Posto a 1 forza velocità ed Override Rapido sugli ASSI CNC in movimento. L'azione del segnale IRAP deve essere abilitata tramite l'apposita variabile di sistema prevista nel sottoprogramma COST.
<b>OBAT#</b>	<b>CNC → PLC</b>	<b>OBAT#= 1</b> Attenzione su asse
<b>OBOK#</b>	<b>CNC → PLC</b>	<b>OBOK# =1</b> Nessun allarme grave sull'asse. <b>OBOK# =0</b> Asse in emergenza.
<b>OBALAX#</b>	<b>CNC → PLC</b>	<b>OBALAX# =1</b> Asse in allarme; se l'allarme è grave, viene anche messo <b>OBOK# =0</b> .
<b>IEME#</b>	<b>PLC → CNC</b>	Posto a 1 richiede di porre l'asse in emergenza. Nel passare in emergenza l'asse viene istantaneamente disabilitato senza alcuna decelerazione controllata.
<b>ISGA#</b>	<b>PLC → CNC</b>	Posto a 1 richiede di sganciare l'asse in fase di Reset. Questo segnale è preso in considerazione dal CNC solo dopo che stata attivata una richiesta di Reset (pulsante [RESET] o IRES=1). Il CNC durante la fase di Reset sgancia e poi riaggancia sulla quota attuale gli assi che hanno <b>ISGA#=1</b> , in modo da annullare l'eventuale errore d'inseguimento accumulato. Si consiglia l'uso del segnale <b>ISGA#</b> quando si fermano gli assi con mezzi esterni (non tramite il CNC) tenendo presente che: Deve essere il PLC con i suoi mezzi a fermare l'asse il più rapidamente possibile. Deve essere il PLC a mantenere il CNC in blocco ( <b>IBLAXMND=1</b> ) sino al Reset. Il segnale <b>ISGA#</b> deve essere mantenuto a 1 sino alla caduta di <b>OBAG#</b> (asse agganciato). In questo modo si evita che il CNC recuperi di colpo l'errore di inseguimento, nel caso non sia intervenuto l'allarme di fuga.



<b>OBLC#</b>	<b>CNC → PLC</b>	<p>Posto = 1 significa che l'asse è bloccabile meccanicamente dal PLC. Perché ciò avvenga l'asse deve essere stato precedentemente dichiarato bloccabile (tipicamente in <b>COSTUSER</b>) tramite l'istruzione <b>&lt;BLC:sigla_asse&gt;</b>.</p> <p>Sequenza di operazioni in caso di movimentazione di un asse dichiarato bloccabile meccanicamente:</p> <p>-Avvio movimento: Il CNC pone il segnale <b>OBLC#=0</b> a segnalare di sbloccare l'asse. Il CNC attende che il PLC ponga il segnale <b>IBLC#=0</b> a segnalare asse sbloccato. Se il PLC usa la routine <b>aggax()</b>, deve essere usato <b>BLCAX#</b>.</p> <p>-Arresto movimento: Con asse arrivato in posizione, il CNC pone il segnale <b>OBLC#=1</b> (bloccare l'asse) e passa all'istruzione successiva senza attendere che l'asse sia effettivamente bloccato. Se non si vogliono muovere gli assi con un asse in fase di bloccaggio, è necessario imporre Feed-Hold (<b>IFEEDHLD=1</b>).</p>
<b>IBLC# BLCAX#</b>	<b>PLC → CNC</b>	<p>Posto a 1, segnala asse bloccato meccanicamente.</p> <p>Per tutti gli assi CNC, il CNC attende, prima di iniziare il movimento, che gli assi da muovere abbiano tutti IBLC=0. Durante il movimento, se per uno degli assi in moto diventa IBLC=1, il CNC ferma tutti gli assi (come FEED-HOLD) senza uscire dalla traiettoria.</p> <p>Se il PLC usa la routine <b>aggax()</b>, deve essere usato <b>BLCAX#</b>.</p>
<b>ILS# RILS#</b>	<b>PLC → CNC</b>	<p>Posto a 1 = Richiesta di Libera Servo. Asse sganciato (non più controllato in spazio dal CNC).</p> <p>Posto a 0 = Asse di proprietà del CNC che lo controlla in spazio in base a come è stato configurato.</p> <p>Tale segnale deve essere utilizzato tutte le volte che il CNC deve trattare assi destinati alla movimentazione di dispositivi (Testine o Tavole), che devono poter essere anche rimossi (sconnessi dal CNC).</p> <p>Il CNC riscontrando una variazione 0→1 su un segnale <b>ILS#</b> procede come segue: Non interpreta più blocchi in avanti e si ferma. Attende che l'interpolatore termini di eseguire quello che era stato interpretato Analizza se la richiesta può essere onorata eventualmente segnalando la ragione del rifiuto al PLC Sgancia l'asse ponendo <b>OBAG#=0</b> e <b>OLS#=1</b> (Vedi oltre) Riprende ad interpretare il Part Program. L'asse può ora essere mosso da PLC attraverso riferimenti esterni. Se il PLC usa la routine <b>aggax()</b>, deve essere usato <b>RILS#</b>. La Richiesta di Libera Servo non deve essere rivolta ad un asse con <b>IAXOFF#=1</b> che non potrà settare <b>OLS#</b>.</p>
<b>OLS#</b>	<b>CNC → PLC</b>	<p>E' la risposta del CNC alla richiesta del PLC di Libera Servo attuata tramite <b>ILS#=1</b>. <b>OLS#=0</b> Asse gestito dal CNC <b>OLS#=1</b> Asse abbandonato dal CNC e sganciato (Segnale <b>OBAG=0</b>).</p>



<b>IAXOFF#</b>	<b>PLC → CNC</b>	<p>Posto a 1 = Richiesta di Asse OFF. Asse sganciato (non più controllato in spazio dal CNC).</p> <p>Posto a 0 = Asse di proprietà del CNC che lo controlla in spazio in base a come è stato configurato.</p> <p>Tale segnale può essere utilizzato (in alternativa a ILS#) tutte le volte che sia necessario spegnere l'azionamento e/o sconnettere il motore di un asse del CNC, continuando a lavorare regolarmente con gli altri assi.</p> <p>Il CNC, riscontrando una variazione 0→1 su un segnale <b>IAXOFF#</b>, verifica se l'asse relativo è fermo e poi sgancia l'asse ponendo <b>OBAG#=0</b> e <b>OAXOFF#=1</b>.</p> <p>Se l'asse è Digitale, l'azionamento relativo viene disabilitato. L'asse in stato di OFF può ora essere mosso attraverso riferimenti esterni o spento e scollegato. Per evitare l'intervento di allarmi, occorre mettere <b>INAL#=1</b> ed eventualmente <b>IDGOFF#=1</b> (vedi più avanti).</p>
<b>OAXOFF#</b>	<b>CNC → PLC</b>	<p>E' la risposta del CNC alla richiesta del PLC di Asse OFF attuata tramite <b>IAXOFF#=1</b>.</p> <p><b>OAXOFF#=0</b> Asse gestito dal CNC</p> <p><b>OAXOFF#=1</b> Richiesta asse OFF accettata. Per verificare che l'asse sia effettivamente abbandonato dal CNC e sganciato controllare anche il segnale <b>OBAG=0</b> (<b>OBVV=0</b> se mandrino).</p>
<b>INAL#</b>	<b>PLC → CNC</b>	<p><b>INAL#=1</b> sospende la segnalazione di Allarme e l'Hold CNC, anche se l'asse è in allarme per mancanza trasduttore o sovraconteggio. E' valido solo per gli assi attualmente FRE o con <b>ILS#=1</b> o con <b>IAXOFF#=1</b>.</p> <p>Il CNC non va in Hold anche quando l'allarme riguarda un asse dichiarato indipendente in <b>AXS.TAR</b> e con il campo ALL=N.</p> <p>Questo segnale è da utilizzare, assieme a <b>ILS#</b> e <b>IAXOFF#</b> per assi in cui viene staccato temporaneamente il trasduttore.</p> <p>Ricollegando il trasduttore prima di togliere <b>INAL#</b> e <b>ILS#</b> o <b>IAXOFF#</b> l'asse sarà ripreso in carico dal CNC senza segnalazione di allarmi; ovviamente, se c'era stata una situazione di allarme, il SET è stato perso (<b>OBSE#=0</b>) ed occorre rifarlo.</p>
<b>OBM0#</b>	<b>CNC → PLC</b>	<p><b>OBM0#= 1</b> in presenza di marker del trasduttore ottico.</p>
<b>OBM1#</b>	<b>CNC → PLC</b>	<p><b>OBM1# = 1</b> alla chiusura del micro di zero associato all'asse.</p>
<b>OBSE#</b>	<b>CNC → PLC</b>	<p><b>OBSE# = 1</b> con SET asse eseguito. Se <b>OBSE# = 0</b> è consentito muovere l'asse solo nei modi JOG e SET.</p>
<b>IBSET#</b>	<b>PLC → CNC</b>	<p><b>IBSET# = 1</b> Forza la condizione di SET da fare (<b>OBSE# = 0</b>; <b>OBSET# = 1</b>)</p>
<b>OBSET#</b>	<b>CNC → PLC</b>	<p><b>OBSET# = 1</b> Condizione di SET da fare, forzata da IBSET#.</p>



<b>OBAG#</b>	<b>CNC → PLC</b>	<b>OBAG# = 1</b> Asse agganciato. E' utilizzato per abilitare l'azionamento. Non cade allorché <b>OBOK#</b> va a 0. Con <b>OBAG#=0</b> , il PLC deve mantenere fermi gli assi con mezzi propri (bloccaggi, freni, ecc.).
<b>OBCN#</b>	<b>CNC → PLC</b>	<b>OBCN# =1</b> Asse agganciato in contouring. Nel caso di un asse Indipendente ciò avviene solo in modo JOG.
<b>OBVV#</b>	<b>CNC → PLC</b>	<b>OBVV# =1</b> Asse agganciato in modo velocità. Valido per asse Mandrino.
<b>OBIN#</b>	<b>CNC → PLC</b>	<b>OBIN# = 1</b> sta ad indicare che l'asse si sta muovendo indietro (verso quote decrescenti). Ad asse arrivato il segnale <b>OBIN#</b> mantiene l'ultimo valore assunto.
<b>OBMT#</b>	<b>CNC → PLC</b>	Asse in moto <b>OBMT# = 1</b> se asse in moto ovvero se velocità calcolata diversa da zero. <b>OBMT# = 0</b> se velocità calcolata = 0, anche in seguito a Hold, Feed-Hold, Override = 0, asse bloccato. Per assi in modo velocità (es. Mandrino in rotazione), viene considerata la velocità effettiva se l'asse ha il trasduttore o la velocità calcolata se manca il trasduttore.
<b>OPSGL#</b>	<b>CNC → PLC</b>	Asse entrato in soglia <b>OPSGL#=1</b> se l'asse non è in moto ( <b>OBMT#=0</b> ) e se l'errore è ≤ <b>SGL</b> (Soglia di posizionamento definita in <b>AXS.TAR</b> ). In caso di KV=0 (anello di spazio aperto) <b>OPSGL#</b> assume lo stesso valore del segnale <b>OPFRM#</b> (asse in soglia = asse fermo).
<b>OPFRM#</b>	<b>CNC → PLC</b>	Asse fermo L'asse viene considerato fermo ( <b>OPFRM#=1</b> ) se non è in moto ( <b>OBMT#=0</b> ) e rimane in soglia ( <b>OPSGL=1</b> ) per un tempo di "ntau"*Tau (taratura in <b>AXS.TAR</b> , campo <b>SGL=sgl;ntau</b> ). In caso di KV=0 (anello di spazio aperto) <b>OPFRM#</b> assume il valore "complementato" di quello del segnale <b>OBMT#</b> (asse fermo = asse non in moto).
<b>ONBAGM#</b>	<b>CNC → PLC</b>	Asse non agganciato in movimento (non fermo) Valido quando <b>OBAG# = 0</b> , segnala che l'asse #, pur essendo sganciato, si sta muovendo (la media delle ultime quote varia di una quantità superiore alla soglia di fuga asse fermo definita in <b>AXS.TAR</b> , campo <b>FFS</b> ). In caso di KV=0 (anello di spazio aperto) è sempre = 0. Per il mandrino viene usata la soglia <b>MFR=</b> .
<b>OBNA#</b>	<b>CNC → PLC</b>	Asse non arrivato <b>OBNA# messo a 1</b> con asse in moto ( <b>OBMT#=1</b> ). <b>OBNA# messo a 0</b> quando l'asse è fermo ( <b>OPFRM=1</b> ) ed è arrivato alla quota comandata. Per assi CNC questo risulta essere sempre vero quando: - Non ci sono blocchi di programma in carico all'interpolatore - Non sono stati programmati spostamenti di assi CNC nel blocco attualmente in esecuzione, in carico all'interpolatore.



<b>OASF</b>	<b>CNC → PLC</b>	Assi CNC fermi <b>Messo a 1</b> quando tutti gli assi CNC hanno OBNA# = 0.
<b>OLAV</b>	<b>CNC → PLC</b>	Assi CNC in movimento di lavoro (G01, G02, G03) <b>OLAV = 1</b> almeno un asse CNC viene mosso in lavoro <b>OLAV = 0</b> almeno un asse CNC viene mosso in rapido (G00). A fine movimento, OLAV mantiene l'ultimo stato acquisito. Può essere usato per far intervenire un allarme in caso di programmazione di un movimento di lavoro con mandrino fermo. Per un esempio, vedere più avanti al paragrafo : ESEMPIO D'USO DELLA ROUTINE AGGAX
<b>ABLAVMF</b>	<b>CNC → PLC</b>	Abilitazione movimenti di lavoro a mandrino fermo Settato/resettato dalle routines che fanno movimenti non in rapido senza avviare il mandrino: cicli di misura, procedure test assi, ... Può essere usato per inibire l'eventuale allarme di movimento di lavoro con mandrino fermo visto sopra.
<b>IAXPLC#</b>	<b>PLC → CNC</b>	<b>IAXPLC# = 1</b> Richiesta di gestire l'asse da PLC <b>IAXPLC# = 0</b> Richiesta di riconsegnare l'asse al CNC Per i dettagli di funzionamento e per la relativa gestione, si rimanda a quanto scritto nel capitolo ASSI INDIPENDENTI (FRE).
<b>OAXPLC#</b>	<b>CNC → PLC</b>	<b>OAXPLC# = 1</b> Risposta OK alla richiesta di gestire l'asse da PLC.
<b>IJOGP#</b>	<b>PLC → CNC</b>	<b>IJOGP# = 1</b> Richiesta di JOG+ (positivo) di velocità in Rapido. La richiesta è accettata solo con assi gestiti da PLC (con <b>OAXPLC# = 1</b> ). Per i dettagli di funzionamento di <b>IJOGP#</b> e <b>IJOGN#</b> si rimanda a quanto scritto nel capitolo ASSI INDIPENDENTI (FRE).
<b>IJOGN#</b>	<b>PLC → CNC</b>	<b>IJOGN# = 1</b> Richiesta JOG— (negativo) di velocità in Rapido. La richiesta è accettata solo con assi gestiti da PLC (con <b>OAXPLC# = 1</b> ).
<b>OCPLC#</b> (short)	<b>CNC → PLC</b>	<b>OCPLC#</b> contiene il codice di ritorno dell'ultima richiesta di movimentazione effettuata da PLC. Per i dettagli si rimanda a quanto scritto nel capitolo ASSI INDIPENDENTI (FRE).
<b>ISAJ</b> <b>COMAXS</b> (char)	<b>PLC → CNC</b>	Selezione dell'asse CNC da muovere in JOG - digit Low: = numero asse (0...15) per JOG - digit High: = non usato ISAJ è usato anche per selezionare gli assi da muovere con il volantino 1 e 2 (vedi paragrafo 2.9.2 "Volantini"). Se si usa la routine modijog(), il numero va scritto in COMAXS
<b>IJSR</b>	<b>PLC → CNC</b>	Richiesta JOG di SPAZIO in RAPIDO
<b>IJSPL</b>	<b>PLC → CNC</b>	Richiesta JOG di SPAZIO in LAVORO
<b>IJVL</b>	<b>PLC → CNC</b>	Richiesta JOG di VELOCITÀ in RAPIDO



<b>IJVLL</b>	<b>PLC → CNC</b>	Richiesta JOG di VELOCITÀ in LAVORO
<b>IVJ</b> <b>INJS</b> (long)	<b>PLC → CNC</b>	Valore, in micron (o millesimi di grado), dell'incremento da eseguire in JOG di SPAZIO. IVJ è una doppia word; range: 0...+ 4.294.967.295 Se si usa la routine modijog(), lo spazio va scritto in INJS.
<b>IHLD#</b> <b>HLDFRE#</b>	<b>PLC → CNC</b>	Posto a 1 richiede di mettere in HOLD l'asse. E' imposta velocità 0 con rampa di decelerazione nominale. L'esecuzione del movimento riprenderà allorché viene nuovamente posto IHLD#=0. Se il PLC usa la routine <b>hldfre()</b> , occorre usare il corrispondente segnale <b>HLDFRE#</b> .

Indichiamo qui di seguito la corretta sequenza di abilitazione / disabilitazione, in funzione dell'accensione dei teleruttori di potenza:

Accensione:

- 1) IDRNON#=0
- 2) Accensione teleruttori
- 3) IDRNONn=1

Spegnimento:

- 1) IDRNONn=0
- 2) Spegnimento teleruttori

Ovviamente con IDRNON# = 0, il motore (se privo di freno) è libero di muoversi; muovendo il motore, se l'errore supera la soglia di fuga, il CNC emette il relativo allarme; riabilitando il DRIVE (IDRNON# = 1), l'eventuale errore viene recuperato.

**Il CNC, alla commutazione 0→1 del segnale IDRNON# forza FDF=0 per un tempo di 100ms, al fine di evitare "bruschi" recuperi dell'errore eventualmente accumulato a DRIVE disabilitato.**

## 2.3 AGGANCIAMENTO ASSI E GESTIONE ORIGINI

### 2.3.1 Modo JOG

Tutti gli assi, siano questi CNC o FRE, vengono agganciati in contouring (OBCN#=1). Se l'asse mandrino è in rotazione rimane in rotazione e agganciato in velocità (OBVV#=1); in queste condizioni non si può fare JOG sul mandrino. Se il mandrino è orientato o fermo viene agganciato in contouring e si può muovere in JOG.

I movimenti degli assi possono essere comandati in modo continuo (JOG in VELOCITÀ) oppure ad incrementi (JOG di SPAZIO), alla velocità di RAPIDO o all'ultima feed di LAVORO programmata, tramite i segnali visti al paragrafo 2.2.1 "SINONIMI...".

I movimenti possono essere comandati anche tramite volantini ( vedi più avanti).

La routine di libreria PLC **tasfuecs()**, che gestisce i tasti funzione del PLC nella configurazione standard ECS definita in BASMENU.MEN, in abbinamento alle routines **modijog()** e **aggvol()**, risolve in maniera facile e sicura la gestione delle varie modalità di movimento assi in modo JOG. L'uso delle routines suddette è spiegato più avanti in questo capitolo.

Se non vengono usate le routines standard ECS, per muovere un asse in JOG occorre:





- Attivare il modo JOG:
  - Scrivere su ICTS il codice (esadecimale) relativo:
    - \* **0x91** richiesta di modo JOG:
  - e complementare il segnale INCTS (nuovo codice tasto)
  - Il CNC risponde complementando lo stato del segnale di output **OAKCT**.
  - Quando il modo JOG è attivo, il CNC mette ad 1 l'uscita **OJOG**.
- Selezionare il tipo di movimento desiderato, scelto tra uno dei seguenti:
  - IJSPR = 1 Richiesta JOG di SPAZIO in RAPIDO
  - IJSPL = 1 Richiesta JOG di SPAZIO in LAVORO
  - IJVLR = 1 Richiesta JOG di VELOCITÀ in RAPIDO
  - IJVLL = 1 Richiesta JOG di VELOCITÀ in LAVORO
- Se selezionato un JOG di SPAZIO:
  - scrivere su IVJ (long) l'incremento desiderato in mm/1000 o gradi/1000
- Selezionare l'asse da muovere scrivendo su ISAJ (byte) Digit Low il suo numero (di AXS.TAR).
- Comandare START/STOP del movimento:
  - scrivere su ICTS il codice (esadecimale) relativo
    - \* 0xA3 start JOG +
    - \* 0xB5 stop JOG +
    - \* 0xA4 start JOG -
    - \* 0xA3 stop JOG -
- e complementare il segnale INCTS (nuovo codice tasto)

#### NOTE

*Il CNC può muovere in JOG un solo asse per volta.*

*Il segnale INCTS può essere complementato solo se OAKCT e INCTS stesso sono ambedue a 0 o a 1 (la routine di libreria `tastcnc()` setta CNOK in questa situazione).*

*Ad un codice di START far **sempre** seguire il relativo codice di STOP.*

*Il codice di STOP ha effetto solo su movimenti in JOG di VELOCITÀ.*

*In JOG di VELOCITÀ, il movimento continua fino al ricevimento del relativo codice di STOP (o RESET).*

*Un asse mosso in JOG non oltrepasserà mai i propri fincorsa software, purché siano definiti in AXS.TAR e sia stato fatto il SET.*

*La velocità dell'asse in JOG è la rapida (IJxxR) o l'ultima Feed programmata (IJxxL) ed è controllabile con il relativo Override.*

### 2.3.2 Modo SET

Tutti gli assi sono resi indipendenti (FRE), siano questi già FRE o CNC. Fa eccezione l'asse mandrino che si comporta in maniera analoga al modo JOG.

### 2.3.3 Modi MDI, AUTOMATICO, RIPOSIZIONAMENTO

- Tutti gli assi sono agganciati come definito dal file AXS.TAR e/o eventualmente modificato nel file COST o, meglio, COSTUSER. Durante l'esecuzione o lettura di programmi la situazione aggancio assi in contouring o punto-a-punto varia in seguito delle istruzioni <CNC:...>, <FRE:...>, ma con il RESET viene ripristinata la situazione come definita in AXS.TAR. Le istruzioni <FRE:...>; <CNC:...> hanno effetto immediato anche in LETTURA o RICERCA.
- Gli assi dichiarati nel file AXS.TAR come INDIPENDENTI (IND=Y) non possono essere messi in contouring (l'istruzione <CNC:...> è rifiutata). Questo non vale per l'asse dichiarato MANDRINO.



- Per gli assi dichiarati INDIPENDENTI (IND=Y) nel file AXS.TAR, valgono le seguenti regole (escluso asse mandrino):
  1. La Feed è quella dichiarata con le istruzioni <FED:...> (per default = 100 mm/' per asse lineare o 360°/' per asse tondo, se non modificato nel file COST o, meglio, COSTUSER).
  2. Non ha effetto l'istruzione <MIR:...>: specularità.
    - Non ha effetto l'istruzione G58 o G59: spostamento origine
    - Non ha effetto l'istruzione G54.1÷G54.20: cambio origine
    - Non ha effetto l'istruzione G25; G26: variazione finecorsa software; vengono considerati i solo i finecorsa definiti in AXS.TAR.
    - Non ha effetto l'istruzione G91 programmazione incrementale
- In modo JOG, gli ASSI dichiarati INDIPENDENTI in AXS.TAR ed il MANDRINO si muovono con la F programmata per gli assi CNC. Valgono le considerazioni fatte al punto 2.
- ASSE MANDRINO (dichiarato INDIPENDENTE in AXS.TAR). Normalmente si comporta come ai punti 1 e 2. Agganciato in contouring da Programma (quindi con modo CNC diverso da JOG o SET), si comporta come un asse CNC a tutti gli effetti. Sente quindi: G58, G59, G54.1÷G54.20, G25, G26, <MIR:...>, G91 (non ha senso per gli ASSI TONDI parlare di Fine Corsa).
- Per gli assi resi indipendenti <FRE:...> da Programma mentre sono dichiarati CNC in AXS.TAR, vengono considerate le istruzioni G58, G59, G54.1÷G54.20, G25, G26, <MIR:...>, G91. In modo JOG non si tiene conto di G91 e <MIR:...>
- Con le istruzioni tipo G800 e G900 (escluse le G del mandrino), per tutti gli assi, siano questi CNC o FRE, non vengono considerati: G25, G26, finecorsa software, G58, G59, G54.1÷G54.20, <MIR:...>.
- ORIGINI G54.1 ÷ G54.20  
Con cambi modo del CNC da: MAN a JOG o SET, rimane sempre l'ultima origine selezionata.  
Passando da MAN o JOG o SET ad AUTO o RIPOS., si hanno due comportamenti:
  - Se non si stava prima eseguendo in AUTO un Programma, l'origine non viene variata.
  - Se in AUTO si stava eseguendo un Programma e durante l'esecuzione si è passati in MAN-SET-JOG ritornando in AUTO, RIPOSIZ. viene sempre ripristinata l'origine utilizzata dal Programma.Premendo il pulsante [RESET], in qualsiasi modo sia il CNC, rimane l'ultima origine selezionata (se non diversamente cambiata nel file RESET).  
Con l'istruzione % viene selezionata l'origine di default G54.1 o quella eventualmente diversa stabilita nel file COST o, meglio, COSTUSER.

## 2.3.4 Routines a disposizione del PLC

### 2.3.4.1 Routine "AGGAX"

Chiamata: **aggax()**;

Scopo: Gestisce l'abilitazione / disabilitazione di assi CNC.

Parametri in Ingresso :

**RABIL# = 1** RICHIESTA ABILITAZIONE ASSE ORDINE #

Tale richiesta sarà resa vera dal PLC in presenza di tutte le condizioni (ausiliari inseriti, azionamento OK, ecc.) necessarie per abilitare ogni specifico asse.

Per gli assi che prevedono bloccaggi (e definiti Bloccabili con <BLC:nome\_asse>) occorre passare in input alla routine **aggax()** anche le seguenti informazioni:

**BLCAX# = 1 ASSE DI ORDINE # BLOCCATO**

Il segnale **BLCAX#** deve rimanere ad "1" fintanto che l'asse è bloccato meccanicamente.

Quando esso va a zero, il CNC può dare riferimento all'asse interessato.

Se un asse è definito bloccabile, è consigliabile impostare un ritardo nel bloccare l'asse dopo che esso è "arrivato" in posizione. Se tale ritardo non è gestito, l'asse verrà immediatamente bloccato e disabilitato appena arrivato in posizione.

Per far questo occorre caricare, con la funzione **lotime(...)**, il ritardo desiderato nei temporizzatori appositamente previsti:

```
lotime(#,xx);
```

dove: # = numero del temporizzatore, che corrisponde al numero dell'asse, per gli assi 0...11; per gli assi 12...15, i temporizzatori sono quelli di numero 112...115 .

xx = tempo di ritardo espresso in secondi (tipico = 0.3 sec.)

La routine gestisce anche la richiesta di porre l'asse nello stato di "Libera Servo":

**RILS# = 1 RICHIESTA LIBERA SERVO PER L'ASSE DI ORDINE #**

Non va quindi comandato direttamente **ILS#** per evitare conflitti con quanto fatto nella routine stessa.

La routine, una volta ricevuto la richiesta di abilitazione, verifica, per ciascun asse, se esso è in gestione al CNC (OBAG#=1) e se non è in allarme (OBOK#=1); in caso affermativo fornisce in uscita al PLC il consenso ad abilitare l'asse.

Parametri di uscita:

**ABIL# = 1 CONSENSO AD ABILITARE ASSE DI ORDINE #**

Quando si ha l'uscita ABIL# = 1, il PLC può abilitare l'azionamento dell'asse.

Per un asse Digitale (ECSLINK, Sercos, Mechatrolink), l'abilitazione è fatta tramite il segnale IDRNON#.

All'arrivo in quota di un asse "bloccabile" il CNC si comporta come segue:

- il segnale **OBLC#** va ad "1" per avvisare che è possibile bloccare l'asse.
- il segnale **ABIL#** ha uno dei seguenti comportamenti:
- ABIL# = va a 0 istantaneamente (se non definito un ritardo)
- ABIL# = va a 0 dopo il ritardo impostato attraverso la funzione **lotime(...)**

Il secondo approccio è più corretto in quanto lascia un po' di tempo all'asse per recuperare un eventuale errore di posizionamento prima di venire bloccato e disabilitato.

Da tenere presente che il CNC, una volta che l'asse è arrivato in quota (OBLC#=1), non aspetta che l'asse sia effettivamente bloccato per passare al blocco successivo, se occorre aspettare, deve essere cura del PLC mettere in FEED HOLD gli assi.

I parametri di ingresso e uscita della routine sono tutti di tipo bit; i relativi sinonimi sono definiti in DF\_SYN.INC e sono allocati nell'area dati B (512 word riservate ad appoggi per le routines di libreria PLC fornite da ECS).

**{T}** Negli esempi dati, si fa' tipicamente riferimento ad una Fresa con 3 assi di nome X Y Z associati ai primi tre assi di AXS.TAR; normalmente tali esempi sono applicabili anche ai Torni tenendo però in conto che gli assi di un tornio tipico sono X Z associati ai primi due di AXS.TAR .



```
/*-----  
ESEMPIO D'USO DELLA ROUTINE AGGAX  
-----*/  
if (pon)  
{  
  /* SETTA TEMPORIZZATI AL pon (Power ON) */  
  lotime(1,0.3) ;  
  /* SETTA RITARDO DI 0.3 SEC PER ASSE Y DI ORDINE 1 */  
  lotime(2,0.4) ;  
  /* SETTA RITARDO DI 0.4 SEC PER ASSE Z DI ORDINE 2 */  
}  
RABIL0= MUON && AZXOK && ... ;  
  /* RICHIESTA ABILITAZIONE ASSE X ORDINE 0 */  
RABIL1= MUON && AZYOK && ... ;  
  /* RICHIESTA ABILITAZIONE ASSE Y ORDINE 1 */  
RABIL2= MUON && AZZOK && ... ;  
  /* RICHIESTA ABILITAZIONE ASSE Z ORDINE 2 */  
BLCAX1= PBLOCY ;  
  /* ASSE Y BLOCCATO (PBLOCY= PRESSOSTATO ASSE BLOCCATO) */  
BLCAX2= PBLOCZ ;  
  /* ASSE Z BLOCCATO (PBLOCZ= PRESSOSTATO ASSE BLOCCATO) */  
  
aggax() ;  
  /* ROUTINE STANDARD AGGANCIO ASSI */  
  
ABILX= ABIL0 ;  
  /* ABILITAZIONE ASSE X (SENZA BLOCCAGGI) */  
ABILY= ABIL1 || ABILY && !PBLOCY ;  
  /* ABILITAZIONE ASSE Y */  
EVSBY= ABIL1 ;  
  /* ELETTRIVALVOLA SBLOCCO ASSE Y */  
ABILZ= ABIL2 || ABILZ && !PBLOCZ ;  
  /* ABILITAZIONE ASSE Z */  
  
IDRVON0= ABILX ;  
  /* DRIVE ON PER CNC ED AZIONAMENTO DIGITALE ASSE X */  
IDRVON1= ABILY ;  
  /* DRIVE ON PER CNC ED AZIONAMENTO DIGITALE ASSE Y */  
IDRVON2= ABILZ ;  
  /* DRIVE ON PER CNC ED AZIONAMENTO DIGITALE ASSE Z */  
  
ALR54= OLAV && OEXE &&!ABLAVMF !OJOG && !OMAN &&!AM03 && !AM04 && !G84  
  && !G184 || ALR54 && !ORST ;  
  /* ASSI IN LAVORO CON MANDRINO FERMO */
```

### 2.3.4.2 Routine "MODIJOG"

Chiamata: **modijog()**;

Scopo: Gestisce la selezione dell'asse da muovere in modo JOG e la modalità di movimento.

Parametri in Ingresso :

**COMAXS =...** (unsigned char) Numero d'ordine dell'Asse da muovere in JOG.



---

**INJS =...** : (long) Incremento desiderato (mm/1000 o gradi/1000) per movimenti in JOG di SPAZIO

La routine lavora in abbinamento con **tasfuecs()** che setta COMAXS e INJS in base alle scelte fatte dall'operatore tramite i tasti funzione del PLC nella configurazione standard ECS definita in BASMENU.MEN; tasfuecs() setta anche i segnali (IJSPL IJSPL IJVLR IJVLR) per la scelta della modalità di movimento.

La routine modijog() gestisce anche la scelta dell'asse e la modalità di movimento richieste dall'utility di TARATURA AZIONAMENTI presente sul CNC.

Variabili PLC non riutilizzabili:

### SEQ28

SEQ28 è usato dalla routine aggvol() descritta più avanti ed in particolare segnala, quando SEQ28==0, che non sono attivi i volantini.

I valori di ingresso della routine sono del tipo indicato più sopra; i relativi sinonimi sono definiti in DF\_SYN.INC e sono allocati nell'area dati B (512 word riservate ad appoggi per le routines di libreria PLC fornite da ECS).

```
/*-----  
ESEMPIO D'USO ROUTINE IN QAL.C  
-----*/  
  
tasfuecs();  
/* gestione standard tasti funzione PLC */  
/* oppure: */  
/* IJSPL= R0.bit._00; */  
/* IJSPL= R0.bit._01; */  
/* IJVLR= R0.bit._02; */  
/* IJVLR= R0.bit._03; */  
/* COMAXS= R0.byte.h ; */  
/* INJS= (long)SPAZIO; */  
modijog ();  
/* gestione modo JOG */
```

## 2.4 BLOCCAGGIO ASSI

Il CNC permette di definire dinamicamente quali assi bloccare o no. All'accensione, il CNC considera tutti gli assi vivi (se non dichiarato diversamente nel file COST o COSTUSER) e quindi sbloccati. Lo stato di bloccabile / non bloccabile definito nel file COST è preso come stato di base degli assi ad ogni ingresso in qualsiasi modo di funzionamento del CNC (AUTO, MDI, SET, JOG, RIP) e sempre dopo RESET CNC. La sintassi delle istruzioni per sbloccare e bloccare gli assi è:

**<SBL: sigla\_asse[;...;...]>**  
**<SBL: numero\_asse[;...;...]>**  
**<SBL:<EXP>[;...;...]>**

gli assi sono sbloccati anche se fermi; viene annullata l'istruzione <BLC:...> per gli assi indicati.

**<BLC: sigla\_asse[;...;...]>**  
**<BLC: numero\_asse[;...;...]>**  
**<BLC:<EXP>[;...;...]>**

abilita il PLC a bloccare gli e assi indicati quando questi non debbono muoversi.

La sincronizzazione CNC ↔ PLC avviene attraverso i segnali **OBLC#**, **BLCAX#**, **IBLC#**.

---



Le istruzioni <SBL:...> <BLC:...> hanno lo stesso effetto su ogni tipo di asse, sia esso dichiarato CNC o FRE da Programma o FRE di Taratura.

### 2.4.1 Bloccaggio/Sbloccaggio assi: funzionamento e temporizzazioni

Quando il CNC deve muovere un asse che precedentemente è stato dichiarato BLOCCABILE attraverso l'istruzione <BLC:...>, la sequenza è:

- 1) Il CNC pone il segnale  $OBLC\# = 0$ : avverte cioè di sbloccare asse.
- 2) Il CNC attende che il PLC metta il segnale  $IBLC\# = 0$  prima di emettere il riferimento (se si usa  $aggax()$ , il segnale da porre  $=0$  è  $BLCAX\#$ ).
- 3) Con il segnale di asse arrivato ( $OBNA\#=0$ ) il CNC mette il segnale  $OBLC\#="1"$  e passa all'istruzione successiva SENZA attendere che l'asse sia effettivamente BLOCCATO. Occorre forzare Feed-Hold ( $IFEEDHLD=1$ ) se non si vuole che si muovano gli assi con un asse in fase di bloccaggio.

#### NOTE

- Il CNC può muovere assi anche non programmati esplicitamente; ad es. nel caso di istruzione <TCT:...> o di inserzione raccordo automatico con programmazione GAP, ...
- Un asse è da muovere anche quando viene programmata la quota dove si trova attualmente.
- Prima di inserire la compensazione raggio utensile G41 G42 si debbono escludere i bloccaggi sugli assi interessati dalla contornatura, altrimenti il CNC arriva con velocità 0 sulla quota di partenza delle varie N... ove deve controllare lo stato di sbloccaggio degli assi

#### ATTENZIONE:

Se non si vuole sbloccare/abilitare un asse che il CNC chiede di sbloccare ma che non sarà da muovere perché la sua quota finale è uguale a quella attuale, occorre utilizzare il segnale **OBNA#** (oltre naturalmente ad **ABIL#**) per abilitare e sbloccare l'asse stesso. In questo caso non va utilizzato **BLCAX#** che dovrà essere posto sempre  $=0$ ; per NON fare muovere gli assi dal CNC quando **OBNA#=1** e asse ancora NON sbloccato/abilitato occorre, in questo caso, usare **IFEEDHLD**.

#### 2.4.1.1 Esempi

##### Esempio 1

- X Y Z assi CNC interessati alla contornatura

```
N10 <BLC:X;Y;Z;W>
...
N50 <FRE:W>
...
N100 <SBL:X;Y;Z>
N101 G41 X...Y...
N102 X...
N103 X...Z...
N104 X... Z... W...
```

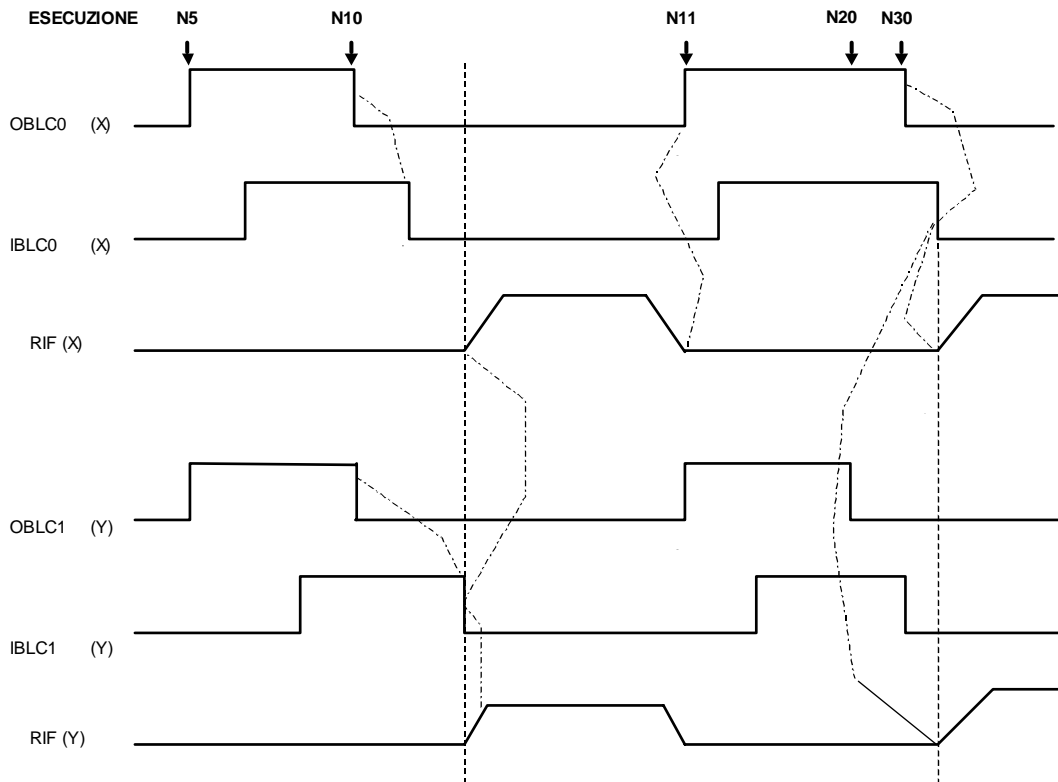
Alla fine della N103 gli assi X e Z arrivano sul punto programmato con velocità zero perché nella N104, prima di muovere i tre assi, si deve attendere che W sia sbloccato.

##### Esempio 2

```
N5 <BLC:X;W>
```



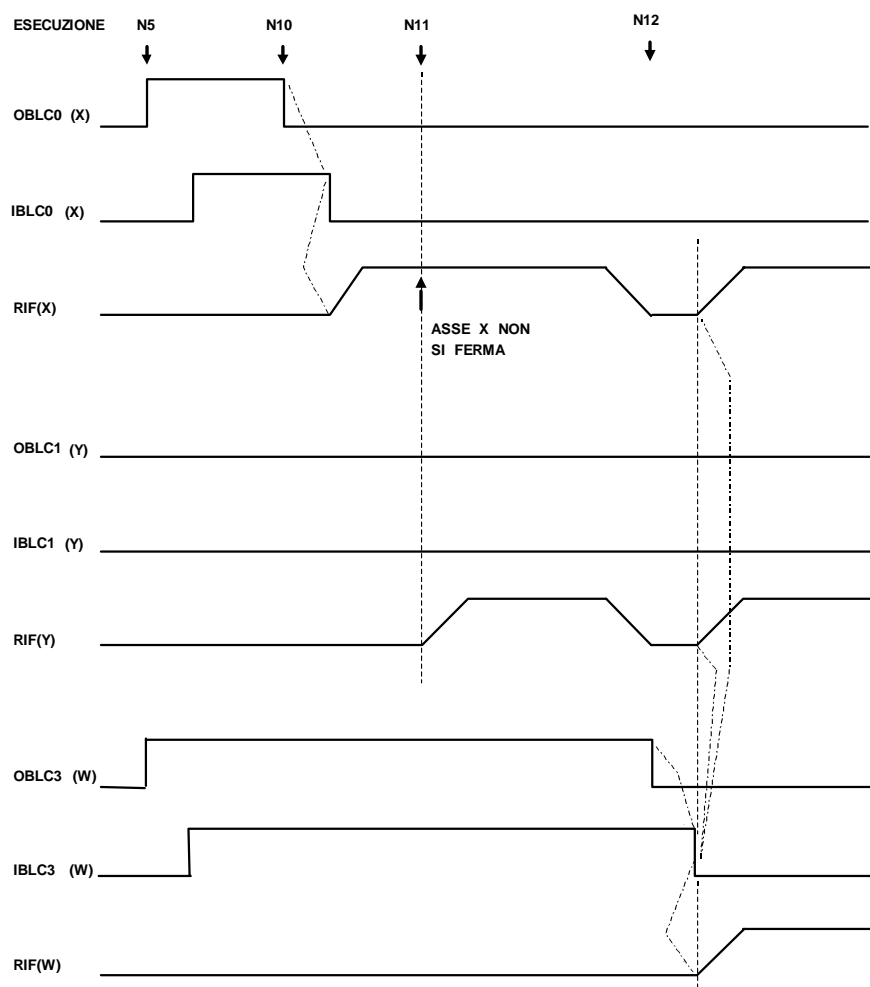
```
N10 X... Y...  
N11..  
N12..  
N15..  
N20<SBL:Y>  
N30 X...Y...
```



### Esempio 3

Y su quota 10; W asse indipendente:

```
N5 <BLC:X;W>  
..  
N10 X100  
N11 X200 Y10.1  
N12 X300 Y10.2 W200
```



### 2.4.1.2 Sottoprogrammi

Il CNC contiene, in LAVMACRO, i sottoprogrammi M10 (bloccaggio assi) e M11 (sbloccaggio assi). Per l'utilizzo, inserire le sigle degli assi interessati. Nel caso si volesse creare una funzione di bloccaggio e una di sbloccaggio dedicata a ciascun asse, occorre creare i nuovi sottoprogrammi utilizzando i codici delle F.A.. M disponibili ricopiandoli dal M10 e M11.

Dichiarare le F.A. utilizzate nel file FA.TAB (vedi MANUALE TARATURE 720P385).





## 2.5 SET ASSI

Sui CNC ECS serie WIN sono presenti vari files da usare per la procedura di ricerca zero (SET) degli assi, denominati **SET0 SET1 SET2** ecc. rispettivamente per gli assi d'ordine 0-1-2 ecc., ed il file **SETAL**.

I files **SET#** sono predisposti per la ricerca zero di assi lineari con trasduttori ottici dotati di una o più MARCHE DI ZERO, dette anche MARKER; prevedono la presenza di una camma posizionata dal lato del fincorsa positivo e di un micro-switch con contatto normalmente aperto, detto anche MICRO DI ZERO. Per gli assi che richiedono una procedura di presa zero diversa da quella predisposta, occorre modificare il relativo file SET# usando le opportune istruzioni.

Il file **SETAL** automatizza il lancio e quindi la sequenza di esecuzione dei files di set relativi ai singoli assi oppure, opportunamente riscritto, può comandare la presa di zero di più assi contemporaneamente.

I MICRO DI ZERO vanno collegati sugli appositi ingressi previsti sulle schede assi. Con assi digitali, i micro di zero vanno collegati sugli appositi ingressi degli azionamenti:

Azionamenti ECS con ECSLINK	Morsetto <b>MICROSET</b> (C8)
Azionamenti Indramat con Sercos	Morsetto <b>Homing switch</b> (X3-1)
Azionamenti Yaskawa con Mechatrolink	Pin <b>EXT1</b> (CN1-44).

Sono presenti inoltre i files:

**RIGABS**, predisposto per righe ottiche assolutizzate. Ricopiare RIGABS nel file SET# relativo. Modificare i parametri in funzione del tipo di riga assoluta e di come si vuole eseguire il ciclo.

**ENCABST**, predisposto per encoder ottici assolutizzati applicati ad assi tondi (ad es. tavole). Ricopiare ENCABST nel file SET# relativo. Modificare i parametri in funzione del tipo di encoder e di come si vuole eseguire il ciclo.

**ENCABSL**, predisposto per encoder ottici assolutizzati applicati ad assi lineari (su vite). Ricopiare ENCABSL nel file SET# relativo. Modificare i parametri in funzione del tipo di encoder e di come si vuole eseguire il ciclo.

La funzione di questi files ed il relativo contenuto sono dettagliatamente spiegati nel MANUALE TARATURE 720P385.

Tutti i files di set sono modificabili utilizzando una programmazione di tipo ISO per i movimenti assi, delle apposite funzioni G8nn, G9nn ed istruzioni LIP per adattarli alle specifiche esigenze di ciascun asse della macchina utensile.

I files verranno automaticamente eseguiti dal CNC quando, tramite l'apposito menù attivo in modo SET, l'operatore richiederà la presa di zero degli assi. Il menù permette la libera scelta del singolo asse, per cui l'ordine di esecuzione del SET è a scelta dell'operatore. L'operatore può scegliere anche la voce "**TUTTI**"; in questo caso verrà automaticamente eseguito il file SETAL che richiama i vari files di SET secondo la sequenza decisa dal costruttore della M.U. e riportata all'interno del file stesso.

Con **OBSE# = 1** è segnalato lo stato di presa di zero effettuata.

Con **OBSE# = 0** è consentito muovere l'asse solo nei modi JOG e SET e, naturalmente, non sono attivi i FINECORSA SOFTWARE.

Con **IBSET# = 1** viene forzata la condizione di SET da rifare; forza **OBSE# = 0** e **OBSET# = 1**. Il segnale OBSET# verrà rimesso = 0 quando sarà di nuovo effettuato il SET dell'asse.

Lo stato di SET eseguito non viene perso con il passaggio EXE⇒ALT⇒EXE; viene perso, ovviamente, spegnendo il CNC, in caso di Allarme FUGA ASSE ed in caso di problemi sui trasduttori degli assi.



Dalla release SW Fresa V2.00 (Tornio V1.0), è stata inserita la possibilità di NON perdere lo stato di SET dell'asse in caso di Allarme FUGA ASSE (ALL. n. 32708) che interviene quando l'errore di inseguimento supera le soglie definite, asse per asse, nel file di taratura assi AXS.TAR (SFS=... valida con asse in moto e FFS=... valida con asse fermo). A questo scopo, occorre abilitare (togliendo le parentesi graffe { }) l'apposita istruzione presente sul file COST o, meglio, scriverla in COSTUSER:

```
N... { <%V1912=%V1912|'4000'> }  
N... {bit14=1 fa in modo che la condizione di fuga non generi la perdita}  
N... {del SET. Valido per assi con allarme trasduttore abilitato ATR=Y }
```

Si precisa che per il controllo dei trasduttori assi sono presenti due specifici allarmi:

- a) ALL. n. 32710: "*Trasduttore asse .. non collegato, non bilanciato o guasto*" (viene abilitato ponendo **ATR=Y** in AXS.TAR per ogni singolo asse) che interviene quando l'hardware della scheda assi rileva problemi di livello sui segnali provenienti dal trasduttore.
- b) ALL. n. 32713: "*Sovraconteggio Trasduttore asse ..*" che segnala una frequenza di conteggio troppo alta per la circuiteria della scheda assi e viene abilitato, per tutti gli assi, tramite l'apposita istruzione presente sul file COST, da ricopiare, senza graffe, in COSTUSER:

```
N... { <%V1912=%V1912|'8000'> }  
N... {bit15=1 Abilita allarme "sovracconteggio" trasduttore}
```

Riassumendo, il SET viene perso se:

- 1- interviene l'Allarme a) di *trasduttore guasto* abilitato da ATR=Y in AXS.TAR .
- 2- interviene l'Allarme b) di *sovracconteggio* abilitato dal bit 15 di %V1912 visto sopra.
- 3- interviene l'Allarme di *FUGA ASSE* e NON è settato il bit 14 di %V1912 .
- 4- interviene l'Allarme di *FUGA ASSE* ed è stato posto ATR=N in AXS.TAR .
- 5- viene dato il comando specifico mettendo IBSET# = 1.

Di conseguenza, il SET NON viene perso solo se:

- 1- interviene *FUGA ASSE*, è stato posto ATR=Y in AXS.TAR ed è settato il bit 14 di %V1912.

Il ciclo di presa di zero viene eseguito con il CNC in modo SET e quindi con gli assi posti automaticamente nello stato FRE; per gli assi sono quindi attivi i segnali relativi a tale stato (**IHL#**, **HLD#**, **OVR#**); non sono attivi i segnali validi per gli assi CNC (**IBLAXMND**, **IBLAX**, **IFEEDHLD**, **IRAP**, **IOVR**, **IOVF**).

## 2.5.1 ASSI INCREMENTALI SERCOS IN GANTRY {W}

La gestione del SET (homing) di un qualsiasi asse, con driver Indramat dotato di interfaccia Sercos verso il CNC, è demandata dal CNC al driver stesso. Per assi "normali" non ci sono differenze, per la procedura di SET, rispetto a quanto visto sopra. Per assi agganciati in Gantry si rende necessaria una procedura particolare di SET. Al riguardo, si rimanda al paragrafo "Homing assi Sercos incrementali in Gantry" (vedi capitolo successivo).

## 2.5.2 ASSI ASSOLUTI {W}

Dalla release di software WIN V1.00, il CNC gestisce in modo automatico gli assi assoluti ECS con interfaccia SSI, e tutti i trasduttori assoluti gestiti da Azionamenti Indramat con interfaccia Sercos verso il CNC. Per assi "normali" non occorre eseguire, evidentemente,



---

alcuna procedura di SET. Per assi agganciati in Gantry vero si rende necessaria una procedura particolare di SET. Al riguardo, si rimanda al paragrafo "Assi assoluti in Gantry" (vedi capitolo successivo).

## 2.6 LIBERA SERVO

Tale prestazione si rende necessaria tutte le volte che il CNC deve abbandonare il governo di un suo asse, sia questo dichiarato CNC o FRE; l'esempio più classico è quando ci si trova a dover trattare assi per movimentazione di dispositivi (TESTINE o TAVOLE) che devono poter essere anche rimossi (sconnessi dal CNC). I segnali di input relativi a questi assi sono ILS#, e INAL#, mentre il segnale di output è OLS#.

Tali segnali sono già stati sommariamente descritti nel paragrafo 2.2.1 "SINONIMI . . ." . Qui viene approfondito il loro uso e le precauzioni da prendere per un corretto utilizzo.

### 2.6.1 Segnali ILS# e OLS# per LIBERA SERVO

Il CNC controlla ciclicamente (per tutti gli assi, sia CNC che FRE) se vi sono state variazioni al segnale ILS (Input Libera Servo).

#### TRANSIZIONE DEL SEGNALE ILS DA FALSO A VERO

- Il CNC non interpreta più i blocchi in avanti, e si ferma.
- Attende che l'interpolatore abbia terminato di eseguire quello che era stato interpretato. Al massimo vengono accettate F.A. o comunicazioni generiche che erano in corso di emissione verso il PLC.
- Superato i primi due punti, il CNC analizza se la richiesta è lecita e se quindi può essere onorata ed eseguita; se le condizioni non sono OK interviene un ALLARME SPECIFICO e la relativa segnalazione è inviata al PLC come comunicazione generica sulle words di Output OVGENH, OVGENL. Le condizioni per cui il CNC può rifiutare il comando sono:
  1. Asse gestito dal PLC (OAXPLC#=1).
  2. CNC in HOLD.
  3. KEY in corso o richiesta KEY in corso.
  4. Il CNC sta interpretando un Part-Program con istruzioni GAP oppure con compensazione Raggio Utensile inserita.
  5. Asse con matrici dinamiche inserite.
  6. Asse mandrino in rotazione o in cambio gamma.
  7. Asse con volantino inserito.
  8. Modo COPIA attivo
  9. Asse non fermo OBNA#=1 e/o OBMT#=1
  10. Sono in corso Istruzioni o Comandi ad un qualsiasi asse indipendente (FRE).
  11. Traiettoria degli assi CNC non terminata (il CNC attende 20 sec. poi dà Allarme)
- Se sono soddisfatte tutte le condizioni elencate al punto precedente, il CNC sgancia l'asse (OBAG#=0) e mette OLS#=1; se si stava interpretando un Part-Program, il CNC riprende ad interpretarlo. A questo punto il PLC può muovere l'asse attraverso riferimenti non generati dalla scheda ASSE del CNC oppure può pilotare l'asse attraverso il CNC con le istruzioni:
  - \* <FRE:...>
  - \* <SAX:...>.
  - \* Tutte le G800 e G900 per assi indipendenti.
  - \* Programmandolo come ASSE INDIPENDENTE per esempio X quota se fosse l'asse X.



**ATTENZIONE!**

*Gli assi che hanno il segnale OLS =1, non sono presi in considerazione nelle istruzioni <WAT:...> (anche le istruzioni che effettuano un wait implicito non considerano gli assi che hanno il segnale OLS=1)*

**TRANSIZIONE DEL SEGNALE ILS DA VERO A FALSO**

- Il CNC esegue le fasi descritte per passaggio da FALSO a VERO (da 1 a 3), soddisfatte tutte queste condizioni sgancia l'asse e invia nuovamente tutti i parametri di inizializzazione della scheda Assi dichiarati in AXS.TAR per quell'asse, riaggancia l'asse come voluto dal CNC e rende FALSO il segnale OLS.
- Se si era interrotta l'interpretazione dell'eventuale Programma, il CNC riprende ad interpretare.

**2.6.2 Note generali**

- Gli assi con OLS=1 sono sconosciuti al CNC durante i cambi di MODO.
- Quando un asse viene ripreso in carico dal CNC (passaggio di OLS da 1 a 0), il CNC lo riaggancia sulla quota ove questo si trova e fino a che non viene programmato un movimento su quota assoluta G90 per questo asse non viene gestito l'eventuale riposizionamento.
- Il segnale ILS è sentito dal CNC anche all'accensione del CNC o nei passaggi dallo stato di ALT allo stato di EXE.
- Durante la fase di RESET il CNC sente sempre lo stato del segnale IEME (anche per assi che hanno il segnale OLS=1) e poi lo stato di ILS#. Il segnale ISGA non è preso in considerazione per assi che hanno il segnale OLS=1. Se la logica d'interfaccia, durante il RESET, desidera modificare lo stato degli assi, deve SETTARE/RESETTARE il segnale ILS prima di restituire l'OK al RESET tramite il segnale IRES. Durante il RESET, il CNC reinizializza con le tarature date da AXS.TAR anche gli assi che pur avendo il segnale OLS=1 hanno OBOK=0.
- Il CNC considera gli assi che hanno OLS =1 come assi INDIPENDENTI.
- I segnali IEME#, IANA#, IHLD, IBLC# sono gestiti dal CNC nel modo consueto anche se l'asse ha OLS#=1.
- Durante l'esecuzione di Part-Program in G65, il CNC ignora le variazioni del segnale ILS.
- Qualora il PLC abbia la necessità di rendere VERO o FALSO il segnale ILS in un punto ben preciso del Part-Program (sincronizzato), una modalità consigliata è la seguente:

dichiarare un F.A., ad esempio M50, WAIT in FA.TAB; quando il PLC intercetta tale M deve togliere il FATTOM (Fatto delle F.A. "M"), variare lo stato del segnale ILS, attendere che il CNC risponda con il segnale OLS congruente, restituire il FATTOM relativo alla M emessa. In sintesi si deve rendere VERO/FALSO il segnale ILS appena si intercetta la M, togliendo immediatamente il FATTOM e restituendolo solo quando arriverà il segnale OLS congruente con lo stato di ILS. Se il PLC ha la necessità di fare cose diverse dal semplice SGANCIO dell'asse, si può dichiarare subroutine la M50 e definire in tale sottoprogramma cosa fare prima e dopo la \$M50 che agisce su ILS.



---

## 2.7 ASSE OFF

Tale prestazione è molto simile a quanto ottenuto con il comando ILS# visto al paragrafo precedente; la differenza sostanziale consiste nel fatto che il CNC non effettua tutti i controlli visti sopra per ILS# ma accetta immediatamente il comando. Sono quindi l'operatore e/o il PLC a preoccuparsi di richiedere la disinserzione dell'asse quando la situazione operativa della M.U. lo permette. Il segnale di input relativo a questo comando è **IAXOFF#** mentre il segnale di output è **OAXOFF#**.

Il CNC, riscontrando una variazione 0→1 su un segnale **IAXOFF#**, verifica se l'asse relativo (ed il suo asse correlato in caso di GANTRY) è fermo e poi sgancia l'asse ponendo **OBAG#=0** e **OAXOFF#=1**; ciò accade anche per l'eventuale asse GANTRY correlato.

Se l'asse è Digitale, l'azionamento relativo viene disabilitato.

L'asse in stato di OFF (**OBAG#=0** e **OAXOFF#=1**) può ora essere mosso attraverso riferimenti esterni o spento e scollegato. Poiché il CNC controlla continuamente l'integrità del sistema di misura, per evitare l'intervento di allarmi occorre mettere **INAL#=1** ed eventualmente, per assi comandati da driver digitali (ECSLINK, Sercos e Mechatrolink) **IDGOFF#=1**.

Il CNC comunica eventualmente l'allarme 10020 "si tenta di muovere l'asse attualmente non disponibile" in caso rilevi movimenti programmati sull'asse interessato.

Prima di rimettere IAXOFF#=0 per riattivare l'asse, occorre ricollegare motore e trasduttore dell'asse, riaccendere il relativo azionamento, togliere IDGOFF# e, se tutto OK, togliere INAL#. Il CNC invia nuovamente tutti i parametri di inizializzazione dichiarati in AXS.TAR per quell'asse (gli assi master e slave in caso di GANTRY), riaggancia l'asse come voluto dal CNC e rende FALSO il segnale OAXOFF.

Nel caso il comando di riattivare l'asse sia richiesto dal Part-program (o da una subroutine), prima di un'eventuale istruzione <FRE:...> è necessario inserire una <WAT:CNC> per attendere il riaggancio.

## 2.8 FINECORSO SOFTWARE E HARDWARE

Il CNC permette di definire (vedi MANUALE TARTURE 720P385: AXS.TAR) i limiti estremi della corsa di ciascun asse della Macchina Utensile (Finecorsa software). Tali limiti non possono essere superati, in quanto già prima dell'inizio del movimento è individuato, e segnalato con specifico allarme, quale asse oltrepasserebbe la propria corsa automatica se il movimento fosse effettuato. Con l'uso dei finecorsa software si ottiene perciò uno sfruttamento ottimale della corsa utile degli assi della M.U.

### PRECISAZIONI

- E' opportuno, comunque, montare micro-interruttori di emergenza (extracorsa hardware) che disalimentino, se azionati, tutto l'impianto in modo da provocare l'arresto della M.U. nella maniera più sicura e più veloce possibile (usando, ad esempio, la frenatura dinamica degli azionamenti). La distanza tra micro di extracorsa e battuta meccanica deve essere tale che ogni asse riesca a fermarsi pur arrivando sul micro stesso in velocità rapida.
- La distanza che intercorre tra i limiti imposti dai finecorsa software (FCP FCN in AXS.TAR) ed il relativo micro di extracorsa può essere molto ridotta (al limite = 1 micron).



- I finecorsa software sono DISATTIVI se non è stata effettuata la presa di zero (SET) dell'asse (OBSE#=0); in questa situazione sono consentiti solo movimenti in JOG. Si può limitare la velocità massima degli assi ponendo un limite (ad es. 20%) agli Override (vedi paragrafo OVERRIDE).
- I finecorsa software sono sempre attivi (una volta effettuato il SET, OBSE#=1) sia per gli assi CNC che per quelli INDIPENDENTI. Il CNC non può eseguire nessuna diagnostica preventiva o limitazione per gli assi per i quali non sono stati definiti.
- Ricordiamo inoltre che l'utilizzo dei finecorsa software rende disponibili le istruzioni G25 e G26 per restringere il campo di lavoro da Part-Program (vedi "Norme di Programmazione").

## 2.9 VOLANTINI

### 2.9.1 Generalità

Questa prestazione consente all'operatore di muovere gli assi della M.U. tramite volantini. Un volantino è un trasduttore ottico simile a quelli usati per gli assi; normalmente fornisce 100 impulsi a giro.

- Il CNC può gestire fino a tre volantini contemporaneamente (quattro in modo DRF).
- I volantini possono essere associati dinamicamente a qualunque asse del CNC.
- È possibile definire il fattore moltiplicativo (FATTORE DI SCALA) tra movimento manovella volantino e spostamento asse.
- Il FATTORE DI SCALA è un numero che va da 1 a 255 ed è il fattore moltiplicativo di ciascun conto del volantino; determina l'entità di spostamento dell'asse (in micron o millesimi di grado) per ciascun conto del volantino.
- Un giro completo di volantino corrisponde normalmente, con FATTORE DI SCALA = 1, a 0.1 mm (o grado), con incrementi di 0.001 mm (o gradi).
- La velocità raggiungibile è funzione della velocità di rotazione della manovella del volantino ed al massimo uguale alla velocità rapida definita in taratura (AXS.TAR).
- L'accelerazione dell'asse è quella di taratura (AXS.TAR).
- L'inserimento/disinserimento dei volantini è a cura del PLC.
- Esistono vari modi di gestire i volantini: in Spazio, in Feed, in modo DRF.
- Il funzionamento in Spazio ed in Feed è ammesso solo nel modo JOG (l'asse da agganciare in modo volantino deve essere fermo) ed è equivalente a quello ottenuto tramite comandi di velocità o di spazio in JOG.
- Il funzionamento DRF (Differential Resolver Function), attivabile solo su assi dichiarati CNC in AXS.TAR, è ammesso anche in modo MDI e AUTOMATICO.

### 2.9.2 Sinonimi utilizzati per i VOLANTINI

Per la gestione dei volantini tramite il PLC, sono presenti vari segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Molti di questi segnali non sono da utilizzare se viene usata la routine standard **aggvol()** descritta più avanti. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).



<b>ISAJ</b> (char)	<b>PLC → CNC</b>	Selezione degli assi associati ai volantini 1 e 2 - digit Low: = numero asse (0...15) per volantino 1 - digit High: = numero asse (0...15) per volantino 2 Il digit Low è usato anche per selezionare l'asse da muovere in JOG (vedi paragrafo 2.2.1 "SINONIMI. . .") <i>Non utilizzare se usata aggvol().</i>
<b>ISAJ3</b> (char)	<b>PLC → CNC</b>	Selezione dell'asse associato al volantino 3 - digit Low: = numero asse (0...15) per volantino 3 - digit High: = non usato. <i>Non utilizzare se usata aggvol().</i>
<b>IVJ1</b> IVJL (short)	<b>PLC → CNC</b>	IVJ1= valore, in conti, del volantino 1 Il valore va prelevato dal registro opportuno secondo le predisposizioni fatte in configurazione. <i>Non utilizzare se usata aggvol().</i>
<b>IVJ2</b> IVJH (short)	<b>PLC → CNC</b>	Valore, in conti, del volantino 2. (vedi IVJ1) <i>Non utilizzare se usata aggvol().</i>
<b>IVJ3</b> (short)	<b>PLC → CNC</b>	Valore, in conti, del volantino 3. (vedi IVJ1) <i>Non utilizzare se usata aggvol().</i>
<b>IVOLF</b>	<b>PLC → CNC</b>	Volantini in FEED IVOLF=0: volantini gestiti in SPAZIO IVOLF=1: volantini gestiti in FEED. Feed su IVJ1, IVJ2, IVJ3 o da Tastatore, se ITSTFE=1) <i>Non utilizzare se usata aggvol().</i>
<b>OVL#</b>	<b>CNC → PLC</b>	Volantino # (1, 2, 3) agganciato. Il CNC ha accettato la richiesta di attivare il volantino #.
<b>ITSTFE</b> {F}	<b>PLC → CNC</b>	Gestione tastatore continuo come volantini Feed. Posto a 1 predispone a muovere gli assi di Copiatura, in modo JOG, come se fosse attivo il modo COPIA e cioè tramite le deflessioni del tastatore. Vedere al paragrafo 2.9.4.1 un esempio d'uso.
<b>IVFM</b> (char)	<b>PLC → CNC</b>	Modo SPAZIO: Fattore di scala (1...255) comune per i volantini, in micron (o gradi/1000) ad impulso. <i>Non utilizzare se usata aggvol().</i> Modo FEED: i tre bit più leggeri definiscono, ciascuno, un fattore fisso di moltiplicazione della feed per i tre volantini: bit0=0: fattore 1; bit0=1: fattore 10; per volantino 1. bit1 : idem per volantino 2. bit2 : idem per volantino 3.
<b>IDRF12</b> (char)	<b>PLC → CNC</b>	Selezione degli assi associati ai volantini 1 e 2 in modo DRF. - digit Low : = numero asse (0...15) per volantino 1 - digit High : = numero asse (0...15) per volantino 2 <i>Non utilizzare se usata aggvol().</i>



<b>IDRF3</b> (char)	<b>PLC → CNC</b>	Selezione degli assi associati ai volantini 3 e 4 in modo DRF. - digit Low: = numero asse (0...15) per volantino 3 - digit High : = numero asse (0...15) per volantino 4 <i>Non utilizzare se usata aggvol().</i>
<b>ODRF</b>	<b>CNC → PLC</b>	Modalità volantini DRF attivata. Il CNC ha accettato la richiesta di attivare il modo volantini DRF.
<b>ODRF#</b>	<b>CNC → PLC</b>	Volantino # (1, 2, 3, 4) agganciato in modo DRF.
<b>IFMDRF</b> (char)	<b>PLC → CNC</b>	Fattore di scala per volantini in modo DRF Fattore (1...255) comune per i volantini attivi, in micron (o gradi/1000) ad impulso. <i>Non utilizzare se usata aggvol().</i>
<b>IBLDRF#</b>	<b>PLC → CNC</b>	Richiesto blocco conteggio volantino # (1, 2, 3, 4) in modo DRF Anche se il volantino viene mosso, non viene modificata la quota DRF accumulata sull'asse associato; l'asse non si sposta. Lo stesso accade con CNC in blocco o in feed-hold, se asse bloccato, non agganciato, con set non fatto o in finecorsa
<b>VVOL</b> (unsigned short)	<b>CNC → PLC</b>	Valore, in conti, del volantino collegato sulla tastiera.
<b>VVOL#</b> (short)	<b>CNC → PLC</b>	Valore, in conti, dei volantini collegati su schede assi; # può valere 1, 2, 3, 4. La corrispondenza tra numero ed input fisico è definita in SWCNF.TAR (vedi MANUALE TARATURE 720P385). <i>Non utilizzare se usata aggvol().</i> Quando sono gestiti da aggvol(): - se VOLTAS=0, VVOL1,2,3,4 corrispondono a volantino 1,2,3,4 - se VOLTAS=1, VVOL1,2,3 sono i volantini 2,3,4.
<b>VOL#IKR</b> (short)	<b>CNC → PLC</b>	Valore, in conti, dei volantini collegati su IKR (Interfaccia Keyboard Remota). # può valere 1, 2, 3. <i>Non utilizzare se usata aggvol().</i>
<b>CR_VVOL1</b> (short)	<b>CNC → PLC</b>	Valore, in conti, del volantino montato sulla console remota. <i>Non utilizzare se usata aggvol().</i>

## 2.9.3 Gestione Volantini

Per la gestione dei volantini è disponibile la routine **aggvol()** che risolve in maniera facile e sicura la gestione delle varie modalità di movimento assi tramite volantini. L'uso della routine suddetta è spiegato più avanti in questo capitolo (vedi par. 2.9.5.1).

Qui di seguito, vengono date alcune precisazioni sul come gestire i volantini, se non si usa la routine suddetta o se si vuole approfondirne il funzionamento.

### 2.9.3.1 Inserimento Volantini

Per l'inserimento dei volantini in modalità SPAZIO occorre:





- Scrivere su ISAJ digit Low il numero asse per volantino 1, e/o su digit High per volantino 2 e/o su ISAJ3 digit Low per volantino 3.
- Scrivere su IVFM il fattore di scala; tale fattore, con volantini inseriti, è acquisito dinamicamente. Il fattore di scala è lo stesso per tutti i volantini.
- Scrivere su ICTS il codice (esadecimale) relativo al tipo di inserimento richiesto:
  - \* 0xC1 Inserimento VL1
  - \* 0xC2 Inserimento VL2
  - \* 0xC3 Inserimento VL1 + VL2
  - \* 0xC4 Inserimento VL3
  - \* 0xC5 Inserimento VL1 + VL3
  - \* 0xC6 Inserimento VL2 + VL3
  - \* 0xC7 Inserimento VL1 + VL2 + VL3
- e complementare il segnale INCTS (nuovo codice tasto).

Il CNC risponde complementando lo stato del segnale di output **OAKCT**.

Quando il VL1 e/o VL2 e/o VL3 sono inseriti, il CNC mette ad 1 le uscite **OVL1** e/o **OVL2** e/o **OVL3**. Tali uscite sono a 0 per volantini disinseriti.

Il PLC trasferisce il valore dei "conti" dei volantini VL1, VL2 e VL3 su **IVJ1**, **IVJ2**, **IVJ3**. Tale valore è binario (sono utilizzabili 15 bits) e ogni bit equivale ad 1 conto che sarà moltiplicato per **IVFM** per ottenere i micron (o gradi/1000) di spostamento. I conti dei volantini sono disponibili sulle word **VVOL**, **VVOL#**, **VOL#IKR** secondo l'hardware presente.

Per attivare i volantini in modalità DRF occorre:

- Definire tipo e scala per i 4 volantini; si può usare la routine di libreria PLC `iniz_drf()` descritta più avanti.
- Testare se il CNC sta lavorando; si può usare la routine di libreria PLC `is_cnc_work()`:
  - ret = is\_cnc\_work();**
  - \* ret = 1 : CNC lavora
  - \* ret = 0 : CNC non lavora
- Con ret = 0, scrivere su ICTS il codice (esadecimale) relativo al comando:
  - \* 0xC9 Richiesta attivazione volantini DRF
- e complementare il segnale INCTS (nuovo codice tasto).
- Attendere la conferma ODRF = 1. Se il CNC rifiuta il comando, risponde con:  
`NEWGEN && (OCGEN == 1)`
- Scrivere su IDRF12 digit Low il numero asse per volantino 1, e/o su digit High per volantino 2 e/o su IDRF3 digit L per volantino 3 e/o su IDRF3 digit H per volantino 4.
- Scrivere su IFMDRF il fattore di scala; tale fattore, con volantini inseriti, è acquisito dinamicamente. Il fattore di scala è lo stesso per tutti i volantini.
- Chiamare la routine di libreria PLC `avl1_drf()` con la sintassi:
  - ret = avl1\_drf(cvol, nax1, nax2, nax3, nax4)** con i parametri:
    - \* **cvol** = codice numerico dei volantini da inserire:
      - 1 Inserimento VL1
      - 2 Inserimento VL2
      - 3 Inserimento VL1 + VL2
      - 4 Inserimento VL3
      - 5 Inserimento VL1 + VL3
      - 6 Inserimento VL2 + VL3
      - 7 Inserimento VL1 + VL2 + VL3
      - 8 Inserimento VOL4
      - 9 Inserimento VOL1+VOL4
      - 10 Inserimento VOL2+VOL4
      - 11 Inserimento VOL1+VOL2+VOL4



- 12 Inserimento VOL3+VOL4
- 13 Inserimento VOL1+VOL3+VOL4
- 14 Inserimento VOL2+VOL3+VOL4
- 15 Inserimento VOL1+VOL2+VOL3+VOL4

- \* **nax1** = numero asse per VL1 ( uguale a IDR12 digit Low)
- \* **nax2** = numero asse per VL2 ( uguale a IDR12 digit High)
- \* **nax3** = numero asse per VL3 ( uguale a IDR3 digit Low)
- \* **nax4** = numero asse per VL4 ( uguale a IDR3 digit High)

- Testare il codice di ritorno:

- \* ret = 0 o 1 : Richiesta accettata
- \* ret != 0 o 1: Richiesta rifiutata:
  - 2: altro comando in corso
  - 1: codice cvol errato; deve essere 0-:-15
  - 2: asse inesistente per vol. 1
  - 3: asse inesistente per vol. 2
  - 4: asse inesistente per vol. 3
  - 5: stato asse non ok per vol. 1 (non disponibile, in libera servo, indipendente, mandrino, in allarme, disabilitato con <DRF:OFF;asse>.
  - 6: stato asse non ok per vol. 2
  - 7: stato asse non ok per vol. 3
  - 8: solito asse per più volantini
  - 9: opzione non disponibile
  - 10: DRF non abilitato
  - 11: KEY in corso o CNC in allarme
  - 12: modalità volantino già attiva
  - 13: tipo di input volantino non OK: chiamare correttamente iniz\_drf()
  - 14: rapporto scala = 0: chiamare correttamente iniz\_drf()
  - 15: asse inesistente per vol. 4
  - 16: stato asse non ok per vol. 4

Quando il VL1 e/o VL2 e/o VL3 e/o VL4 sono inseriti, il CNC mette ad 1 le uscite **ODRF1** e/o **ODRF2** e/o **ODRF3** e/o **ODRF4**. Tali uscite sono a 0 per volantini DRF disinseriti.

Il CNC legge direttamente il valore dei "conti" dei volantini VL1, VL2, VL3 e VL4 sugli input definiti con iniz\_drf(). Tale valore sarà moltiplicato per **IFMDRF** per ottenere i micron (o gradi/1000) di spostamento. I conti dei volantini sono disponibili sulle word **VVOL**, **VVOL#**, **VOL#IKR** secondo l'hardware presente.

### 2.9.3.2 Disinserimento Volantini

Per disinserire i volantini agganciati in modalità SPAZIO occorre:

- scrivere su ICTS il codice relativo al disinserimento:
  - \* 0xC0 disinserisce tutti i volantini,
- complementare il segnale INCTS (nuovo codice tasto).

Il CNC risponde complementando lo stato del segnale di output **OAKCT**.

Quando VL1 e/o VL2 e/o VL3 sono disinseriti, il CNC mette ad 0 le uscite **OVL1** e/o **OVL2** e/o **OVL3**.

Per disinserire i volantini agganciati in modalità DRF occorre:

- Chiamare la routine di libreria PLC avl1\_drf() con la sintassi:  
**ret = avl1\_drf(0, 0, 0, 0)**
- Testare se il CNC sta lavorando, con la routine di libreria PLC is\_cnc\_work():  
**ret = is\_cnc\_work();**



- Con `ret=0`, scrivere su ICTS il codice relativo al disinserimento:
  - \* **0xC8** disinserisce tutti i volantini DRF
- Complementare il segnale INCTS (nuovo codice tasto).
- Mettere a 0 il segnale CNOK.

Il CNC risponde complementando lo stato del segnale di output **OAKCT** e rimette **CNOK=1**.

Quando il modo DRF è disattivato, il CNC mette ad 0 l'uscita **ODRF**.

Se il CNC rifiuta il comando, risponde con:

```
NEWGEN && (OCGEN == 1)
```

### PRECISAZIONI

- Nel movimento tramite volantini sono attivi i finecorsa software, ma non quelli modificati da Part-Program
- Con [RESET] i volantini vengono disinseriti.
- La richiesta di esecuzione dei KEY viene ignorata fintanto che i volantini sono inseriti.

### 2.9.4 Volantini gestiti in FEED

Con questa prestazione è possibile muovere fino a 3 assi contemporaneamente in modalità simile a quella del JOG di VELOCITÀ. Tale modalità viene attivata dal CNC secondo le normali regole del volantino, quando c'è l'apposita richiesta settata tramite il segnale **IVOLF**.

Tale segnale viene testato dal CNC in fase di aggancio volantini; se è vero, è richiesta la modalità volantini gestiti in FEED, (JOG di VELOCITÀ su più assi).

In questo caso gli ingressi per i volantini:

- **IVJ1** per il volantino 1
- **IVJ2** per il volantino 2
- **IVJ3** per il volantino 3

non esprimono più lo spazio espresso in conti dell'encoder, bensì la FEED desiderata (feed positive = spostamenti positivi, feed negative = spostamenti negativi).

Anche **IVFM** cambia significato.

Nella modalità volantini "NORMALE" (**IVOLF=0**), **IVFM** esprime il fattore di scala del valore dei volantini (valori validi da 1 a 255) cioè il coefficiente moltiplicativo che definisce il peso unitario dei conti del volantino.

Nella modalità "VOLANTINI GESTITI IN FEED" (**IVOLF=1**) **IVFM** stabilisce il fattore moltiplicativo della FEED scritta su **IVJ1 / IVJ2 / IVJ3**.

I tre bit più leggeri definiscono, ciascuno, un fattore fisso di moltiplicazione della feed per i tre volantini:

bit0=0: fattore 1; bit0=1: fattore 10; per volantino 1.

bit1 : idem per volantino 2.

bit2 : idem per volantino 3.

Ad esempio, se **IVFM=2** (bit0=0, bit1=1, bit2=0) e gli ingressi per i volantini valgono:

**IVJ1=** 3000 per il primo volantino

**IVJ2=** 200 per il secondo volantino

**IVJ3=** 1000 per il terzo volantino

le FEED desiderate risulteranno:

**3000** mm/min. per il primo volantino (3000 x 1 = 3000)

**2000** mm/min. per il secondo volantino (200 x 10 = 2000)



1000 mm/min. per il terzo volantino (1000 x 1 = 1000)

La FEED desiderata sarà poi moltiplicata per l'override di lavoro (IOVF).

- Sarà la logica PLC a dover impostare gli ingressi per i volantini congruentemente con la FEED desiderata. Non essendo ammessa una modalità "RAPIDA", l'override interessato ai movimenti da volantini gestiti in FEED sarà sempre e comunque IOVF.
- Per ottenere la max. velocità dell'asse, occorrerà agire sul valore della FEED desiderata, impostando ad esempio il max. positivo (32767 = 0x7FFF) oppure il max. negativo (-32767 = 0xFFFF); il CNC limita comunque la velocità effettiva alla sua RAPIDA.
- L'interpolatore, nel caso di movimentazione da volantini gestiti in feed, seguirà le caratteristiche di accelerazione/decelerazione nominali proprie dell'asse (non viene eseguita l'accelerazione "con curve a S").

#### 2.9.4.1 Gestione tastatore continuo come volantini Feed {F}.

Sulle macchine dotate di tastatore continuo per la digitalizzazione (copiatura) è possibile utilizzare il tastatore stesso per muovere la macchina, come se fosse una penna, senza attivare la modalità COPIA. Può essere utile per rilevare le quote di una serie di punti su un pezzo ed eventualmente memorizzarle con la funzione di "autoapprendimento" per ricostruire un percorso nello spazio.

Ponendo ITSTFE = 1 si predispone il CNC a muovere gli assi selezionati, in modo JOG, come se fosse attivo il modo COPIA e cioè tramite le deflessioni del tastatore.

Vediamo un esempio di uso:

```
/*-----  
TASTATORE COME VOLANTINI FEED  
-----*/  
if(OJOG && RICTVF)  
{ /* RIChiesta attivazione Tastatore come Volantini Feed */  
  if((OVL1 || OVL2 || OVL3) && RICVOL && !ITSTFE)  
  { /* Se Volantini attivi li disattivo */  
    RICVOL = 0;  
  }  
  if(!(OVL1 || OVL2 || OVL3) && !ITSTFE && !RKEYSN && !OKEYSN)  
  {  
    APVOL=0x10;  
    /* Vol1: asse X; Vol2: asse Y */  
    APVOL3=0x02;  
    /* Vol3: asse Z */  
    IVFM= 7;  
    /* Attiva fattore 10 per tastatore */  
    SVOL1=SVOL2=SVOL3=1 ;  
    RICVOL=1;  
    IVOLF=1;  
    ITSTFE=1;  
    /* Attiva gestione tastatore come volantino in feed */  
  }  
}  
else if(ITSTFE)  
{  
  SVOL1=SVOL2=SVOL3=0 ;  
  RICVOL=0;  
  IVOLF=0;  
  ITSTFE=0;  
}
```



```
if(ITSTFE)
{
  DEF001=DEF010=DEF100=0;
  /* annulla selezione scala da tasfuecs() */
}
aggvol();
/* Routine Standard Gestione Volantino */
/*-----*/
```

## 2.9.5 Routines a disposizione del PLC

### 2.9.5.1 Routine "AGGVOL"

Chiamata: **aggvol()**;

Scopo: Gestisce i volantini nelle varie modalità di funzionamento.

Segnali in Ingresso :

**ABILIKR= 1** abilitati volantini su IKR

**ABIKRCAN= 1** abilitati volantini della IKR su CAN

**VOLTAS= 1** abilitato volantino su tastiera

**CR\_ACTIV = 1** è attiva la console remota

**RIMPV1=...** rapporto scala tra impulsi hardware e conti per volantino 1 (valore di default: 4)

**RIMPV2=...** rapporto scala tra impulsi hardware e conti per volantino 2 (valore di default: 4)

**RIMPV3=...** rapporto scala tra impulsi hardware e conti per volantino 3 (valore di default: 4)

I valori ammessi sono: 1, 2, 4.

Per il volantino 4 (usabile solo in DRF) il rapporto scala tra impulsi hardware e conti e' fisso a 4.

**APVOL =...** (char) numero d'ordine (da AXS.TAR) degli assi associati a volantino 1 e 2:

Digit 0 = num. ordine vol. 1 (0...15)

Digit 1 = num. ordine vol. 2 (0...15)

**APVOL3=...** (char) numero d'ordine (da AXS.TAR) dell'asse associato al volantino 3:

Digit 0 = non usato

Digit 1 = num. ordine vol. 3 (0...15)

**APVOL4=...** (char) numero d'ordine (da AXS.TAR) dell'asse associato al volantino 4.

**SVOL1= 1** selezionato volantino 1

**SVOL2= 1** selezionato volantino 2

**SVOL3= 1** selezionato volantino 3

**SVOL4= 1** selezionato volantino 4 (solo per DRF)

**DEF001= 1** definizione = 1/1000 [mm o gradi] a conto

**DEF010= 1** definizione = 10/1000 a conto



---

**DEF100= 1** definizione = 100/1000 a conto

**RICVOL= 1** richiesta aggancio volantini

**RICDRF= 1** richiesta modalità DRF, (Differential Resolver Function)

**RICVFED = 1** richiesta modalità FEED (JOG di velocità su più assi)

Con RICVOL=1, se RICDRF= 0 e RICVFED=0, viene attivata la modalità "Spazio".

**CODITAS=...** (long) filtro per volantino su tastiera (default: (0.040/TAUPLC) + 2)

**CODIKR=...** (long) filtro per volantino su IKR (default: (0.040/TAUPLC) + 2)

**CODICR=...** (long) filtro per volantino su console remota (default: (0.040/TAUPLC) + 17)

I valori dei filtri sono inizializzati la prima volta che si chiama la aggvol() e non vanno, normalmente, modificati; se necessario, possono essere ridefiniti dopo la prima chiamata.

Variabili PLC non riutilizzabili:

### SEQ28

SEQ28 è usato dalla routine ed in particolare segnala, quando SEQ28==0, che non sono attivi i volantini.

I segnali di ingresso della routine RIMPV# sono di tipo unsigned char mentre APVOL e APVOL3 sono di tipo char. I relativi sinonimi sono definiti in DF\_SYN.INC e sono allocati nell'area dati B (512 word riservate ad appoggi per le routines di libreria PLC fornite da ECS).

NOTE:

- La routine gestisce i volantini collegati su SCHEDE ASSI, su TASTIERA, su IKR (Interfaccia Keyboard Remota), su IKR su CAN-Bus o sulla console remota.
- Se si attiva ABIKRCAN, i volantini debbono essere collegati nel seguente modo:
  - NON c'è VOLTAS e ci sono canali asse liberi
    - \* Volantino 1 su IKR
    - \* Volantino 2 su IKR
    - \* Volantino 3 su primo canale encoder (VL0 in SWCNF.TAR)
  - NON c'è VOLTAS e non ci sono canali asse liberi
    - \* Volantino 1 su IKR
    - \* Volantino 2 su IKR
  - C'è VOLTAS
    - \* Volantino 1 su Tastiera
    - \* Volantino 2 su IKR
    - \* Volantino 3 su IKR
- Se si attiva ABILIKR, tutti i volantini debbono essere collegati al modulo IKR.
- Se si attiva VOLTAS, il volantino 1 è quello collegato alla TASTIERA e gli eventuali altri vanno collegati su SCHEDE ASSI:
  - volantino 2 su primo canale encoder (VL0 in SWCNF.TAR)
  - volantino 3 su secondo canale encoder (VL1)
- Con ABIKRCAN, ABILIKR e VOLTAS non attivi, tutti i volantini vanno collegati su SCHEDE ASSI:
  - volantino 1 su primo canale encoder (VL0 in SWCNF.TAR)
  - volantino 2 su secondo canale encoder (VL1)
  - volantino 3 su terzo canale encoder (VL2)



- Se è attiva la console remota (CR\_ACTIV=1) il volantino 1 è quello della console e gli eventuali altri seguono le regole viste sopra.
- Prima di attivare RICDRF è necessario eseguire **iniz\_drf()** (vedi par. 2.9.5.2) per inizializzare correttamente la gestione della modalità DRF. La funzione **iniz\_drf()** è chiamata direttamente da **aggvol()**.
- Se il PLC varia dinamicamente l'associazione assi/volantini o il numero di volantini attivi, la routine dapprima li disinserisce tutti e poi li riaggancia nella nuova configurazione; le variazioni del fattore di scala vengono invece acquisite dinamicamente

**USO:**

- a. definire tipo di input: su tastiera o IKR o assi (VOLTAS, ABILIKR, ABIKRCAN)
- b. definire eventuale rapporto non standard tra conti e impulsi hardware (RINPV#)
- c. definire abbinamento numero ordine assi con volantini impiegati (APVOL, APVOL3, APVOL4)
- d. selezionare i volantini desiderati (SVOL#)
- e. definire fattore moltiplicativo (DEFxxx)
- f. settare richiesta di aggancio volantino "normale" (RICVOL) o DRF (RICDRF e RICVOL)
- g. attivare la routine

**NOTA:**

Se viene usata la routine **tasfuecs()**, che gestisce i tasti funzione del PLC nella configurazione standard ECS definita in BASMENU.MEN, non vanno attuati i punti c. d. e. f. poiché è la routine stessa che vi provvede, selezionando il volantino 1, abbinandolo via via ad uno degli assi CNC, definendo un fattore di scala scelto tra x1 x10 x100 ed attivando o disattivando RICVOL in base ai tasti funzione PLC premuti dall'operatore. Volendo modificare le scelte effettuate con **tasfuecs()**, è possibile agire dopo la chiamata a **tasfuecs()** stessa e prima di chiamare la **aggvol()**.

Per attivare i volantini in modalità DRF, occorre settare RICDRF prima o assieme a RICVOL.

```
/*-----  
ESEMPIO D'USO ROUTINE AGGVOL IN QAL.C  
-----*/  
tasfuecs();  
/* ROUTINE GESTIONE TASTI P1-P8 COME DEFINITI IN BASMENU.MEN */  
if( SELDRFZ )  
/* RICHIESTA SELEZIONE MODALITÀ DRF SU ASSE Z */  
{  
SVOL1= 1; SVOL2= 0; SVOL3= 0;  
/* SOLO VOLANTINO 1 */  
APVOL= 0x02;  
/* SU ASSE Z = 2 */  
DEF001=0;  
DEF010=1;  
DEF100=0;  
/* 10 MICRON A IMPULSO */  
RICDRF=1;  
RICVOL=1;  
}  
else if(RICDRF)  
/* RICHIESTA SGANCIO E DRF ANCORA ATTIVO */  
{  
RICDRF=0;  
RICVOL=0;  
/* DISATTIVA IL VOLANTINO */
```



```

    }
    aggvol();
    /* ROUTINE STANDARD GESTIONE VOLANTINO */
    /*-----*/

```

### 2.9.5.2 Routine "INIZ\_DRF"

Chiamata: **esito=iniz\_drf(type1,scal1,type2,scal2,type3,scal3,type4,scal4);**

Scopo: inizializza i dati necessari per i Volantini in modalità DRF.

La routine è chiamata direttamente da **aggvol()** se è abilitata l'opzione DRF.

Parametri in Ingresso :

**type#** = (unsigned short) Tipo di input per il volante #. Valori ammessi:

- 1 input su SCHEDA ASSI
- 2 input su IKR
- 3 input su TASTIERA

*Attenzione!: con un unico volante collegato alla tastiera è possibile mettere type# = 3 in tutti i campi ed attivare uno qualsiasi dei volantini (SVOL1, SVOL2, SVOL3); non attivare più di un volante alla volta!*

**scal#** = (unsigned short) Rapporto scala tra impulsi hardware e conti per il volante #. Deve essere diverso da 0; il CNC divide il valore dei contatori hardware (normalmente 4 conti ogni ciclo del trasduttore ottico) per questo rapporto; il risultato sarà poi moltiplicato per il fattore di scala **IFMDRF** che può essere variato dinamicamente. **Valori ammessi: 1, 2, 4, 8.**

*Attenzione!: con un unico volante, mettere scal# = 1 (ad es.) in tutti i campi non usati.*

Parametro di uscita:

**esito** = (short) valore di ritorno della routine . Può assumere uno dei seguenti valori:

- 0 Successo – inizializzazione OK
- 1 Insuccesso – primo parametro errato (diverso da 1, 2, 3)
- 2 Insuccesso – secondo parametro errato (diverso da 1, 2, 4, 8)
- 3 Insuccesso – terzo parametro errato
- 4 Insuccesso – quarto parametro errato
- 5 Insuccesso – quinto parametro errato
- 6 Insuccesso – sesto parametro errato
- 7 Insuccesso – settimo parametro errato
- 8 Insuccesso – ottavo parametro errato

```

/*-----
ESEMPIO D'USO ROUTINE INIZ_DRF IN QAL.C
-----*/
if(pon)
{
    ESITODRF= iniz_drf(3,4,1,4,1,4,1,4);
    /* INIZIALIZZA VOLANTINI DRF */
    /* VOL1 SU TASTIERA; ALTRI SU SCHEDE ASSI */
    if(ESITODRF)
    { /* INSUCCESSO */
        INIDRFOK= 0;
        . . . . .
    }
    else

```





```
    { /* INIZIALIZZAZIONE OK */  
      INIDRFOK= 1;  
    }  
  }  
/*-----*/
```

## 2.10 OVERRIDE

I CNC ECS serie WIN permettono all'operatore di regolare dinamicamente la velocità rapida e di lavoro degli assi e la speed del mandrino. A questo scopo sono dedicati tre commutatori (di tipo incrementale ad encoder) posti sulla tastiera BASIC del CNC. In altri tipi di tastiera, sono presenti due potenziometri, uno per rapido e feed assi, l'altro per speed mandrino.

Le routine di libreria PLC **GestOvr3()** e **gestovr()** risolvono in maniera facile la gestione dei potenziometri e dei commutatori usati per il controllo degli override. Nella libreria PLC è presente anche la routine **gestovrc()** da usare in alternativa a **gestovr()** quando è attiva la console remota.

### 2.10.1 Sinonimi utilizzati per gli OVERRIDE

Per la gestione degli override sono presenti vari segnali di scambio tra TASTIERA, CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

I segnali sotto indicati non vanno normalmente utilizzati se vengono usate le routines **GestOvr3()** e/o **gestovr()** e/o **gestovrc()**.

<b>AIOVR</b> (unsigned char)	<b>CNC → PLC</b>	Con tastiera Basic contiene il valore di incremento per l'override di RAPIDO degli assi.
<b>AIOVF</b> (unsigned char)	<b>CNC → PLC</b>	Valore Override F% (0...255) proveniente dal potenziometro assi della tastiera del CNC. Può essere usato sia per l'override RAPIDO che di LAVORO degli assi. Con tastiera Basic contiene il valore di incremento per l'override di LAVORO.
<b>AIOVS</b> (unsigned char)	<b>CNC → PLC</b>	Valore Override S% (0...255) proveniente dal potenziometro Speed della tastiera del CNC. È normalmente usato per l'override Speed del Mandrino. Con tastiera Basic contiene il valore di incremento per l'override Speed.
<b>IOVR</b> (unsigned char)	<b>PLC → CNC</b>	Valore Override RAPIDO Assi in % (0...100) per il CNC.
<b>IOVF</b> (unsigned char)	<b>PLC → CNC</b>	Valore Override Feed di LAVORO Assi in % (0...255) per il CNC.
<b>IOVS</b> (unsigned char)	<b>PLC → CNC</b>	Valore Override Speed Mandrino principale in % (0...255) per il CNC.
<b>IOVSS</b> (unsigned char)	<b>PLC → CNC</b>	Valore Override Speed Mandrino secondario in % (0...255) per il CNC.



<b>OVRFRE#</b> (unsigned char)	<b>PLC → CNC</b>	Valore Override velocità in % (0...255) per l'asse indipendente #. Valido anche in fase di SET asse.
-----------------------------------	------------------	---

## 2.10.2 Routines a disposizione del PLC

### 2.10.2.1 Routine "GESTOVR3"

Chiamata: **GestOvr3()**;

Scopo: Gestisce gli Override assi e mandrino con input dai tre commutatori (di tipo incrementale ad encoder) posti sulla tastiera BASIC del CNC. Gestisce anche l'override assi comandato dall'eventuale console remota ECS.

Parametri in Ingresso :

<b>MAXOVF</b>	Valore massimo dell'override assi in <b>Feed</b>	(0-:-255; default: 200)
<b>MAXOVR</b>	Valore massimo dell'override assi in <b>Rapido</b>	(0-:-100; default 100)
<b>MAXOVS</b>	Valore max dell'override <b>Speed</b> mandrino principale	(0-:-255; default: 200)
<b>MAXOVSS</b>	Valore max dell'override <b>Speed</b> mandrino secondario	(0-:-255; default: 200)
<b>MINOVS</b>	Valore minimo override <b>Speed</b> mandrino principale	(0-:-255; default: 30)
<b>MINOVSS</b>	Valore minimo override <b>Speed</b> mandrino secondario	(0-:-255; default: 30)
<b>AOVSS = 1</b>	L'input per l'override Speed mandrino secondario è separato da quello per il mandrino principale.	
<b>COMOVSS</b>	Valore di input per l'override Speed Mandrino secondario (0-:-255); valido se AOVSS=1.	
<b>FILTRO</b>	Valore di filtro per i disturbi in input sul convertitore A/D (su COMOVSS) per l'override <b>Speed</b> del mandrino secondario. (0-:-255; default: 1)	
<b>FILTROSS</b>	Valore di filtro dell'override <b>Speed</b> mandrino secondario. (0-:-255; default: 0)	
<b>CR_ACTIV</b>	Console remota attiva; l'override Rapido ricopia l'override Feed.	

I valori di ingresso della routine sono tutti di tipo unsigned char; i relativi sinonimi sono definiti in DF\_SYN.INC e sono allocati nell'area dati B (512 word riservate ad appoggi per le routines di libreria PLC fornite da ECS).

**MAXOVR**, **MAXOVF**, **MAXOVS** e **MAXOVSS** sono i valori massimi (in %) che si desiderano per gli override, rispettivamente per i movimenti in rapido ed in lavoro degli assi e per la rotazione del mandrino principale e secondario. **MINOVS** e **MINOVSS** sono l'override minimo che si desidera per il mandrino principale e secondario. Lo Speed override, a differenza di quelli associati agli assi, non dovrebbe mai fermare il mandrino.

Se **AOVSS=1**, il valore di override per la Speed del mandrino secondario viene "prelevato" da **COMOVSS** dove può essere scritto, ad esempio, il valore proveniente da uno dei convertitori A/D (Analogico/Digitale) delle schede Input/Output presenti o della IKR; più avanti (Cap. 6 e 7) sono descritti in dettaglio i tipi, sinonimi e range dei vari convertitori.



**FILTRO** può essere usato per minimizzare i disturbi (variazioni random di poche unità) che possono arrivare dal convertitore A/D; le variazioni inferiori a **FILTRO** non hanno effetto in fase di inversione; si consiglia di non usare valori maggiori di 3 o 4.

**FILTROSS** è da usare per evitare variazioni troppo brusche di velocità agendo sull'override. Se definito (valore > 0), le variazioni sono limitate, nel trasferimento sugli input del CNC, a **FILTRxx** ad ogni **TAUPLC**.

Valori bassi generano una variazione "lenta" dell'override. Valori più alti rendono più pronta la variazione. Valori consigliati: 1 -:- 10.

La routine è utilizzabile solo nel caso di override per rapido, feed e speed di tipo incrementale ad encoder (da 30 scatti ogni giro).

La scalatura di default, cioè la corrispondenza tra posizione encoder e valore associato, è predefinita negli arrays **ovr\_tab[]**, **ovf\_tab[]**, **ovs\_tab[]**:

```
static short ArrSizeR = 30;
/* Numero di elementi dell'array Rapidi */
static unsigned char ovr_tab[256] = {
    0, 1, 2, 3, 4, 5, 6, 7, 8, 10,
    12, 14, 17, 20, 23, 26, 30, 34, 38, 43,
    48, 53, 58, 63, 68, 74, 80, 86, 93,100 };

static short ArrSizeF = 35;
/* Numero di elementi dell'array Feed */
static unsigned char ovf_tab[256] = {
    0, 1, 2, 4, 8, 14, 20, 25, 30, 40,
    50, 60, 70, 75, 80, 85, 90, 92, 94, 96,
    98,100,102,104,106,110,120,130,140,150,
    160,170,180,190,200 };

static short ArrSizeS = 35;
/* Numero di elementi dell'array Speed */
static unsigned char ovs_tab[256] = {
    30, 40, 45, 50, 55, 60, 65, 70, 75, 80,
    85, 90, 92, 94, 96, 98,100,102,104,106,
    108,110,115,120,125,130,135,140,145,150,
    160,170,180,190,200 };

/* INIZIALIZZA CON VALORI DI DEFAULT OVR OVF E OVS */
```

La scalatura del mandrino principale e secondario è identica.

Se si desidera cambiare la scalatura, prima della chiamata alla routine **GestOvr3()** vanno chiamate le routine **ovF\_set()**, **ovS\_set()**, **ovR\_set()** con l'indirizzo degli arrays (di max 256 elementi) dei valori desiderati (vedi esempio più avanti).

I valori percentuali massimi e minimi possono essere modificati anche dinamicamente (ad esempio in caso di set non fatto o di ripari aperti).

```
/*-----
ESEMPIO D'USO ROUTINE GestOvr3 IN QAL.C
- per scalatura del RAPIDO su 21 scatti del 5%
- per scalatura non lineare della FEED su 40 scatti
- per scalatura non lineare della SPEED su 50 scatti
-----*/
```



```
static unsigned char ovR[ ] = {
    0, 5, 10, 15, 20, 25, 30, 35, 40, 45,
    50, 55, 60, 65, 70, 75, 80, 85, 90, 95,
    100 };

static unsigned char ovF[ ] = {
    0, 1, 2, 4, 8, 14, 20, 25, 30, 40,
    50, 60, 70, 75, 80, 85, 90, 92, 94, 96,
    98,100,102,104,106,110,115,120,125,130,
    135,140,150,160,170,180,190,200,225,250 };

static unsigned char ovS[ ] = {
    10, 15, 20, 25, 30, 35, 40, 45, 50, 55,
    60, 65, 70, 75, 80, 82, 84, 86, 88, 90,
    92, 94, 96, 98,100,102,104,106,108,110,
    112,114,116,118,120,125,130,135,140,145,
    150,155,160,165,170,175,180,185,190,200 };

void gal( void )
{

if(pon)
{
    lotime(46,0.3);
    /* SETTA RITARDO DI 0.3 SEC PER MANDRINO */
    ovR_set(&ovR[0],sizeof(ovR));
    ovF_set(&ovF[0],sizeof(ovF));
    ovS_set(&ovS[0],sizeof(ovS));
    MAXOVR = 100;
    MAXOVF = 250;
    MINOVS = 10;
    MAXOVS = 200;
    MINOVSS = 10;
    MAXOVSS = 200;
}

if(!(OBSE0 && OBSE1 && OBSE2))
{
    MAXOVR= MAXOVF= 40;
    /* OVERRIDE LIMITATO AL 40% PRIMA DEL SET ASSI */
}
else
{
    MAXOVR= 100;
    MAXOVF= 250;
}

GestOvr3();
/* ROUTINE STANDARD GESTIONE TRE OVERRIDE AD ENCODER */

TIME_46= ABMND;
/* RITARDO 0.3 sec ABILITAZIONE MANDRINO */
if((AM03 || AM04) && ABMND && !RITP_46)
    IOVS= 0;
/* FORZA OVERRIDE SPEED A ZERO ALL'ABILITAZIONE */
/*-----*/
```



### 2.10.2.2 Routine "GESTOVR"

Chiamata: **gestovr()**;

Scopo: Gestisce gli Override assi e mandrino con input da potenziometri.

Parametri in Ingresso :

<b>MAXOVF</b>	Valore massimo dell'override assi in <b>Feed</b>	(0-:-255; default: 120)
<b>MAXOVR</b>	Valore massimo dell'override assi in <b>Rapido</b>	(0-:-100; default 100)
<b>MAXOVS</b>	Valore max dell'override <b>Speed</b> mandrino principale	(0-:-255; default: 150)
<b>MAXOVSS</b>	Valore max dell'override <b>Speed</b> mandrino secondario	(0-:-255; default: 100)
<b>MINOVS</b>	Valore minimo override <b>Speed</b> mandrino principale	(0-:-255; default: 50)
<b>MINOVSS</b>	Valore minimo override <b>Speed</b> mandrino secondario	(0-:-255; default: 50)
<b>FILTRO</b>	Valore di filtro per i disturbi in input sui convertitori A/D	(0-:-255; default: 1)
<b>FILTROF</b>	Valore di filtro dell'override assi in <b>Feed</b>	(0-:-255; default: 0)
<b>FILTROR</b>	Valore di filtro dell'override assi in <b>Rapido</b>	(0-:-255; default: 0)
<b>FILTROS</b>	Valore di filtro dell'override <b>Speed</b> mandrino princip.	(0-:-255; default: 0)
<b>FILTROSS</b>	Valore di filtro dell'override <b>Speed</b> mandrino second.	(0-:-255; default: 0)
<b>AOVR = 1</b>	L'input per l'override Rapido è separato da quello per la Feed.	
<b>COMOVR</b>	Valore di input per l'override Rapido (0-:-255); valido se AOVR=1.	
<b>AOVSS = 1</b>	L'input per l'override Speed mandrino secondario è separato da quello per il mandrino principale.	
<b>COMOVSS</b>	Valore di input per l'override Speed Mandrino secondario (0-:-255);valido se AOVSS=1.	

I valori di ingresso della routine sono tutti di tipo unsigned char; i relativi sinonimi sono definiti in DF\_SYN.INC e sono allocati nell'area dati B (512 word riservate ad appoggi per le routines di libreria PLC fornite da ECS).

MAXOVR, MAXOVF, MAXOVS e MAXOVSS sono i valori massimi (in %) che si desiderano per gli override, rispettivamente per i movimenti in rapido ed in lavoro degli assi e per la rotazione del mandrino principale e secondario. MINOVS e MINOVSS sono l'override minimo che si desidera per il mandrino principale e secondario. Lo Speed override, a differenza di quelli associati agli assi, non deve infatti fermare il mandrino.

FILTROR, FILTROF, FILTROS e FILTROSS sono da usare per evitare variazioni troppo brusche di velocità agendo sugli overrides. Se definiti (valore > 0), le variazioni dei corrispondenti override sono limitate, nel trasferimento sugli input del CNC, a FILTRxx ad ogni TAUPLC.

Valori bassi generano una variazione "lenta" dell'override. Valori più alti rendono più pronta la variazione. Valori consigliati: 1 -:- 10.



La routine gestovr() fa in modo che i valori assunti dagli override varino in modo lineare tra i minimi e massimi impostati, in base al valore assunto dall'ingresso analogico associato.

La routine legge il valore di input per gli override assi da **AIOVF** (unsigned char), scritto dal driver di tastiera e lo filtra per eventuali disturbi ( le variazioni inferiori a **FILTRO** non hanno effetto in fase di inversione; si consiglia di non usare valori maggiori di 3 o 4).

Il dato viene poi messo in scala in base al valore massimo impostato (MAXOVx) e trasferito sugli inputs del CNC **IOVR** e **IOVF**; ad esempio, con i valori di default, con potenziometro a fondo scala (AIOVF=255) si ottiene IOVR=100 e IOVF=120.

In modo simile vengono elaborati **AIOVS** (e **COMOVSS**) per ottenere **IOVS** (e **IOVSS**); in questo caso c'è anche da considerare il valore minimo MINOVS (e MINOVSS); con i valori di default, con potenziometro a zero si ha IOVS=IOVSS=50 ed a fondo scala (AIOVS=COMOVSS=255) si ottiene IOVS=150 e IOVSS=100.

Se **AOVR=1**, il valore di override per i Rapidi degli assi viene "prelevato" da **COMOVR** anziché da AIOVF; in COMOVR può essere scritto, ad esempio, il valore proveniente da uno dei convertitori A/D (Analogico/Digitale) delle schede Input/Output presenti o della IKR; più avanti (Cap. 6 e 7) sono descritti in dettaglio i tipi, sinonimi e range dei vari convertitori.

Se **AOVSS=1**, il valore di override per la Speed del Mandrino Secondario viene "prelevato" da **COMOVSS** anziché da AIOVS.

Volendo porre dei limiti alle velocità massime degli assi in particolari situazioni, si può agire direttamente su IOVR e/o IOVF dopo la chiamata alla routine (vedi esempio).

```
/*-----  
ESEMPIO D'USO ROUTINE gestovr() IN QAL.C  
-----*/  
if(pon)  
{  
/* SETTAGGI AL POWER ON) */  
lotime(46,0.3);  
/* SETTA RITARDO DI 0.3 SEC PER MANDRINO */  
MAXOVS= 140;  
MAXOVF= 110;  
MINOVS= 20;  
/* DEFINIZIONE VALORI MAX E MIN DEGLI OVERRIDE */  
}  
. . . .  
FILTRO= 2;  
/* VALORE FILTRO PER DISTURBI */  
FILTROR= 4;  
/* VALORE FILTRO PER VARIAZIONE OVERRIDE RAPIDO */  
FILTROF= 4;  
/* VALORE FILTRO PER VARIAZIONE OVERRIDE FEED */  
FILTROS= 5;  
/* VALORE FILTRO PER VARIAZIONE OVERRIDE MANDRINO */  
AOVR= 0;  
/* OVERRIDE RAPIDO DA POTENZIOMETRO OVERRIDE FEED */  
gestovr();  
/* ROUTINE STANDARD GESTIONE OVERRIDE */  
TIME_46= ABMND;  
/* RITARDO 0.3 sec ABILITAZIONE MANDRINO */  
if((AM03 || AM04) && ABMND && !RITP_46)  
IOVS= 0;
```



```
/* FORZA OVERRIDE SPEED A ZERO ALL'ABILITAZIONE */
if(!(OBSE0 && OBSE1 && OBSE2) && IOVF>=40)
  IOVR= IOVF= 40;
/* OVERRIDE LIMITATO AL 40% PRIMA DEL SET ASSI */
if(AMODREG && IOVF>=25)
  IOVR= IOVF= 25;
/* OVERRIDE LIMITATO AL 25% IN MODO REGOLAZIONE */
/*-----*/
```

### 2.10.2.3 Routine "GESTOVR"

Chiamata: **gestovrc()**;

Scopo: Gestisce gli Override assi nel caso di override per feed e rapido comandati da console remota ECS in presenza di tastiera ECS con override a potenziometri gestita dalla routine standard `gestovr()`. Non va usata se si usa la `GestOvr3()` per tastiere con override ad encoder.

Parametri in Ingresso :

**MAXOVF** Valore massimo dell'override assi in Feed (0:-255)

#### Attenzione!

La routine `gestovrc()` è incompatibile con la `GestOvr()` e va chiamata, in alternativa a `gestovr()`, quando è attiva la console remota ECS (**CR\_ACTIV=1**).

La scalatura di default, cioè la corrispondenza tra posizione encoder e valore associato, è predefinita nell'array `ovf_tab[]`:

```
static short ArrSizeF = 35;
/* Numero di elementi dell'array Feed */
static unsigned char ovf_tab[256] = {
    0, 1, 2, 4, 8, 14, 20, 25, 30, 40,
    50, 60, 70, 75, 80, 85, 90, 92, 94, 96,
    98,100,102,104,106,110,120,130,140,150,
    160,170,180,190,200 };

/* INIZIALIZZA CON VALORI DI DEFAULT OVF */
```

Se si desidera cambiare la scalatura, prima della chiamata alla routine `gestovrc()` va chiamata la routine **ovF\_setc()** con l'indirizzo dell'array (di max 256 elementi) dei valori desiderati (vedi esempio).

```
/*-----
ESEMPIO D'USO ROUTINE gestovrc IN QAL.C
- per scalatura non lineare su 40 scatti
-----*/

static unsigned char ovF[ ] = {
    0, 1, 2, 4, 8, 14, 20, 25, 30, 40,
    50, 60, 70, 75, 80, 85, 90, 92, 94, 96,
    98,100,102,104,106,110,115,120,125,130,
    135,140,150,160,170,180,190,200,225,250 };
```



```
void gal( void )
{
    if(pon)
    {
        ovF_setc(&ovF[0],sizeof(ovF));
        MAXOVF = 250;
    }

    if(CR_ACTIV)
    {
        gestovrc();
    }
    else
    {
        gestovr();
        /* ROUTINE STANDARD GESTIONE OVERRIDE A POTENZIOMETRI */
    }
}
/*-----*/
```

## 2.11 Gestione per assi con azionamenti digitali ECSSLINK, SERCOS o Mechatrolink.

Dalla Release SW Fresa V2.00 (Tornio V1.0), è stata inserita la gestione degli azionamenti digitali ECS di tipo **ECSSLINK IDD**; tali azionamenti sono adatti per il controllo dei Motori per assi e sono completamente gestiti dal CNC che colloquia con essi tramite segnali digitali attraverso un collegamento realizzato con Flat-cable. Tale collegamento permette una più completa e precisa gestione della comunicazione tra CNC ed Azionamento, evitando i problemi tipicamente associati alla circuiteria Analogica (differenze di masse, disturbi, bassa risoluzione, numero di "fili" e quindi di I/O impegnati, ecc...).

Dalla Release SW Fresa V2.3 (Tornio V3.0), è stata inserita la gestione degli azionamenti digitali ECS di tipo ECSSLINK **EDD**, con encoder anziché resolver.

Dalla Release SW Fresa V2.01 (sul Tornio dalla V3.0), è stata inserita la gestione degli azionamenti digitali **Sercos** con collegamento su fibra ottica ad anello (**Sercans**) e chiusura del ciclo di spazio interna all'azionamento stesso. Tali azionamenti permettono un miglior controllo dell'asse eliminando anche tutte le problematiche derivanti da collegamenti elettrici.

Dalla Release SW V4.00, è stata inserita la gestione degli azionamenti digitali, per assi, **Yaskawa** con collegamento su Bus con doppiino twistato/schermato (**Mechatrolink**) e chiusura del ciclo di spazio interna all'azionamento stesso. Tali azionamenti permettono un miglior controllo dell'asse eliminando anche tutte le problematiche derivanti da collegamenti di tipo analogico. Tali azionamenti non sono, per ora, adatti per assi di copiatura (opzione digitalizzazione).

### 2.11.1 Sinonimi utilizzati per gli assi digitali

Per la gestione di ciascun asse dotato di azionamento digitale ECSSLINK, Sercos o Mechatrolink tramite il PLC, sono presenti vari segnali di scambio tra CNC e PLC che sono





stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

Nei sinonimi dove compare il carattere "#", occorre sostituire ad esso il numero d'ordine dell'asse al quale il segnale è riferito. Tale numero, secondo il tipo di CNC, sarà compreso tra 0 e 2 (1801, 1802), tra 0 e 5 (2801) o tra 0 e 15 (4801, 4802).

Esempi:

**DGDROK0** Driver Digitale relativo all'asse di ordine 0 OK.

**DGDROK2** Driver Digitale relativo all'asse di ordine 2 OK.

**DGDROK11** Driver Digitale relativo all'asse di ordine 11 OK.

Tutti i segnali qui sotto descritti (eccetto IDGTAR) sono usati dagli azionamenti comandati tramite ECSLINK, per gli assi Sercos valgono solo quelli indicati con **{S}**, per gli assi Mechatrolink valgono solo quelli indicati con **{M}**.

<b>DGMICR#</b>	<b>CNC → PLC</b>	<u>Micro di Zero Asse # da Driver Digitale (ECSLINK)</u> Messo a 1 quando il Micro di Zero dell'asse # (collegato direttamente sulla morsettiera del Driver) è chiuso.
<b>DGVERS#</b>	<b>CNC → PLC</b>	<u>Versione del Driver Digitale (ECSLINK) Asse #.</u> 0 = IDD, 1 = EDD
<b>DGSETP#</b>	<b>CNC → PLC</b>	<u>Set parametri velocità del Driver Digitale (ECSLINK) Asse # in corso.</u>
<b>DGMON#</b>	<b>CNC → PLC</b>	<u>Dati Monitor Validi Driver Digitale (ECSLINK) Asse #.</u> Segnala che i dati presenti sulla word <b>ODGMON#</b> corrispondente sono validi e possono essere letti.
<b>DGPOW#</b> <b>{S}{M}</b>	<b>CNC → PLC</b>	<u>Potenza Presente sul Driver Digitale Asse #.</u> Segnala che l'alimentazione di potenza del Driver è connessa.
<b>DGDROK#</b> <b>{S}{M}</b>	<b>CNC → PLC</b>	<u>Driver Digitale dell'Asse # OK.</u> DGDROK#=1 quando il Driver funziona correttamente e non sono presenti Allarmi. Il segnale è anche presente in "forma Hardware" sull'apposita uscita della morsettiera dei Drive ECS. Con DGDROK#=1, il PLC può richiedere l'abilitazione "via Hardware" del Drive ECS stesso tramite l'apposito ingresso presente sulla sua morsettiera. Con DGDROK#=0, il PLC <u>deve mettere in Blocco il CNC</u> e deve poi (per Drivers ECS) comandare il reset della situazione di Allarme (ad esempio quando l'operatore comanda un RESET del CNC) tramite il segnale DRESET#.
<b>DGABIL#</b> <b>{S}{M}</b>	<b>CNC → PLC</b>	<u>Driver Digitale Asse # Abilitato.</u> DGABIL#=1 quando è presente sia <u>l'abilitazione Software</u> generata dal CNC e "comunicata" tramite Flat-cable sia <u>l'abilitazione Hardware (se configurata)</u> sull'apposito ingresso presente sulla sua morsettiera ed il Driver non è in allarme. Per Driver Sercos e Mechatrolink, esiste solo l'abilitazione Software, comandata tramite collegamento digitale.



<p><b>ODGMON#</b> (short)</p>	<p><b>CNC → PLC</b></p>	<p><u>Variabile Monitor di Output del Driver Digitale (ECS) per Asse #.</u> Se <b>DGMON#=1</b> (Dati Monitor Validi), contiene i dati che il Driver comunica al CNC e PLC (può essere la corrente massima o media o la velocità a seconda delle predisposizioni fatte sul Driver stesso). In caso di Allarme sul Driver ( <b>DGDROK#=0</b>) sulla parte leggera è presente il codice dell'allarme stesso.</p>
<p><b>DRESET#</b></p>	<p><b>PLC → CNC</b></p>	<p><u>Comando di Reset Allarme Driver Digitale (ECS) Asse # KO.</u> Il PLC, mettendo <b>DRESET#=1</b> per l'asse che segnala una situazione di allarme con <b>DGDROK#=0</b>, richiede il reset dell'allarme stesso. Se l'allarme è "ripristinabile" e cioè dovuto ad una situazione transitoria non più presente, il drive risponderà rimettendo <b>DGDROK#=1</b> e sarà pronto a funzionare nuovamente. In caso contrario, <b>DGDROK#</b> rimarrà =0 e sarà necessario intervenire per eliminare la situazione di allarme. Il codice di allarme può essere letto sulla parte leggera della word <b>ODGMON#</b> corrispondente purché sia <b>DGMON#=1</b> (Dati Monitor Validi).</p>
<p><b>IDGOFF#</b> {S}{M}</p>	<p><b>PLC → CNC</b></p>	<p><u>Richiesta disinserzione Driver Digitale Asse #.</u> Mettere <b>IDGOFF=1</b> per permettere lo spegnimento del Driver Digitale <b>ECSLINK</b> usato per l'asse # senza che intervengano allarmi per mancanza colloquio tra CNC e Driver stesso. Permette, ad esempio, di spegnere l'azionamento di una tavola o testina motorizzata quando non sono montati sulla Macchina Utensile. Se tale situazione può essere presente anche in fase di accensione del CNC, occorre anche mettere <b>ALR_AXDG=OFF</b> in <b>SWCNF.TAR</b> . Quando la testina o tavola vengono nuovamente montati, collegati ed il relativo Driver alimentato, il PLC deve rimettere <b>IDGOFF#=0</b> per chiedere al CNC di riattivare il controllo del colloquio. Naturalmente il SET dell'asse viene perso ed occorre rifarlo. Per assi con Driver Digitali <b>Sercos</b>, l'azionamento relativo viene disabilitato ma non può essere spento altrimenti si interrompe la continuità del collegamento sulla fibra ottica che collega ad anello i vari azionamenti presenti. In questo caso il SET non viene perso, poiché, per gli assi Sercos, la perdita del SET viene gestita dal Drive. Quanto suddetto vale anche per i Driver Digitali <b>Mechatrolink</b>. Il CNC non comunica, anche in caso di "certi" allarmi, il SET "perso" se anche il Drive non comunica tale stato. <b>IDGOFF#</b> è quindi un segnale analogo a <b>IAXOFF#</b>; in più non viene comunicato allarme in caso di spegnimento del DRIVE con <b>IDRVON# = 1</b>.</p>



<b>IDGTAR#</b> {S}	<b>PLC → CNC</b>	<p>Richiesta modo tarature per Driver Digitale Sercos Asse #.</p> <p>Mettere IDGTAR=1 per permettere la taratura del Driver Digitale Sercos usato per l'asse #, tramite l'apposito Applicativo Software INDRAMAT "DRIVETOP", senza che intervengano allarmi. Il CNC sgancia l'asse (<b>OBAG#=0</b>), rendendolo non disponibile (come se fosse DSP=N in AXS.TAR) e resetta l'eventuale SET già fatto (<b>OBSE#=0</b>).</p> <p>Il segnale IDGTAR# non viene preso in considerazione per assi agganciati in GANTRY, oppure dichiarati GANTRY da taratura (COST o COSTUSER).</p> <p>N.B.: questo segnale può essere utilizzato per l'integrazione "pulita" dell'applicativo "DRIVETOP", potendo gestire in modo dinamico la modalità "TARATURA" ed il ritorno a quella "CNC". <b>In caso non si utilizzi tale segnale è opportuno e consigliabile spegnere e riaccendere il CNC una volta terminata la fase di taratura e prima di rientrare in modalità "CNC"</b>; questo al fine di evitare pericolosi "colpi" sugli assi dovuti al "disallineamento" delle quote interne al CNC, non essendo quest'ultimo stato informato della modalità "TARATURA".</p>
<b>OCURR#</b> {S} (short)	<b>CNC → PLC</b>	<p>Valore percentuale della corrente assorbita dall'asse # (mandrino od asse). # identifica il numero d'ordine dell'asse.</p> <p>Il valore di OCURR# è espresso in "per cento" della corrente continuativa motore con asse bloccato.</p> <p>Dato valido solo per azionamenti SERCOS. Sul driver il parametro corrispondente è S-0-0084 (dato percentuale). In effetti il driver trasmette normalmente il valore moltiplicato per 10 e quindi, ad esempio, un valore di 123 su OCURR# indica una corrente del 12,3%. Il valore della corrente continuativa motore con asse bloccato può essere visto, tramite l'apposito Applicativo Software INDRAMAT "DRIVETOP", sul parametro S-0-0111.</p>

## 2.11.2 Routines a disposizione del PLC

### 2.11.2.1 Routine "IDDDRIVE"

Chiamata: **idddrive()**;

Scopo: Gestisce i segnali per gli assi con azionamenti digitali ECSLINK, Sercos o Mechatrolink.

Segnali in Ingresso :

**DRESET#= 1** Comando di Reset Allarme Driver Digitale Asse # KO (solo per ECSLINK).

Segnali in Uscita:

- Tutti quelli segnati **CNC → PLC** nella tabella del paragrafo precedente
- **MECS10= 1** Uno o più Driver Digitali IDD sono in Allarme; la causa (codice e descrizione sintetica) è visualizzata nella finestra "Allarmi del PLC" sul video del CNC.



---

- **MECS11= 1** Uno o più Driver Digitali EDD sono in Allarme; la causa (codice e descrizione sintetica) è visualizzata nella finestra "Allarmi del PLC" sul video del CNC.

NOTA:

Gli allarmi sopra indicati non vengono attivati per gli assi messi OFF dal PLC (IDGOFF#= 1).

```
/*-----  
ESEMPIO D'USO ROUTINE idddrive IN QAL.C  
-----*/  
iddrive();  
/* Routine standard gestione Drives Digitali */  
AZXOK= DGDROK0;  
AZYOK= DGDROK1;  
AZZOK= DGDROK2;  
/* Azionamenti OK */  
DRESET0= !AZXOK && ORST;  
/* Reset Azionamento Asse X Drive 0 */  
DRESET1= !AZYOK && ORST;  
/* Reset Azionamento Asse Y Drive 1 */  
DRESET2= !AZZOK && ORST;  
/* Reset Azionamento Asse Z Drive 2 */  
  
IDGOFF0= !MUON;  
IDGOFF1= !MUON;  
IDGOFF2= !MUON;  
IDGOFF3= !MUON;  
/* Sospensione Allarmi Drivers con Ausiliari Spenti */  
  
RABIL0= AZXOK && ABIPOT ;  
/* RICHIESTA ABILITAZIONE ASSE X ORDINE 0 */  
RABIL1= AZYOK && ABIPOT ;  
/* RICHIESTA ABILITAZIONE ASSE Y ORDINE 1 */  
RABIL2= AZZOK && ABIPOT ;  
/* RICHIESTA ABILITAZIONE ASSE Z ORDINE 2 */  
  
aggax() ;  
/* Routine Standard Aggancio ASSI */  
  
ABILX= ABIL0 ;  
/* OUTPUT ABILITAZIONE ASSE X */  
ABILY= ABIL1 ;  
/* OUTPUT ABILITAZIONE ASSE Y */  
ABILZ= ABIL2 ;  
/* OUTPUT ABILITAZIONE ASSE Z */
```

## CAPITOLO 3

### 3 ASSI: FUNZIONI SPECIALI

In questo capitolo vengono trattati alcuni aspetti "speciali" della gestione assi realizzata sui CNC ECS della serie WIN, non descritti oppure solo accennati nel capitolo ASSI.

Queste funzioni non sono quasi mai usate sulle "normali" Macchine Utensili.

#### 3.1 COMPENSAZIONE TERMICA

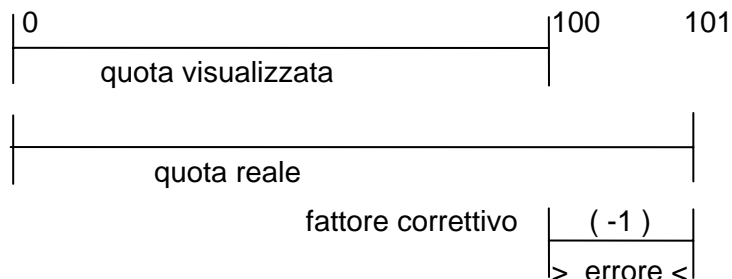
I CNC ECS della serie WIN, oltre alla correzione tabellare errori assi, descritta nel Manuale Tarature 720P385, permettono una correzione da input esterno, tramite il PLC.

La variazione di temperatura può produrre un'effettiva variazione di posizione di un asse. Questo valore, da rilevare tramite appositi strumenti di misurazione (LASER, ...), è il parametro che la correzione dell'errore asse da PLC aggiunge algebricamente alla quota effettiva dell'asse per posizionarlo alla quota desiderata lasciando inalterata la quota visualizzata.

Prendiamo in esame quanto segue: se tutte le volte che si muove l'asse X alla quota 100 si esegue una misurazione della quota tramite un'apparecchiatura esterna e si rileva che l'asse non si trova alla quota 100 (quota visualizzata) ma alla quota 101 (quota reale), si è verificato un errore:

$$\text{ERRORE} = \text{quota visualizzata (100)} - \text{quota reale (101)} = -1$$

Il valore -1 è il parametro da dare in input al CNC, tramite il PLC, per fare in modo che l'asse X sia posizionato realmente sulla quota X=100.



Se inoltre, effettuando varie misure, si rileva che l'errore varia al variare della temperatura, si può determinare una regola (un algoritmo) di correzione in base alla temperatura.

#### 3.1.1 Sinonimi utilizzati per la compensazione da PLC

Per la gestione della compensazione degli assi tramite il PLC, sono presenti appositi input che sono stati associati ai sinonimi qui di seguito elencati.

Dove compare il carattere "#", occorre sostituire ad esso il numero d'ordine dell'asse al quale il segnale è riferito. Tale numero, secondo il tipo di CNC, sarà compreso tra 0 e 2 (1801, 1802), tra 0 e 5 (2801) o tra 0 e 15 (4801, 4802).

Esempi:

**ICORAX0** correzione relativa all'asse di ordine 0.

**ICORAX4** correzione relativa all'asse di ordine 4.



<b>ICORAX#</b> (double)	<b>PLC → CNC</b>	Valore di correzione esterna per l'asse di ordine #. Il valore è di tipo double (numero reale in doppia precisione) ed è espresso in <b>micron</b> ; può assumere quindi anche valori frazionari di micron.
----------------------------	------------------	--

La gestione di tali compensazioni è totalmente a cura del PLC. I tempi di risposta delle compensazioni esterne, per tutti gli assi configurati a tale prestazione, sono garantiti essere il tau successivo a quello della logica PLC.

L'attivazione e la disattivazione delle compensazioni è comandata dal PLC tramite apposite funzioni della libreria PLC. Le compensazioni vengono in ogni modo azzerate dal Controllo Numerico all'accensione ed al passaggio ALT/EXE.

Le correzioni sono eseguite dal CNC anche se l'asse è indipendente (anche configurato Indipendente in taratura, ad esclusione dell'asse mandrino). Esse sono congelate se l'asse è bloccato meccanicamente, è in libera servo (OLS#=1) o in emergenza (OBOK#=0) o fuori servizio o non agganciato o agganciato in modo velocità.

Le correzioni attuali non influiscono sulla presa di zero dell'asse (sono cioè considerate nulle mentre l'asse esegue la procedura di SET).

Sugli stessi assi sui quali è attiva la compensazione da PLC, può essere contemporaneamente attiva la compensazione tabellare; in questo caso, il risultato ottenuto è la somma algebrica dei due valori.

### 3.1.2 Funzioni per Attivare/Disattivare la compensazione Asse da PLC

#### 3.1.2.1 start\_corr\_ext: Attiva la compensazione Asse da PLC

Chiamata: **esito = start\_corr\_ext (num\_ax, cvmed)**

Parametri di ingresso:

**num\_ax** = (unsigned short) Numero d'ordine dell'asse per il quale s'intende attivare la compensazione. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**cvmed** = (unsigned short) Flag di input indicante se calcolare il valore medio dei valori di correzione arrivati da PLC, scartando il valore minimo ed il valore massimo.

**cvmed != 0** indica che occorre calcolare il valore medio.

**cvmed = 0** indica che non occorre calcolare il valore medio.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- 0**      Successo; attivata la correzione sull'asse indicato.
- 1**      Numero di asse errato (errore sul primo parametro).
- 2**      C'è già un comando in corso.
- 3**      Correzione errore già attiva sull'asse specificato.

Il CNC effettua la compensazione errori gestita da PLC prendendo in considerazione un valore di errore (per ogni asse) calcolato secondo una media degli ultimi 10 valori accumulati scartando i valori minimo e massimo. Questo filtraggio presenta il vantaggio, e la sicurezza, che il CNC possa escludere eventuali disturbi arrivati, ad esempio, in input dalla sonda termica, ma anche il limite che il CNC non



---

---

possa acquisire un valore di correzione esplicitamente voluto da PLC in un tempo di reazione immediato. A tal fine è stata inserita la possibilità di includere oppure escludere il filtraggio da parte del CNC degli errori di INPUT da PLC tramite il flag **cvmed**.

Dalla release software V2.02 non è più compito del PLC evitare variazioni brusche del valore della correzione per non provocare colpi o allarme fuga sull'asse interessato; il CNC attua una "interpolazione" della correzione, applicando un valore massimo, al Tau, pari all'accelerazione di taratura dell'asse.

### 3.1.2.2 stop\_corr\_ext: Disattiva la compensazione Asse da PLC

Chiamata: **esito = stop\_corr\_ext (num\_ax, command)**

Parametri di ingresso:

**num\_ax** = (unsigned short) Numero d'ordine dell'asse per il quale s'intende disattivare la compensazione. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**command** = (unsigned short) Specifica il tipo di comando di disattivazione:

- 1 disattiva la correzione esterna ed impone correzione = 0.
- 4 disattiva la correzione esterna non alterando il valore dell'ultima correzione.
- 5 disattiva la correzione esterna rendendola permanente (fino a nuovo SET).

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- 0 Successo; disattivata la correzione sull'asse indicato.
- 1 Numero di asse errato (errore sul primo parametro).
- 2 C'è già un comando in corso.
- 4 Correzione errore non attiva sull'asse specificato.
- 5 Codice su command non previsto.

Note sul comando con **command=5**:

Tale comando "rifasa" i valori di QS, cioè somma alla posizione attuale dell'asse la quota correzione comandata, annullando allo stesso tempo il valore ultimo di correzione passatogli dal PLC. Tale comando è immediatamente acquisito dal CNC, ma viene eseguito solamente in condizioni di CNC fermo (CNC non in esecuzione, part-program non sospeso, buffers interni di interscambio svuotati). Fintanto il CNC non è fermo, il comando rimane "appeso", non consentendo perciò altri comandi di attivazione/disattivazione delle correzioni esterne. In questa circostanza, le chiamate a start\_corr\_ext() o stop\_corr\_ext() sopra citate, ritornerebbero il codice 2 di comando in corso.

E' possibile controllare lo stato di CNC fermo tramite la funzione is\_cnc\_work():

```
ret = is_cnc_work();  
* ret = 1 : il CNC lavora  
* ret = 0 : il CNC non lavora
```

### 3.1.2.3 is\_corr\_ext: Testa se è attiva la compensazione Asse da PLC

Chiamata: **esito = is\_corr\_ext (num\_ax)**

Parametro di ingresso:

**num\_ax** = (unsigned short) Numero d'ordine dell'asse per il quale testare se è attiva la compensazione. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- 0 Correzione non attiva o numero di asse errato.



1      Correzione attiva sull'asse specificato.

### 3.1.2.4 is\_cmd\_corr: Testa se c'è un comando comp. asse da PLC in corso

Chiamata: **esito = is\_cmd\_corr (num\_ax)**

Parametro di ingresso:

**num\_ax** = (unsigned short) Numero d'ordine dell'asse per il quale testare se è in corso un comando per la compensazione. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- 0**      Comando non in corso o numero di asse errato.
- 1**      Comando in corso sull'asse specificato.

### 3.1.3 Esempio di compensazione termica assi da PLC

Qui di seguito è riportato un frammento di PLC che effettua una semplice correzione termica degli assi Y e Z di una fresatrice:

```
/*-----*/  
/* CORREZIONE TERMICA ASSI Y    Z */  
/*-----*/  
if (pon)  
{  
  start_corr_ext(1,1);  
  start_corr_ext(2,1);  
  /* Attiva correzione termica assi Y e Z con filtro */  
  read_var(VL_FLOT,10000,600,-1, & CTEMPY);  
  /* Coeff. dilatazione termica Y da #300 scritto in COSTUSER */  
  read_var(VL_FLOT,10000,602,-1, & CTEMPZ);  
  /* Coeff. dilatazione termica Z da #301 scritto in COSTUSER */  
}  
ICORAX1= (double)(RSTZC-RSTZF)*CTEMPY ;  
/* CORREZIONE ASSE Y */  
/* RSTZF: Sonda termica zona fredda */  
/* RSTZC: Sonda termica zona calda */  
ICORAX2= (double)(RTC2-RTC1)*TEMPZ ;  
/* CORREZIONE ASSE Z */  
/*-----*/
```

## 3.2 FUNZIONI PER QUOTE ASSI

In questo paragrafo è descritta una famiglia di funzioni utilizzabili per leggere le quote degli assi del CNC.

### 3.2.1 qvax: Legge la quota visualizzata di un asse

Chiamata: **esito = qvax (nasse, &p\_qv, flag)**

Parametri di ingresso:





**nasse** = (short) Numero d'ordine dell'asse per il quale s'intende avere la quota. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**p\_qv** = (double \*) Indirizzo dove scrivere il valore della quota visualizzata

**flag** = (short) Selezione tipo di quota; normalmente = 0.

= 0 Si vogliono quote "normali" o PUNTA (Centro Utensile) se Twist attivo

= 1 Si vogliono quote PIVOT con Twist attivo.

= 2 Si vogliono quote PUNTA (Centro Utensile) anche se Twist non attivo.

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

**-2** Valore della quota non aggiornato. La funzione è stata chiamata con **flag** = 2, ma non è mai stato programmato <ATW: . . . >.

**-1** Valore della quota non aggiornato (caso di asse mandrino attualmente agganciato in velocità ed in movimento).

**>=0** Valore della quota aggiornato; il valore di **esito** indica il numero di caratteri decimali visualizzati.

NOTA:

Il valore della quota è in mm. o pollici o gradi secondo il tipo di asse e se attiva o no la programmazione in pollici.

Esempio di utilizzo:

```
if( espressione )
{
  NASSE= 0;
  ESITO= qvax(NASSE, &QUOTA, 0);
  if(ESITO < 0)
  {
    ALR00= 1;
  }
}
/* leggo quota asse ordine 0 ed emetto diagnostica se lettura KO */
```

### 3.2.2 get\_qr\_pintl: Legge la quota qr ed il parametro pintl

Chiamata: **esito = get\_qr\_pintl (nasse, &qr, &pintl)**

Parametri di ingresso:

**nasse** = (unsigned short) Numero d'ordine dell'asse per il quale s'intende avere i dati. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**qr** = (double \*) Indirizzo dove scrivere il valore della quota qr (quota assoluta di riferimento dell'asse) calcolata da Babut.

**pintl** = (double \*) Indirizzo dove scrivere il valore del parametro d'interpolazione lineare (delta spostamento asse / traiettoria).

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

**0** Valore dei parametri aggiornato.

**1** Numero di asse errato.

**2** L'asse non è CNC.



- 3 Non ci sono buffers bi in carico al Babut.
- 4 L'asse non è da muovere.

NOTA: Con esito = 3 o 4, la quota qr è comunque valida.

### 3.3 RTCP (Rotational Tool Center Point) {F}

Le caratteristiche meccaniche della testa RTCP (TWIST) sono descritte nella monografia 720P385 "MANUALE TARATURE", file SPC.TAR. Nella libreria routine PLC è presente la funzione **tartwt()** per leggere/scrivere i parametri relativi.

#### 3.3.1 tartwt: Legge/scrive i parametri RTCP

Chiamata: **esito = tartwt (&param, fread)**

Parametri di ingresso:

**param** = (double \*) Indirizzo di un array di 7 double da dove leggere o dove scrivere i parametri della testa RTCP. I parametri sono:

- param[0] = valore di EC1 (vedi SPC.TAR)
- param[1] = valore di EC2 (vedi SPC.TAR)
- param[2] = valore di EC3 (vedi SPC.TAR)
- param[3] = valore di EAU (vedi SPC.TAR)
- param[4] = valore di DPV (vedi SPC.TAR)
- param[5] = valore dell'angolo tra utensili principale/attivo (V1024)
- param[6] = valore dell'angolo tra assi trascinati principale/attivo (V1028)

**fread** = (short) Flag di lettura/scrittura; può assumere i valori:

- 10 Scrittura nelle variabili di taratura. (param[0] . . [6] )
- 11 Lettura dalle variabili di taratura. (param[0] . . [6] )

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- 2 Valore di **fread** errato
- 1 Comando di scrittura non accettato; è attiva la modalità Twist.
- 1 Lettura/scrittura eseguita.

Esempio di utilizzo:

in PLC\_SYN.INC:

```
#define TWTEC1 F_A100
#define TWTEC2 F_A102
#define TWTEC3 F_A104
#define TWTEAU F_A106
#define TWTEPV F_A108
#define TWTANM F_A110
#define TWTANT F_A112
/* Sette double per i parametri testa TWIST */
```

in QAL.C:

```
if( . . . )
    ESITO= tartwt(&TWTEC1, 11);
    /* Legge parametri testa TWIST e scrive in TWTEC1 ... TWTANT */
if(ESITO != 1)
{
    ALR00= 1;
}
```



### 3.4 ASSE PENDOLANTE {F}

I CNC ECS 1801 2801 4801(Fresa), permettono di gestire il pendolamento di un asse tra due quote impostate (Q1, Q2) o intorno alla quota attuale definendo il delta desiderato. Questa funzionalità è controllata da apposite istruzioni LIP (vedi più avanti).

Il PLC può conoscere se e come è attiva questa funzionalità tramite alcuni segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati.

Dove compare il carattere "#", occorre sostituire ad esso il numero d'ordine dell'asse al quale il segnale è riferito. Tale numero, secondo il tipo di CNC, sarà compreso tra 0 e 2 (1801), tra 0 e 5 (2801) o tra 0 e 15 (4801).

Esempi:

**OPND0** l'asse di ordine 0 è attualmente pendolante.

**OPND4** l'asse di ordine 4 è attualmente pendolante.

<b>OPND#</b>	<b>CNC → PLC</b>	<b>OPND#=0</b> L'asse di ordine # non è attualmente pendolante. <b>OPND#=1</b> L'asse di ordine # è attualmente pendolante.
<b>OPQ12#</b>	<b>CNC → PLC</b>	<b>OPQ12#=1</b> L'asse di ordine # è pendolante tra due quote. Valido solo se il corrispondente <b>OPND#=1</b>
<b>OPVQ1#</b>	<b>CNC → PLC</b>	<b>OPVQ#=1</b> L'asse di ordine # deve raggiungere la quota Q1 <b>OPVQ#=0</b> L'asse di ordine # deve raggiungere la quota Q2 Valido solo se i corrispondenti <b>OPND#=1</b> e <b>OPQ12#=1</b>

#### 3.4.1 Programmazione di un asse pendolante

La movimentazione dell'asse pendolante viene comandata tramite le seguenti istruzioni LIP.

##### 3.4.1.1 Comando per il pendolamento tra due quote

**<PND: asse; tipoq; Q1; Q2; feed; ...>**

Parametri:

**asse** = Sigla dell'asse su cui attivare il pendolamento  
Può essere anche il numero d'ordine dell'asse oppure un'espressione LIP che ritorni un numero intero.

**tipoq** = Un carattere che indica il tipo di quote programmate; può essere:  
**A** : Quote Q1 e Q2 assolute  
**O** : Quote Q1 e Q2 relative all'origine attiva

**Q1** = Prima quota di pendolamento  
Può essere anche un'espressione LIP che ritorni un numero float.

**Q2** = Seconda quota di pendolamento  
Può essere anche un'espressione LIP che ritorni un numero float.

**feed** = Indica la feed di pendolamento.  
Può essere anche un'espressione LIP che ritorni un numero float.



NOTA:

E' permesso programmare più di un asse nella stessa istruzione, ripetendo più volte i cinque parametri necessari.

### 3.4.1.2 Comando per il pendolamento attorno alla quota attuale

**<AMP: asse; deltaq; feed ; ...>**

Parametri:

- asse** = Come per <PND: ...>
- deltaq** = Delta quota di pendolamento rispetto la posizione attuale  
Può essere anche un'espressione LIP che ritorni un numero float.  
L'oscillazione è quindi di ampiezza doppia di deltaq.
- feed** = Come per <PND: ...>

NOTA:

E' permesso programmare più di un asse nella stessa istruzione, ripetendo più volte i tre parametri necessari.

### 3.4.1.3 Comando per disattivare il pendolamento

**<DMP: asse; ...>**

Parametri:

- asse** = Come per <PND: ...>

NOTA:

E' permesso programmare più di un asse nella stessa istruzione, ripetendo più volte il parametro.

### 3.4.1.4 Note e precisazioni

In uno stesso blocco si possono programmare più comandi <PND:...>, <AMP:...>, <DMP:...> contemporaneamente; il CNC controlla che i comandi non siano riferiti agli stessi assi.

Il CNC controlla che gli assi programmati siano indipendenti (FREE) e non in Velocità; non si può rendere pendolante l'asse mandrino.

Attivando il pendolamento su di un asse con il comando <PND:...>, l'asse sarà dapprima posizionato sulla quota Q1 e poi inizierà a pendolare. Con il comando <AMP:...> la partenza è immediata, in quanto la quota attuale dell'asse viene considerata come quota centrale di pendolamento.

Quando é attivo un pendolamento, sull'asse interessato sono accettate solo istruzioni di fine pendolamento (<DMP:...>) oppure altre istruzioni <PND:...> o <AMP:...> per modificare uno dei parametri d'ingresso.

Non vengono accettati comandi per cambiare "al volo" il tipo di pendolamento attualmente attivo sull'asse; occorre passare per la disattivazione.

Con il comando di disattivazione <DMP:...>, il movimento dell'asse pendolante si arresta sulla quota Q1 in caso di comando di attivazione <PND:...> oppure sulla quota "centrale" in caso di comando <AMP:...>.

I comandi <PND:...> e <AMP:...>, in caso di prima attivazione, ed il comando <DMP:...>, sono comandi con WAIT, cioè attendono la fine del movimento di assi indipendenti.

- Il pendolamento rimane attivo anche durante i cambi modo.

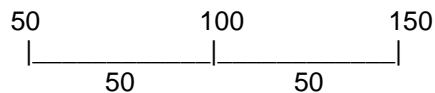


- Il JOG sull'asse pendolante non viene accettato.
- Il pendolamento viene disattivato tramite istruzione <DMP:...> oppure tramite RESET.
- Durante il pendolamento rimangono attivi gli OVERRIDE, l'HOLD, ed il BLOCCAGGIO per l'asse indipendente interessato.

Per comandare da PLC un asse pendolante occorre programmare i relativi comandi LIP di attivazione e disattivazione tramite KEY, su richiesta, ad esempio, di una funzione ausiliaria.

Vediamo un esempio per un asse "Z", dichiarato CNC in AXS.TAR

```
<FRE:Z> {L'asse Z viene reso indipendente }
Z100 {Quota centrale di pendolamento }
<AMP:Z;50;1000> {Attiva Movimento Pendolare a Feed= 1000 mm/min }
```



```
.....
.....
<DMP:Z> {Disattiva Movimento Pendolare }
```

Lo stesso effetto di pendolamento si avrebbe programmando:

```
<PND:Z;0;50;150;1000> {Pendolamento da 50 a 150; Feed= 1000 mm/min }.
```

### 3.5 COPIA {F}

Il CNC ECS 4801WIN può essere dotato dell'opzione **COPIA** (chiamata anche Digitalizzazione) che aggiunge prestazioni particolari per le Fresatrici, quali la copiatura mediante scansione di modelli, la digitalizzazione dei punti rilevati ed il loro salvataggio su HARD DISK.

Il Part Program ottenuto può essere riprodotto dalla sezione CNC per la lavorazione dello stampo (rilevato dal modello). Per i particolari di funzionamento e le tarature necessarie si rimanda alla specifica monografia 720P364 (OPZIONE DIGITALIZZAZIONE - NORME DI PROGRAMMAZIONE ED USO).

La routine di libreria PLC **copia()**, in abbinamento alla routine **tasfuecs()**, che gestisce i tasti funzione del PLC nella configurazione standard ECS definita in BASMENU.MEN, risolve in maniera facile e sicura la relativa funzionalità. L'uso della routine suddetta è spiegato più avanti in questo capitolo.

#### 3.5.1 Sinonimi utilizzati per la funzione COPIA

Per la gestione del funzionamento del CNC in modo COPIA tramite il PLC, sono presenti vari segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati.

<b>ISSTCPY</b> <b>ISST</b> <b>RISST</b>	<b>PLC → CNC</b>	Ingresso Sospensione Sicurezze Tasto. Posto a 1sospende le sicurezze che normalmente intervengono in caso di deflessione del tasto con il CNC non in modo Copia. <i>Se il PLC usa la routine <b>copia()</b>, deve essere usato <b>RISST</b>.</i>
---	------------------	--



		<i>La routine <b>tasfuecs()</b> gestisce <b>RISST</b> tramite tasto dedicato.</i>
<b>ISTEPCPY</b> <b>ISTEP</b> <b>RSTEP</b>	<b>PLC → CNC</b>	Richiesta STEP ASSI BLOCCATI in Copiatura manuale. Posto impulsivamente a 1, comanda un movimento incrementale degli assi di Copiatura attualmente bloccati. Il segno e l'entità dello step sono impostati dall'operatore tramite apposita finestra. <i>Se il PLC usa la routine <b>copia()</b>, deve essere usato <b>RSTEP</b>.</i>
<b>MODE_CPY</b> <b>OCOP</b>	<b>CNC → PLC</b>	<b>MODE_CPY=1</b> CNC in modo COPIA Automatico o Manuale
<b>OMDGPCY</b> <b>OMDG</b>	<b>CNC → PLC</b>	<b>OMDGPCY=0</b> CNC non in modo COPIA Manuale <b>OMDGPCY=1</b> CNC in modo COPIA Manuale
<b>OSTOCPY</b> <b>OSTO</b>	<b>CNC → PLC</b>	<b>OSTOCPY=1</b> Il CNC è in modo COPIA e sta registrando le quote di scansione modello (STORE attivo).
<b>ITF02</b>	<b>PLC → CNC</b>	Input TRU/FALSE numero 2. Usato per segnalare al CNC che il tastatore ha toccato il riscontro durante il ciclo di taratura delle deflessioni. <i>Se il PLC non usa la routine <b>copia()</b>, deve avere una istruzione: <b>ITF02= OLIP202</b>;</i>

NOTA:

**ISSTCPY** (o **RISST**) posto a 1 sospende le sicurezze che normalmente intervengono in caso di superamento della deflessione massima del tasto ammessa con il CNC **non in modo Copia**.

Il valore massimo ammesso, prima che intervenga l'allarme di supero deflessioni, è definito in COST con, ad es., <%V4216=200>, con 200 espresso in "conti" del trasduttore-tasto.

In caso di arresto della macchina con il tasto deflesso, ad esempio dopo un'emergenza in fase di copiatura, è attivo l'allarme CNC di "supero deflessioni" che impedisce i movimenti degli assi; l'allarme non è resettabile finché permane la situazione di tasto deflesso. Se non è possibile smontare il tasto stesso, ponendo l'ingresso **ISSTCPY=1** è allora possibile resettare l'allarme ed allontanarsi dal modello per liberare il tastatore.

Si raccomanda di rimettere, appena possibile, **ISSTCPY=0** per riattivare la sicurezza anticollisione.

### 3.5.2 Tarature sul file COST

Nel file di taratura COST (vedi monografia 720P385) sono presenti alcune variabili da inizializzare ad ogni accensione del CNC. Le relative istruzioni sono simili alle seguenti e possono o debbono essere modificate per inserire i valori più adatti alla specifica Macchina:

```
N1600 {----- Variabili per Copia -----}
N1610
N1620 <DBL=50>
N1630 {Dimensione blocchetto per qualifica tasto copia, in mm.}
N1640
N1650 <%V4216=200>
N1660 {Max deflessione tasto non in modo COPIA (conti)}
N1670
N1680 <DEM=0.150>
```



```
N1690 {Deflessione minima per contatto pezzo in copia, in mm.}
N1700
N1710 <FER=100>
N1720 {Feed ridotta per autoapprendimento profilo, in mm./min.}
N1730
N1740 <ANL=10>
N1750 {Angolo limite sul profilo, in gradi}
N1760
N1770 <SFR=5>
N1780 {Spazio da percorrere a feed ridotta FER, in mm.}
N1790
N1800 <DPM=2>
N1810 {Distanza punti per la misura degli angoli sul profilo, in mm.}
```

**N.B.:** si consiglia di non modificare direttamente i records del file COST ma di copiarli e modificarli nel file COSTUSER esplicitamente previsto per le personalizzazioni legate ad una macchina specifica. In questo modo si facilita il "trasporto" delle personalizzazioni da una release di software ad una successiva

### 3.5.3 Routines a disposizione del PLC

Come accennato più sopra, il PLC dispone della routine di libreria **copia()**. Questa, in abbinamento alla routine **tasfuecs()**, che gestisce i tasti funzione del PLC nella configurazione standard ECS definita in BASMENU.MEN, risolve in maniera facile e sicura la gestione del modo COPIA. Per completare la gestione, occorre usare anche la routine **hldfre()** in quanto gli assi di copiatura sono automaticamente messi FRE in Velocità; le suddette routines vanno richiamate come nell'esempio qui sotto indicato:

```
/*-----
  Gestione tastiera
-----*/
tastcnc();
/* Chiamata Routine Gestione Tastiera */
tasfuecs();
/* Routine gestione tasti P1-P8 come definiti in BASMENU.MEN */

/* ----- PER COPIA ----- */

RISST= PSDPT && !RISST || RISST && !PSDPT;
/* DISABILITA PROTEZIONE TASTO */
copia();
/* Chiamata Routine Gestione Comandi Copia */
hldfre();
/* Chiamata routine gestione Hold assi Indipendenti */
ATT06= ISSTCPY;
/* Sicurezza Tasto Sospesa */
/* ----- FINE PER COPIA ----- */
```

#### 3.5.3.1 Routine "COPIA"

Chiamata: **copia()**;

Scopo: gestisce i comandi riguardanti la copiatura (Digitalizzazione) e pertanto va' usata se la relativa opzione è presente sul CNC.

Parametri in Ingresso:



**RSTEP = 1** richiesta STEP asse bloccato in MaDig (copiatura manuale)

**RISST = 1** richiesta sospensione sicurezza tasto

La routine **copia()** provvede a richiudere su ITF02 (True/False bit 02) l'uscita OLIP202, gestita dalla procedura di taratura deflessioni, ad inoltrare le richieste di STEP ASSI BLOCCATI e quella di SOSPENSIONE SICUREZZA TASTO.

Provvede inoltre ad attivare il KEY98 per la copiatura manuale ( L<MADIG> ); il KEY suddetto è del tipo sincrono ed è gestito dalla routine di sistema varie(), richiamata automaticamente ad ogni loop del PLC.

- Lo STEP ASSE BLOCCATI avviene sul fronte in salita di RSTEP
- Con RISST attivo, non interviene più l'allarme SUPERO DEFLESSIONI TASTO
- E' d'obbligo inserire anche la chiamata a **hldfre()**.

### 3.6 ASSE STEP (Opzionale)

Attraverso un'istruzione di tipo **<SAX: .... >** è possibile configurare un'asse del CNC (definito come caratteristiche in AXS.TAR) come ASSE STEP, avente quindi un riferimento analogico adatto al pilotaggio di un asse con motore passo-passo. Questa istruzione deve essere programmata una volta prima di inviare una richiesta di movimento all'asse STEP. L'istruzione da usare è la seguente:

```
<SAX:ASSE;1;'89';0;1;0>
```

dove:

ASSE = sigla dell'asse o espressione LIP con il numero d'ordine dell'asse.

Si consiglia di scrivere questa istruzione nel sottoprogramma di presa di zero (SETnn) dell'asse relativo, in modo che sia eseguita automaticamente all'accensione del CNC e in caso di ripresa di zero a seguito di un allarme sull'asse.

L'asse STEP deve essere definito INDIPENDENTE in AXS.TAR oppure reso tale attraverso l'istruzione <FRE:...>. Un asse STEP viene pilotato con un riferimento "ad onda quadra" alla frequenza opportuna; il segno del movimento è dato dall'uscita **OBSF#** ( 1 = negativo).

#### ATTENZIONE!

La gestione di "Assi Step" è opzionale e non può essere attivata su un CNC con la normale procedura di abilitazione opzioni ma richiede l'installazione di un Software speciale da parte della ECS. Si prega di contattare gli uffici commerciali ECS qualora ne sia prevista la necessità.

### 3.7 ASSE DUMMY {F}

L'asse **Dummy** è un asse utilizzato per **generare punti** (calcolati dall'interpolatore del CNC) leggibili dal PLC. In base al percorso compiuto ed ad altre variabili (calcolate da algoritmi implementati nel PLC), si potrà, ad esempio, compensare in tempo reale l'usura di un utensile, modificando la posizione degli assi nel piano o nello spazio, attraverso gli input della compensazione da PLC (ICORAX#).

#### 3.7.1 Caratteristiche dell'asse definito Dummy

- I fincorsa SW non sono definibili.
- Non può appartenere al piano di contornatura definito con G16...





- Non può essere compensato in lunghezza
- Sono ammesse solo le istruzioni LIP logiche:  
<CNC:....> <FRE:....> <FED:....> <WAT:FRE;....>

### 3.7.2 Funzionamento asse Dummy

Il funzionamento dell'asse Dummy varia secondo il modo in cui opera il CNC. Il funzionamento "DUMMY" è però possibile soltanto nel modo AUTOMATICO o MANUALE.

#### Asse Dummy attualmente Free (Indipendente)

Quando è programmato come asse indipendente, l'asse Dummy segue le regole di tutti gli altri assi indipendenti.

#### Modo SET - Modo JOG

In questi due modi l'asse Dummy funziona come un qualsiasi altro asse del CNC

#### Modo AUTOMATICO - Modo MANUALE

Quando l'asse Dummy (attualmente CNC) è programmato in un blocco dove non sia stato programmato nessun altro asse CNC, viene effettuato un posizionamento dell'asse nello stesso modo in cui avviene per gli altri assi CNC. Viene tenuto conto cioè delle tarature di accelerazione e velocità dell'asse, viene quindi effettuata la rampa di accelerazione e decelerazione propria dell'asse; la velocità dell'asse non potrà superare quella rapida.

Se invece l'asse Dummy è programmato insieme ad altri assi CNC, abbiamo un tipo di interpolazione (interpolazione lineare asse Dummy) che non segue le regole degli altri assi agganciati in contouring.

L'asse dichiarato Dummy di fatto non partecipa alla contornatura, vuole solamente produrre quote incrementali (lineari) proporzionali al movimento degli altri assi che stanno interpolando, ma senza interferire con questi. In altre parole la velocità di contornatura e lo spazio da percorrere degli assi effettivamente legati alla lavorazione (nel piano o nello spazio), non sono in nessun modo legati allo spazio da percorrere (alla quota da raggiungere) dall'asse Dummy per quel blocco. È però implicito che l'asse Dummy deve raggiungere la quota programmata insieme agli altri assi di contornatura.

Da tutto questo si deduce che quando si muove l'asse Dummy insieme ad altri assi CNC non vengono tenute conto le tarature di accelerazione e di velocità di questo asse; vengono perciò esclusi i controlli sui limiti di accelerazione dell'asse.

Per quanto riguarda l'allarme fuga, questo non viene escluso; è necessario quindi definire KV=0 in AXS.TAR.

Le quote programmate per l'asse Dummy sono assolute oppure incrementali a seconda del modo di programmazione attiva (G90 / G91).

Solo quando è attiva l'interpolazione Dummy, è possibile rendere incrementali le quote per l'asse Dummy a prescindere se è attiva G90 o G91, abilitando il seguente parametro nel file COST:

```
<%V4194=%V4194|'8000'>
```

La visualizzazione dell'asse Dummy è abilitata dal seguente parametro, sempre in COST:

```
<%V4194=%V4194|'4000'>
```

**N.B.:** si consiglia di non modificare direttamente i records del file COST ma di copiarli e modificarli nel file COSTUSER esplicitamente previsto per le personalizzazioni legate ad una macchina specifica. In



questo modo si facilita il “trasporto” delle personalizzazioni da una release di software ad una successiva.

### 3.7.3 Parametri dell'asse Dummy

Le variabili descritte di seguito sono aggiornate dal CNC solo in modo AUTO ESECUZIONE ed interpolazione Dummy in corso. Queste variabili vengono resettate con START Programma in AUTOMATICO (start dopo selezione, dopo fine programma, dopo reset).

Si ricorda che le variabili sono accessibili tramite le funzioni di libreria PLC **read\_var(...)** e **write\_var(...)** ; per una descrizione si rimanda al relativo capitolo.

<b>AXDUMMY</b>	<p>Parametro <b>%V1003</b>          Numero d'ordine asse Dummy oppure -1 (asse Dummy non definito).          Esprime il valore del campo AXDM del record "VARI:" di GEN.TAR</p>
<b>IPOCOR</b>	<p>Parametro <b>V1004</b>          Delta spostamento attuale rispetto ad inizio blocco:          esprime il valore dell'interpolazione lineare dei punti dell'asse Dummy per il blocco attuale elaborato. Tale valore è resettato ad inizio esecuzione blocco di programma ed è calcolato poi dall'interpolatore secondo la relazione:  <math display="block">(1 - \text{Traiett.residua} / \text{Traiett.totale}) * \text{DELTA\_AXDUMMY}</math>         dove per traiettoria residua s'intende lo spostamento che gli assi CNC devono ancora percorrere in questo blocco di Programma e, per traiettoria totale, lo spostamento totale degli assi CNC relativo a questo blocco di Programma.</p>
<b>T_IPOCOR</b>	<p>Parametro <b>V1005</b>          Somma degli spostamenti programmati precedentemente per asse Dummy (accumulo dei valori di IPOCOR finali). Il valore è calcolato secondo la relazione:  <math display="block">T\_IPOCOR = T\_IPOCOR + \text{DELTA\_AXDUMMY}</math>         La variabile viene aggiornata ad inizio blocco programma.</p>
<b>ASS_IPOCOR</b>	<p>Parametro <b>V1006</b>          Spostamento assoluto:          esprime il delta spostamento attuale rispetto ad inizio programma. Il valore è calcolato dall'interpolatore appena dopo il calcolo di IPOCOR secondo la relazione:  <math display="block">\text{ASS\_IPOCOR} = T\_IPOCOR + \text{IPOCOR}</math>         ed è anche aggiornato ad inizio blocco programma ricopiandovi il valore dell'accumulo T_IPOCOR appena aggiornato (rifasamento valori).</p>
<b>NBLK_AXDUMMY</b>	<p>Parametro <b>%%V1007</b>          Numero totale di blocchi contati da inizio programma in cui è stato programmato in asse Dummy.          La variabile viene incrementata ad inizio blocco appena dopo aver resettato IPOCOR, aver calcolato T_IPOCOR ed ASS_IPOCOR, aver inizializzato DELTA_AXDUMMY.</p>



<b>DELTA_AXDUMMY</b>	<b>Parametro V1008</b> Delta spostamento programmato per asse Dummy per questo blocco (quota IPOCOR finale per questo blocco). La variabile viene inizializzata ad inizio blocco programma, subito dopo il calcolo dell'accumulo T_IPOCOR, ricopiando il valore ricavato dalla differenza tra quota fine blocco e quota inizio blocco (QCF0 - QC0) dell'asse Dummy. Serve per il calcolo degli accumuli T-IPOCOR e per il calcolo di IPOCOR stesso.
<b>FL_IPOCOR</b>	<b>Parametro %V1009</b> Variabile relativa all'interpolazione asse Dummy. Attualmente non utilizzata ed inizializzata a zero. Disponibile per eventuali usi futuri.
<b>QR_IPOCOR</b>	<b>Parametro V1010</b> Quota di riferimento (QR) dell'asse Dummy calcolata dall'interpolatore. Tale variabile NON VIENE RESETTATA AD INIZIO PROGRAMMA. Corrisponde alla quota assoluta dell'asse, relativa allo zero macchina. Se ad inizio P.P. l'asse è stato azzerato e non sono attive origini per l'asse, il valore di QR_IPOCOR coincide con quello di ASS_IPOCOR.

### 3.7.4 Esempio di configurazione asse Dummy

#### File di configurazione GEN.TAR

Nel record VARI: al campo AXDM viene assegnato il valore 4. Questo significa che il quinto asse dichiarato in AXS.TAR (di ordine ORD=4) sarà quindi un asse Dummy.

#### File di configurazione AXS.TAR

- Definisce 5 assi di nome X, Y, Z, C, P
- I primi tre assi X, Y, Z, sono quelli relativi alla contornatura (CNC)
- Il quarto asse C, è l'asse mandrino
- Il quinto asse P, di ordine ORD=4 è l'asse dichiarato CNC (IND=N) e configurato come asse Dummy in GEN.TAR

#### PROGRAMMA

```
%  
N0 G16XYZ+  
N10 P0  
N20 X0 Y0  
N25  
N30 X10 Y20 P50  
N40 X50 Y40 P175  
N50 X100 Y100 P230  
N60 X10 Y30  
N70 P100
```

Se attiva programmazione assoluta G90 e programmazione asse Dummy incrementale oppure

Se attiva programmazione incrementale G91 (ad esempio programmando N25 G91) abbiamo:



	IPOCOR	T_IPOCOR	ASS_IPOCOR	NBLK_AXDUMMY	DELTA_AXDUMMY
INIZIO P.P.	0	0	0	0	0
N30 Inizio blocco	0....	0	0....	1	50
N30 Fine blocco	....50	0	....50	1	50
N40 Inizio blocco	0....	50	50....	2	175
N40 Fine blocco	....175	50	....225	2	175
N50 Inizio blocco	0....	225	225....	3	230
N50 Fine blocco	....230	225	....455	3	230
N60 Inizio blocco**	230	225	455	3	230
N60 Fine blocco**	230	225	455	3	230
N70 Inizio blocco**	230	225	455	3	230
N70 Fine blocco**	230	225	455	3	230

\*\*Le variabili non vengono aggiornate perché non c'è interpolazione asse Dummy.

Se attiva programmazione assoluta G90 e programmazione asse Dummy assoluta  
abbiamo:

	IPOCOR	T_IPOCOR	ASS_IPOCOR	NBLK_AXDUMMY	DELTA_AXDUMMY
INIZIO P.P.	0	0	0	0	0
N30 Inizio blocco	0....	0	0....	1	50 (P50-P0)
N30 Fine blocco	....50	0	....50	1	50
N40 Inizio blocco	0....	50	50....	2	25 (P175-P50)
N40 Fine blocco	....125	50	....175	2	125
50 Inizio blocco	0....	175	175....	3	55 (P230-P175)
50 Fine blocco	....55	175	....230	3	55
N60 Inizio blocco**	55	175	230	3	55
N60 Fine blocco**	55	175	230	3	55
N70 Inizio blocco**	55	175	230	3	55
N70 Fine blocco**	55	175	230	3	55

\*\* Le variabili non vengono aggiornate perché non c'è interpolazione asse Dummy

#### NOTA

I puntini .... stanno a significare una variazione lineare, calcolata dinamicamente dall'interpolatore. QR\_IPOCOR, che non è stata inserita nella tabella, NON VIENE RESETTATA AD INIZIO PROGRAMMA. Corrisponde alla quota assoluta dell'asse, relativa allo zero macchina. Se ad inizio programma l'asse è stato azzerato (programmato ad una quota zero) e non sono attive origini per l'asse, il valore di QR\_IPOCOR coincide con quello di ASS\_IPOCOR.



## 3.8 ASSI GANTRY

### 3.8.1 Funzione degli assi Gantry

Due assi (denominati Master e Slave) si dicono "**agganciati in Gantry**" (o più semplicemente "**in Gantry**") se la posizione dell'uno "segue" quella dell'altro in base ad una funzione stabilita. La più semplice (e più comune) di queste funzioni è quella lineare ossia:

$$QR_m = K_{OM} * QR_s$$

essendo:

QR<sub>m</sub>: quota riferimento Master

QR<sub>s</sub>: quota riferimento Slave

K<sub>OM</sub>: costante di proporzionalità

Tipicamente si ha  $K_{OM} = 1$  per cui i due assi in Gantry hanno la stessa quota di riferimento. Ciò significa, in altri termini, che essi si muovono in modo sincrono, realizzando in pratica un cosiddetto "albero elettrico", cioè un "cinematismo" virtuale tra i due assi analogo ad un vero e proprio albero meccanico che colleghi i due assi. Per questo si può affermare che, quando due assi sono in Gantry, diventano in pratica un solo asse sia dal punto di vista meccanico che dal punto di vista "controllo" (CNC). Si conviene perciò, dal momento in cui i due assi sono agganciati in Gantry, di riferirsi al solo asse Master per i comandi asse.

Si distingue talvolta un "Gantry Vero" ed un "Gantry Falso" intendendo per il primo una situazione di assi che devono essere SEMPRE solidali (esempio tipico un Portale) e per il secondo una situazione in cui gli assi possono lavorare sia in Gantry che no (esempio due tavole che si muovono in modo sincrono quando deve essere lavorato un pezzo unico che le occupa entrambe).

L'asse Master può essere definito in AXS.TAR sia CNC che indipendente. L'asse Slave di un "Gantry Vero" è normalmente dichiarato indipendente, ma può essere anche CNC.

Dalla release di Software V4.00, gli assi Slave di uno stesso Master possono essere più di uno (Gantry Multislave), naturalmente con lo stesso K<sub>OM</sub>. Questa possibilità è valida, di norma, per un "Gantry Vero" con assi assoluti, ma può essere attivata anche per assi incrementali e/o per un "Gantry Falso".

### 3.8.2 Realizzazione di assi Gantry sui CNC ECS serie WIN

La gestione degli assi Gantry viene controllata tramite apposite istruzioni di Part-Program scritte in linguaggio LIP.

#### 3.8.2.1 Aggancio assi in GANTRY

```
<GAN:ON; Master; Slave; %KV; SFG; Scale>
```

oppure, dalla V4.00:

```
<GAN:ON; Master; N=nAxS; Slave; Slave2;... ;%KV; SFG; Scale>
```

**ON** Il primo parametro deve essere "ON" per l'istruzione di "Aggancio in GANTRY"



---

---

<b>Master</b>	Nome o numero d'ordine dell'asse MASTER
<b>N=nAxS</b>	nAxS: numero di assi Slave; da 2 a numero totale assi -1; opzionale
<b>Slave</b>	Nome o numero d'ordine dell'asse SLAVE (unico o primo)
<b>Slave2; Slave3;...</b>	Nome o numero d'ordine degli assi SLAVE successivi; opzionali. I campi SlaveN debbono essere tanti quanto definito con nAxS.
<b>%KV</b>	Guadagno del ciclo GANTRY espresso come percentuale del KV del singolo asse. Deve essere un numero intero compreso tra 0 e 100 (consigliato da 50 a 100)
<b>SFG</b>	Soglia di Fuga GANTRY espressa in [mm] o [gradi]. Se l'errore Gantry (differenza tra le due quote) supera questa soglia, interviene un apposito Allarme di Fuga Gantry.
<b>Scale</b>	Rapporto di scala GANTRY [V2.00]; è il rapporto meccanico o di velocità tra asse(i) SLAVE ed asse MASTER. Deve essere un numero reale (double) diverso da zero. Valori positivi (>0) indicano verso concorde di movimento relativo tra l'asse Master e l'asse Slave. Valori negativi (<0) indicano invece verso discorde di movimento relativo tra l'asse Master e l'asse Slave. Con un valore =-1 si ottiene un movimento dei due assi speculari. Per rel. SW precedenti, erano accettati solo i valori 1 e -1. Il rapporto Scale deve essere necessariamente uguale a 1 per un "Gantry Vero" (con aggancio comandato direttamente in COSTUSER). Parimenti, il rapporto deve essere =1 (o -1) in caso di assi definiti "Tondi" in AXS.TAR .

### 3.8.2.2 Sgancio assi in GANTRY

<GAN:OFF; Master>

<b>OFF</b>	Il primo parametro deve essere "OFF" per l'istruzione di "Sgancio da GANTRY"
<b>Master</b>	Nome o numero d'ordine dell'asse MASTER

### 3.8.2.3 Compensa assi in GANTRY

<CGA:Axis; DeltaQ[;'Modo']>

<b>Axis</b>	Nome o numero d'ordine dell'asse MASTER o SLAVE
<b>DeltaQ</b>	Indica il Delta Quota di compensazione del disallineamento meccanico tra i due assi (MASTER e SLAVE) agganciati in GANTRY. Deve essere un numero reale (double) espresso in [mm] o [gradi].
<b>Modo</b>	Modo di applicazione del disallineamento. <b>A</b> (Assoluto): il valore di <b>DeltaQ</b> è considerato <u>assoluto</u> ; programmando più volte l'istruzione CGA, il nuovo valore di compensazione <u>sostituisce</u> quello precedente. <b>I</b> (Incrementale): il valore di <b>DeltaQ</b> è considerato <u>incrementale</u> ; programmando più volte l'istruzione CGA, il nuovo valore di compensazione <u>si somma</u> (con il segno relativo) a quello precedente. Parametro opzionale; se omesso, il funzionamento corrisponde ad <b>A</b> .

---

---



Con tale istruzione si ottiene uno spostamento dell'asse SLAVE dello spazio DeltaQ (se Modo=I) o della differenza tra DeltaQ attuale e DeltaQ precedente (se Modo=A), mentre l'asse MASTER rimane fermo.

La compensazione viene applicata in maniera graduale per evitare "colpi" sull'asse SLAVE.

L'istruzione deve essere rivolta all'asse MASTER, che deve essere, prima, reso indipendente (se non lo fosse già); le quote dei due assi "visivamente" rimangono comunque uguali.

Per un Gantry MULTISLAVE occorre specificare obbligatoriamente uno degli assi Slave.

#### 3.8.2.4 Compensazione automatica assi in GANTRY

<AGC:Asse[ ;DP=DeltaQ]>

**Asse** Nome o numero d'ordine dell'asse MASTER o SLAVE.

**DeltaQ** Master Slave Delta Position. Indica la differenza programmata e voluta tra quote "effettive" del Master e quote "effettive" dello Slave. Deve essere un numero reale (double) espresso in [mm] o [gradi].  
Il parametro è opzionale; se omissso viene considerato il valore di default = 0.

Tramite la funzione <AGC:...> il CNC provvede automaticamente ad effettuare una corretta compensazione per posizionare gli assi Master/Slave ad una distanza fisica "programmata"; entrambi gli assi Master e Slave devono avere il "SET fatto", pena l'allarme 2319.

L'istruzione <AGC:...>, se programmata sull'asse Slave, viene comunque "riportata" sull'asse Master; il comando di compensazione produce sempre uno spostamento sull'asse Slave, mentre l'asse Master rimane FERMO. Il valore di Default = 0 indica che l'asse Slave deve essere posizionato sulla stessa quota effettiva del Master.

Note:

Per un Gantry MULTISLAVE occorre specificare obbligatoriamente uno degli assi Slave.

A differenza dell'istruzione <CGA:...> (Compensa assi in Gantry), dove viene programmato il valore assoluto od incrementale di compensazione, nell'istruzione <AGC:...> (Compensazione Automatica Gantry) tale valore viene automaticamente calcolato dal CNC in base alle posizioni "reali" degli assi ed alla differenza programmata di tali valori (Master-Slave).

Mentre l'istruzione <CGA:...> viene normalmente aggiunta nella procedura di ZERO (file di SET per assi incrementali) dell'asse Master di un Gantry "vero", a seguito della ricerca micro-marca sia dell'asse Master, sia dell'asse Slave, l'istruzione <AGC:...> deve necessariamente essere aggiunta nel file di "SET" di un'asse ASSOLUTO, agganciato come Master di un GANTRY "Vero".

Le due funzioni <CGA:...> e <AGC:...> sono interpretate dal Controllo Numerico con una WAIT (interrompono la lettura "in avanti" del Part-Program), non sono lecite nei programmi COST, COSTUSER e RESET, possono essere programmate solamente per assi già agganciati in GANTRY ed hanno effetto solo se ambedue gli assi sono sbloccati.

L'istruzione "Disallinea assi in GANTRY"

<DGA:Asse; DeltaQ>

esistente sulle release di software precedenti non è più presente.

La Compensazione è azzerata con lo "SGANCIO" dal modo GANTRY.

#### 3.8.3 Sinonimi utilizzati per gli Assi Gantry

Per la gestione degli assi Gantry tramite il PLC, sono presenti alcuni output che sono stati associati ai sinonimi qui di seguito elencati.

Dove compare il carattere "#", occorre sostituire ad esso il numero d'ordine dell'asse al quale il segnale è riferito. Tale numero, secondo il tipo di CNC, sarà compreso tra 0 e 2 (1801, 1802), tra 0 e 5 (2801) o tra 0 e 15 (4801, 4802).

Esempi:

**OBGG0** l'asse di ordine 0 è agganciato in Gantry.

**OBGM4** l'asse di ordine 4, agganciato in Gantry, è il Master



<b>OBGG #</b>	<b>CNC → PLC</b>	<b>OBGG#= 1</b> L'asse # è agganciato in Gantry. Questo segnale deve essere usato nel PLC per avere la certezza che gli assi desiderati siano effettivamente agganciati in Gantry.
<b>OBGM #</b>	<b>CNC → PLC</b>	<b>OBGM#= 0</b> L'asse #, agganciato in Gantry, è lo Slave. <b>OBGM#= 1</b> L'asse #, agganciato in Gantry, è il Master. Il segnale è valido se <b>OBGG# = 1</b> .
<b>OBGSET #</b>	<b>CNC → PLC</b>	<b>OBGSET#= 1</b> L'asse <u>Assoluto</u> #, agganciato in Gantry, è da <u>riallineare</u> . Occorre attivare la compensazione automatica Gantry (istruzione <AGC:...>) prima di poter muovere gli assi stessi.

### 3.8.4 Funzionamento degli assi Gantry

Le istruzioni <GAN:ON;...> e <GAN:OFF;...> sono istruzioni supermodali, agiscono cioè permanentemente dal momento della programmazione fino alla loro riprogrammazione (ON/OFF) oppure fino allo spegnimento del controllo.

Qualora l'istruzione <GAN:ON;...> fosse programmata in COST o COSTUSER, gli assi configurati come Master e Slave risulteranno agganciati in GANTRY fin dall'accensione e dalla fase di SET del CNC (si tratta quindi di un **Gantry Vero**). In questo caso non è ammesso un valore di Rapporto Gantry (Scale di <GAN:ON;...>) diverso da 1.

L'istruzione <GAN:OFF;...> non è invece lecita nel programma COST o COSTUSER.

L'aggancio in Gantry tra due o più assi è possibile solo se tutti sono dello stesso "tipo" (tutti TONDI oppure NON TONDI); l'aggancio è possibile anche tra assi con caratteristiche "meccaniche" diverse (campo RSM= di AXS.TAR con valori differenti), purché dello stesso "tipo".

Per assi Master e Slave definiti TONDI, non è ammesso un valore di Rapporto di scala GANTRY (Scale) diverso da 1 oppure -1.

L'istruzione di aggancio in Gantry, nel caso di "**Gantry Falso**" (programmata cioè fuori da COST) è lecita anche se uno od entrambi gli assi, Master e Slave, non hanno il SET fatto (vedi più avanti al paragrafo 3.8.10).

Quando l'istruzione <GAN:ON;...> ha successo, i segnali di aggancio Gantry (OBGG#) vengono settati su tutti gli assi interessati.

Le istruzioni di aggancio/sgancio non avranno successo nei seguenti casi:

- Sintassi errata
- Asse non disponibile.
- Asse di proprietà del PLC ( OLS# = 1).
- Asse gestito dal PLC ( OAXPLC# = 1)
- Asse già agganciato in GANTRY (solo per aggancio)
- Asse sganciato (OBAG# = 0)

Quando gli assi sono in GANTRY, lo Slave si "aggancia" sulla quota del Master. L'asse Master può essere programmato sia come asse CNC (in contouring), sia come asse FRE (in punto-a-punto). La programmazione sull'asse Slave potrà avvenire solamente con assi sia Master sia Slave agganciati FRE (in punto-a-punto); in questo caso lo spostamento dello





Slave si "rifletterà" sul Master alla velocità programmata per lo Slave. Questo permette di comandare sull'asse Slave gli spostamenti di ricerca micro/marca (G8nn).

Programmare un asse Slave CNC non provocherà alcun effetto; programmare un asse Slave FRE quando il Master risulti essere CNC provocherà un allarme.

Non è ammesso eseguire la PROVA STEP o la PROVA BANDA per assi in GANTRY: una tale richiesta provoca un allarme specifico. Questo viene fatto per evitare che in caso di GANTRY vero, l'eventuale risposta differente dei due assi provochi danni alla struttura della macchina.

Con l'emissione della funzione <GAN:OFF;Master>, ossia con lo sgancio dalla modalità GANTRY, gli assi Master e Slave vengono agganciati come definiti in taratura AXS.TAR (CNC o FRE) in funzione però del modo attuale del CNC; la quota dell'asse Slave viene ripristinata sulla posizione attuale effettiva, "sganciandosi" quindi dalla quota del Master. La Compensazione ed il Disallineamento sono azzerati con lo "SGANCIO" dal modo GANTRY.

Non è ammesso programmare la funzione di sgancio <GAN:OFF;Master> in caso di Gantry vero, ossia attivato in COST.

#### 3.8.4.1 Allarmi

La gestione degli allarmi su assi in Gantry è tale per cui un allarme su uno dei due assi si "riflette" anche sull'altro.

#### 3.8.4.2 Fine Corsa Software

In caso di "Gantry Vero", ossia attivato in COST, non ci sono normalmente problemi, poiché gli assi configurati come Master e Slave risulteranno agganciati in GANTRY fin dall'accensione e dalla fase di SET del CNC; la posizione reciproca dei due assi sarà sempre la stessa ed i Fine Corsa Software saranno coincidenti (stessi valori in AXS.TAR).

In caso di "Gantry Falso", ossia attivato NON in COST, potrebbero sorgere dei problemi, perché l'aggancio può avvenire con i due assi in posizioni qualsiasi, anche molto diverse tra di loro. Il movimento dell'asse Master, pur rispettando i limiti del suo campo di lavoro, potrebbe portare l'asse Slave oltre i suoi fine corsa.

Per ovviare a quest'inconveniente, al momento dell'aggancio in Gantry il CNC "ricalcola" i Fine Corsa Software del Master tenendo conto della corsa "residua" dello Slave nelle due direzioni. Ovviamente il CNC tiene conto del "Rapporto di scala GANTRY" programmato, compreso il segno.

Analoghe verifiche sono effettuate nel caso siano usate le istruzioni G25 e G26 che riducono il campo di lavoro.

All'atto dello sgancio Gantry, i Fine Corsa Software eventualmente modificati sono ripristinati ai valori di taratura definiti in AXS.TAR .

Nelle release SW precedenti alla V2.00, i finecorsa software sull'asse SLAVE NON erano controllati.

#### 3.8.4.3 Rapidi ed Accelerazioni

Al momento dell'aggancio in Gantry, le velocità rapide e le accelerazioni degli assi Master e Slave vengono limitate in funzione dei valori "peggiori" tra i due assi, considerando anche il rapporto Gantry (Scale di <GAN:ON;...>); in questo modo non saranno superati i limiti dell'asse Slave per spostamenti del Master e viceversa.

---



---

Per essere precisi, detti "ValMaster" i valori di rapido ed accelerazione del Master e "ValSlave" i corrispondenti dello Slave:

Se  $(\text{ValMaster} * \text{Scale}) > \text{ValSlave}$  Allora  $\text{NewValMaster} = \text{ValSlave} / \text{Scale}$   
Vengono cioè limitati i valori dell'asse Master

Se  $(\text{ValSlave} / \text{Scale}) > \text{ValMaster}$  Allora  $\text{NewValSlave} = \text{ValMaster} * \text{Scale}$   
Vengono cioè limitati i valori dell'asse Slave

In fase di sgancio da Gantry, vengono ripristinati per ambedue gli assi i relativi valori di Rapido ed Accelerazione definiti in AXS.TAR .

#### 3.8.4.4 Bloccaggi

Per gli assi agganciati in Gantry, lo stato dei bloccaggi di un asse si riflette anche sull'altro asse della coppia. Il movimento avviene solamente se entrambi gli assi risultano essere sbloccati (vedi **IBLC#** e **BLCAX#** al cap. 2).

Per un Gantry MULTISLAVE, le considerazioni suddette valgono per tutti gli assi interessati.

#### 3.8.5 Assi in Gantry in ciclo di digitalizzazione

Durante il ciclo di digitalizzazione, gli assi interessati ( X, Y, Z ) al movimento del tastatore sono chiusi in ciclo di spazio dal Software di digitalizzazione che si cura di muoverli in modo tale da mantenere costante la deflessione del tastatore (ovvero il tastatore a contatto col pezzo). Per questo il riferimento assi è calcolato da tale Software ed inviato al convertitore D/A direttamente. Per consentire quindi il movimento in Gantry di due assi (di cui ovviamente solo il Master è uno degli assi X, Y, Z), l'algoritmo di controllo è modificato in modo tale che il riferimento Master non viene alterato dal ciclo Gantry (ciò sposterebbe il tastatore dal profilo del pezzo), mentre il riferimento Slave è uguale a quello Master corretto dalla compensazione Gantry raddoppiata.

#### 3.8.6 Presa di zero

Data la sua particolarità, riteniamo utile suggerire una procedura per la presa di zero su una coppia di assi Gantry. La procedura indicata si riferisce ad un Gantry Vero ma può essere usata come esempio anche per due assi CNC agganciati in Gantry per motivi di lavoro.

Supponiamo di avere assi dotati di micro e marca (ottica) di zero, ma ciò che segue vale anche per assi con encoder simulato.

Obiettivo della presa di zero è il seguente:

- posizionare l'asse Master sulla sua marca ed eseguire il SET;
- posizionare l'asse Slave sulla sua marca e verificare che la quota, o meglio la distanza fra le due marche, sia quella "giusta", ossia uguale (a meno di una soglia desiderata) con quella misurata una prima volta (e conservata).

Quest'obiettivo garantisce la verifica del mantenimento della posizione relativa dei due assi rispetto ad "una prima volta", che quindi fa' fede riguardo all'allineamento meccanico desiderato.

La procedura di set si sviluppa quindi nei seguenti passi:



- 1) Si agganciano in Gantry i due assi (in COSTUSER).
- 2) Si fa, in Gantry, il set ordinario dell'asse Master, vale a dire si cerca il micro, poi la marca e si fa lo zero su di essa; il segnale di SET fatto (OBSE#) diventa 1 su entrambi gli assi. Quando il Master trova la sua marca, campiona le quote di ambedue gli assi e le deposita nelle variabili %%V4209/# (con # corrispondente ai numeri d'ordine dell'asse Master e dello Slave). Tali variabili contengono quindi la misura (ud) della posizione dei due assi in corrispondenza della marca Master.
- 3) Si sposta il Gantry rivolgendo i comandi di spostamento all'asse Slave ricercando la SUA marca/micro (G8nn). I comandi inviati allo Slave vengono in realtà da questo dirottati al Master che li effettua avendo cura di osservare i micro/marca dello Slave. Quando il Master trova la marca dello Slave campiona, ancora una volta, le quote di ambedue gli assi e le deposita nelle variabili %%V4209/#. Tali variabili contengono ora la misura (ud) della posizione dei due assi sulla marca Slave. La differenza di quota può a questo punto essere confrontata con l'analoga differenza misurata una "prima volta" (memorizzata in una variabile ritentiva). In caso di verifica KO la procedura darà un allarme (oppure potrà compensare lo scostamento, se entro una soglia ammessa).

#### File SETGAN per PRESA DI ZERO Assi GANTRY

```
N0 { SET per assi agganciati in GANTRY VERO }
{-----}
{ASSE Master: CNC con Micro M1 lato FC+ }
{ASSE Slave: FRE con Micro M1 lato FC+ }
{ #300 = Flag SET prima volta eseguito }
{ #301 = Memoria per Misura prima volta }
{ R1 = Appoggio per Misura altre volte }
{ R2 = Diff. fra prima volta e successive }
{ - Inserire in COSTUSER l'istruzione, ad es.: }
{ N... <GAN:ON;W;V;75;0.5;1> }
{ Aggancio in Gantry assi W e V }
{-----}
N10 <LCK:OFF>
N20 <##40=nm>
{ Numero ordine Asse Master }
N30 <##41=ns>
{ Numero ordine Asse Slave }
{---- SET asse Master -----}
N110 <FED:<##40>;2000>
N120 G885 <MVM:<##40>;-1000> G91
{ Cerca M1 Master in apertura senso negativo in incrementale }
N130 <WAT:FRE>
N140 G887 <MVM:<##40>;1000>
{ Cerca M1 Master in chiusura in incrementale }
N150 <WAT:FRE>
N160 <FED:<##40>;100>
N170 G885 <MVM:<##40>;-50>
{ Cerca M1 Master in apertura lento }
N180 <WAT:FRE>
N190 G883 <MVM:<##40>;-100>
{ Cerca Marker Master in chiusura }
N200 <WAT:FRE>
```



```
N210 G862 <MVM:<#40>;0>
  {Impone quota ORG= di AXS.TAR sul Marker}
N220 <WAT:FRE>
  {---- Verifica per asse Slave -----}
N240 <FED:<#41>;100>
N250 G885 <MVM:<#41>;-1000> G91
  { Cerca M1 Slave in apertura senso negativo in incrementale }
N260 <WAT:FRE>
N270 G887 <MVM:<#41>;1000>
  { Cerca M1 Slave in chiusura in incrementale }
N280 <WAT:FRE>
N290 <FED:<#41>;100>
N300 G885 <MVM:<#41>;-50>
  { Cerca M1 Slave in apertura lento }
N310 <WAT:FRE>
N320 G883 <MVM:<#41>;-100>
  { Cerca Marker Slave in chiusura }
N330 <WAT:FRE>
N340 <IFF<#300>;N500;N400;N500>
  { Prima volta ? SI -> N400; NO -> N500 }
  {---- Prima Volta -----}
N400 <#301 = LTF(%V4209/<#40>) / V4056/<#40>>
  {Quota Asse Master su Marker Slave; 1^ volta}
N410 <#300 = 1>
N420 <LCK:ON>
N430 <RET>
  {---- Volte Successive -----}
N500 <R1= LTF(%V4209/<#40>) / V4056/<#40>>
  {Quota Asse Master su Marker Slave; altre volte}
N510 <R2= #301 - R1>
  {Differenza fra PRIMO SET e SET SUCCESSIVI }
N520 <IFF< ABS(R2) - 1 >;N600;N600;N800>
  {Diff. minore della massima (1mm) ammessa ? SI -> N600; NO -> N800 }
  {---- Compensazione Disallineamento Gantry ----}
N600 <CGA:<#40>;<R2>;'A'>
N610 <LCK:ON>
N620 <RET>
  {---- Errore > SOGLIA -----}
N800 <#10= 2800n> {definire in ALLARxxx.MU, AL_MUn, il testo dell'allarme}
N810 L<ERROR>
N820 <RET>
```

NOTE:

- 1) L'aggancio in Gantry deve essere fatto in COSTUSER.
- 2) Nella N20 e nella N30 vengono assegnati i numeri d'ordine degli assi Master e Slave alle variabili d'appoggio %V40 e #41, che saranno usate nelle successive istruzioni parametriche FED ed MVM.
- 3) Le variabili permanenti #300 e #301 sono, in quest'esempio, riservate alla procedura di SET del Gantry e non vanno usate per altro scopo; possono essere comunque usate altre variabili nel range #300 . . . #497 (variabili ritentive ad accesso diretto a disposizione del costruttore; vedi manuale PROGRAMMAZIONE AVANZATA 720P371).
- 4) In fase di messa in servizio della macchina, occorre fare il "SET prima volta" che richiede un preventivo allineamento meccanico dei due assi ed una verifica a Set completato.
- 5) Mettendo uguale a 0 la variabile flag #300, si può rifare il "SET prima volta" che deve essere rifatto in caso d'intervento sugli encoder (o sui motori in caso d'encoder simulati), ricaricando o aggiornando il Software del CNC o dopo un "Reset Navetta".



- 6) Dalla N500 vengono fatte le verifiche di posizione relativa, nel caso di "Set non prima volta"; nell'esempio dato, si ammette un disallineamento massimo di 1 mm, entro il quale viene dato il comando di riallineamento <CGA:<#40>;<R2>;'A'> eseguito dal CNC in modo graduale.

### 3.8.7 Assi assoluti Gantry

Per un Gantry "falso" problemi non ce ne sono; difatti, prima di essere agganciati, i due assi Master e Slave vengono comandati ad una posizione/disallineamento voluto, per poter lavorare in maniera corretta.

Per un Gantry "vero", i due o più assi risultano invece essere agganciati in Gantry sin dall'accensione del CNC e possono essere, anche se per pochi micron, "disallineati".

In queste condizioni, pur trattandosi di assi Assoluti, il CNC "richiede" una procedura di SET, intesa come "procedura di compensazione Gantry automatica"; in mancanza di tale procedura, il CNC permetterà di muovere gli assi solamente in modo SET ed in modo JOG, pena l'allarme 10202.

Per gli assi assoluti agganciati in Gantry "vero", è necessario quindi un file di SET contenente la sola istruzione <AGC:...> per la Compensazione Automatica Gantry, cioè per l'allineamento dei due o più assi Master e Slave da effettuarsi all'accensione del CNC.

Per un Gantry MULTISLAVE occorre programmare l'istruzione <AGC:...> per ognuno degli assi Slave.

Esempio 1 SET0:

```
<AGC:X> {Gli assi Master e Slave sono allineati ad una quota "reale" }  
<RET>   {uguale per entrambi gli assi QMaster-QSlave=0 }
```

Esempio 2 SET0:

```
<AGC:X;DP=10.27> {Gli assi Master e Slave sono allineati quando le quote}  
<RET>           {"reali" di Master-Slave differiscono di 10.27 mm }
```

N.B:

Il CNC richiede la "procedura di compensazione Gantry automatica" per assi assoluti nella stessa maniera in cui richiede il SET per assi incrementali, cioè tramite una lista degli assi interessati.

La visualizzazione delle "Quote Reali" per l'asse SLAVE mostra la posizione reale dell'asse, non quella del MASTER.

### 3.8.8 Homing assi Sercos incrementali in Gantry

La gestione dell'homing di un qualsiasi asse, con driver Indramat, è demandata dal CNC al driver stesso. Pertanto il CNC non gestisce lo stop dell'asse quando viene incontrata la marca encoder (istruzione G883 per assi lineari, G903 G907 G911 per assi tondi).

Per due assi in Gantry ciò significa che un asse si può fermare, appena trovata la marca, prima che il suo asse correlato abbia anch'esso terminato il movimento in seguito a marca trovata; il risultato sarebbe un disastroso disallineamento meccanico tra i due assi. È però possibile agire su uno dei parametri "S-0-0150, Reference Offset 1" (in caso di trasduttore sul motore) oppure "S-0-0151, Reference Offset 2" (in caso di trasduttore esterno) per "ritardare" e quindi "traslare" la fermata dell'asse sul quale la marca viene trovata per prima, di uno spazio pari alla distanza tra le marche dei due assi.



Per calcolare automaticamente la distanza tra le marche dei due assi è stata sviluppata l'apposita procedura **G888** per assi lineari, oppure una delle **G951-G954** per assi tonde.

Programmando da CNC una G888 per assi lineari, oppure una delle G951-G954 per assi tonde, viene attivato, su entrambi gli assi master e slave, il comando SERCOS specifico per la determinazione delle posizioni di Marca (P-0-0014 - D500 Command Determine Marker Position A). Tale comando deve essere utilizzato nella procedura di SET di assi in Gantry per automatizzare il calcolo dei parametri di Homing.

Basandosi sul suddetto comando, è stata così personalizzata la procedura di SET per assi Gantry Incrementali.

### 3.8.8.1 SET0 - Esegue il SET degli assi Sercos in GANTRY

Supponendo che i due assi in Gantry siano X l'asse Master di ordine 0, e Y l'asse Slave di ordine 1, la procedura di set per l'asse master è la seguente.

```
N10 G91                {Posizionamenti incrementali}
N20 <FED:X;2000>      {Imposta posizionamento VELOCE}
N30 G885 X-50 G91     {Cerca apertura M1, senso negativo}
N40 <WAT:FRE>
N50 G887 X10000       {Cerca chiusura M1, senso positivo}
N60 <WAT:FRE>
N70 <FED:X;50>        {Imposta posizionamento LENTO}
N80 G885 X-50         {Cerca apertura M1, senso negativo}
N90 <WAT:FRE>
N100 G888 X-20        {Comando D500 su entrambi gli assi, senso negativo}
N110 <WAT:FRE>
N120 <FED:X;200>      {Imposta posizionamento MEDIO}
N130 G887 X10000     {Cerca chiusura M1, senso positivo}
N140 <WAT:FRE>
N150 <FED:X;50>       {Imposta posizionamento LENTO}
N160 G885 X-50       {Cerca apertura M1, senso negativo}
N170 <WAT:FRE>
N180 G883 X-10        {Cerca Marker in chiusura, senso negativo}
N190 <WAT:FRE>
N200 G862 X0          {Impone quota ORG= di AXS.TAR sul Marker}
N210 <WAT:FRE>
N220 ....            Segue Allineamento Gantry (vedi di seguito)
N1000 <RET>
```

I blocchi da N100 a N170 sono i blocchi aggiunti rispetto ad un comune SET assi senza Gantry.

È importante sottolineare che, per ottenere la ripetibilità del SET, il comando D500, attivato tramite la N100 G888, deve essere eseguito nella stessa posizione (nel nostro caso ad uscita micro) e nella stessa direzione (nel nostro caso senso negativo), con le quali si effettuerà in seguito la ricerca Marker (N180 G883).

Eseguito il SET dell'asse, occorre procedere con l'allineamento Gantry. Per fare questo occorre dotare anche l'asse Slave di un Micro di riferimento.

Ecco il seguito della procedura di SET:

```
{ESEGUIE ALLINEAMENTO GANTRY TRA X MASTER ED Y SLAVE}
{N.B: Attenzione ai comandi rivolti all'asse SLAVE }
N220 <FED:Y;2000>    {Imposta posizionamento VELOCE}
```



```
N230 G885 Y-50 G91 {Cerca apertura M1 SLAVE, senso negativo}
N240 <WAT:FRE>
N250 G887 Y10000 {Cerca chiusura M1 SLAVE, senso positivo}
N260 <WAT:FRE>
N270 <FED:Y;50> {Imposta posizionamento LENTO}
N280 G885 Y-50 {Cerca apertura M1 SLAVE, senso negativo}
N290 <WAT:FRE>
N300 <CGA:X;<LTF(3560 - %%V4209/0) / V4056/0>>
      {Allineamento Gantry, quote asse MASTER}
N310 <WAT:FRE>
```

La procedura di allineamento prevede di portarsi su di una posizione nota dell'asse MASTER utilizzando però come punto di riferimento il MICRO dell'asse SLAVE.

L'uscita lenta dal MICRO dell'asse SLAVE è la posizione di riferimento per il confronto delle quote dell'asse MASTER. Terminato il comando G885 della N280, le variabili %%V4209/0 e %%V4209/1, contengono rispettivamente il valore delle quote dell'asse MASTER e dell'asse SLAVE, espresse in conti [ud], relative alla posizione di "uscita MICRO".

Confrontando il valore dell'asse MASTER con quello acquisito (e quindi scritto nel file di SET) la prima volta, quando si è certi dell'allineamento degli assi, è possibile calcolare il disallineamento.

La prima volta si commenterà quindi, nel file di SET, l'istruzione <CGA:...> per non effettuare l'allineamento:

```
{N300 <CGA:X;<LTF(XXXX - %%V4209/0) / V4056/0>>}
```

Eseguito il SET dovrà essere letta la variabile %%V4209/0 dell'asse MASTER; il valore contenuto (es. 3560) dovrà essere scritto sulla N300 al posto di "xxxx":

```
N300 <CGA:X;<LTF(3560 - %%V4209/0) / V4056/0>>
```

La variabile V4056/numero\_asse contiene il coefficiente di conversione tra mm ed unità di misura SA [ud].

### 3.8.9 Homing Assi Gantry SERCOS con righe incrementali assolute

L'Homing degli assi SERCOS viene comandato direttamente dagli azionamenti INDRAMAT in funzione dei dati impostati nel parametro S-0-0147, Homing Parameter. Gli ultimi Firmware ECODRIVE e DIAX supportano il comando "Run Path", impostato attraverso il parametro S-0-0147 con i bit07=0 e bit08=1, specifico per le righe codificate; in questo caso il comando di Homing prevede uno spostamento pari alla distanza tra tre Marker di Riferimento. I due DRIVE correlati degli Assi agganciati in Gantry cominceranno e finiranno lo spostamento in maniera sincrona non generando alcun problema in termini di disallineamento.

I file di SET dell'asse X master (ordine 0), e W slave (ordine 5) sono, ad esempio:

#### File SET5

```
N10 L<SET0> {Richiama SET0}
N20 <RET>
```

#### File SET0

```
N10 <FED:X;1000>
N20 G883 X-10 {Cerca Marker in chiusura}
N30 <WAT:FRE>
N40 G862 X0
N50 <WAT:FRE>
N60 <AGC:X;DP=0> {Compensazione Gantry}
```



---

N70 <WAT:FRE>  
N80 <RET>

### 3.8.10 Gestione riaggancio degli assi in Gantry

Essendo la gestione del Gantry una funzione "supermodale" particolare, il CNC attua degli automatismi di "riaggancio" specifici, qualora la condizione di "assi agganciati in Gantry" venga persa per particolari condizioni di allarme con eventuale perdita del SET di uno dei due assi Master o Slave.

In caso di "Gantry Vero", ossia attivato in COST, la condizione di "assi agganciati in Gantry" viene ripristinata subito, in seguito a RESET; difatti in queste condizioni l'eventuale SET dovrà essere eseguito solamente sull'asse Master e la quota di SET dell'asse Slave verrà imposta pari a quella del Master (come deve essere per assi effettivamente e permanentemente solidali tra loro).

In caso di "Gantry Falso", ossia NON attivato in COST, la condizione di "assi agganciati in Gantry" non è ripristinata automaticamente ma può essere comandata dall'operatore o dal PLC anche prima di rifare il SET. Questo permette, in caso di perdita del SET per spegnimento della Macchina o per fuga Assi, di rifare il SET del Master (se la posizione relativa degli assi lo permette) e di completare la lavorazione del pezzo montato, ad esempio, su due tavole parallele. Naturalmente, in questo caso, non è possibile controllare i fine-corsa software dell'asse Slave ed è responsabilità dell'operatore verificare che non sorgano problemi. Terminata la lavorazione, smontato il pezzo e sganciati gli assi dalla situazione di Gantry, occorrerà fare il SET dell'asse Slave per poterlo riutilizzare normalmente.





### 3.9 BLOCK RETRACE {F}

La funzione di BLOCK-RETRACE rende possibile, durante la lavorazione di un profilo disposto nel piano di contornatura, mettendo la macchina in HOLD, di muoversi indietro/avanti sul percorso fatto fino a quel momento e riprendere quindi a lavorare da un punto qualsiasi di tale percorso.

La prestazione è presente dalla release di Software V2.0 .

Dalla release di Software V2.03, la prestazione può investire anche assi che non appartengono al piano di contornatura; la quantità di assi associati al retrace può essere al massimo 5 e minimo 1. Fra questi debbono esserci ambedue gli assi del piano di contornatura oppure nessuno di essi; infatti se si abilita il retrace su un solo asse del p.c. è impossibile gestire eventuali cerchi.

Normalmente la prestazione è disattivata; per renderla operativa occorre agire tramite il PLC che ha a disposizione alcune funzioni che permettono di gestire il Block-retrace. Inoltre, il PLC deve mettere a disposizione dell'operatore almeno due tasti:

- tasto "RETRACE-" per andare indietro
- tasto "RETRACE+" per andare avanti

Per un funzionamento sicuro, i tasti suddetti debbono avere un funzionamento del tipo "monostabile" in modo che il rilascio del tasto stesso (anche nel caso di tasto-funzione) arresti il movimento in corso.

Possono essere aggiunti anche altri tasti per:

- "MODIFICA FEED" di movimentazione avanti/indietro.
- "TAGLIA/RIATTIVA" il percorso già fatto.

La libreria PLC mette a disposizione una serie di funzioni appositamente scritte per gestire il funzionamento di BLOCK-RETRACE.

#### 3.9.1 Funzioni a disposizione del PLC

##### 3.9.1.1 Funzione " plcrtrc "

Chiamata: **esito = plcrtrc (cmd\_cod);**

Scopo: Invia al CNC un **comando** di BLOCK-RETRACE.

Parametro di ingresso:

**cmd\_cod** = (short) Codice del comando. Può assumere uno dei seguenti valori:

- |          |  |
|----------|--|
| <b>0</b> | Disabilita la funzione di Block-retrace        |
| <b>1</b> | Abilita la funzione                            |
| <b>2</b> | Muovi gli assi in avanti sul percorso fatto    |
| <b>3</b> | Muovi gli assi all'indietro sul percorso fatto |
| <b>4</b> | Arresta il movimento indietro/avanti in corso  |

Parametro di uscita:

**esito** = (short) Valore di ritorno della routine. Può assumere uno dei seguenti valori:

- |          |                   |
|----------|-------------------|
| <b>1</b> | Comando accettato |
|----------|-------------------|



- 1 Codice **cmd\_cod** errato; valori ammessi: 0 . . . 4 .
- 2 E' stato richiesto un movimento ma il Block-retrace non è abilitato
- 3 Si vuole disabilitare il Block-retrace mentre è in esecuzione il Part-program.
- 4 Si tenta di abilitare retrace ma tra gli assi attivati c'è un solo asse del p.c.
- 5 Richiesta di movimento non accettata. CNC non in hold, part-program non in esecuzione, modo attuale non automatico.
- 6 Si vuole disabilitare il Block-retrace mentre è in fase di ricerca della N da cui ripartire.

### 3.9.1.2 Funzione " plcrtrcf "

Chiamata: **esito = plcrtrcf (feed);**

Scopo: stabilisce con quale feed fare i movimenti di BLOCK-RETRACE.

Parametro di ingresso:

**feed** = (double) esprime la velocità in mm/min. Se questa funzione non è usata, la feed è posta automaticamente uguale a 100 mm/min.

Parametro di uscita:

**esito** = (short) Valore di ritorno della routine. Può assumere uno dei seguenti valori:

- 1 Comando accettato
- 1 Comando non accettato: Block-retrace non attivo

### 3.9.1.3 Funzione " fasertrc "

Chiamata: **stato = fasertrc ( );**

Scopo: Legge lo stato attuale della funzionalità di BLOCK-RETRACE.

Parametro di uscita:

**stato** = (unsigned short) valore di ritorno della routine. La word contiene alcuni bits con il seguente significato:

- bit 0** =1: il retrace è abilitato.
- bit 1** =1: il retrace è in fase di movimentazione (fermo o no); siamo cioè nella fase in cui l'operatore sta cercando il punto giusto per ripartire a lavorare.
- bit 2** =1: il retrace è in fase di riletture del Part-program.
- bit 3** =1: il retrace è in fase di riposizionamento
- bit 4** =1: il retrace è in fase di reset automatico

#### Nota:

Con la release SW V3.02 è stato aggiunto il bit 4 in modo che il PLC, chiamando continuamente la funzione *fasertrc()*, possa distinguere la premuta del tasto RESET, dal reset emesso automaticamente dal CNC durante la fase di retrace+ oppure retrace-.

### 3.9.1.4 Funzione " assirtrc "

Chiamata: **esito = assirtrc (&assi, qassi);**

Scopo: Definisce gli assi oggetto di BLOCK-RETRACE (da V2.03).

Parametri di ingresso:



**assi** = (char \*) Puntatore a un array di char dove ciascun carattere deve essere inizializzato col numero di ordine dell'asse da attivare.

**qassi** = (unsigned short) Quantità di assi da attivare

Parametro di uscita:

**esito** = (short) Valore di ritorno della routine. Può assumere uno dei seguenti valori:

- 1 Comando accettato
- 1 errore: il retrace è attivo; non si possono cambiare assi "al volo"
- 2 errore: superata la massima quantità di assi consentita: 5
- 3 errore: il numero di ordine dell'asse non è corretto
- 4 errore: gli assi p.c. debbono esserci entrambi o nessuno

Se il PLC non usa questa funzione per definire gli assi oggetto di retrace, il CNC associa al retrace i due assi del piano di contornatura.

Il RESET, "%", ":" mantengono gli ultimi assi associati.

Non è possibile programmare nello stesso blocco assi retrace e assi non sottoposti al retrace.

### 3.9.1.5 Funzione " cuttrc "

Chiamata: **esito = cuttrc (cmd\_cod);**

Scopo: Invia al CNC un comando di "taglio" del percorso di BLOCK-RETRACE (da V2.03).

Parametro di ingresso:

**cmd\_cod** = (short) Codice del comando. Può assumere uno dei seguenti valori:

- 1 Disabilita la parte di profilo, memorizzata nel file di retrace, precedente al punto in cui si trova attualmente la Macchina
- 0 Riabilita la parte di profilo che era disabilitata

Parametro di uscita:

**esito** = (short) Valore di ritorno della routine. Può assumere uno dei seguenti valori:

- 1 Comando accettato
- 1 errore: il retrace non è attivo o si trova in uno stato non idoneo all'operazione richiesta

## 3.9.2 Note di funzionamento

Il PLC normalmente abiliterà il Block-retrace al "pon" (power-on) chiamando la funzione:

```
ESITO= plcrtrc (1);
```

definendo eventualmente una FEED di default:

```
plcrtrcf (500);
```

Con questo comando il CNC si pone nello stato di Block-retrace abilitato.

Quando il CNC è nel modo Automatico/Esecuzione ed è eseguito un Part-program, la funzione di Block-retrace copia in un apposito file gli spostamenti fatti dagli assi abilitati al retrace (normalmente quelli del piano di contornatura); facendo RESET o iniziando un nuovo programma il file viene riscritto dall'inizio, mentre il retrace rimane abilitato.

Durante l'esecuzione del programma, l'operatore, notando qualcosa di non corretto, preme HOLD (la lavorazione è così sospesa) poi preme "RETRACE-" ed eventualmente "RETRACE+" per tornare su un punto del profilo da cui ripartire.



Il PLC, tramite la funzione `plcrtrc (..)`, invia i comandi relativi al CNC (codici `cmd_cod`: 2 = avanti, 3 = indietro, 4 = stop; vedi esempio di PLC più avanti).

In questa fase (detta di movimentazione) il CNC emette un RESET tutte le volte che si preme RETRACE- o RETRACE+, vengono così annullate tutte le funzioni ausiliarie, spostamenti origini e quant'altro emesso da programma tranne le matrici dinamiche. Le quote visualizzate, durante questa fase, sono riferite all'origine base (G54.1 o quella eventualmente definita in COSTUSER).

Durante questa fase l'operatore può uscire dal modo AUTO e, ad esempio, passare in JOG per andare fuori del percorso e fare dei controlli. Volendo poi continuare la fase di movimentazione, l'operatore dovrà ritornare in AUTO; il CNC richiederà in questo caso il riposizionamento sul punto del percorso lasciato in precedenza prima di poter riprendere la movimentazione avanti/indietro.

Scelto il punto da cui ripartire a lavorare, l'operatore deve premere il tasto REL; il CNC inizia così la fase di rilettura del Part-Program fino al punto determinato dalla posizione in cui si trova attualmente la macchina. La rilettura inizia con l'emissione di un RESET; segue quindi la lettura del Part-Program con l'emissione di tutte le funzioni incontrate. Se tra queste non vi sono funzioni ausiliarie, la macchina ricomincia subito a lavorare, altrimenti il CNC passa automaticamente alla fase di riposizionamento: si apre la tendina con l'elenco degli organi da riposizionare ed all'operatore è richiesto di comandare il riposizionamento gli organi stessi. Fatto ciò, l'operatore deve premere il tasto per tornare in AUTO e con questo la lavorazione riparte. Gli organi da riposizionare sono tutti quelli incontrati nella fase di lettura anche se le relative funzioni ausiliarie erano state eseguite in precedenza; questo perché sicuramente tali funzioni sono state annullate dai precedenti RESET.

Durante l'esecuzione del programma, l'operatore può decidere di precludere al movimento indietro la parte di percorso che va dall'inizio al punto attuale; per fare questo, mentre l'utensile sta percorrendo il profilo, può usare, ad esempio, un apposito pulsante, previsto dal costruttore della Macchina, che attiva nel PLC la funzione `cuttrc (1)`. Allo stesso modo, il retrace può essere riattivato fino ad inizio part-program tramite la funzione `cuttrc (0)`.

### 3.9.3 Esempio di gestione Block-retrace da PLC

Qui di seguito è riportato un frammento di PLC che realizza la funzione di Block-retrace:

```
/*-----*/
/*   GESTIONE BLOCK-RETRACE   */
/*-----*/
if (pon)
{
  plcrtrc (1);
  /* Abilita la funzione di BLOCK-RETRACE */
  plcrtrcf (500);
  /* Feed per i movimenti di BLOCK-RETRACE */
  ....
}
...
LMP1001= PUSER && (CODTAS == 1001) ||
        LMP1001 && !(PUSER && ((short)CODTAS == -1001));
/* Pulsante RETRACE+ */

LMP1002= PUSER && (CODTAS == 1002) ||
        LMP1002 && !(PUSER && ((short)CODTAS == -1002));
/* Pulsante RETRACE- */
```



```
e_u_wrw(21,WLAMP1);
/* Word lampade per MENU verticale */

if(LMP1001 && !ALMP1001)
{
  plcrtrc(2);
  /* Muovi gli assi in avanti sul percorso fatto */
  ALMP1001= 1;
}
else if(!LMP1001 && ALMP1001)
{
  plcrtrc(4);
  /* Arresta il movimento in corso */
  ALMP1001= 0;
}

if(LMP1002 && !ALMP1002)
{
  plcrtrc(3);
  /* Muovi gli assi all'indietro sul percorso fatto */
  ALMP1002= 1;
}
else if(!LMP1002 && ALMP1002)
{
  plcrtrc(4);
  /* Arresta il movimento in corso */
  ALMP1002= 0;
}
/*-----*/
```

### 3.10 MODIFICA ACCELERAZIONE ASSI

La gestione della modifica dell'accelerazione degli assi è realizzata tramite apposite istruzioni di Part-Program scritte in linguaggio LIP e con funzioni di libreria PLC.

#### 3.10.1 Gestione da Part-Program

##### 3.10.1.1 Modifica Accelerazione

<ACC:Asse; AccRap; AccLav>

<b>Asse</b>	Nome o numero d'ordine dell'asse.
<b>AccRap</b> (double)	Indica l'accelerazione NOMINALE DI RAPIDO in [mm/(min x sec)] per assi lineari o in [giri/(min x sec)] per assi rotanti.
<b>AccLav</b> (double)	Indica l'accelerazione di LAVORO espressa come coefficiente moltiplicativo dell'ACCELERAZIONE NOMINALE.

##### 3.10.1.2 Ripristina Accelerazione di taratura

<TAC:Asse>



**Asse** Nome o numero d'ordine dell'asse.

I parametri dinamici assi "ACC" (accelerazione) e "KV", modificati durante l'esecuzione di un part-program oppure in MANUALE, tramite le istruzioni LIP <ACC:...> e <PKV:...> vengono ripristinati ai valori "nominali" di taratura nei seguenti casi:

- Con l'istruzione "%"
- Con RESET
- Al passaggio Manuale–Automatico se condizioni iniziali = 0 (part-program non in esecuzione).

NON vengono ripristinati i valori di taratura in caso di KEY, ed al passaggio MANUALE–AUTOMATICO se condizioni iniziali != 0 (part-program in esecuzione sospeso); questo significa che in teoria è possibile, durante l'esecuzione di un part-program, passare in ESECUZIONE-SINGOLA, andare in MANUALE, cambiare i valori di ACCELERAZIONE, tornare in AUTOMATICO, quindi lavorare con i nuovi parametri di accelerazione. Questa risulta comunque essere una manovra fortemente sconsigliata, da poter effettuare solamente previa sincronizzazione tramite F.A. (WAIT) da part-program, in quanto il CNC acquisisce i parametri di accelerazione in fase di interpretazione, cioè guardando "in avanti" rispetto all'esecuzione del blocco effettivamente in corso; al riguardo il CNC non effettua alcun controllo.

### 3.10.2 Gestione da PLC

#### 3.10.2.1 AxAcc: Modifica Accelerazione Asse

Chiamata: **esito = AxAcc (IndAx,AccRap,AccLav)**

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

**AccRap** = (double) Accelerazione Nominale di Rapido dell'asse espressa in [mm/(min x sec)] o [gradi/(min x sec)]. In caso AccRap sia < 0 il CNC impone il suo valore positivo AccRap = -AccRap.

**AccLav** = (double) Accelerazione di Lavoro dell'asse espressa come coefficiente moltiplicativo di AccRap. In caso AccLav sia < 0 il CNC impone il suo valore positivo AccLav = -AccLav. Dovendo essere l'Accelerazione di Lavoro inferiore a quella Rapida, il CNC controlla che AccLav sia >0 e <=1; in caso contrario impone AccLav=1.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |  |
|----------|--|
| <b>0</b> | Successo – comando inviato               |
| <b>1</b> | Insuccesso – numero d'ordine asse errato |
| <b>2</b> | Insuccesso – canale comandi occupato     |

N.B.:

Tale funzione è supermodale; una volta cambiati, i valori d'accelerazione di AXS.TAR potranno essere ripristinati solamente con OFF/ON CNC (o ALT/EXE), oppure utilizzando la funzione SavAxAccTar() per salvarne i valori e quindi ripristinarli con un'ulteriore AxAcc().

Occorre predisporre il PLC affinché chiami la funzione AxAcc() solamente se il CNC non è in RUN (is\_cnc\_work() == 0) oppure sincronizzare il tutto con una funzione ausiliaria (WAIT).

#### 3.10.2.2 SavAxAccTar: Salva Parametri di Taratura Accelerazione Asse

Chiamata: **esito = SavAxAccTar (IndAx,&SAccRap,&SAccLav)**

Parametri di ingresso:



**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

**SAccRap** = (double \*) Puntatore alla variabile dove salvare l'Accelerazione Nominale di Rapido dell'asse.

**SAccLav** = (double \*) Puntatore alla variabile dove salvare l'Accelerazione di Lavoro dell'asse.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 0        Successo – comando inviato
- 1        Insuccesso – numero d'ordine asse errato

### 3.11 MODIFICA DELLA FEED E SPEED

La gestione della modifica dei valori di default, dei valori limite e dei valori attuali della Feed Assi e della Speed Mandrino (Principale e Secondario) è realizzata tramite apposite funzioni di libreria PLC, riassunte nelle tabelle seguenti e descritte più avanti in questo stesso capitolo.

<b>SetAxDefFeed( )</b>	Imposta la velocità (Feed) di Default Asse Indipendente
<b>SetDefFeed()</b>	Imposta la velocità (Feed) di Default Assi CNC
<b>SetAxLimFeed()</b>	Limita la velocità massima Asse
<b>SetAxTarFeed()</b>	Ripristina la velocità Asse di taratura
<b>SetLimFeedOn()</b>	Limita la velocità di percorso della punta utensile
<b>SetLimFeedOff()</b>	Annula il limite di velocità di percorso della punta utensile
<b>SetFeed()</b>	Setta la velocità (Feed) degli assi CNC (Contouring)
<b>ValueAxDefFeed()</b>	Legge il valore della Feed di Default Asse Indipendente
<b>ValueDefFeed()</b>	Legge il valore della Feed di Default Assi CNC
<b>ValueAxLimFeed()</b>	Legge il valore limite attuale per la velocità Asse
<b>ValueAxTarFeed()</b>	Legge il valore di taratura per la velocità Asse
<b>ValueLimFeed()</b>	Legge il valore limite attuale per la velocità di percorso
<b>ValueFeed()</b>	Legge il valore della Feed in esecuzione degli assi CNC
<b>ValueDefModeCNC()</b>	Legge il modo G94/96 di Default del CNC
<b>ValueModeCNC()</b>	Legge il modo G94/96 in esecuzione del CNC
<b>DisableRqAxes()</b>	Disabilita richieste varie da PLC

<b>SetDefSpeed()</b>	Imposta la velocità di rotazione (Speed) di Default Mandrino
<b>SetLimSpeed()</b>	Limita la velocità massima Mandrino
<b>SetTarSpeed()</b>	Ripristina la velocità di taratura del Mandrino
<b>SetSpeed()</b>	Setta il valore di velocità di rotazione (Speed) del Mandrino
<b>ValueDefSpeed()</b>	Legge il valore della Speed di default
<b>ValueLimSpeed()</b>	Legge il valore limite attuale per la Speed
<b>ValueTarSpeed()</b>	Legge il valore di taratura per la Speed
<b>ValueSpeed()</b>	Legge il valore di velocità di rotazione (Speed) del Mandrino

Le seguenti funzioni riguardano il mandrino secondario ed hanno lo stesso significato di quelle viste sopra per il mandrino principale, alle quali si rimanda per la sintassi e la descrizione.

<b>SetDefSpeedSec()</b>	Imposta la velocità di rotazione (Speed) di Default Mand.Sec.
<b>SetLimSpeedSec()</b>	Limita la velocità massima Mand.Sec.
<b>SetTarSpeedSec()</b>	Ripristina la velocità di taratura del Mand.Sec.
<b>SetSpeedSec()</b>	Setta il valore di velocità di rotazione (Speed) del Mand.Sec.
<b>ValueDefSpeedSec()</b>	Legge il valore della Speed di default del Mand.Sec.
<b>ValueLimSpeedSec()</b>	Legge il valore limite attuale per la Speed del Mand.Sec.



---

---

<b>ValueTarSpeedSec()</b>	Legge il valore di taratura per la Speed del Mand.Sec.
<b>ValueSpeedSec()</b>	Legge il valore di velocità di rotazione (Speed) del Mand.Sec.

### 3.11.1 Feed Assi

#### 3.11.1.1 SetAxDefFeed: Imposta la velocità (Feed) di Default Asse Indipendente

Chiamata: **esito = SetAxDefFeed (IndAx, DefFeed)**

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

**DefFeed** = (double) Valore della velocità (Feed) di default dell'asse FRE espressa in [mm/min] o [gradi/min].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 0 Successo – comando inviato
- 1 Insuccesso – numero d'ordine asse errato
- 2 Insuccesso – canale comandi occupato

#### 3.11.1.2 SetDefFeed: Imposta la velocità (Feed) di Default Assi CNC

Chiamata: **esito = SetDefFeed (DefFeed)**

Parametro di ingresso:

**DefFeed** = (double) Valore della velocità (Feed) di default degli assi CNC espressa in [mm/min] o [gradi/min] o [mm/giro].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 0 Successo – comando inviato
- 2 Insuccesso – canale comandi occupato

#### 3.11.1.3 SetAxLimFeed: Limita la velocità massima Asse

Chiamata: **esito = SetAxLimFeed (IndAx, LimFeed)**

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

**LimFeed** = (double) Valore limite di velocità dell'asse espressa in [mm/min] o [gradi/min].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 0 Successo – comando inviato
- 1 Insuccesso – numero d'ordine asse errato
- 2 Insuccesso – canale comandi occupato

#### 3.11.1.4 SetAxTarFeed: Ripristina la velocità Asse di taratura

Chiamata: **esito = SetAxTarFeed (IndAx)**

Parametro di ingresso:





**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |  |
|----------|--|
| <b>0</b> | Successo – comando inviato               |
| <b>1</b> | Insuccesso – numero d'ordine asse errato |
| <b>2</b> | Insuccesso – canale comandi occupato     |

### 3.11.1.5 SetLimFeedOn: Limita la velocità di percorso della punta utensile

Chiamata: **esito = SetLimFeedOn (LimFeed)**

Parametro di ingresso:

**LimFeed** = (double) Valore limite di velocità della punta utensile espressa in [mm/min] o [gradi/min].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine.

- |          |                            |
|----------|----------------------------|
| <b>0</b> | Successo – comando inviato |
|----------|----------------------------|

### 3.11.1.6 SetLimFeedOff: Annulla il limite di velocità di percorso della punta utensile

Chiamata: **esito = SetLimFeedOff ()**

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine.

- |          |                            |
|----------|----------------------------|
| <b>0</b> | Successo – comando inviato |
|----------|----------------------------|

### 3.11.1.7 SetFeed: Setta la velocità (Feed) degli assi CNC (Contouring)

Chiamata: **esito = SetFeed (Feed)**

Parametro di ingresso:

**Feed** = (double) Valore della velocità programmata per gli assi CNC espressa in [mm/min] o [gradi/min].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine.

- |          |                                      |
|----------|--------------------------------------|
| <b>0</b> | Successo – comando inviato           |
| <b>2</b> | Insuccesso – canale comandi occupato |

### 3.11.1.8 ValueAxDefFeed: Legge il valore della Feed di Default Asse Indipendente

Chiamata: **esito = ValueAxDefFeed (IndAx, &SDefFeed)**

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

**SDefFeed** = (double \*) Puntatore alla variabile dove salvare il valore della Feed di default dell'asse FRE IndAx.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

---



---

0	Successo – valore letto
1	Insuccesso – numero d'ordine asse errato

### 3.11.1.9 ValueDefFeed: Legge il valore della Feed di Default Assi CNC

Chiamata: **esito = ValueDefFeed (&SDefFeed)**

Parametro di ingresso:

**SDefFeed** = (double \*) Puntatore alla variabile dove salvare il valore della Feed di default degli assi CNC.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Assume sempre il seguente valore:

0	Successo – valore letto
---	-------------------------

### 3.11.1.10 ValueAxLimFeed: Legge il valore limite attuale per la velocità Asse

Chiamata: **esito = ValueAxLimFeed (IndAx, &SLimFeed)**

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

**SLimFeed** = (double \*) Puntatore alla variabile dove salvare il valore limite attuale di velocità

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

0	Successo – valore letto
1	Insuccesso – numero d'ordine asse errato

### 3.11.1.11 ValueAxTarFeed: Legge il valore di taratura per la velocità Asse

Chiamata: **esito = ValueAxTarFeed (IndAx, &STarFeed)**

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

**STarFeed** = (double \*) Puntatore alla variabile dove salvare il valore di taratura di velocità.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

0	Successo – valore letto
1	Insuccesso – numero d'ordine asse errato

### 3.11.1.12 ValueLimFeed: Legge il valore limite attuale per la velocità di percorso

Chiamata: **esito = ValueLimFeed (&SLimFeed)**

Parametro di ingresso:

**SLimFeed** = (double \*) Puntatore alla variabile dove salvare il valore limite attuale di velocità di percorso della punta utensile in [mm/min] o [gradi/min].

Parametro di uscita:



---

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 0**      Successo – valore letto
- 1**      Insuccesso – limitazione non attiva

#### 3.11.1.13      **ValueFeed: Legge il valore della Feed degli assi CNC**

Chiamata: **esito = ValueFeed (&SFeed)**

Parametro di ingresso:

**SFeed** = (double \*) Puntatore alla variabile dove salvare il valore attuale di Feed dell'interpolatore per gli assi CNC (in Contouring) in [mm/min] o [gradi/min] o [mm/giro].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine.

- 0**      Successo – valore letto

#### 3.11.1.14      **ValueFeedEx: Legge il valore della Feed in esecuzione degli assi CNC**

Chiamata: **esito = ValueFeedEx (&SFeed)**

Simile alla precedente funzione: la Feed restituita è quella attuale dell'interpolatore.

Questa funzione non è più presente dalla release software V4.00 .

#### 3.11.1.15      **ValueDefModeCNC: Ritorna la modalità G94\_96 di Default del CNC**

Chiamata: **esito = ValueDefModeCNC (&Mode)**

Parametro di ingresso:

**Mode** = (unsigned short \*) Puntatore alla variabile dove salvare il valore di default della modalità di avanzamento assi del CNC.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine.

- 0**      Successo – valore letto

Nella variabile Mode viene scritto:

- 94**      Modalità di avanzamento di default G94 F[mm/min], S[gradi/min]
- 95**      Modalità di avanzamento di default G95 F[mm/giro], S[gradi/min]
- 96**      Modalità di avanzamento di default G96 F[mm/giro], S[metri/min]

#### 3.11.1.16      **ValueModeCNC: Ritorna la modalità G94\_96 del CNC**

Chiamata: **esito = ValueModeCNC (&Mode)**

Simile alla precedente funzione: la modalità restituita è quella attuale dell'interpolatore.

#### 3.11.1.17      **ValueModeCNCEx: Ritorna la modalità G94\_96 eseguita del CNC**

Chiamata: **esito = ValueModeCNCEx (&Mode)**

Simile alle precedenti funzioni: la modalità restituita è quella attuale dell'interpolatore.

Questa funzione non è più presente dalla release software V4.00 .



---

### 3.11.1.18 DisableRqAxes: Disabilita alcune richieste del PLC

Chiamata: **DisableRqAxes ( )**

Eventualmente disabilita le richieste:

- "Nuova Limitazione Feed"
- "Nuova Limitazione Speed"
- "Nuova Feed di Default per gli assi indipendenti"
- "Nuovi Fine Corsa da PLC"
- "Nuove Accelerazioni da PLC"
- "Nuove Soglie di Posizionamento e di Fuga da PLC"

di tutti gli assi; può essere utile per velocizzare il processo dell'interprete del Part-Program (MAMUT).  
Le singole funzionalità vengono disabilitate se non ci sono richieste in corso.

### 3.11.2 Speed Mandrino

#### 3.11.2.1 SetDefSpeed: Imposta la velocità di rotazione (Speed) di Default Mandrino

Chiamata: **esito = SetDefSpeed (DefSpeed)**

Parametro di ingresso:

**DefSpeed** = (double) Valore della velocità di rotazione del mandrino (Speed) di default espressa in [giri/min] in gamma diretta.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |                                      |
|----------|--------------------------------------|
| <b>0</b> | Successo – comando inviato           |
| <b>1</b> | Errore – asse Mandrino non presente  |
| <b>2</b> | Insuccesso – canale comandi occupato |

**Nota:**

Il valore di default della Speed viene ripristinato ad inizio Part-Program (con l'istruzione "%"), con la condizione di "Mandrino Fermo".

#### 3.11.2.2 SetLimSpeed: Limita la velocità massima Mandrino

Chiamata: **esito = SetLimSpeed (LimSpeed)**

Parametro di ingresso:

**LimSpeed** = (double) Valore limite di velocità del mandrino (Speed) espressa in [giri/min] in gamma diretta.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |                                      |
|----------|--------------------------------------|
| <b>0</b> | Successo – comando inviato           |
| <b>1</b> | Errore – asse Mandrino non presente  |
| <b>2</b> | Insuccesso – canale comandi occupato |

#### 3.11.2.3 SetTarSpeed: Ripristina la velocità di taratura del Mandrino

Chiamata: **esito = SetTarSpeed ( )**

Parametro di uscita:



**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |                                      |
|----------|--------------------------------------|
| <b>0</b> | Successo – comando inviato           |
| <b>1</b> | Errore – asse Mandrino non presente  |
| <b>2</b> | Insuccesso – canale comandi occupato |

#### 3.11.2.4 SetSpeed: Setta il valore di velocità di rotazione (Speed) del Mandrino

Chiamata: **esito = SetSpeed (Speed)**

Parametro di ingresso:

**Speed** = (double) Valore della velocità di rotazione del mandrino (Speed) espressa in [giri/min].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |   |
|----------|---|
| <b>0</b> | Successo – comando inviato                |
| <b>1</b> | Errore – asse Mandrino non presente       |
| <b>2</b> | Insuccesso – canale comandi occupato      |
| <b>3</b> | Insuccesso – non è attiva la modalità G94 |

**Nota:**

La funzione SetSpeed() è completamente asincrona rispetto all'esecuzione di una "S" da Part-Program e "sovrascrive" il valore dell'ultima Speed ("S") "vista dal Software Assi", che può essere diversa dall'ultima programmata (ad esempio una M05 impone zero sul valore "visto dal Software Assi"). Occorre anche considerare che la funzione SetSpeed(), a differenza di una programmazione esplicita S... nel Part-Program, non comanda un eventuale cambio gamma che sarebbe necessario per realizzare la nuova Speed mandrino; in questo caso, la Speed mandrino viene comunque limitata al massimo valore ammesso per la gamma attuale.

#### 3.11.2.5 ValueDefSpeed: Legge il valore della Speed di default

Chiamata: **esito = ValueDefSpeed (&SDefSped)**

Parametri di ingresso:

**SDefSped** = (double \*) Puntatore alla variabile dove salvare il valore di default della velocità di rotazione (Speed) del mandrino in gamma diretta.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |                                     |
|----------|-------------------------------------|
| <b>0</b> | Successo – valore letto             |
| <b>1</b> | Errore – asse Mandrino non presente |

#### 3.11.2.6 ValueLimSpeed: Legge il valore limite attuale per la Speed

Chiamata: **esito = ValueLimSpeed (&SLimSped)**

Parametri di ingresso:

**SLimSped** = (double \*) Puntatore alla variabile dove salvare il valore limite attuale della velocità di rotazione (Speed) del mandrino in gamma diretta.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |                                     |
|----------|-------------------------------------|
| <b>0</b> | Successo – valore letto             |
| <b>1</b> | Errore – asse Mandrino non presente |
-



---

### 3.11.2.7 ValueTarSpeed: Legge il valore di taratura per la Speed

Chiamata: **esito = ValueTarSpeed (&STarSped)**

Parametri di ingresso:

**STarSped** = (double \*) Puntatore alla variabile dove salvare il valore di taratura della velocità di rotazione (Speed) del mandrino in gamma diretta.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

<b>0</b>	Successo – valore letto
<b>1</b>	Errore – asse Mandrino non presente

### 3.11.2.8 ValueSpeed: Legge il valore di velocità di rotazione (Speed) del Mandrino

Chiamata: **esito = ValueSpeed (&SSped)**

Parametri di ingresso:

**SSped** = (double \*) Puntatore alla variabile dove salvare il valore della velocità di rotazione (Speed) programmata del mandrino in [giri/min].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

<b>0</b>	Successo – valore letto
<b>1</b>	Errore – asse Mandrino non presente

### 3.11.3 Note sulla gestione della modifica Feed Assi e Speed Mandrino

Tutte le funzioni "Set...()" sono "supermodali"; i parametri modificati vengono ripristinati solamente con OFF/ON CNC (o ALT/EXE) oppure con una nuova riprogrammazione.

Per gli Assi Indipendenti, cioè agganciati in punto a punto, i valori limite di Feed/Speed impostati con le suddette funzioni "Set...()" agiscono a "valle" degli override (OVRFREnn); l'eventuale risultato "*Feed Desiderata = Feed Programmata x Override*" viene confrontato con il limite attuale di velocità impostato e limitato in caso quest'ultimo venga superato.

Per gli Assi CNC, il limite di velocità dei singoli assi viene utilizzato per costruire la velocità di movimento sul percorso programmato (la "risultante" dei singoli spostamenti degli assi interpolanti), agendo quindi a "monte" dell'override sia per i movimenti in lavoro (IOVF) sia per quelli in rapido (IOVR).

Ne deriva che per gli ASSI INDIPENDENTI le funzioni di limitazione "Set...()" della Feed o della Speed vengono acquisite anche dinamicamente, ovvero ad assi in moto; per gli ASSI CNC invece, essendo la velocità max. ammessa acquisita in fase in "interpretazione" del Part-Program o del comando in manuale, è opportuno sincronizzare la chiamata a tali funzioni tramite una F.A. (di tipo WAIT), che agisca in un determinato blocco del Part-Program.

Per gli Assi CNC, è possibile però impostare la limitazione della Feed di percorso utensile (SetLimFeedOn() ) in qualsiasi momento in quanto agisce dinamicamente, con CNC in RUN, "a valle" degli override, limitando, se occorre, la velocità desiderata sulla punta utensile (*Feed Desiderata = Feed Programmata x Override*).

Le Feed e Speed di Default, impostate tramite le funzioni:

`SetAxDefFeed()` per gli assi indipendenti



---

---

SetDefFeed() per la velocità di percorso degli assi in contouring  
SetDefSpeed() per la velocità di rotazione del mandrino

vengono acquisite nei cambi modo AUTO→MDI oppure AUTO→JOG oppure quando viene interpretata l'istruzione "%" di inizio Part-Program in modo AUTO.

Le Feed e Speed "di Default" sono infatti i valori che il CNC usa in mancanza di una programmazione esplicita tramite le apposite istruzioni <FED:..., F..., S... .

È opportuno ricordare che, per gli assi CNC, la Feed programmata è valida solamente per gli spostamenti "in lavoro" in quanto per gli spostamenti "in rapido" (G00) sono valide le velocità rapide definite in AXS.TAR con l'eventuale limite "imposto" tramite la funzione SetLimFeedOn().

### 3.11.4 Esempio di uso delle funzioni di limitazione nel PLC

Si vuole limitare, con il segnale LIMITA, la Feed dell'asse X (asse di ordine 0 in AXS.TAR) al 25 % e la Speed del Mandrino al 50% della massima:

```
if (LIMITA && !SETLIM)
{ /* La Speed e la Feed vengono limitate */
double LimFeed, LimSpeed;

ValueAxTarFeed( 0, &LimFeed );
/* Legge il valore di Taratura dell'asse X */
ValueTarSpeed( &LimSpeed );
/* Legge il valore di Taratura dell'asse Mandrino */

LimFeed= LimFeed * 0.25;
/* Feed limitata al 25 % */
LimSpeed= LimSpeed * 0.5;
/* Speed limitata al 50 % */

SetAxLimFeed( 0, LimFeed );
SetLimSpeed( LimSpeed );
/* Assegno nuovi limiti */

SETLIM= 1;
/* Limiti settati */
}
else if (SETLIM && !LIMITA)
{ /* La Speed e la Feed vengono ripristinate */
SetTarSpeed();
/* Ripristina Speed di Taratura dell'asse Mandrino */
SetAxTarFeed( 0 );
/* Ripristina Feed di Taratura dell'asse X */

SETLIM= 0;
/* Valori di taratura ripristinati */
}
```



### 3.12 SINCRONIZZAZIONE ASSI

In questo paragrafo è descritto come realizzare la sincronizzazione del movimento di un asse con il raggiungimento di determinate posizioni (quote) di un altro asse.

Tale funzionalità consente di programmare il movimento su una o più quote di un asse che chiameremo "Slave", in funzione del raggiungimento di una o più quote di un altro asse che chiameremo "Master" (da non confondere con gli analoghi nomi usati per gli assi in modalità Gantry).

La partenza del movimento dell'asse Slave avviene al raggiungimento della quota predefinita del Master; tale condizione ("quota Master di sincronizzazione raggiunta") nel seguito sarà indicata anche come segnale di "trigger". È importante sottolineare che non si vuole sincronizzare il raggiungimento delle quote Master/Slave, bensì l'inizio del moto dell'asse Slave.

Per la programmazione occorre fornire in input un certo numero di "punti di sincronizzazione", ciascuno dei quali è costituito da tre variabili:

Quota asse Master            espressa in [mm] o [gradi]  
Quota asse Slave espressa in [mm] o [gradi]  
Velocità Asse Slave        espressa in [mm/min] o [gradi/min]

Le quote assi Master e Slave sono "assolute", cioè relative allo zero macchina.

La definizione del "Punto di Sincronizzazione" avviene in "C" tramite la struttura SYNCH\_POINT così definita dal CNC e "vista" dal PLC:

```
typedef struct {  
    double QMaster; /* Quota Asse Master in [mm] o [gradi] */  
    double QSlave; /* Quota Asse Slave in [mm] o [gradi] */  
    double VSlave; /* Velocità Asse Slave in [mm/min] o [gradi/min] */  
} SYNCH_POINT;  
#define MAX_SYNCH_POINT 4 /* Numero max. di punti di Sincronizzazione */
```

È possibile programmare tali "punti di sincronizzazione" sia per movimenti "in Avanti", sia per movimenti "Indietro" dell'asse Master, differenziandoli tra loro, quindi con la possibilità di effettuare "attacchi" e "stacchi" dell'asse Slave su di una quota prefissata del Master.

Attualmente il numero max. di "punti di sincronizzazione" è limitato a quattro (vedi definizione MAX\_SYNCH\_POINT), sia per movimenti "in Avanti", sia per movimenti "Indietro".

#### 3.12.1 Predisposizioni

Per predisporre il funzionamento sincronizzato, l'asse Slave deve risultare "Gestito e Movimentato da PLC" (richiesta IAXPLCnn=1 e stato OAXPLCnn=1); non ci sono limitazioni per l'asse Master che dovrà però essere agganciato in ciclo di spazio.





### 3.12.2 Sinonimi utilizzati per la funzione SINCRONIZZAZIONE ASSI

Per la gestione del funzionamento del CNC in modo SINCRONIZZAZIONE ASSI tramite il PLC, sono presenti vari segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati.

Dove compare il carattere "#", occorre sostituire ad esso il numero d'ordine dell'asse al quale il segnale è riferito. Tale numero, secondo il tipo di CNC, sarà compreso tra 0 e 2 (1801, 1802), tra 0 e 5 (2801) o tra 0 e 15 (4801, 4802).

<b>OAXSLV#</b>	<b>CNC → PLC</b>	<b>Predisposizione del sincronismo sull'asse slave attiva.</b> Posto a 1 se le funzioni <b>AxsFnzAxmOn()</b> e <b>AxSynch()</b> ritornano un codice di successo. <i>Vedere più oltre i particolari del funzionamento.</i>
<b>OIDSLV#</b> (short)	<b>CNC → PLC</b>	<b>Indice del movimento attuale</b> Se l'asse "Slave" si sta muovendo (OBMT#=1), rappresenta l'indice, a partire da 1 (o -1), della struttura dei punti di sincronizzazione (SYNCH_POINT), che identifica quale movimento l'asse Slave sta effettuando in funzione di quale posizione dell'asse Master. Un codice positivo identifica la struttura dei movimenti "in Avanti"; un codice negativo identifica la struttura dei movimenti "Indietro"; un codice uguale a zero significa che dal momento della sua attivazione, non è ancora stata raggiunta una quota di sincronismo.
<b>OSQSLV#</b> (short)	<b>CNC → PLC</b>	<b>Contatore di "perdita di sequenza".</b> Viene incrementato ogni volta che l'asse Slave non segue gli spostamenti dell'asse Master. <i>Vedere più oltre i particolari del funzionamento.</i>

### 3.12.3 Funzioni a disposizione del PLC

#### 3.12.3.1 AxsFnzAxmOn: attiva il sincronismo dell'asse Slave

Chiamata: **esito=AxsFnzAxmOn (IndAxSlv, IndAxMst, NpntAv, &PntAv, NpntIn, &PntIn)**

Parametri di ingresso:

- IndAxSlv** = (unsigned short) Numero d'ordine dell'asse Slave. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.
- IndAxMst** = (unsigned short) Numero d'ordine dell'asse Master. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.
- NpntAv** = (unsigned short) Numero di "punti di sincronizzazione" per movimenti in Avanti dell'asse Master. Tale valore è limitato ad un massimo di 4.
- PntAv** = (SYNCH\_POINT \*) Indirizzo (puntatore) della struttura dei "punti di sincronizzazione" per i movimenti dell'asse Master in Avanti. Sono previsti un numero max. di 4 "punti di sincronizzazione".
- NpntIn** = (unsigned short) Numero di "punti di sincronizzazione" per movimenti Indietro dell'asse Master. Tale valore è limitato ad un massimo di 4.



**PntIn** = (SYNCH\_POINT \*) Indirizzo (puntatore) della struttura dei “punti di sincronizzazione” per i movimenti dell’asse Master Indietro. Sono previsti un numero max. di 4 “punti di sincronizzazione”.

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno delle routine. Può assumere uno dei valori:

- 0 Successo – richiesta accettata
- >0 Insuccesso – codici sottoindicati
- 1 Numero d’ordine asse Slave errato
- 2 Numero d’ordine asse Master errato
- 3 Numero di punti “in Avanti” errato (da 0 a 4)
- 4 Numero di punti “Indietro” errato (da 0 a 4)
- 5 Numero di punti “in Avanti” e/o “Indietro” non definito
- 6 L’asse Slave non risulta essere “Gestito da PLC” (richiesta IAXPLC#=0 e/o stato OAXPLC#=0)
- 7 Comando in corso
- 8 Quote asse Master “in Avanti” non in ordine crescente
- 9 Quote asse Master “Indietro” non in ordine crescente
- 10 Superato fine corsa neg. Quote asse Slave “in Avanti”
- 11 Superato fine corsa neg. Quote asse Slave “Indietro”
- 12 Superato fine corsa pos. Quote asse Slave “in Avanti”
- 13 Superato fine corsa pos. Quote asse Slave “Indietro”

### 3.12.3.2 AxsFnzAxmOff: disattiva il sincronismo dell'asse Slave

Chiamata: **esito=AxsFnzAxmOff (IndAxSlv)**

Parametro di ingresso:

**IndAxSlv** = (unsigned short) Numero d’ordine dell’asse Slave. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno delle routine. Può assumere uno dei valori:

- 0 Successo – richiesta accettata
- >0 Insuccesso
- 1 Numero d’ordine asse Slave errato
- 2 Comando in corso

### 3.12.3.3 AxSynch: comanda un movimento sincronizzato dell'asse Slave

Chiamata: **esito=AxSynch (IndAxSlv, IndAxMst, QMst, QSlv, VSlv, Trig)**

Parametri di ingresso:

**IndAxSlv** = (unsigned short) Numero d’ordine dell’asse Slave. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**IndAxMst** = (unsigned short) Numero d’ordine dell’asse Master. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**QMst** = (double) Quota di sincronismo dell’asse Master in [mm] o [gradi].

**QSlv** = (double) Quota di movimento dell’asse Slave in [mm] o [gradi].

**VSlv** = (double) Velocità di movimento dell’asse Slave in [mm/1'] o [gradi/1'].

**Trig** = (short) Tipo di Trigger da usare per il movimento; può essere:



- 0 Sincronismo attivato sia per movimenti in Avanti sia per movimenti Indietro dell'asse Master
- 1 Sincronismo attivato solo per movimenti in Avanti del Master
- 2 Sincronismo attivato solo per movimenti Indietro del Master

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno delle routine. Può assumere uno dei valori:

- 0 Successo – richiesta accettata
- >0 Insuccesso – codici come per la funzione **AxsFnzAxmOn()**

**Nota:**

Effettuato il sincronismo, la modalità viene disattivata (funzione One shot)

### 3.12.4 Note di funzionamento

Con la programmazione delle funzioni **AxsFnzAxmOn()** e **AxSynch()**, l'asse **Slave** risponde con **OAXSLV#=1** solamente se le funzioni ritornano un codice = 0 (di successo) e contemporaneamente sono verificate le seguenti condizioni:

- a) L'asse **SLAVE** deve essere gestito da PLC
- b) Deve avere il SET fatto
- c) Deve essere agganciato in punto a punto, NON in velocità
- d) L'asse **MASTER** deve avere a sua volta il SET fatto
- e) Deve essere agganciato in Contouring oppure Punto a Punto, NON in velocità.

Se le suddette condizioni non sono verificate, le richieste effettuate vengono “respinte”, non attivando la predisposizione del sincronismo sull'asse Slave (**OAXSLV#=0**); in queste condizioni occorre un nuovo comando per attivare la sincronizzazione.

Quando le funzioni **AxsFnzAxmOn()** e **AxSynch()** sono accettate con successo e viene settato lo stato **OAXSLV#=1**, il “trigger” di inizio moto dell'asse Slave viene attivato solamente al primo movimento dell'asse Master.

Il Software analizza la possibilità di muovere l'asse Slave solamente quando quest'ultimo è arrivato ed è fermo (**OBNA#=0** e **OPFRM#=1**); se vengono programmati spostamenti “lunghi” dell'asse Slave, in relazione a spostamenti ravvicinati dell'asse Master, alcune sequenze di movimentazione possono essere perse (in questo caso viene incrementato il **Contatore di “perdita di sequenza” OSQSLV#**).

Il CNC a riguardo non effettua alcuna segnalazione di allarme, incrementando solamente il contatore di “perdita di sequenza”; spetta perciò all'applicazione la corretta programmazione della funzionalità e l'eventuale verifica sulla “perdita di sequenza”.

L'asse Slave, a prescindere dalla sua predisposizione di funzionamento tramite la suddetta **AxsFnzAxmOn()**, può comunque essere mosso da PLC con tutte le funzioni descritte per “l'Asse Gestito e Movimentato da PLC”; è possibile altresì annullare lo spostamento dell'asse Slave con **AxPlcAn()** e gestire gli override (**OVRFRE#**).

### 3.12.5 Esempio di sincronizzazione

Si vuole muovere l'asse di profondità Z come asse Slave in funzione delle posizioni raggiunte dall'asse X visto da Z come Master.

Per i movimenti in Avanti di X, si vuole “staccare” l'asse di profondità Z alla quota Z80 al raggiungimento di X50 e X150 ed “attaccare” alla quota Z30 al raggiungimento di X100 e X200; al contrario per i movimenti Indietro di X, si vuole “staccare” l'asse di profondità Z alla



quota Z80 al raggiungimento di X200 e X100 ed “attaccare” alla quota Z30 al raggiungimento di X150 e X50.

Lo “stacco” deve avvenire a velocità rapida (ad esempio 10000 mm/min), “l’attacco” deve avvenire a 1000 mm/min.

1) Si definiscono le due strutture che descrivono i 4 “punti di sincronizzazione” per in movimenti “in Avanti” ed “Indietro” dell’asse X:

```
#define NUM_PNT 4
/* Numero totale dei punti di sincronizzazione */

SYNCH_POINT   PntAv[NUM_PNT] = {
  /* QuotaAsseMaster   QuotaAsseSlave   VelocitàAsseSlave */
  /* [mm] o [gradi]    [mm] o [gradi]   [mm/min] o [gradi/min] */
  50, 80, 10000,
  100, 30, 1000,
  150, 80, 10000,
  200, 30, 1000 };

SYNCH_POINT   PntIn[NUM_PNT] = {
  50, 30, 1000,
  100, 80, 10000,
  150, 30, 1000,
  200, 80, 10000 };
```

**N.B.:**

*Le “Quote assi Master e Slave” sono “assolute” cioè relative allo zero macchina.*

*Le “Quote Asse Master” debbono essere programmate in ordine crescente, pena il ritorno di errore dalla funzione AxsFnzAxmOn().*

2) Vengono programmate le funzioni per attivare/disattivare la funzionalità

```
if(AUXSYNC && !OAXSLV2)
{
  ret= AxsFnzAxmOn(2, 0, NUM_PNT, PntAv , NUM_PNT, PntIn );
}
else if(!AUXSYNC && OAXSLV2)
{
  ret= AxsFnzAxmOff(2);
}
```

Nella funzione abbiamo supposto che l’asse X Master e Z Slave abbiano rispettivamente numero d’ordine (in AXS.TAR) X=0 e Z=2.

La funzione ritorna in ret zero in caso di successo, un valore maggiore di zero in caso di errore.

3) è anche possibile modificare “dinamicamente” una o più quote e/ velocità

```
if(NEWSYNC && OAXSLV2 && !ANEWSYNC)
{
  PntAv[0].QMaster = 45;
  PntAv[0].QSlave = 90;
  PntAv[1].QMaster = 110;
  PntAv[1].QSlave = 20;
  PntIn[1].QMaster = 110;
  PntIn[1].QSlave = 90;
  PntIn[2].QMaster = 150;
  PntIn[2].QSlave = 20;
  ret= AxsFnzAxmOn(2, 0, NUM_PNT, PntAv , NUM_PNT, PntIn );
  ANEWSYNC= 1 ;
}
```



### 3.13 MODIFICA SOGLIE DI POSIZIONAMENTO E DI FUGA DEGLI ASSI

La gestione della modifica delle soglie di posizionamento e di fuga degli assi è realizzata tramite apposite istruzioni di Part-Program scritte in linguaggio LIP e con funzioni di libreria PLC.

#### 3.13.1 Gestione da Part-Program

Con queste funzione è possibile modificare temporaneamente i valori delle soglie di POSIZIONAMENTO e di FUGA **attuali**, relative cioè al Part-Program in corso. I parametri modificati "attuali" vengono ripristinati al loro valore di taratura nei seguenti casi:

- RESET
- Passaggio dal modo [MAN] al modo [AUTO]
- Programmazione %
- Sovrascrittura da PLC tramite la funzione SetAxThresold()

##### 3.13.1.1 Modifica valori ATTUALI delle soglie di POSIZIONAMENTO e di FUGA.

```
<SGL:Asse;[SGL];[SFS];[FFS]>  
<SGL:Asse0;[SGL0];[SFS0];[FFS0];Asse1;[SGL1];[SFS1];[FFS1];...>
```

<b>Asse</b>	Nome o numero d'ordine dell'asse.
<b>SGL</b> (double)	Soglia di Posizionamento asse fermo/arrivato [mm] [gradi]
<b>SFS</b> (double)	Soglia Fuga di spazio asse in Moto [mm] [gradi]
<b>FFS</b> (double)	Soglia Fuga di spazio asse Fermo [mm] [gradi]

È possibile programmare più assi contemporaneamente. SGL, SFS, FFS sono opzionali; non programmando tali campi, viene automaticamente assegnato il valore attuale, lasciando quindi inalterata la soglia corrispettiva.

La funzione internamente verifica ed eventualmente "aggiusta" automaticamente le seguenti condizioni, senza dare alcuna diagnostica:

<b>Condizione</b>	<b>Valore Assegnato</b>
SGL < 0	Valore positivo di SGL
SFS < 0	Valore positivo di SFS
FFS < 0	Valore positivo di FFS
SFS < SGL	SFS=SGL
FFS < SGL	FFS=SGL
FFS > SFS	FFS=SFS

##### 3.13.1.2 Ripristina le soglie di POSIZIONAMENTO e di FUGA.

```
<TSG:Asse>  
<TSG:Asse0; Asse1;...>
```

<b>Asse</b>	Nome o numero d'ordine dell'asse.
-------------	-----------------------------------

È possibile programmare più assi contemporaneamente.



### 3.13.2 Gestione da PLC

Le funzioni per la gestione delle soglie di posizionamento e di fuga da PLC, sono “modali”; i **parametri di taratura**, modificati tramite la funzione da PLC, vengono ripristinati solamente con ALT/EXE oppure con una nuova riprogrammazione da PLC; i **parametri attuali** possono invece essere modificati temporaneamente da Part-Program tramite l'istruzione <SGL:....>

#### 3.13.2.1 ValueAxTarThresold: Legge i valori Attuali di Taratura delle soglie di Posizionamento e di Fuga

Chiamata: **esito= ValueAxTarThresold (IndAx, &SGL, &SFS, &FFS)**

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

**SGL** = (double \*) Puntatore alla variabile dove salvare il valore della Soglia di Posizionamento asse fermo-arrivato [mm] [gradi].

**SFS** = (double \*) Puntatore alla variabile dove salvare il valore della Soglia Fuga di spazio asse in Moto [mm] [gradi].

**FFS** = (double \*) Puntatore alla variabile dove salvare il valore della Soglia Fuga di spazio asse Fermo [mm] [gradi].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |   |  |
|---|--|
| 0 | Successo – comando eseguito              |
| 1 | Insuccesso – numero d'ordine asse errato |

#### 3.13.2.2 SetAxThresold: Modifica i valori di Taratura ed Attuali delle soglie di Posizionamento e di Fuga

Chiamata: **esito= SetAxThresold(IndAx, SGL, SFS, FFS)**

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse (come definito in AXS.TAR).

**SGL** = (double) Soglia di Posizionamento asse fermo-arrivato [mm] [gradi].

**SFS** = (double) Soglia di Fuga di spazio asse in Moto [mm] [gradi].

**FFS** = (double) Soglia di Fuga di spazio asse fermo [mm] [gradi].

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |   |  |
|---|--|
| 0 | Successo – comando inviato               |
| 1 | Insuccesso – numero d'ordine asse errato |
| 2 | Insuccesso – canale comandi occupato     |

**Note:**

Assegnando un valore uguale a zero ai parametri di input SGL, SFS, FFS, il campo relativo viene lasciato inalterato, viene cioè automaticamente preso il valore attuale di taratura.

La funzione verifica, ed eventualmente “aggiusta” automaticamente le seguenti condizioni, senza dare alcuna diagnostica:



---

---

Condizione	Valore Assegnato
SGL < 0	Valore positivo di SGL
SFS < 0	Valore positivo di SFS
FFS < 0	Valore positivo di FFS
SFS < SGL	SFS=SGL
FFS < SGL	FFS=SGL
FFS > SFS	FFS=SFS

### 3.14 MODIFICA DEI FINE CORSA ASSI

I valori dei **Fine Corsa attuali** (di seguito chiamati "fcmin", "fcmax" ed assegnati rispettivamente alle variabili V5073/na e V5075/na), cioè quelli applicati durante l'esecuzione di un Part-Program, tengono conto dei valori programmati da PLC, dei valori assegnati in AXS.TAR, della restrizione di campo programmata tramite G25/G26. Quest'ultima non viene mai applicata per assi indipendenti da taratura e per assi controllati da PLC.

Il criterio per assegnare "fcmin" e "fcmax" segue la logica:

- In caso siano stati programmati, vengono usati i valori dei FINE CORSA impostati da PLC; questi difatti escludono tutti gli altri, cioè quelli di taratura eventualmente impostati in AXS.TAR e l'eventuale limitazione dei FINE CORSA programmata tramite G25/G26.
- Vengono altrimenti usati i valori minimi tra i FINE CORSA di taratura eventualmente impostati in AXS.TAR e l'eventuale limitazione dei FINE CORSA programmata tramite G25/G26.
- Se nessun fine corsa è programmato da PLC, nessun fine corsa è impostato in AXS.TAR, nessuna limitazione è programmata tramite G25/G26, alla variabile "fcmax" viene assegnato il valore +3.40e+38 ed alla variabile "fcmin" viene assegnato il valore -3.40e+38

#### 3.14.1 Funzioni a disposizione del PLC

Tutte le funzioni per la gestione dei Fine Corsa da PLC, sono "modali"; i parametri modificati vengono ripristinati solamente con ALT/EXE oppure con una nuova riprogrammazione da PLC. **I valori dei Fine Corsa impostati da PLC escludono tutte le altre predisposizioni attive.**

##### 3.14.1.1 ValueAxTarTravelLimit: Legge il valore dei FINE CORSA di Taratura

Chiamata: **esito= ValueAxTarTravelLimit (IndAx, &Tlpos, &TLneg)**

Parametro di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**TLpos** = (double \*) Indirizzo dove scrivere il valore FINE CORSA POSITIVO di AXS.TAR [mm] o [gradi]

**TLneg** = (double \*) Indirizzo dove scrivere il valore FINE CORSA NEGATIVO di AXS.TAR [mm] o [gradi]

Parametro di uscita:



---

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |   |
|----------|---|
| <b>0</b> | Successo – comando eseguito                     |
| <b>1</b> | Insuccesso – numero d'ordine asse errato        |
| <b>2</b> | Insuccesso – Fine Corsa non definiti in AXS.TAR |

### 3.14.1.2 ValueAxTravelLimit: Legge il valore dei FINE CORSA (Travel Limit) attuali

Chiamata: **esito= ValueAxTravelLimit (IndAx, &Tlpos, &TLneg)**

Parametro di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**TLpos** = (double \*) Indirizzo dove scrivere il valore FINE CORSA POSITIVO attuale "fcmax" [mm] o [gradi]

**TLneg** = (double \*) Indirizzo dove scrivere il valore FINE CORSA NEGATIVO attuale "fcmin" [mm] o [gradi]

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |  |
|----------|--|
| <b>0</b> | Successo – comando eseguito              |
| <b>1</b> | Insuccesso – numero d'ordine asse errato |

### 3.14.1.3 ExcludeAxTravelLimit: Esclude i FINE CORSA (Travel Limit)

Chiamata: **esito= ExcludeAxTravelLimit (IndAx)**

Parametro di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |  |
|----------|--|
| <b>0</b> | Successo – comando inviato               |
| <b>1</b> | Insuccesso – numero d'ordine asse errato |
| <b>2</b> | Canale occupato                          |

### 3.14.1.4 SetAxTravelLimitOn: Imposta i FINE CORSA (Travel Limit) da PLC

Chiamata: **esito= SetAxTravelLimitOn (IndAx, TLpos, TLneg)**

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**TLpos** = (double) Valore FINE CORSA POSITIVO [mm] o [gradi]

**TLneg** = (double) Valore FINE CORSA NEGATIVO [mm] o [gradi]

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

---





- 0 Successo – comando inviato
- 1 Insuccesso – numero d'ordine asse errato
- 2 Canale occupato

N.B.:

TLpos e TLneg esprimono il campo di lavoro; il valore minore tra i due viene sempre associato al Fine Corsa Negativo; viceversa il valore maggiore tra i due viene sempre associato al Fine Corsa Positivo.

### 3.14.1.5 SetAxTravelLimitOff: Ripristina i FINE CORSA (Travel Limit) di TARATURA

Chiamata: **esito= SetAxTravelLimitOff (IndAx)**

Parametro di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 0 Successo – comando inviato
- 1 Insuccesso – numero d'ordine asse errato
- 2 Canale occupato

La funzione disabilita il controllo dei FINE CORSA da PLC, disabilita l'eventuale esclusione e ripristina i FINE CORSA di TARATURA.

## 3.15 EVENTI ESTERNI E PROBE ON/OFF

I CNC ECS serie WIN possono gestire eventi esterni al CNC stesso come, ad esempio, un tastatore ON/OFF (PROBE) necessario per l'esecuzione di cicli di misura sulla Macchina Utensile. A questo scopo sono previste apposite istruzioni (<TRU...> <FLS...> descritte nel manuale di Programmazione Avanzata) che permettono l'interruzione del movimento in corso, il salto di programma ad una Nxxx definita e l'eventuale memorizzazione della quota attuale degli assi. Quanto sopra viene ottenuto valutando lo stato o la commutazione di particolari segnali previsti a questo scopo.

### 3.15.1 Sinonimi utilizzati dal PLC per eventi esterni

Per la gestione degli eventi esterni sono presenti appositi input che sono stati associati ai sinonimi qui di seguito elencati.

<b>ITFnn</b>	<b>PLC → CNC</b>	16 ingressi TRU/FALSE (vero/falso) dell'evento esterno da tenere sotto controllo. <b>nn</b> vale 00 ... 15. Lo stato di questi segnali viene testato tramite le istruzioni <TRU <sub>nn</sub> :...> <FLS <sub>nn</sub> :...>.
<b>IOSTPB</b>	<b>CNC → PLC</b> oppure <b>PLC → CNC</b>	Ingresso TOUCH-PROBE delle schede assi analogiche Touch-Probe (040F361 e successive) o degli Azionamenti Digitali con collegamento Sercos o Mechatrolink. Lo stato di questi segnali viene testato tramite le istruzioni <TRU<STP>:...> <TRU<CTP>:...> e le analoghe <FLS<...>:...>



### 3.15.2 Cicli di misura

Sui CNC con release di software precedenti la V2.00, erano presenti solo i segnali ITFnn e la precisione delle quote di commutazione era influenzata da numerosi fattori:

- Velocità di esecuzione del ciclo
- Tempo di acquisizione del segnale in ingresso dal PROBE
- Tempo di risposta del PLC al segnale di commutazione ITFnn
- Tempo di ritardo della logica PLC rispetto alla base dei tempi del CNC
- Quote errori degli assi

Un primo ritardo è intrinseco nel tempo di logica del PLC e le quote di commutazione (memorizzate nelle variabili V6006/na) indicano semplicemente la quota “calcolata” e non le quote “trasduttore” al momento della commutazione. Le quote suddette erano inoltre calcolate solamente per gli assi CNC (in contouring) e non per tutti gli assi configurati.

Sfruttando l'hardware proprio delle schede assi analogiche Touch-Probe (040F361) oppure degli azionamenti Indramat in caso di connessione digitale Sercos, è ora possibile ovviare ai suddetti inconvenienti rendendo fattibile effettuare cicli di misura assolutamente ripetitivi e più precisi a pari velocità di toccata.

Per utilizzare la nuova funzionalità i cicli di misura debbono usare le nuove sintassi <TRU<STP>:....>, <TRU<CTP>:....>, <FLS<STP>:....>, <FLS<CTP>:....>.

Le “chiavi” <STP> e <CTP> predispongono il CNC all'ascolto dello “Stato del contatto Touch-Probe (STP)” oppure sulla “Commutazione del contatto Touch-Probe (CTP)” proprio della scheda 040F361 (l'input denominato “Micro di Evento” o MCREVNT) oppure l'ingresso PROBE1 dell'azionamento Indramat in caso di connessione digitale Sercos.

Tale input è visibile dal PLC attraverso il sinonimo IOSTPB, aggiornato ciclicamente dal CNC solamente in caso di connessione analogica; in caso di connessione Sercos il segnale IOSTPB viene aggiornato solamente durante l'esecuzione del ciclo; in queste condizioni il segnale IOSTPB è quindi un Output del CNC.

Le istruzioni TRU<STP> e TRU<CTP> hanno uguale logica, ma funzionamenti differenti in caso siano o non siano presenti le schede assi analogiche Touch-Probe (040F361) e connessioni Sercos.

Per gli azionamenti Yaskawa con collegamento Mechatrolink valgono le stesse considerazioni fatte per gli assi Sercos considerando che non sono però accettate le istruzioni FLS e che il contatto Touch-Probe deve essere collegato sugli ingressi EXT2 (CN1-45).

#### 3.15.2.1 Presenza di assi Sercos, connessi ad azionamenti Indramat

In questo caso il micro del probe deve necessariamente essere connesso in parallelo su gli ingressi PROBE1 di tutti gli azionamenti SERCOS presenti per avere un campionamento (“latch”) simultaneo su tutti gli assi connessi ad azionamenti INDRAMAT.

Nella eventualità di voler escludere alcuni Drive dal cablaggio, e quindi da un latch hardware delle quote ad opera del drive INDRAMAT, occorre agire sul campo "AX\_TP\_MASK" di SWCNF.TAR con la seguente sintassi:

```
AX_TP_MASK=0xhexvalue
```



“hexvalue” rappresenta il valore esadecimale della “maschera” degli assi NON connessi in parallelo all’input PROBE1 per il latch hardware delle quote. Il valore 0x0000 significa che tutti gli assi SERCOS sono connessi in parallelo all’input PROBE1 (valore di default).

Un valore diverso da zero, significa che gli assi indicati con i bit a 1 vengono esclusi dal campionamento hardware delle quote ad opera del drive INDRAMAT; in quest’ultimo caso le loro quote vengono campionate dal CNC con la risoluzione del tempo di ciclo; ad esempio il valore AX\_TP\_MASK=0x0002 esclude l’asse di ordine 1 dal campionamento hardware delle quote.

In presenza di assi SERCOS, connessi ad azionamenti INDRAMAT, le due istruzioni <STP> e <CTP> si comportano in maniera molto simile, dovendo attivare il ciclo di probing all’interno degli azionamenti anche solamente per testare lo stato del contatto PROBE1.

Il CNC sincronizza la movimentazione assi con l’attivazione del ciclo; nel blocco precedente l’attivazione dell’istruzione TRU/FLS viene cioè provocata una fermata di fine blocco, gli assi riprendono il movimento solamente quando, nel blocco successivo, il ciclo di probing risulta realmente comandato ed attivato all’interno degli azionamenti.

Le istruzioni <STP> e <CTP> predispongono gli azionamenti SERCOS per un nuovo latch hardware delle quote degli assi ad essi collegati. Le istruzioni <TRU<STP>:...> e <TRU<CTP>:...> risultano vere con il contatto PROBE CHIUSO (IOSTPB=1) (commutazione 0→1 del contatto; fronte di salita); le istruzioni <FLS:<STP>:...> e <FLS:<CTP>:...> risultano vere con il contatto PROBE APERTO (IOSTPB=0) (commutazione 1→0 del contatto; fronte di discesa);

L’istruzione <STP> non provoca la memorizzazione delle quote nelle variabili del CNC e viene utilizzata solamente per testare lo stato attuale del contatto del probe.

A fine istruzione <CTP>, il CNC memorizza le quote di tutti gli assi configurati (CNC e FREE):

*V6006/na quote misurate di tutti gli assi configurati nel CNC, espresse come quote “ASSOLUTE” (relative all’origine macchina).*

*V6007/na quote misurate di tutti gli assi configurati nel CNC, espresse come quote “VISUALIZZATE” (relative al part-program in corso).*

Le quote degli assi SERCOS, essendo memorizzate dall’hardware proprio dell’azionamento INDRAMAT, risultano prive di “ritardi” e quindi precise e ripetitive.

Le stesse considerazioni valgono per gli assi Mechatrolink

Le quote degli assi non connessi agli azionamenti INDRAMAT o Mechatrolink e le quote degli assi “esclusi” dal ciclo di probing attraverso il campo “AX\_TP\_MASK”, vengono campionate con la precisione propria del tempo di campionamento (TCYC= Servo Loop Cycle Time), comunque molto più precise del precedente sistema la cui precisione, in termini di tempo, era dovuta al ciclo di PLC (4-8 volte superiore).

### 3.15.2.2 Presenza di schede assi analogiche Touch-Probe (senza assi SERCOS)

In questo caso il micro del probe deve necessariamente essere connesso sull’input MCREVNT della prima scheda assi.

In caso di BUS esteso (back panel di connessione sulle schede assi) non è necessario altro cablaggio; le eventuali altre schede assi analogiche Touch-Probe vengono difatti automaticamente configurate come “Slave” dal software del CNC rispetto alla prima scheda “Master” ed il segnale MCREVNT viene propagato attraverso il bus esteso dalla scheda Master verso le schede Slave.

Se non è possibile la connessione attraverso il BUS esteso, occorre connettere il contatto in parallelo su tutte le schede Touch-Probe ed informare il sistema di questa predisposizione attraverso la chiave BUS\_SLAVE=1 (default è BUS\_SLAVE=0) in SWCNF.TAR (è possibile modificare SWCNF.TAR ed inserire il record con la chiave BUS\_SLAVE=1 in una qualunque posizione nel file).

L’istruzione <STP> non provoca la memorizzazione delle quote, ma viene utilizzata solamente per testare lo stato attuale del contatto del probe (segnale IOSTPB); l’istruzione <TRU<STP>:...> risulta vera con il contatto PROBE CHIUSO (IOSTPB=1), l’istruzione <FLS:<STP>:...> risulta vera con il contatto PROBE APERTO (IOSTPB=0).



---

L'istruzione <CTP> predispone le schede Touch-Probe per un nuovo latch hardware delle quote degli assi collegati a quest'ultime; in queste condizioni l'ascolto di avvenuta commutazione viene effettuata dal CNC ogni "tau/2"; ad avvenuta commutazione, il CNC memorizza le quote di tutti gli assi configurati (CNC e FREE) nelle variabili:

*V6006/na quote misurate di tutti gli assi configurati nel CNC, espresse come quote "ASSOLUTE" (relative all'origine macchina).*

*V6007/na quote misurate di tutti gli assi configurati nel CNC, espresse come quote "VISUALIZZATE" (relative al part-program in corso).*

Le quote degli assi connessi alle schede analogiche Touch-Probe, essendo memorizzate dall'hardware proprio della scheda, risultano prive di "ritardi" e quindi precise e ripetitive.

Le quote degli assi non connessi alle schede analogiche Touch-Probe, vengono campionate con la precisione propria del tempo di campionamento (TCYC= Servo Loop Cycle Time), comunque molto più precise del precedente sistema la cui precisione, in termini di tempo, era dovuta al ciclo di PLC (4-8 volte superiore).

### 3.15.2.3 Assenza di schede assi analogiche Touch-Probe e di assi Sercos o Mechatrolink

In questo caso il funzionamento delle istruzioni <TRU<STP>:...> e <TRU<CTP>:...> è analogo ai precedenti cicli di misura, con la differenza che il CNC è ora in ascolto sul segnale IOSTPB (in queste condizioni il segnale IOSTPB è quindi un Input del CNC aggiornato dal PLC), in modo analogo ai precedenti segnali ITF00...ITF15.

L'istruzione <STP> non provoca la memorizzazione delle quote, ma viene utilizzata solamente per testare lo stato attuale del contatto del probe (segnale IOSTPB gestito dal PLC); l'istruzione <TRU<STP>:...> risulta vera con il contatto PROBE CHIUSO (IOSTPB=1), l'istruzione <FLS:<STP>:...> risulta vera con il contatto PROBE APERTO (IOSTPB=0).

L'istruzione <CTP> ascolta la commutazione del segnale IOSTPB proveniente dal PLC; in particolare, l'istruzione <TRU<CTP>:...> sente la commutazione 0→1 del suddetto segnale, mentre l'istruzione <FLS<CTP>:...> sente la commutazione 1→0 dello stesso. Ad avvenuta commutazione, il CNC memorizza le quote di tutti gli assi configurati (CNC e FREE) nelle variabili:

*V6006/na quote misurate di tutti gli assi configurati nel CNC, espresse come quote "ASSOLUTE" (relative all'origine macchina).*

*V6007/na quote misurate di tutti gli assi configurati nel CNC, espresse come quote "VISUALIZZATE" (relative al part-program in corso).*

Non essendo presente alcun asse Sercos o Mechatrolink né alcuna scheda analogica Fast Input/Touch-Probe, la commutazione del contatto deve essere comunicata dal PLC in modo analogo a quanto avveniva per i segnali ITF00..ITF15; il campionamento delle quote avviene con un ritardo "intrinseco" della logica PLC e del tempo di campionamento; la precisione ottenuta sulle quote riflette quindi tali ritardi.

### 3.15.2.4 Note di applicazione

#### **Caso di Assi Digitali SERCOS o Mechatrolink:**

- Se il sistema dispone di assi SERCOS è necessario connettere in parallelo il micro del probe su tutti gli ingressi PROBE1 di tutti gli azionamenti INDRAMAT presenti nel sistema; in alternativa configurare il campo "AX\_TP\_MASK" di SWCNF.TAR, escludendo gli assi desiderati dal ciclo di probing.
- Se il sistema dispone di assi Mechatrolink è necessario connettere in parallelo il micro del probe su tutti gli ingressi EXT2 (CN1-45) di tutti gli azionamenti YASKAWA presenti nel



sistema; in alternativa configurare il campo "AX\_TP\_MASK" di SWCNF.TAR, escludendo gli assi desiderati dal ciclo di probing.

- Il sinonimo IOSTPB è un segnale di Output, letto dal PLC ed aggiornato dal CNC solamente durante l'esecuzione del ciclo, non ciclicamente come accade in presenza di assi analogici

#### **Caso di Assi Analogici, senza Assi Digitali SERCOS:**

- Se il sistema dispone di schede analogiche Touch-Probe è sufficiente cablare il segnale del Probe sull'ingresso MCREVNT della prima scheda (indirizzo fisico "più basso")
- Se il sistema non dispone di "Bus Esteso" (back panel di espansione schede assi e schede I/O) occorre connettere il contatto in parallelo su tutte le schede Touch-Probe ed informare il sistema di questa predisposizione attraverso la chiave BUS\_SLAVE=1 (default è BUS\_SLAVE=0) in SWCNF.TAR.
- Se il sistema dispone di schede analogiche Touch-Probe il sinonimo IOSTPB è un segnale di Output aggiornato ciclicamente dal CNC e letto dal PLC.
- Se il sistema non dispone di schede analogiche Touch-Probe il sinonimo IOSTPB è un segnale di Input aggiornato dal PLC e letto dal CNC.
- Se il sistema non dispone di schede analogiche Touch-Probe, per aggiornare un "vecchio" PLC con i nuovi cicli di misura, è necessario solamente aggiungere, ad esempio, l'equazione IOSTPB= ITF00, oppure utilizzare direttamente il nuovo sinonimo.
- Al completamento delle istruzioni <CTP>, in ogni caso vengono campionate le quote assolute (V6006/na) e le quote relative al part-program (V6007/na "quote visualizzate") di tutti gli assi configurati, a differenza del precedente sistema nel quale venivano campionate solamente le quote degli assi CNC.

I cicli esistenti possono continuare a funzionare con il precedente sistema, oppure essere aggiornati al nuovo funzionamento, semplicemente cambiando la sintassi <TRU<0>:...> in <TRU<STP>:...> oppure <TRU<CTP>:...> a seconda del caso, e rispettare il cablaggio come specificato nei primi due punti.

### **3.16 ASSI VIRTUALI {F}**

Gli assi virtuali sono 3 assi che individuano un sistema cartesiano ortogonale comunque disposto nello spazio, tale sistema fisicamente non esiste ed è realizzato applicando una matrice dinamica a 3 assi macchina che sono ortogonali fra loro.

Questa prestazione si può attivare su frese dotate di testa birotativa o monorotativa: la matrice è calcolata partendo dalle quote degli assi rotanti che determinano l'inclinazione dell'utensile.

Per definire gli assi rotanti e gli assi fisici da associare alla matrice, occorre programmare l'istruzione <ATW:...>, con gli stessi parametri e sintassi della programmazione TWIST; del TWIST si mantiene anche la modalità con la quale si effettua il SET degli assi interessati.

Gli assi virtuali possono essere gestiti da programma tramite istruzioni LIP o da PLC tramite alcune funzioni.

Le istruzioni LIP sono:

<VAX:ON>	attiva il sistema virtuale;
<VAX:OFF>	disattiva il sistema virtuale;

#### **3.16.1 Funzioni per Attivare/Disattivare gli Assi Virtuali da PLC**

##### **IsVaxOn: Testa se sono attivi gli Assi Virtuali**

Chiamata: **esito = IsVaxOn ()**

##### **VaxPlcOn: Attiva gli Assi Virtuali**



---

Chiamata: **esito = VaxPlcOn ()**

**VaxPlcOff: Disattiva gli Assi Virtuali**

Chiamata: **esito = VaxPlcOff ()**

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- 1 Gli assi virtuali sono attivi
- 2 Gli assi virtuali sono disattivi
- 3 Il PLC deve ripetere la funzione, in attesa che il comando impartito sia terminato
- 2 Canale occupato: il CNC deve terminare il comando precedentemente emesso
- 3 il CNC ha dei blocchi in esecuzione
- 4 non è stata programmata l'istruzione <ATW:...>
- 5 c'è una matrice dinamica attiva
- 6 trasmit attivo
- 7 opzione non abilitata
- 8 esaurito spazio per matrici
- 10 matrice su assi associati a seconda o terza funzione ausiliaria
- 11 matrice su assi mandrino
- 12 asse trascinate ha origine su posizione non consentita
- 13 un asse è non disponibile, oppure pendolante, oppure FREE, oppure di proprietà del PLC (Libera Servo), oppure gestito dal PLC.

### 3.17 RIPOSIZIONAMENTO CONTEMPORANEO DI TUTTI GLI ASSI

Dalla release software V3.01 è stata introdotta la possibilità di riposizionare contemporaneamente tutti gli assi del CNC. Questa funzionalità è attiva solo se viene abilitata dal PLC tramite la funzione di libreria **SetAllAxs()**. Sempre da PLC, tramite la funzione di libreria **FeedAllAxs()**, è possibile stabilire con quale velocità eseguire il riposizionamento contemporaneo.

#### 3.17.1 Funzioni per il riposizionamento contemporaneo di tutti gli assi

##### 3.17.1.1 SetAllAxs: Attiva/disattiva il riposizionamento contemporaneo

Chiamata: **SetAllAxs (allaxs)**

Parametro di ingresso:

**allaxs** = (short) Flag di input che indica se attivare o disattivare la funzionalità.

**allaxs != 0** indica che occorre attivare la funzionalità.

**allaxs = 0** indica che occorre disattivare la funzionalità.

Quando la funzionalità è attiva, nella tendina del riposizionamento compare anche la voce "TUTTI GLI ASSI" che permette di comandare il riposizionamento contemporaneo di tutti gli assi del CNC. Naturalmente, rimane la possibilità di riposizionare i singoli assi uno per volta.

Quando la funzionalità non è attiva, tale voce non è presente e quindi gli assi possono essere riposizionati solo uno per volta.

##### 3.17.1.2 FeedAllAxs: Definizione della Feed per il riposizionamento contemporaneo

Chiamata: **FeedAllAxs (feed)**

Parametro di ingresso:



**feed** = (double) Velocità di riposizionamento desiderata (in mm/1') .

Se questa funzione non viene chiamata o la **feed** è =0, come velocità vale quella attiva nel punto di riposizionamento (l'ultima presente nel part-program).

Si ricorda che il segnale **ORIPOK** è vero quando, in modo Riposizionamento, tutti gli organi programmati sono riposizionati.

### 3.18 FUNZIONE ANTISLITTAMENTO PER PIEGATUBI {F}

Dalla release software V3.03 è stata introdotta la possibilità di gestire gli assi di una piegatubi caratterizzata dal fatto che il trasduttore dell'asse dedicato all'avanzamento del tubo non è collegato al motore o alla meccanica dell'asse stesso ma "legge" direttamente i movimenti del tubo da piegare; questo permette di muovere correttamente gli assi dedicati alla piegatura anche se il tubo stesso "slitta" sui rulli che ne comandano l'avanzamento.

Per gestire l'asse (supponiamo X) deputato all'avanzamento del tubo si deve definire tale asse come appartenente ad una nuova tipologia denominata "asse T" ed identificata in AXS.TAR con un campo TYP specifico ("TAX").

La particolare interpolazione che è applicata all'asse X coinvolgerà anche gli altri assi programmati nello stesso blocco.

Descrizione del "meccanismo di controllo" implementato:

L'interpolatore allorché verifica che la quota reale di X (nel caso particolare l'encoder che indica il reale avanzamento del tubo) differisce da quella teorica, riduce la feed degli altri assi interpolati con X tramite un meccanismo simile all'override Feed. Al contempo continua a comandare l'asse X sino a che il movimento non raggiunge la quota prevista. Potendo tale condizione protrarsi per un tempo "indefinito" è ovvio che il movimento avviene con allarmi fuga disabilitati. La ripresa del movimento su X fa automaticamente ridurre l'effetto dell'override feed sugli altri assi. In caso d'arresto o mancanza del tubo la feed di X sarà automaticamente ridotta ad un valore, impostabile e indipendente da quello programmato, nel seguito definito "Feed a vuoto".

Per consentire il test di Part-Program, in assenza di tubo, sarà possibile, ponendo KV=0, disabilitare i controlli incrociati sopra descritti interpolando l'asse X con gli altri assi a loop aperto.

Considerando che l'asse X potrà montare rulli di trascinamento con differenti diametri (secondo il tipo di tubo/profilato da curvare) per garantire la Feed d'avanzamento programmata il CNC dovrà necessariamente tenere conto del reale "diametro primitivo" del rullo di trascinamento.

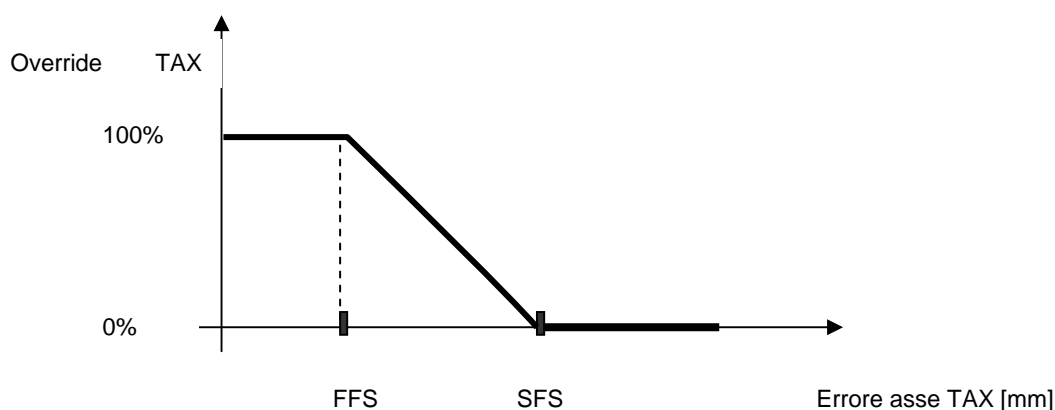
A tal fine, con questa particolare tipologia d'assi, il parametro di taratura **RAP** in **AXS.TAR** dovrà essere configurato con il numero massimo di giri del sistema di trascinamento (quindi espresso in rpm). Analogamente anche la sua accelerazione sarà espressa in giri/min sec.

#### 3.18.1 File AXS.TAR per "Antislittamento"

L'Asse **T** è trattato dal CNC come un asse lineare caratterizzato dai seguenti "particolari" dichiarati in AXS.TAR:

CAMPO	Sigla oppure unità di	Descrizione
-------	-----------------------	-------------

	misura	
<b>TYP</b>	<b>TAX</b>	Definisce l'asse come "Asse T"
<b>RAP</b>	<b>giri/min</b>	max. velocità di rotazione della puleggia di Trascinamento
<b>ACC</b>	<b>giri/min*sec</b>	max. accelerazione della puleggia di Trascinamento
<b>RSM</b>	<b>mm/n. impulsi * 8</b>	indica lo spostamento lineare in [mm] del tubo / profilo per ogni impulso del trasduttore deputato a misurare il reale avanzamento
<b>KV</b>	<b>1/sec</b>	Guadagno proporzionale del di ciclo di spazio. Se = 0: Ciclo di spazio aperto; in queste condizioni l'asse non controlla le soglie FFS e SFS e neppure lo spostamento rispetto agli altri assi interpolati imponendo un coefficiente moltiplicativo dell'Override Feed = 1
<b>KI</b>	<b>Numero intero &gt;= 0</b>	Guadagno integrativo del ciclo di spazio =0 Taratura disabilitata >0 Taratura abilitata quando l'asse NON è in Moto (OBMTx=0)
<b>FFS</b> <b>SFS</b>	<b>mm</b>	Soglie d'errore (FFS <= SFS) nel cui campo è modulato il valore di Feed Override (vedi figura)



**Note:**

- la feed di T è automaticamente limitata al valore impostato come "**Velocità a vuoto**" allorché l'errore dell'asse supera il valore di **SFS**.
- Per un asse T non è controllata l'entrata in soglia a fine movimento
- I valori dei segnali OPSGLx e OPFRMx seguono il complemento del segnale **OBMTx**
- Non sono diagnosticati gli allarmi "fuga" e "fuori soglia".





---

Questa funzionalità è attiva solo se viene abilitata dal PLC tramite la funzione di libreria **init\_T\_Axis()**.

### 3.18.2 init\_T\_Axis: inizializza i parametri per asse Piegatubi

Chiamata: **esito = init\_T\_Axis(IndAx,Diameter,maxRpm)**;

Scopo: Definisce il diametro della puleggia di trascinamento e la velocità massima di rotazione in caso di slittamento.

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse T, dichiarato TYP=TAX in AXS.TAR. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**Diameter** = (double) Diametro della puleggia di trascinamento, espresso in [mm]; attraverso il parametro "Diameter" e le tarature RAP=[giri/min] ed ACC=[giri/min\*sec] di AXS.TAR, il CNC calcola la massima velocità ed accelerazione "lineare" dell'Asse T:

$$V_{max} = RAP * \pi * Diameter \text{ [mm/min]}$$

$$A_{max} = ACC * \pi * Diameter \text{ [mm/min*sec]}$$

Con tali valori l'**Asse T** effettua l'interpolazione con gli altri assi.

**maxRpm** = (double) "**Feed a vuoto**". Velocità di rotazione della puleggia a vuoto in caso di "slittamento" del tubo. Espresso in giri/min. Tale velocità di rotazione è garantita anche in caso l'asse T sia stato programmato ma il tubo non sia ancora stato inserito nella puleggia oppure sia "inceppato", per favorirne il "recupero". Il valore è comunque limitato al valore di RAR di taratura.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- |   |   |
|---|---|
| 0 | Successo.   |
| 1 | Numero di asse errato (errore sul primo parametro). |
| 2 | C'è già un comando in corso.                        |
| 3 | L'asse non è di tipo TYP=TAX in AXS.TAR.            |
| 4 | Diameter = 0 o negativo                             |
| 5 | maxRpm = 0 o negativo                               |

La funzione dovrà essere riprogrammata, ogni qualvolta sia necessario cambiare la puleggia di trascinamento e quindi ricalcolare la massima velocità d'avanzamento dell'Asse T (del tubo). Se la funzione non è mai programmata, l'asse T non potrà essere mosso, pena l'allarme 10206 emesso dal CNC.

#### Note applicative:

E' possibile "modulare" la prontezza della retroazione dell'Asse T, rispetto agli altri assi interpolanti, attraverso le soglie FFS e SFS; più vicine saranno tra di loro tali soglie, più pronta sarà la retroazione.

Sono altrettanto importanti i valori d'accelerazione (ACC=) e di Jerk (<T1A:...>) impostati; un'accelerazione alta ed un tempo T1A basso renderanno il sistema più rigido e pronto.

L'asse T può essere reso indipendente (gestito da PLC come un normale asse).



L'override applicato agli assi contemporaneamente interpolati con un asse **T** sarà in "cascata" all'override eventualmente impostato dall'operatore in console (prodotto dei 2 valori)

### 3.18.3 get\_T\_Axis\_Override: legge l'override dell'asse T

Chiamata: **TAxOver = get\_T\_Axis\_Override();**

Scopo: Legge il valore attuale dell'override di velocità dovuto all'asse T.

Parametro di ritorno:

**TAxOver** = (double) Valore attuale dell'override assi dovuto allo slittamento dell'asse T.

## 3.19 COMPENSAZIONE DINAMICA DELL'USURA UTENSILE

In talune applicazioni, in particolare allorché si utilizzino come utensili mole caratterizzate da usura particolarmente elevata, è necessario adeguare dinamicamente la compensazione alla continua riduzione del raggio e della lunghezza dell'utensile. Il caso tipico è quello di mole lucidanti per vetro.

A tal fine i CNC ECS della serie WIN dispongono di una speciale funzionalità definita appunto "**Compensazione Dinamica dell'Usura Utensile**" descritta al Capitolo 9 del manuale "NORME DI PROGRAMMAZIONE 720P395".

Tramite due parametri, uno per il raggio e l'altro per la lunghezza, l'utente può infatti quantificare l'usura dell'utensile montato sul mandrino e quindi istruire il CNC su come compensare al volo tali variazioni.

Il comando per attivare da part-program la prestazione è:

```
<USR:ON;c_r_usr;c_l_usr>
```

dove:

"**c\_r\_usr**" è il coefficiente di usura radiale dell'utensile espresso in mm/metro e

"**c\_l\_usr**" è il coefficiente di usura assiale dell'utensile espresso in mm/metro

L'usura radiale corregge gli assi del piano di contornatura, mentre quella assiale corregge la quota dell'asse di profondità; come conseguenza, l'attivazione della funzionalità deve essere preceduta dalla definizione del piano di contornatura (G17, G18, G19 o G16xxx) e dell'asse su cui applicare la compensazione in lunghezza ( <CFF=CFx> ).

Il PLC, tramite alcune funzioni presenti in libreria PLC descritte più avanti, può modificare dinamicamente i coefficienti definiti nel part-program ed inserire un valore aggiuntivo di correzione.

### 3.19.1 Funzioni per la compensazione dinamica da PLC

#### 3.19.1.1 plcrusr: legge/scrive il coefficiente di usura radiale da PLC

Chiamata: **plcrusr (&valore, flag)**

Parametri di ingresso:

**valore** = (double \*) Indirizzo dove scrivere o leggere il valore del coefficiente di usura radiale, in mm/metro.



**flag** = (short) Flag di lettura/scrittura; può assumere i valori:

- 0** Scrittura del nuovo coefficiente
- !=0** Lettura del coefficiente attuale

### 3.19.1.2 plclusr: legge/scrive il coefficiente di usura longitudinale da PLC

Chiamata: **plclusr (&valore, flag)**

Parametri di ingresso:

**valore** = (double \*) Indirizzo dove scrivere o leggere il valore del coefficiente di usura longitudinale, in mm/metro.

**flag** = (short) Flag di lettura/scrittura; può assumere i valori:

- 0** Scrittura del nuovo coefficiente
- !=0** Lettura del coefficiente attuale

### 3.19.1.3 scarusr: legge/scrive il valore aggiuntivo di usura radiale da PLC

Chiamata: **scarusr (&valore, flag)**

Parametri di ingresso:

**valore** = (double \*) Indirizzo dove scrivere o leggere il valore aggiuntivo di usura radiale, in mm.

**flag** = (short) Flag di lettura/scrittura; può assumere i valori:

- 0** Scrittura del nuovo valore
- !=0** Lettura del valore attuale

### 3.19.1.4 scalusr: legge/scrive il valore aggiuntivo di usura longitudinale da PLC

Chiamata: **scalusr (&valore, flag)**

Parametri di ingresso:

**valore** = (double \*) Indirizzo dove scrivere o leggere il valore aggiuntivo di usura longitudinale, in mm.

**flag** = (short) Flag di lettura/scrittura; può assumere i valori:

- 0** Scrittura del nuovo valore
- !=0** Lettura del valore attuale

## ATTENZIONE!

I valori scritti con le funzioni scalusr() e scarusr() vengono applicati "a scalino" e quindi senza nessuna "gradualità"; si raccomanda quindi di introdurre variazioni "piccole", minori o uguali a 0.01 mm. Le due funzioni suddette possono essere utili per correggere le dimensioni della mola dopo un ciclo di "rattivatura" o su comando dell'operatore che giudica non corretta l'attuale compensazione.

Valori crescenti provocano un accostamento mola/pezzo, mentre valori decrescenti provocano uno scostamento.

La compensazione viene disattivata da Part-Program con l'istruzione:

**<USR:OFF>**

Al momento della disattivazione, il CNC addebita l'usura accumulata e quella aggiuntiva all'utensile attuale ed azzerà i valori memorizzati. L'addebito può essere indirizzato a raggio

---



e lunghezza utensile (per default) oppure ai relativi offset; la scelta è fatta in base al valore del bit di peso ottale 400 della variabile %V4075, che può essere settato/resettato in COSTUSER con istruzioni del tipo:

```
N160 <%V4075=(%V4075 | 'OCT0400')>  
N162 {Addebito compensazione usura agli offset raggio e lunghezza}  
N164  
N166 {<%V4075=(%V4075 & 'OCT177377')>}  
N168 {Addebito compensazione usura a raggio e lunghezza }
```

### 3.20 CORREZIONE TABELLARE ERRORE ASSI

I CNC ECS della serie WIN, permettono la correzione tabellare errori assi, descritta nel capitolo 4 del Manuale Tarature 720P385. In alcune macchine, può essere necessario avere diverse tabelle di correzione, da applicare in situazioni differenti e da selezionare tramite il PLC.

Dalla release software V4.00 è stata introdotta la possibilità di definire più configurazioni di tabelle di correzione; oltre alla configurazione base definita su COR.TAR, possono essere presenti COR1.TAR, COR2.TAR, eccetera, da attivare tramite PLC.

La gestione delle compensazioni tabellare è comandata dal PLC tramite apposite funzioni della libreria PLC.

#### 3.20.1 Funzioni per gestire le compensazioni tabellare Assi da PLC

##### 3.20.1.1 open\_CorrFile: Seleziona il set di compensazioni Assi

Chiamata: **esito = open\_CorrFile (num\_corr)**

Scopo: Invia al CNC un **comando** di selezione del set di correzioni da usare.

Parametro di ingresso:

**num\_corr** = (unsigned short) Numero del set di correzioni da usare: 0=COR.TAR;  
n=CORn.TAR.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- 0 Successo; comando inoltrato al CNC.
- 1 Correzioni errori assi da file non attivate da GEN.TAR attraverso il record "CVIT: CERR=Y"
- 2 C'è già un comando in corso.

##### 3.20.1.2 get\_CorrFileNumber: Legge il set di compensazioni Assi attivo

Chiamata: **esito = get\_CorrFileNumber ()**

Scopo: Legge il numero di file correzioni attualmente selezionato.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- 0 Attivo il set definito in COR.TAR



---

n Attivo il set definito in CORn.TAR

### 3.20.1.3 get\_CorrFileCode: Verifica l'esito del comando di selezione

Chiamata: **esito = get\_CorrFileCode()**

Scopo: Legge l'esito dell'ultima selezione effettuata con open\_CorrFile().

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

< -1	Errore di attivazione della funzione
-1	Comando in corso
0	Successo; nuovo set correzioni attivato
> 0	Errore: il significato può essere visto tramite il tasto funzione [Allarmi] del menu orizzontale [Tools] del CNC.

Per attivare uno specifico set di correzioni, il PLC deve quindi chiamare la funzione **esito=open\_CorrFile(num\_corr)** con il **num\_corr** desiderato; testando il valore di **esito**, il PLC dovrà:

- se =2 : ripetere il comando al "prossimo giro di PLC"
- se =1 : segnalare l'errore all'operatore
- se =0 : passare alla fase successiva

Il PLC deve poi controllare se tutto OK tramite la funzione **esito=get\_CorrFileCode()**; testando nuovamente il valore di **esito**, il PLC dovrà:

- se =-1: ripetere il test al "prossimo giro di PLC"
- se < -1: segnalare l'errore all'operatore
- se =0 : considerare terminato tutto OK
- se >0 segnalare il problema all'operatore

La funzione **get\_CorrFileNumber()** permette di evitare la sequenza suddetta se il set desiderato è già attivo.

## 3.21 Gestione RETRACT ASSI in EMERGENZA

Questa prestazione (presente dalla release software V4.00) permette di **gestire situazioni di emergenza** dove non è sufficiente la sola richiesta di "BLOCCO ASSI" o "BLOCCO ASSI + MANDRINO" o "RESET" o "EMERGENZA M.U.", ma occorre eseguire in contemporanea anche il posizionamento di uno o più assi in quota di sicurezza.

### 3.21.1 Funzionamento

Tramite i segnali descritti più oltre e le previste istruzioni **G380** scritte nel Part-Program, si attiva la funzionalità di retract assi; in questa situazione, quando l'operatore preme un pulsante [HOLD], [AXISH] o [RESET], il CNC porta gli assi predisposti sulla quota di sicurezza. A fine retract viene interrotta la lavorazione con emissione di RESET.

**Attenzione !** Trattandosi di EMERGENZA il CNC non garantisce di uscire salvaguardando il profilo della lavorazione in corso. I movimenti degli assi sono asincroni, cioè raggiungono la quota assegnata in tempi diversi in funzione dello spazio da percorrere.



La funzionalità è attiva solo con CNC in **ESECUZIONE** nei modi **Automatico** (MODE\_AUT) e **Manuale** (MODE\_MDI).

Se si attiva in MDI, effettuando un cambio modo (ad es. richiesto JOG) la predisposizione si annulla e non viene ripristinata ritornando in MDI. Se si attiva in AUTO effettuando cambio modo (ad es. richiesto MDI) la predisposizione si annulla e viene riattivata ritornando in AUTO solo se condizioni iniziali OK.

La predisposizione al retract è attivata dal PLC, per cui se il PLC non gestisce questa funzionalità, la programmazione della funzione G380 non ha alcun effetto sul CNC.

### 3.21.2 Sinonimi utilizzati per il Retract assi in Emergenza

Per la gestione del retract assi in emergenza, sono presenti appositi segnali che sono stati associati ai sinonimi qui di seguito elencati.

Dove compare il carattere "#", occorre sostituire ad esso il numero d'ordine dell'asse al quale il segnale è riferito. Tale numero, secondo il tipo di CNC, sarà compreso tra 0 e 2 (1801, 1802), tra 0 e 5 (2801) o tra 0 e 15 (4801, 4802).

Esempi:

**OPARTR0** predisposizione relativa all'asse di ordine 0.

**OAXRTR1** movimento di retract dell'asse di ordine 1.

PRIRTR	PLC → CNC	
		<p>Predisporre richiesta di RETRACT assi in emergenza. Gli assi e le quote sono definiti da Part-Program con la funzione G380; l'esecuzione effettiva sarà eseguita se e quando sarà premuto un tasto [HOLD] o [AXISH] o [RESET], solo se il CNC è in modo ESECUZIONE e assi di retract definiti.</p> <p>Questo segnale può essere messo a 1 fisso nel PLC demandando la gestione del RETRACT esclusivamente al programmatore tramite funzione G380, oppure condizionandolo a particolari situazioni di macchina o di lavorazione in corso.</p> <p>L'equazione deve essere scritta prima della chiamata alla funzione tastcnc(), che realizza la gestione del Retract secondo la modalità standard definita da ECS.</p> <p>Ad esempio, se si vuole gestire il retract assi su un tornio solo quando si esegue la macro di lavorazione Sellatura Flangie G669, è sufficiente scrivere la seguente equazione :</p> <pre>PRIRTR = G669;</pre> <p>Utilizzando questo segnale non si deve gestire il segnale IRTR.</p>



<b>IRTR</b>	<b>PLC → CNC</b>	Richiesta esecuzione RETRACT assi in emergenza. Gli assi e le quote sono definiti da Part-Program con la funzione G380. Il CNC resetta la word OPARTR e comunica al PLC, tramite segnale ORTR=1, retract in corso, solo se ci sono assi predisposti. Inoltre segnala con OAXRTR#=1 gli assi in movimento di RETRACT. Durante la fase di esecuzione RETRACT il CNC rifiuta i tasti [RESET], [START] e richieste di CAMBIO MODO. A fine RETRACT il CNC resetta ORTR, emette un comando di RESET e resetta il segnale OAXRTR# di ciascun asse. Il segnale IRTR viene automaticamente resettato se non ci sono assi predisposti al retract. <b>ATTENZIONE!!!</b> Con IRTR=1 e nessun asse predisposto per retract, il CNC esegue solo RESET. Non usare IRTR se si usa PRIRTR con la tastcnc().
<b>ORTR</b>	<b>CNC → PLC</b>	=1: RETRACT in corso di esecuzione
<b>OPARTR</b> (short)	<b>CNC → PLC</b>	Assi predisposti per Retract dalla G380; i singoli bit (vedi sotto) indicano quali assi sono stati programmati. > 0 : ci sono assi predisposti al retract = 0 : nessun asse predisposto al retract. Azzerata con : <ul style="list-style-type: none"><li>• Pulsante [RESET]</li><li>• Carattere “%”</li><li>• Funzioni M02 e M30</li><li>• Segnale ORTR = 1</li></ul>
<b>OPARTR#</b>	<b>CNC → PLC</b>	L'asse di ordine # è predisposto per il retract.
<b>OAXRTR#</b>	<b>CNC → PLC</b>	L'asse di ordine # è in movimento per il retract.

### 3.21.3 Note Importanti

- 1) Il costruttore della M.U. può gestire la richiesta del RETRACT con modalità diversa da quella standard prevista da ECS e realizzata da tastcnc() in base a PRIRTR.
- 2) Al momento della richiesta di retract, il **CNC ignora la predisposizione** su assi che in quel momento sono:
  - FREE
  - PENDOLANTE
  - Non in carico al CNC ( in Libera Servo, in carico al PLC o BRK )
  - Soggetti a MATRICE DINAMICA
  - TWIST
  - INSEGUITORE
  - BLOCCABILI e FERMI

### 3.21.4 Norme di Programmazione

La predisposizione assi al retract viene fatta tramite la funzione **G380**, per cui programmando :  
**G380 <sigla asse><quota> .....** si definiscono gli assi di retract con relativa quota.  
**G380** si annulla la predisposizione assi al retract.  
Con “%”, M02 e M30 si annulla la predisposizione assi al retract.



### 3.21.4.1 Esempio di Part-Program con gestione standard ECS prevista dal PLC :

```
%  
...  
N100 G380 Z100  
...  
N200 G380 X100  
...  
N300 G380  
...  
N400 G380 X150 Z250  
...  
N500 M02 {o M30}
```

% Reset predisposizioni assi per uscita su quote di sicurezza

N100 Viene predisposto il posizionamento del solo asse Z a quota di sicurezza 100.  
Se il PLC ha abilitato la gestione automatica del RETRACT, quando viene premuto un pulsante [RESET],[HOLD] e [AXISH] il CNC esegue il RETRACT su asse Z. A fine retract viene interrotta la lavorazione con emissione di RESET.

N200 Viene annullata la predisposizione precedente sostituendola con l'attuale.  
Se il PLC ha abilitato la gestione automatica del RETRACT, quando viene premuto un pulsante [RESET],[HOLD] e [AXISH] il CNC esegue il RETRACT su asse X. A fine retract viene interrotta la lavorazione con emissione di RESET.

N300 Reset predisposizioni.  
Viene ripristinato il funzionamento normale dei tasti [RESET], [HOLD] e [AXISH] .

N400 Predisposti assi X e Z  
Se il PLC ha abilitato la gestione automatica del RETRACT, quando viene premuto un pulsante [RESET],[HOLD] e [AXISH] il CNC esegue il RETRACT su assi X e Z. A fine retract viene interrotta la lavorazione con emissione di RESET.

N170 Reset predisposizione assi per uscita su quote di sicurezza.

## 3.22 ASSI "INSEGUITORI"

Prestazione presente dalla release software V2.01

Nella contornatura di un profilo raccordato la velocità all'inizio di un blocco può subire una notevole diminuzione se nel blocco precedente erano mossi solo assi del piano di contornatura mentre nel blocco attuale, oltre agli assi del p.c., è da muovere un terzo asse (es. un asse rotante che orienta l'utensile); questo perché il terzo asse inizia il suo movimento a velocità teoricamente nulla, ed essendo legato dall'interpolazione CNC con gli assi p.c., ferma anche loro.

Per non avere questo inconveniente è stata sviluppata la modalità d'interpolazione degli assi "inseguitori".

Con questo termine si riferiscono gli assi soggetti a una particolare modalità di interpolazione, che gli consente di partire a velocità inferiori a quelle previste dalla normale interpolazione CNC, di rincorre gli assi normali programmati nello stesso blocco e riprenderli prima che l'interpolazione raggiunga metà della traiettoria; all'opposto, nella seconda metà della traiettoria, questi particolari assi, passano avanti agli assi normali, poi rallentano e





arrivano a fine blocco, insieme agli assi normali ma ad una velocità inferiore a quella prevista dalla interpolazione CNC tradizionale.

Durante l'interpolazione gli assi inseguitori si trovano, almeno per una parte della traiettoria, ad una posizione diversa da quella (quota teorica) che avrebbero dovuto avere se fossero stati assi normali, cioè si ha un errore di posizione; la capacità di svincolare i normali assi da quelli inseguitori, dipende, oltre che dall'accelerazione e velocità di taratura di questi (in particolare l'accelerazione), da quanto può essere grande questo errore.

### 3.22.1 Norme di programmazione

Per definire gli assi, l'errore max ammesso di ogni asse e attivare la prestazione occorre programmare l'istruzione:

**<PRS:asse1;err1;asse2;err2;kacc>**

dove:

- **PRS** - è l'abbreviazione di "pursuers" (inseguitori).
- **asse1, asse2** - sono le sigle degli assi (o il loro numero d'ordine in axs.tar) che diventano inseguitori.
- **err1, err2** - sono i relativi errori massimi ammessi.
- **kacc** - è una costante moltiplicativa dell'accelerazione degli assi inseguitori che può servire per aumentare o diminuire la prestazione di questa modalità.

Valore di default =1. Può essere impostato tra 0.5 e 3 compresi, se viene scritto un valore fuori da questo campo il coefficiente è posto a 1.

Il massimo numero di assi inseguitori è 2.

Altre forme di programmazione sono:

**<PRS:asse1;err1;asse2;err2>**

**<PRS:asse1;err1>**

**<PRS:kacc>**

La **disabilitazione** della prestazione avviene per ogni singolo asse, programmando l'errore relativo a zero:

**<PRS:asse1;0;asse2;err2>**

con questa istruzione si ha la disabilitazione sull'asse1 e l'attivazione sull'asse2.

### CARATTERISTICHE E LIMITI

1. Gli assi "inseguitori" non sono disattivati da **RESET** e "%", permangono i valori impostati dall'ultima programmazione.
2. La modalità, anche se impostabile con CNC in MDI, è attiva solo in automatico.
3. Non è compatibile con G67, G69, e con la spirale.



- 
4. In grafica la modalità non è attiva.
  5. Non è possibile mettere inseguitore un'asse del piano di con tornatura.
  6. Non è possibile mettere inseguitore un asse lineare del TWT, perchè possono percorrere archi cerchio.
  7. L'istruzione <PRS:...> non può essere emessa in mezzo a geometria non terminata (in mezzo a frasi GAP non concluse) o con compensazione raggio utensile attiva.

## NOTE

- Quando è attivo il TWT con disassamenti non a zero, il problema descritto all'inizio può sussistere anche se è attivo l'inseguimento, perché il rallentamento può essere dovuto non all'asse rotante posto "inseguitore", ma agli assi lineari che al "pivot" sono soggetti a variazioni di velocità non ammesse dalle loro accelerazioni.
- Sempre con TWT attivo e disassamenti non a zero, se l'inseguimento agisce su un asse rotante l'errore presente su questo asse si propaga alla punta, il profilo non viene esatto e l'errore è difficilmente valutabile; questo perché l'inseguimento agisce a valle dell'interpolazione twt



## CAPITOLO 4

### 4 ASSI INDIPENDENTI (FRE)

Si definisce asse Indipendente un asse che non può essere interpolato con altri, ovvero in cui il raggiungimento della quota programmata non avviene in maniera sincrona con gli altri assi. Un asse Indipendente è talvolta anche definito controllato in Punto-a-Punto.

Un asse può essere configurato Indipendente in taratura ( attraverso il file AXS.TAR) o reso indipendente da programma ( attraverso l'istruzione <FRE: nome asse> ). In base comunque alla modalità con cui l'asse è stato dichiarato Indipendente esistono sostanziali differenze nella sua gestione, che vediamo nei paragrafi seguenti.

#### 4.1 Asse dichiarato INDIPENDENTE in AXS.TAR

Per questi assi viene sempre controllato che il movimento rientri nel campo di lavoro e sia entro i fine corsa software.

La programmazione dell'asse da Part Program è sempre assoluta (non si considera la G91 e le eventuali istruzioni D...).

Non si tiene conto di un'eventuale programmazione in pollici (G70). Le quote sono sempre espresse in mm (o gradi)

Le quote programmate sono sempre riferite all'origine assoluta (G53). Non è possibile definire delle origini da parte dell'operatore.

Non si possono applicare trasformazioni di matrici (Statiche o Dinamiche).

Non si può applicare la specularità (istruzione <MIR:...>).

#### 4.2 Asse dichiarato indipendente da PART PROGRAM (Istruzione <FRE:..>)

Il controllo sui fine corsa software e sul campo di lavoro è eseguito nei movimenti assoluti (G90) e nei movimenti incrementali (G91 e D...), purché si sia specificato (agendo nel file COSTUSER sulla variabile %V4194) di ricondurre ad assolute tali quote.

E' comunque possibile la programmazione di movimenti incrementali se abilitata nel programma tramite G91.

L'asse può essere programmato sia in pollici (G70) che in mm (G71).

Non si possono applicare trasformazioni di matrici (Statiche o Dinamiche).

Si può applicare la specularità (istruzione <MIR:...>) purché non sia stata programmata l'istruzione G53 (Origine assoluta).

Tutti gli assi Indipendenti, sia quelli definiti tali in taratura che quelli definiti CNC e successivamente resi indipendenti dall'istruzione <FRE:...>, possono essere agganciati in tutte le modalità (JOG, MDI, AUTO, SET). Possono essere bloccati ed è possibile variarne la FEED impostata attraverso un proprio override (segnali OVRFRE#)

La movimentazione di un asse Indipendente, tipicamente utilizzato come asse di servizio, può avvenire sia da Part Program sia tramite specifiche funzioni PLC.

### 4.3 Gestione assi Indipendenti da Part Program

In questo caso il valore della Feed alla quale si desidera muovere l'asse può essere impostata tramite le istruzioni LIP:

```
<FED: nome ;valore>
<FED: nome ;valore; nome ;valore; nome ;valore>
```

dove: nome = Nome asse

valore = Feed espressa in [mm/min.] o [gradi /min] a seconda del tipo di asse

Il movimento verrà invece comandato, in modo analogo agli assi CNC, associando alla sigla dell'asse la quota finale. Per un approfondimento su tale argomento fare riferimento al manuale "Programmazione Avanzata LIP".

### 4.4 Gestione assi Indipendenti da PLC

E' opportuno in questo caso approfondire alcuni termini nel seguito spesso utilizzati.

Un asse si definisce di "Proprietà del PLC" allorché è posto in LIBERA SERVO ovvero non è più controllato in posizione dal CNC (è sganciato).

Un asse si definisce invece "Gestito dal PLC" allorché risulta ancora controllato in punto-a-punto dal CNC, ma che viene movimentato da PLC tramite specifiche funzioni.

Premesso ciò le funzioni PLC disponibili nei CNC ECS della serie WIN per gestire Assi Indipendenti sono:

<b>Hldfre( )</b>	Gestione Hold Asse
<b>AxMovAbsRap( )</b>	Movimentazione di asse lineare su quote assolute in rapido .
<b>AxMovAbsWork( )</b>	Movimentazione di asse lineare su quote assolute in lavoro.
<b>AxMovIncRap( )</b>	Movimentazione di asse lineare su quote incrementali in rapido.
<b>AxMovIncWork( )</b>	Movimentazioni di asse lineare su quote Incrementali in lavoro.
<b>AxMovRoundInc( )</b>	Movimentazione di asse tondo su posizione angolare assoluta eseguita in lavoro per quote angolari crescenti.
<b>AxMovRoundDec( )</b>	Movimentazione di asse tondo su posizione angolare assoluta eseguita in lavoro per quote angolari crescenti.
<b>AxMovRoundNoZero( )</b>	Movimentazione di asse tondo su posizione angolare assoluta eseguita in lavoro senza attraversare lo zero.
<b>AxMovRound( )</b>	Movimentazione "perpetua" di asse tondo.
<b>AxJogPVelRap( )</b>	Movimentazione in JOG in senso positivo alla velocità di rapido.
<b>AxJogNVelRap( )</b>	Movimentazione in JOG in senso negativo alla velocità di rapido.
<b>AxJogPVelWork( )</b>	Movimentazione in JOG positivo alla velocità di lavoro.
<b>AxJogNVelWork( )</b>	Movimentazione in JOG negativo alla velocità di lavoro.
<b>AxJogSpRap( )</b>	Movimentazione in JOG di spazio alla velocità di rapido.
<b>AxJogSpWork( )</b>	Movimentazione in JOG di spazio alla velocità di lavoro.
<b>AxMov( )</b>	Movimentazione generica assoluta (consente, tramite le funzioni G800-G900 di eseguire il set dell'asse da PLC).
<b>AxMovInc( )</b>	Movimentazione generica incrementale (consente, tramite le funzioni G800-G900 di eseguire il set dell'asse da PLC).
<b>AxFeed( )</b>	Programmazione nuova Feed per un asse gestito da PLC già in movimento.
<b>AxPlcModeSetOn( )</b>	Predisporre l'asse ad essere gestito da PLC anche con SET non fatto, e a non sganciarsi e tornare CNC, in caso di SET perso.
<b>AxPlcModeSetOff( )</b>	Predisporre l'asse a non passare gestito da PLC in caso di SET non fatto e a tornare automaticamente CNC in caso di SET perso.
<b>IsAxPlcModeSet( )</b>	Ritorna il tipo di predisposizione attiva per quanto concerne le condizioni di SET da fare o SET perso.



<b>AxPlcAn( )</b>	Annulla comandi/istruzioni e movimenti di un asse gestito e movimentato da PLC.
-------------------	---

Per dettagli su tali funzioni fare riferimento più avanti al paragrafo 4.6 "ROUTINES E FUNZIONI DEL PLC".

#### 4.5 Sinonimi utilizzati per gli assi indipendenti

Per la gestione di ciascun asse indipendente tramite il PLC, sono presenti vari segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Sono presenti anche altri segnali, con i relativi sinonimi, validi per tutti gli assi; sono già stati visti nel capitolo relativo agli Assi CNC, al quale si rimanda, e sono qui indicati con "Vedi Assi CNC". Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

Nei sinonimi dove compare il carattere "#", occorre sostituire ad esso il numero d'ordine dell'asse al quale il segnale è riferito. Tale numero, secondo il tipo di CNC, sarà compreso tra 0 e 2 (1801, 1802), tra 0 e 5 (2801) o tra 0 e 15 (4801, 4802).

Esempi:

**IAXPLC0** Richiesta di gestire da PLC l'asse di ordine 0.

**IAXPLC1** Richiesta di gestire da PLC l'asse di ordine 1.

**IAXPLC5** Richiesta di gestire da PLC l'asse di ordine 5.

<b>OBAT#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>OBOK#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>IEME#</b>	<b>PLC → CNC</b>	Vedi Assi CNC
<b>ISGA#</b>	<b>PLC → CNC</b>	Vedi Assi CNC
<b>OBLC#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>IBLC#</b> <b>BLCAX#</b>	<b>PLC → CNC</b>	Posto a 1, segnala asse bloccato meccanicamente. Il CNC ferma l'asse (emulando <b>IHLD#</b> ) lasciando gli altri assi liberi di continuare il movimento. Se il PLC usa la routine <b>aggax()</b> , deve essere usato il corrispondente segnale <b>BLCAX#</b> .
<b>OBNT#</b>	<b>CNC → PLC</b>	Comando di movimento in corso sull'asse. Posto a 1 assieme a <b>OBNA#</b> Posto a 0 a movimento terminato o annullato.
<b>IANA#</b>	<b>PLC → CNC</b>	Posto a 1 richiede di annullare il movimento in corso. L'asse verrà arrestato con l'accelerazione configurata. Tale comando è da utilizzare per arrestare l'asse in caso di allarme sulla Macchina Utensile.



<b>IHLD#</b> <b>HLDFRE#</b>	<b>PLC → CNC</b>	Posto a 1 richiede di mettere in HOLD l'asse. E' imposta velocità 0 con rampa di decelerazione nominale. L'esecuzione del movimento riprenderà allorché viene nuovamente posto IHLD#=0. Se il PLC usa la routine <b>hldfre()</b> , occorre usare il corrispondente segnale <b>HLDFRE#</b> .
<b>ILS#</b>	<b>PLC → CNC</b>	Vedi Assi CNC
<b>OLS#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>INAL#</b>	<b>PLC → CNC</b>	Vedi Assi CNC
<b>OBM0#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>OBM1#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>OBSE#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>IBSET#</b>	<b>PLC → CNC</b>	Vedi Assi CNC
<b>OBSET#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>OBAG#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>OBCN#</b>	<b>CNC → PLC</b>	<b>OBCN# =1</b> Asse agganciato in contouring. Nel caso di un asse Indipendente ciò avviene solo in modo JOG.
<b>OBIN#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>OBMT#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>OPSGL#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>OPFRM#</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>OBNA#</b>	<b>CNC → PLC</b>	Asse non arrivato. <b>Messo a 1</b> con asse in moto (OBMT#=1). <b>Messo a 0</b> quando l'asse è arrivato alla quota comandata ed è fermo (OPFRM#=1).
<b>ONEWP</b>	<b>CNC → PLC</b>	Vedi Assi CNC
<b>IAXPLC#</b>	<b>PLC → CNC</b>	<b>IAXPLC# = 1</b> Richiesta di gestire l'asse da PLC <b>IAXPLC# = 0</b> Richiesta di riconsegnare l'asse al CNC. <b>Per i particolari di funzionamento, vedi note a fine tabella.</b>
<b>OAXPLC#</b>	<b>CNC → PLC</b>	<b>OAXPLC# = 1</b> Risposta affermativa alla richiesta di gestire l'asse da PLC.
<b>IJOGP#</b>	<b>PLC → CNC</b>	<b>IJOGP# = 1</b> Richiesta di JOG+ (positivo) di velocità in Rapido. La richiesta è accettata solo con assi gestiti da PLC (con <b>OAXPLC# =1</b> ). <b>Per i particolari di funzionamento, vedi note a fine tabella.</b>



<b>IJOGN#</b>	<b>PLC → CNC</b>	<b>IJOGN# = 1</b> Richiesta JOG— (negativo) di velocità in Rapido. La richiesta è accettata solo con assi gestiti da PLC (con <b>OAXPLC# =1</b> ). <b>Per i particolari di funzionamento, vedi note a fine tabella.</b>
<b>OCPLC#</b> (short)	<b>CNC → PLC</b>	<b>OCPLC#</b> contiene il codice di ritorno dell'ultima richiesta di movimento attivata dal PLC. <b>Per i particolari di funzionamento, vedi note a fine tabella.</b>
<b>OVRFRE#</b> (unsigned char)	<b>PLC → CNC</b>	Valore Override velocità in % (0...255) per l'asse indipendente #.

#### 4.5.1 Note su IAXPLC#

Il segnale **IAXPLC#** può essere emesso nei confronti di un qualsiasi asse configurato in **AXS.TAR**. La risposta del CNC **OAXPLC#** indica lo stato di “aggancio” attuale dell'asse di ordine #. In particolare **OAXPLC#=1** indica che l'asse è gestito dal PLC. Un asse gestito da PLC viene visto dal CNC come un asse Indipendente (agganciato in punto-a-punto). Il CNC ne controlla solo il ciclo di spazio non potendone comandare, né da Part Program né in MDI, la movimentazione (pena l'emissione di allarme 2158). La movimentazione dell'asse è, infatti, possibile solo da PLC tramite le funzioni **AxMov...()** ed **AxJog...()**. Il Reset ed i cambi modo attuati sul CNC non interferiscono minimamente sullo stato di un asse gestito da PLC, in queste condizioni l'eventuale movimentazione dell'asse non viene, infatti, interrotta e lo stato di aggancio in punto-a-punto rimane invariato. Il PLC ha comunque la possibilità, durante la fase di reset (caratterizzata dal segnale **ORST=1**) e prima di dare il consenso alla prosecuzione del comando (tramite il segnale **IRES**), di annullare la movimentazione ed i comandi in corso sull'asse tramite la funzione **AxPlcAn ()**.

La richiesta di “gestire l'asse dal PLC” e la duale di riconsegnarlo al CNC sono completamente asincrone rispetto al funzionamento del CNC. Possono pertanto essere inoltrate in qualunque momento e stato del CNC, non necessitano di WAIT e possono essere esaudite anche durante l'elaborazione di Part Program ed in presenza di movimentazione assi. Come uniche eccezioni, in fase di:

- CNC in modo SET-ESECUZIONE
- RESET in corso (**ORST=1**) e consenso a procedere da parte del PLC (**IRES=1**)
- KEY in corso

le richieste di aggancio/sgancio vengono sospese, non vengono cioè esaudite immediatamente, ma congelate sino al termine delle condizioni sopraddette; nei primi due casi, anche le richieste di movimentazione vengono sospese.

La richiesta di consegnare l'asse di ordine # al PLC attuata tramite il segnale **IAXPLC#= 1** non è accettata se l'asse si trova in uno dei seguenti stati:

- Asse non disponibile (campo **DSP=N** in **AXS.TAR**)
- Asse in libera servo (**OLS# =1**)
- Asse pendolante (**OPND# = 1**) {F}
- Asse con SET non fatto (**OBSE# =0**) senza aver eseguito la funzione **AxPlcModeSetOn ()** che disabilita il relativo controllo.
- Mandrino senza trasduttore oppure senza Gamma inserita.



La richiesta di ritornare l'asse sotto controllo del CNC attuata tramite il segnale **IAXPLC#= 0** ha successo quando anche il segnale **OAXPLC#** è posto a 0. In questo caso l'asse viene "riagganciato" come specificato in taratura in **AXS.TAR**.

Un asse "gestito da PLC" è automaticamente riconsegnato al CNC allorché:

- non sia stato fatto il **SET** o venga perso e venga chiamata la funzione di predisposizione **AxPlcModeSetOff ( )** che riabilita il relativo controllo.
- venga perso il SET (**OBSE# =0**) senza aver eseguito la funzione **AxPlcModeSetOn ( )** che disabilita il relativo controllo.

### Note per l'asse Mandrino

La richiesta **IAXPLC#=1** sull'asse mandrino provoca l'aggancio del mandrino in ciclo di spazio con la conseguente uscita dal modo velocità e l'orientamento dello stesso come se fosse stata programmata una **M19** con i parametri definiti in **COST** o **COSTUSER**:

- **#299** = Quota posizionamento mandrino in M19, in gradi
- **#272** = Velocità di orientamento in M19, in giri/min.

Sono presenti anche i seguenti parametri per il mandrino secondario:

- **#298** = Quota posizionamento mandrino secondario in M19, in gradi
- **#274** = Velocità di orientamento mandrino secondario in M19, in giri/min.

**N.B.:** si consiglia di non modificare direttamente i records del file COST ma di copiarli e modificarli nel file COSTUSER esplicitamente previsto per le personalizzazioni legate ad una macchina specifica. In questo modo si facilita il "trasporto" delle personalizzazioni da una release di software ad una successiva

Per l'asse mandrino non si tiene conto della predisposizione fatta con **AxPlcModeSetOn()**, in quanto il SET dell'asse viene fatto automaticamente.

### 4.5.2 Note su IJOGP# e IJOGN#

I segnali **IJOGP#** e **IJOGN#** sono, per motivi di sicurezza, resettati al momento della presa in consegna dell'asse dal PLC (transizione ad 1 del bit **OAXPLC#**).

In caso di "ambiguità" dei segnali (ad esempio **IJOGP#** e **IJOGN#** contemporaneamente ad 1), viene mantenuto l'ultimo stato valido acquisito. La movimentazione in JOG di velocità può essere comandata anche tramite le specifiche funzioni **AxJog...()**.

La velocità di movimento dell'asse può essere "modulata", tra 0 e 100% della rapida, tramite **OVRFRE#**

### 4.5.3 Note su OCPLC

**OCPLC#** contiene il codice di ritorno dell'ultima richiesta di movimento attivata dal PLC.

Sono previsti i seguenti valori:

- <0 → Codice di errore ( richiesta non acquisita dal CNC )
- 1 → Richiesta di JOG non accettata perché l'asse non è arrivato e fermo
- 2 → Codice di movimento errato
- 0 → Codice di successo (Comando Terminato; Richiesta di movimento attivata e sono validi i segnali legati al movimento dell'asse (OBMTxx, OBNAxx, ...).)
- >0 → Codice di "in corso"
- 1 → In corso ( il PLC ha inoltrato la richiesta al CNC)
- 2 → In corso (il CNC sta analizzando la richiesta di movimento Normale pervenuta dal PLC)
- 3 → In corso (il CNC sta analizzando la richiesta di movimento in JOG pervenuta dal PLC)





- 12 → Movimento Normale in corso (richiesta acquisita dal CNC ed inoltrata verso il SW di gestione del servoloop).
- 13 → Movimento JOG in corso (richiesta acquisita dal CNC ed inoltrata verso il SW di gestione del servoloop).
- 14 → Istruzione in corso (richiesta acquisita dal CNC ed inoltrata verso il SW di gestione del servoloop).
- 23 → Movimento JOG in corso (richiesta acquisita dal SW di gestione del servoloop e sono validi i segnali legati alla movimentazione dell'asse (OBMTxx, OBNAxx, ...).)

I valori di **OCPLCxx=0** e **OCPLCxx=23** (quest'ultimo solo per comandi JOG) **indicano un successo sull'attivazione del comando**; tali valori sono sincroni con i segnali legati alla movimentazione dell'asse, in particolare con il segnale di ASSE NON ARRIVATO **OBNAxx**.

Viene differenziato il codice 0 dal codice 23, in quanto, a differenza dei movimenti "NORMALI", i movimenti in JOG sono strettamente sequenziali, non viene cioè accettato un nuovo comando di movimentazione fintanto sia in corso un movimento JOG.

Il valore OCPLCxx=23 rimane attivo fino al termine del movimento JOG+/JOG-, il valore OCPLCxx=0 dichiara invece che la movimentazione richiesta è stata accettata ed attivata e che nello stesso momento sono validi i segnali di movimentazione dell'asse relativo.

## 4.6 ROUTINES E FUNZIONI DEL PLC

### 4.6.1 Routine "HLDFRE"

Chiamata: **hldfre ()**

Scopo: Gestisce l'hold su assi Indipendenti (FRE).

Parametri in Ingresso:

- **HLDFRE# = 1** PONE IN HOLD L'ASSE INDIPENDENTE DI ORDINE #

In modalità SET, premendo i tasti HOLD o AXIS.H sulla console del CNC, la routine **hldfre()** blocca automaticamente anche tutti gli assi indipendenti.

In modalità diverse, per bloccare gli assi indipendenti è necessario utilizzare la routine **hldfre()** mettendo ad "1" il relativo input.

La routine **hldfre ()** non prevede parametri in uscita.

```
/*-----  
Esempio d'uso della routine HLDFRE()  
-----*/  
HLDFRE0= R0.bit._00;  
/* HOLD ASSE FREE DI ORDINE 0 DA PULSANTE ESTERNO */  
HLDFRE1= OHLd;  
/* HOLD ASSE FREE DI ORDINE 1 CON BLOCCO CNC */  
HLDFRE2= OHLAX;  
/* HOLD ASSE FREE DI ORDINE 2 CON HOLD ASSI */  
HLDFRE3= logica ...;  
/* HOLD ASSE FREE DI ORDINE 3 RICHIESTO DA LOGICA */  
hldfre();  
/* Chiamata routine gestione Hold assi Indipendenti */
```



---

---

## 4.6.2 Funzioni di movimentazione Assoluta/Incrementale di un Asse PLC

In questo paragrafo è descritta una famiglia di funzioni utilizzabili per la movimentazione di assi assegnati in gestione al PLC (caratterizzati dunque dal relativo bit **OAXPLC#=1**). Le funzioni sono tutte applicabili ad assi Tondi, solo in parte ad assi Lineari.

Contrariamente a molte delle routines e/o funzioni viste precedentemente, che vanno inserite nel loop del PLC per essere richiamate ed eseguite sempre, le funzioni che seguono sono del tipo "one shot", vanno cioè chiamate una sola volta quando serve attivare o disattivare o comandare l'esecuzione di una data funzione.

Per questo tipo di gestione sono utili i Sequenziatori (SEQ), "macchine logiche" realizzate tramite le istruzioni "C" **switch** e **case**, che permettono di definire le azioni da eseguire in ogni "stato" del sequenziatore e le condizioni di passaggio da uno stato ad un altro. Inserendo la chiamata alla funzione in uno stato del sequenziatore usato e definendo un passaggio incondizionato allo stato successivo, si realizza la funzione "one shot" ed anche la corretta logica, a stati successivi, di scelta funzione, attivazione, controllo ed attesa dei risultati.

- Per gli assi assegnati in gestione al PLC, le quote programmate sono sempre riferite all'origine assoluta (G53). Non è possibile attivare le origini definite dall'operatore.
- Viene sempre controllato che il movimento rientri nel campo di lavoro e sia entro i fine corsa software.
- Non si tiene conto di un'eventuale programmazione in pollici (G70); le quote sono perciò sempre espresse in mm (o gradi)
- Non si possono applicare trasformazioni di matrici (Statiche o Dinamiche).
- Non si può applicare la specularità (istruzione <MIR:...>).

### 4.6.2.1 AxMovAbsRap: movimentazione di asse su quote assolute in rapido

Chiamata: **esito = AxMovAbsRap (IndAx, Position)**

Valido sia per assi lineari che tondi. In quest'ultimo caso la posizione angolare indicata in Position è espressa in gradi ed è raggiunta per la via più breve.

### 4.6.2.2 AxMovAbsWork: movimentazione di asse su quote assolute in lavoro

Chiamata: **esito = AxMovAbsWork (IndAx, Position, Velocity)**

Valido sia per assi lineari che tondi. In quest'ultimo caso la posizione angolare indicata in Position è espressa in gradi ed è raggiunta per la via più breve.

### 4.6.2.3 AxMovIncRap: movimentazione di asse su quote incrementali in rapido

Chiamata: **esito = AxMovIncRap (IndAx, Position)**

Valido sia per assi lineari che tondi. In quest'ultimo caso la posizione angolare indicata in Position è espressa in gradi ed è raggiunta per la via più breve.

### 4.6.2.4 AxMovIncWork: movimentazioni di asse su quote Incrementali in lavoro

Chiamata: **esito = AxMovIncWork (IndAx, Position, Velocity)**



Valido sia per assi lineari che tondi. In quest'ultimo caso la posizione angolare indicata in Position è espressa in gradi ed è raggiunta per la via più breve.

#### 4.6.2.5 AxMovRoundInc: movimentazione di asse tondo su pos. angolare assoluta in lavoro

Chiamata: **esito = AxMovRoundInc( IndAx, Position, Velocity)**

In questo caso il movimento è eseguito per quote angolari crescenti.

#### 4.6.2.6 AxMovRoundDec: movimentazione di asse tondo su pos. angolare assoluta in lavoro

Chiamata: **esito = AxMovRoundDec (IndAx, Position, Velocity)**

In questo caso il movimento è eseguito per quote angolari decrescenti.

#### 4.6.2.6.1 AxMovRoundNoZero: movimentazione di asse tondo su pos. angolare assoluta in lavoro

Chiamata: **esito = AxMovRoundNoZero (IndAx, Position, Velocity)**

In questo caso il movimento è eseguito senza attraversare lo zero.

#### 4.6.2.7 AxMovRound: movimentazione "perpetua" di asse tondo

Chiamata: **esito = AxMovRound (IndAx, Velocity)**

Il segno del parametro Velocity (positivo o negativo) determina il senso di rotazione.

#### 4.6.2.8 Parametri in ingresso ed in uscita

Parametri di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse che si intende movimentare. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**Position** = (double) Quota in mm o Posizione angolare in gradi che si vuole l'asse raggiunga.

**Velocity** = (double) Velocità programmata per lo spostamento in mm/min ( in caso di assi lineari), in gradi/min ( in caso di assi tondi)

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno della routine; assume uno dei seguenti valori:

0	Successo – richiesta inoltrata
1	Insuccesso – numero d'ordine errato
2	Insuccesso – asse non gestito da PLC
3	Insuccesso - comando in corso (aggancio /sgancio)
4	Insuccesso – altro comando di movimentazione in corso
5	Insuccesso – comando di annulla movimentazione in corso →AxPlcAn( )
6	Insuccesso – errore di sintassi su comando di movimentazione

Note:

Gli assi lineari in gradi (**IGR**) sono assimilabili a quelli lineari in mm con l'unica differenza che i parametri **Position** e **Velocity** si esprimono rispettivamente in gradi e in gradi/min.

---

---

Quando si esegue una delle funzioni di movimentazione descritte, **esito = 0** significa che la richiesta è stata inoltrata al CNC ma non necessariamente eseguita. Per conoscere l'esatto stato di una richiesta di movimentazione, occorre "interpretare" il valore di **OCPLC#** (Vedi Paragrafo 4.5 "SINONIMI. . .").

In caso di **RESET** o **Cambio Modo** sul CNC, l'eventuale movimentazione dell'asse PLC non viene interrotta (annullata) e lo stato di aggancio in punto a punto dell'asse rimane invariato.

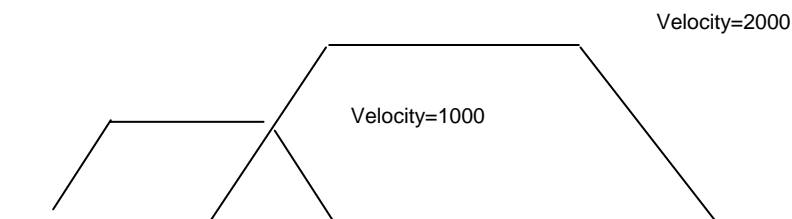
Il PLC ha comunque la possibilità, in fase di **RESET** (caratterizzata dal segnale **ORST** attivo), prima di darne il consenso al proseguimento (tramite il segnale **IRES**), di annullare la movimentazione in corso sull'asse tramite la funzione **AxPlcAn ( )** (per dettagli vedi Paragrafo 4.6.7).

Tutte le richieste di movimentazione rivolte ad un asse del PLC sono completamente asincrone rispetto al funzionamento del CNC, non necessitano di **WAIT** e possono essere esaudite anche durante l'elaborazione di Part-Program ed in presenza di movimentazione di assi CNC.

Per consentire di gestire in modo continuo un asse, i comandi di movimentazione analizzati possono essere "accodati" fino ad un massimo di 3 comandi consecutivi.

Nel caso di due comandi di movimentazione consecutivi il parametro "**Velocity**" del secondo blocco viene acquisito laddove si sarebbe iniziata la fase di decelerazione del primo blocco (per maggiori dettagli vedi **Fig. 1**).

Le funzioni descritte in questo paragrafo tipicamente richiedono che il SET dell'asse sia stato



**Fig. 1**

**AxMovAbsWork(0,100,1000) AxMovAbsWork(0,200,3000)**

già fatto. Volendole eseguire con SET non fatto o qualora sia stato perso occorre predisporre opportunamente il CNC con la funzione **AxPlcModeSetOn( )** (per dettagli vedi Paragrafo 4.6.6.1).

### 4.6.3 Funzioni di movimentazione generica G800-G900

Le funzioni elencate in questo paragrafo consentono di eseguire il SET di un asse gestito da PLC. Si ricorda a tal fine che per consentirne l'esecuzione, in condizioni di SET non fatto o perso in cui tipicamente l'asse non può essere "agganciato" dal PLC, è necessario predisporre il CNC con la specifica funzione **AxPlcModeSetOn( )** descritta al Paragrafo 4.6.6.1.

#### 4.6.3.1 AxMov: movimentazione G800-G900 su quote assolute

Chiamata: **esito = AxMov (IndAx, TypeMov, Position, Velocity)**



#### 4.6.3.2 AxMovInc: movimentazione G800-G900 su quote incrementali

Chiamata: **esito = AxMovInc (IndAx, TypeMov, Position, Velocity)**

#### 4.6.3.3 Parametri in ingresso ed in uscita

Parametri di Ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse che si intende movimentare. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**TypeMov** = (unsigned short) Questo parametro specifica il tipo di movimento richiesto. A tal fine fare riferimento alla seguente tabella:

TypeMov	Funzione espletata
861	Comanda una <b>G861</b> (Forza quota 0 su marker – valida per assi lineari incrementali). Occorre definire anche i parametri <b>Position</b> e <b>Velocity</b> .
862	Comanda una <b>G862</b> (Impone quota su marker – con <b>G878</b> – assi lineari con righe assolutizzate). Occorre definire anche i parametri <b>Position</b> e <b>Velocity</b> .
878	Comanda una <b>G878</b> (Formazione Origini – con <b>G862</b> – valida per assi lineari con righe assolutizzate). Occorre definire anche i parametri <b>Position</b> e <b>Velocity</b> .
883	Comanda una <b>G883</b> (Arresto su marker - assi lineari). Occorre definire anche il parametro <b>Velocity</b> .
885	Comanda una <b>G885</b> (Arresto su apertura micro di zero – valida per assi lineari). Occorre definire anche il parametro <b>Velocity</b> .
887	Comanda una <b>G887</b> (Arresto su chiusura micro di zero – valida per assi lineari). Occorre definire anche il parametro <b>Velocity</b> .
888	Comanda una <b>G888</b> (Ricerca distanza Marker Assi Master/Slave Gantry Sercos – valida per assi lineari). Occorre definire anche il parametro <b>Velocity</b> .
903	Comanda una <b>G903</b> (Arresto su Marker - su via più breve - valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
907	Comanda una <b>G907</b> (Arresto su Marker – per quote crescenti –valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
911	Comanda una <b>G911</b> (Arresto su Marker – per quote decrescenti – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
915	Comanda una <b>G915</b> (Arresto su chiusura micro di zero – senza passaggio per lo zero – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
921	Comanda una <b>G921</b> (Arresto su apertura micro di zero – per via più breve – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
923	Comanda una <b>G923</b> (Arresto su chiusura micro di zero - per via più breve – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
925	Comanda una <b>G925</b> (Arresto su apertura micro di zero – per quote crescenti – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
927	Comanda una <b>G927</b> (Arresto su chiusura micro di zero – per quote crescenti – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
929	Comanda una <b>G929</b> (Arresto su apertura micro di zero – per quote decrescenti - assi tondi).
931	Comanda una <b>G931</b> (Arresto su chiusura micro di zero- per quote decrescenti – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
933	Comanda una <b>G933</b> (Arresto su micro di zero in apertura – movimento senza passare per lo 0 – valida per assi tondi)
935	Comanda una <b>G935</b> (Arresto su chiusura micro di zero – movimento senza passare per lo 0 – valida per assi tondi).
940	Comanda una <b>G940</b> (movimento per la via più breve – valida per assi tondi).



<b>941</b>	Comanda una <b>G941</b> (movimento per quote crescenti – valida per assi tondi).
<b>942</b>	Comanda una <b>G942</b> (movimento per quote decrescenti – valida per assi tondi).
<b>943</b>	Comanda una <b>G943</b> (movimento senza passaggio per lo 0 – valida per assi tondi).
<b>951</b>	Comanda una <b>G951</b> (Ricerca distanza Marker Assi Master/Slave Gantry Sercos – movimento per la via più breve – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
<b>952</b>	Comanda una <b>G952</b> (Ricerca distanza Marker Assi Master/Slave Gantry Sercos – movimento per quote crescenti – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
<b>953</b>	Comanda una <b>G953</b> (Ricerca distanza Marker Assi Master/Slave Gantry Sercos – movimento per quote decrescenti – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .
<b>954</b>	Comanda una <b>G954</b> (Ricerca distanza Marker Assi Master/Slave Gantry Sercos – movimento senza passaggio per lo 0 – valida per assi tondi). Occorre definire anche il parametro <b>Velocity</b> .

**Position** = (double) Quota in mm o posizione angolare in gradi che si vuole l'asse raggiunga. La quota è assoluta per la funzione **AxMov()** mentre è incrementale per la **AxMovInc()**.

Il parametro deve essere scritto anche per i **TypeMov** che non lo richiederebbero; in questo caso, il valore non è significativo.

**Velocity** = (double) Velocità programmata per lo spostamento in mm/min ( in caso di assi lineari), in gradi/min ( in caso di assi tondi)

Il parametro deve essere scritto anche per i **TypeMov** che non lo richiederebbero; in questo caso, il valore non è significativo.

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno della routine . Può assumere uno dei seguenti valori:

- 0** Successo – richiesta inoltrata
- 1** Insuccesso – numero d'ordine errato
- 2** Insuccesso – asse non gestito da PLC
- 3** Insuccesso - comando in corso (aggancio /sgancio)
- 4** Insuccesso – altro comando di movimentazione in corso
- 5** Insuccesso – comando di annulla movimentazione in corso → **AxPlcAn()**
- 6** Insuccesso – errore di sintassi su comando di movimentazione

Note:

Quando si esegue una delle funzioni di movimentazione descritte, **esito = 0** significa che la richiesta è stata inoltrata al Controllo Numerico ma non necessariamente eseguita. Per conoscere infatti l'esatto stato di una richiesta di movimentazione, occorre "interpretare" il valore di **OCPLC#** (Vedi per dettagli Paragrafo 4.5 "SINONIMI").

In caso di **RESET** o **Cambio Modo** sul CNC, l'eventuale movimentazione dell'asse PLC non viene interrotta (annullata) e lo stato di aggancio in punto a punto dell'asse rimane invariato.

Il PLC ha comunque la possibilità, in fase di **RESET** (caratterizzata dal segnale **ORST** attivo), prima di dare il consenso al proseguimento (tramite il segnale **IRES**), di annullare la movimentazione in corso sull'asse tramite la funzione **AxPlcAn()** (Per dettagli vedi Paragrafo 4.6.7).



Tutte le richieste di movimentazione rivolte ad un asse del PLC sono completamente asincrone rispetto al funzionamento del CNC, non necessitano di WAIT e possono essere esaudite anche durante l'elaborazione di Part-Program ed in presenza di movimentazione di assi CNC.

Per maggiori dettagli sulle funzioni **G800-G900** vedi capitolo specifico in **"Manuale Tarature" 720P385**.

#### 4.6.4 Funzioni per la movimentazione di un Asse PLC in JOG

Nel prosieguo è descritta una famiglia di funzioni utilizzabili per la movimentazione in JOG di assi assegnati in gestione al PLC (caratterizzati dunque dal relativo bit **OAXPLC#=1**). Le funzioni sono applicabili sia ad assi Lineari che Tondi

##### 4.6.4.1 AxJogPVelRap: movimento in JOG in senso positivo alla velocità di rapido

Chiamata: **esito = AxJogPVelRap (IndAx)**

##### 4.6.4.2 AxJogNVelRap: movimento in JOG in senso negativo alla velocità di rapido

Chiamata: **esito = AxJogNVelRap (IndAx)**

##### 4.6.4.3 AxJogPVelWork: movimento in JOG in senso positivo alla velocità di lavoro

Chiamata: **esito = AxJogPVelWork (IndAx, Velocity)**

##### 4.6.4.4 AxJogNVelWork: movimento in JOG in senso negativo alla velocità di lavoro

Chiamata: **esito = AxJogNVelWork (IndAx, Velocity)**

##### 4.6.4.5 AxJogSpRap: movimento in JOG di spazio alla velocità di rapido

Chiamata: **esito = AxJogSpRap (IndAx, Position)**

##### 4.6.4.6 AxJogSpWork: movimento in JOG di spazio alla velocità di lavoro.

Chiamata: **esito = AxJogSpWork (IndAx, Position, Velocity)**

##### 4.6.4.7 Descrizione parametri in ingresso ed in uscita

Parametri di Ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse che si intende movimentare. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**Position** = (double) Valore dello step in mm o in gradi di cui si desidera l'asse si muova in JOG di spazio.

**Velocity** = (double) Velocità in mm/min (in caso di assi lineari), in gradi/min (in caso di assi tondi) a cui si desidera muovere l'asse in JOG.

Parametro di uscita:

**esito** = (unsigned short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

- |   |                                     |
|---|-------------------------------------|
| 0 | Successo – richiesta inoltrata      |
| 1 | Insuccesso – numero d'ordine errato |



---

---

2	Insuccesso – asse non gestito da PLC
3	Insuccesso - comando in corso (aggancio /sgancio)
4	Insuccesso – altro comando di movimentazione in corso
5	Insuccesso – comando di annulla movimentazione in corso → <b>AxPlcAn ()</b>
6	Insuccesso – errore di sintassi su comando di movimentazione

Note:

Gli assi lineari in gradi (**IGR**) sono assimilabili a quelli lineari in mm con l'unica differenza che i parametri **Position** e **Velocity** si esprimono rispettivamente in gradi e in gradi/min.

Quando si esegue una delle funzioni di movimentazione descritte **esito = 0** significa che la richiesta è stata inoltrata al CNC ma non necessariamente eseguita. Per conoscere l'esatto stato di una richiesta di movimentazione, occorre "interpretare" il valore di **OCPLC#** (Vedi Paragrafo 4.5 "SINONIMI...").

A differenza dei comandi di movimentazione i comandi **AxJog...()** non possono essere accodati (richiedono infatti per essere accettati che l'asse sia fermo).

In caso di **RESET** o **Cambio Modo** sul CNC, l'eventuale movimentazione dell'asse PLC non viene interrotta (annullata) e lo stato di aggancio in punto a punto dell'asse rimane invariato.

Il PLC ha comunque la possibilità, in fase di **RESET** (caratterizzata dal segnale segnale **ORST** attivo), prima di dare il consenso al prosieguo (tramite il segnale **IRES**), di annullare la movimentazione in corso sull'asse tramite la funzione **AxPlcAn ()** (Vedi Paragrafo 4.6.7).

Tutte le richieste di movimentazione rivolte ad un asse del PLC sono completamente asincrone rispetto al funzionamento del CNC, non necessitano di **WAIT** e possono essere esaudite anche durante l'elaborazione di Part-Program ed in presenza di movimentazione di assi CNC.

#### 4.6.5 AxFeed: modifica della Feed di un Asse PLC in movimento

Chiamata: **esito = AxFeed (IndAx, Velocity)**

Parametri di Ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse che si intende movimentare.

In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**Velocity** = (double) Nuova velocità in mm/min (in caso di assi lineari), in gradi/min (in caso di assi Tondi o IGR) che si desidera applicare all'asse in movimento.

Parametro di uscita:

**esito** = (unsigned short) valore di ritorno delle routine. Può assumere uno dei seguenti valori:

0	Successo – richiesta inoltrata
1	Insuccesso – numero d'ordine errato
2	Insuccesso – asse non gestito da PLC
3	Insuccesso - comando in corso (aggancio /sgancio)

#### 4.6.6 Funzioni di predisposizione valide per Assi PLC

La famiglia di funzioni qui descritte consente di predisporre l'aggancio di un asse come "gestito da PLC" anche in caso di Set da fare o di non farlo sganciare automaticamente in caso di Set perso.





#### 4.6.6.1 AxPlcModeSetOn: predisposizione a mantenere gestito da PLC un asse in caso di SET da fare o perso

Chiamata: **esito = AxPlcModeSetOn (IndAx)**

Parametro di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse che si intende movimentare. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |   |                                     |
|---|-------------------------------------|
| 0 | Successo – richiesta inoltrata      |
| 1 | Insuccesso – numero d'ordine errato |

#### 4.6.6.2 AxPlcModeSetOff: predisposizione per far ritornare gestito da CNC un asse in caso di SET da fare o perso

Chiamata: **esito = AxPlcModeSetOff (IndAx)**

Parametro di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse che si intende movimentare. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |   |                                     |
|---|-------------------------------------|
| 0 | Successo – richiesta inoltrata      |
| 1 | Insuccesso – numero d'ordine errato |

**Nota:**

Questa condizione è quella di default del CNC.

#### 4.6.6.3 IsAxPlcModeSet: fornisce informazioni sulla predisposizione attiva

Chiamata: **esito = IsAxPlcModeSet (IndAx)**

Parametro di ingresso:

**IndAx** = (unsigned short) Numero d'ordine dell'asse che si intende movimentare. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |   |  |
|---|--|
| 0 | L'asse in caso non abbia il SET fatto o lo perda viene gestito da CNC  |
| 1 | L'asse, anche in caso non abbia il SET fatto, può essere comunque "passato in gestione al PLC" e in caso lo perda, se già lo era, rimane in gestione al PLC. |

#### 4.6.7 AxPlcAn: annullamento di movimenti e comandi in corso su un asse PLC

Chiamata: **esito = AxPlcAn (IndAx)**

Parametro di ingresso:

---

---



---

**IndAx** = (unsigned short) Numero d'ordine dell'asse che si intende movimentare. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 0**      Successo – richiesta inoltrata
- 1**      Insuccesso – numero d'ordine errato
- 2**      Insuccesso – l'asse non è gestito dal PLC



## CAPITOLO 5

### 5 MANDRINO

In questo capitolo sono trattati i segnali e le funzioni che riguardano la gestione del mandrino della Macchina Utensile, realizzata sui CNC ECS serie WIN.

#### 5.1 GENERALITÀ

I CNC ECS serie WIN possono gestire **due mandrini** (Principale e Secondario) direttamente controllati dal CNC stesso più un **terzo mandrino** (Utensile Rotante) pilotato dal PLC.

- Il mandrino **PRINCIPALE** gestito dal CNC è considerato di **FRESATURA** oppure di **TORNITURA** in base al tipo d'apparato installato, rispettivamente **CNC FRESA (MILL)** o **CNC TORNIO(LATHE)**. In AXS.TAR, il mandrino Principale è identificato dal campo **AMN=Y...**  
Le funzioni ausiliarie (F.A.) per la rotazione, arresto, velocità, cambio gamma e orientamento sono quelle standard e cioè: M03-M04-M05-Sxxxx-M40-M41-M42-M43-M44-M45-M19.
- Il mandrino **SECONDARIO** gestito dal CNC è configurabile solo se definito il mandrino **principale CNC**. In questo caso il mandrino **secondario** può essere configurato per funzioni diverse in relazione al sistema installato :

Sistema installato	Funzione
<b>MILL</b> (CNC Fresa)  Segnali PLC: <b>OSYSMILL=1</b> <b>OSYSTURN=0</b>	<b>FRESATURA SECONDARIO</b> Definito in AXS.TAR dal campo <b>AMN=S...</b>
	<b>TORNITURA SECONDARIO</b> Definito in AXS.TAR dal campo <b>AMN=T...</b> Tale configurazione è possibile solo se è presente l'opzione OPZ_FRETOR .
<b>LHATE</b> (CNC Tornio)  Segnali PLC: <b>OSYSMILL=0</b> <b>OSYSTURN=1</b>	<b>TORNITURA SECONDARIO</b> Definito in AXS.TAR dal campo <b>AMN=T...</b>
	<b>FRESATURA SECONDARIO</b> Definito in AXS.TAR dal campo <b>AMN=S...</b> Tale configurazione è possibile solo se è presente l'opzione OPZ_TORFRE .

Le funzioni F.A. per la rotazione, arresto, velocità, cambio gamma e orientamento sono quelle standard viste per il mandrino principale. Anche le modalità di controllo sono quelle previste per il mandrino principale: G94, G95 e G96.

- La presenza o meno dell'**Utensile Rotante** è configurabile con valori > 0 nelle variabili **##106**, **##107** e **##108**, definite nel file COSTUSER (per i valori di default e la



sintassi, vedere la sezione relativa nel file COST ed il paragrafo “Rotazione” del presente Capitolo).

Nel file COSTUSER deve essere definito anche la variabile **##110**, per comunicare al CNC l'indirizzo della F.A. (seconda o terza) utilizzata dall'utensile rotante per la gestione della velocità dell'utensile stesso.

- ##110 = -1** Non usata né seconda né terza F.A. (Default)
- = 2** Usata seconda F.A. (normalmente H)
- = 3** Usata terza F.A. (normalmente E)

Il valore scritto nei rispettivi # definisce la Mxx utilizzata per la rotazione oraria, antioraria e lo stop dell'utensile rotante. Il PLC deve gestire i segnali **IRUR** e **ISRUR**. La modalità di lavoro ammessa è G94.

Se non esiste utensile rotante si ha **##106=##107=##108=-1** .

Le stesse F.A. “M” utilizzate per la rotazione oraria/antioraria/stop utensile rotante devono essere definite nel file di taratura **FA.TAB** come funzioni ausiliarie anticipate o posticipate ed organi riposizionabili. Questa definizione serve al CNC per un corretto funzionamento del riposizionamento dell' utensile rotante.

La selezione del **mandrino attivo** è fatta tramite le istruzioni G633, G634, G635 .

**G633** Attiva il mandrino principale. Comunicato al PLC con **OCODSMN=0**

Le F.A. standard relative al mandrino sono rivolte al mandrino principale.

**G634** Attiva il mandrino secondario. Comunicato al PLC con **OCODSMN=1**

Le F.A. standard relative al mandrino sono rivolte al mandrino secondario. In funzione della definizione fatta in AXS.TAR del mandrino secondario questa selezione può essere:

Con **<AMN=S...>** si seleziona mandrino secondario di fresatura

Con **<AMN=T...>** si seleziona mandrino secondario di tornitura

**G635** Attiva l'utensile rotante. Comunicato al PLC con **OCODSMN=2**.

Le F.A. standard relative al mandrino **non** sono permesse.

In effetti le funzioni G63x suddette sono delle macro (scritte con le normali regole di programmazione dei Part-Program) e possono essere quindi modificate dal costruttore della macchina secondo le proprie esigenze. Al loro interno, queste funzioni richiamano le istruzioni elementari LIP **<SMN:...>** che eseguono il cambio mandrino attivo:

**<SMN:P>** Attiva il mandrino principale.

**<SMN:S>** Attiva il mandrino secondario.

**<SMN:U>** Attiva l'utensile rotante.

I cicli fissi e/o macro che richiedono mandrino in rotazione, verificano che il mandrino in rotazione sia solo quello attualmente selezionato, altrimenti è emesso allarme specifico.

Il mandrino CNC attualmente selezionato è segnalato nella finestra di visualizzazione quote assi tramite una diversa icona abbinata all'asse mandrino.

Nella finestra “Stato CNC” viene segnalato il mandrino attivo con le sigle :

**MNP** = mandrino principale

**MNS** = mandrino secondario

**UTR** = utensile rotante



**Note:**

1. Le istruzioni G633, G634, G635 hanno effetto anche con CNC in modo "lettura"
2. All'accensione, il CNC attiva il mandrino principale (file AXS.TAR campo AMN=Y;....) Qualora sia richiesto che il CNC si accenda con lo stesso mandrino selezionato allo spegnimento, è necessario che il PLC scriva l'apposito codice nella variabile IMNDATT (vedere più avanti al paragrafo "segnali del PLC").

Il mandrino principale e quello secondario possono essere dotati di trasduttore di misura oppure esserne sprovvisti. Il sistema di misura, se presente, è di tipo ottico incrementale.

In questo capitolo, quando si parla del Mandrino ci si riferisce sia al mandrino principale che al secondario; quando ci siano differenze, queste saranno indicate esplicitamente.

I mandrini possono essere gestiti in vari modi:

- come assi del CNC con trasduttore
- come assi del CNC senza trasduttore
- tramite il PLC attraverso convertitori D/A su schede assi
- tramite il PLC attraverso hardware esterno al CNC.

### 5.1.1 Parametri di taratura del mandrino

Se il mandrino è un asse del CNC, le sue caratteristiche sono definite nel file di taratura AXS.TAR (vedi "Manuale Tarature" 720P385). Il numero d'ordine contenuto in questo file (abbinato a quanto scritto in SWCNF.TAR) definisce la corrispondenza tra tarature software e circuiteria hardware.

Quando il mandrino è un asse del CNC, l'aggancio in velocità, in spazio e il cambio gamma sono gestiti dal CNC attraverso appositi sottoprogrammi. Le tarature relative sono definite nel sottoprogramma COST, come riportato nel paragrafo 5.2.4.2. Il PLC deve gestire solamente l'abilitazione ed il comando di cambio delle gamme con relativa risposta al CNC. La routine di libreria PLC **aggmnd()** risolve in maniera facile e sicura la gestione delle varie modalità di funzionamento del mandrino. L'uso della routine suddetta è spiegato più avanti in questo capitolo. Se il mandrino è gestito dal PLC, la gestione è completamente a carico del programma PLC.

Qui di seguito, sono riassunti i vari files che contengono tarature o procedure riguardanti il mandrino.

SWCNF.TAR (1)	Configurazione Hardware.
AXS.TAR (1)	Parametri asse-mandrino
COST (1)	Parametri per orientamento e cambio gamma
COSTUSER (1)	Personalizzazione dei parametri di COST
M15; M40 ÷ M45 (2)	Per la gestione del cambio gamma
M03, M04, M05, M13, M14, M19 (2)	Aggancio mandrino in velocità e in spazio
SPEED (2)	Funz.Aus. S e gestione cambio gamma
CGAMMA (2)	Gestione cambio gamma

- 1) La funzione di questi files ed il relativo contenuto sono dettagliatamente spiegati nel "Manuale Tarature" 720P385.
- 2) Questi files sono predisposti da ECS per gestire vari tipi di mandrini e cambi gamma; vanno modificati solo nei rari casi non previsti. Si prega di contattare l'Ufficio Applicazioni ECS quando sorgesse tale necessità.



### 5.1.2 Sinonimi utilizzati per il mandrino

Per la gestione del mandrino tramite il PLC, sono presenti alcuni segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

**{W}** Nella prima colonna, il sinonimo scritto in **GRASSETTO** è quello da preferire; l'eventuale altro sinonimo è comunque valido ed è stato mantenuto per compatibilità con la serie D.

Quando si parla del Mandrino ci si riferisce sia al mandrino principale sia secondario; quando ci siano differenze, queste saranno indicate esplicitamente.

Nei sinonimi dove compare il carattere "#", occorre sostituire ad esso il numero d'ordine dell'asse al quale il segnale è riferito (l'asse dichiarato mandrino, con campo AMN=Y o S o T, in AXS.TAR). Tale numero, secondo il tipo di CNC, sarà compreso tra 0 e 2 (1801, 1802), tra 0 e 5 (2801) o tra 0 e 15 (4801, 4802).

Esempi:

**OSOK3** Velocità mandrino (asse 3) raggiunta.

**OSOK5** Velocità mandrino (asse 5) raggiunta.

<p><b>IBLAXMND</b> IBL</p>	<p><b>PLC → CNC</b></p>	<p><b>Richiesta di BLOCCO CNC.</b> Posto a 1 impone velocità 0 a tutti gli assi CNC ed al mandrino ed impedisce il proseguimento del Part Program. Posto poi a 0, permette la ripartenza di assi, mandrino e Part Program, che avverrà dopo la pressione del tasto [REL] di annullo (RELEASE) del BLOCCO CNC. Mettere IBLAXMND=1 transitoriamente se si vuole bloccare sia assi CNC che il mandrino; esegue le stesse funzioni del pulsante HOLD della tastiera. Mettere IBLAXMND=1 fisso se si vuole rendere inattivo il pulsante [REL].</p>
<p><b>OHLMND</b></p>	<p><b>CNC → PLC</b></p>	<p><b>MANDRINO in HOLD (BLOCCO).</b> Il CNC passa in questo stato su richiesta dell'operatore (a seguito della pressione del pulsante HOLD) o su richiesta del PLC (IBLAXMND) o per altre situazioni rilevate dal CNC stesso; contemporaneamente viene forzato a 1 anche OHLAX. Il PLC deve fermare le "azioni" in svolgimento sulla Macchina Utensile (M.U.) tenendo presente che il CNC provvede automaticamente a portare a zero la velocità del mandrino (e di tutti gli assi CNC, mantenendoli fermi in ciclo di spazio). All'atto dello sblocco dato dal pulsante [REL], il CNC riavvierà il mandrino e le "azioni" interrotte dovranno essere riprese. Lo stato di OHLMND <u>NON</u> cambia con il passaggio AUTO→MDI o MDI→AUTO. OHLMND = 0 all'accensione o dopo RESET o dopo il comando di sblocco dato dal pulsante [REL] solo se l'ingresso IBLAXMND è=0.</p>
<p><b>OHLD</b></p>	<p><b>CNC → PLC</b></p>	<p><b>CNC in HOLD (BLOCCO ASSI e MANDRINO).</b> Il CNC passa in questo stato quando <u>entrambi i segnali OHLMND e OHLAXS sono = 1</u> su richiesta dell'operatore (a seguito della pressione del pulsante HOLD) o su richiesta del PLC (IBLAXMND) o per altre situazioni rilevate dal CNC stesso. OHLD = 0 all'accensione o dopo RESET o dopo il comando di</p>



## MANDRINO

		sblocco dato dal pulsante [REL] o dopo cambio modo, purché l'ingresso IBLAXMND sia = 0. Con <b>OHLD= 1</b> , gli assi ed il mandrino vengono portati a velocità 0. All'atto dello sblocco dato dal tasto [REL], il CNC riavvierà gli assi ed il mandrino. L'arresto e la partenza del mandrino e degli assi avvengono contemporaneamente.
<b>NEWSMN</b>	<b>CNC → PLC</b>	Nuova Selezione Mandrino Messo a 1 dal CNC, per almeno un giro del PLC, quando viene eseguita l'istruzione <SMN:...>. Il PLC può interrompere il proseguimento del Part-Program ponendo <b>FATSMN= 0</b> . Il valore della <SMN:...> è presente su OCODSMN.
<b>OCODSMN</b> (unsigned char)	<b>CNC → PLC</b>	Codice mandrino richiesto <b>0</b> = <SMN:P> (G633) Mandrino Principale (AMN=Y in AXS.TAR) <b>1</b> = <SMN:S> (G634) Mandrino Secondario (AMN=S o T) <b>2</b> = <SMN:U> (G635) Utensile Rotante Il valore è aggiornato con NEWSMN = 1.
<b>FATSMN</b>	<b>PLC → CNC</b>	Fatto per Nuova Selezione Mandrino <SMN: . . .>. Posto a 0 quando NEWSMN= 1, interrompe il proseguimento del Part-Program, ed esempio per dar tempo al PLC di fare le azioni necessarie al cambio di mandrino attivo. Normalmente non usato; forzato a 1 al pon (power-on).
<b>IMNDATT</b> (char)	<b>PLC → CNC</b>	Codice mandrino inserito <b>=0</b> Mandrino Principale (AMN=Y in AXS.TAR) <b>=1</b> Mandrino Secondario <b>=2</b> Utensile Rotante <b>Vedi nota a fine tabella</b>
<b>NEWS</b>	<b>CNC → PLC</b>	Nuova Funzione Ausiliaria S (Speed) Messo a 1 dal CNC, per almeno un giro del PLC, quando viene programmata una F.A. S. Il PLC può interrompere il proseguimento del Part-Program ponendo <b>FATTOS= 0</b> . Il valore della S è presente su OTDS.
<b>OTDS</b> (long)	<b>CNC → PLC</b>	Valore della F.A. S programmata (doppia word ). In tempi diversi, sulla stessa word sono presenti altri dati; il valore è garantito corrispondere alla F.A. S solo con NEWS=1.
<b>FATTOS</b>	<b>PLC → CNC</b>	Fatto per Funzione Ausiliaria S (Speed). Posto a 0 quando NEWS= 1, interrompe il proseguimento del Part-Program, ed esempio per dar tempo al PLC di fare un cambio gamma mandrino. Normalmente non usato; forzato a 1 al pon (power-on).
<b>AIOVS</b> (unsigned char)	<b>CNC → PLC</b>	Valore Override S% (0...255) proveniente dalla tastiera del CNC. È normalmente usato per l'override Speed del Mandrino. Vedi OVERRIDE al Cap. 2.
<b>IOVS</b> (unsigned char)	<b>PLC → CNC</b>	Valore Override Speed Mandrino in % (0...255) per il CNC. Vedi OVERRIDE al Cap. 2.
<b>IOVSS</b>	<b>PLC → CNC</b>	Valore Override Speed Mandrino Secondario, in % (0...255), per



(unsigned char)		il CNC. Vedi OVERRIDE al Cap. 2.
<b>OG33</b>	<b>CNC → PLC</b>	Movimento di Filettatura (G33, G34, G35) in esecuzione. Per i particolari, vedere le note al paragrafo "Filettatura".
<b>OSTM##</b>	<b>CNC → PLC</b>	Velocità mandrino (asse #) minore o uguale a quella definita in AXS.TAR: MFR=C1;C2 Velocità mandrino fermo e <b>stop mandrino</b> . Per i particolari, vedere le note al paragrafo "Stop mandrino".
<b>OSOK#</b> <b>OSOK</b> <b>OSOKS</b>	<b>CNC → PLC</b>	Velocità mandrino (asse #) raggiunta. OSOK#= 1: la velocità comandata per il mandrino è uguale a quella desiderata; non è quindi attiva nessuna variazione del riferimento in uscita dal CNC. Se il PLC usa la routine <b>aggmnd()</b> , usare <b>OSOK OSOKS</b> (per mandrino Principale e Secondario).
<b>OSAMAC#</b>	<b>CNC → PLC</b>	Mandrino (asse#) in fase di accelerazione/frenatura. Il PLC può usare questo segnale per comandare la sospensione allarme fuga ISAM# se il mandrino segue con difficoltà la rampa di accelerazione impostata in taratura. Questo segnale è usato direttamente dalla routine <b>aggmnd()</b> .
<b>OBCG#</b> <b>VELCG</b> <b>VELCGS</b>	<b>CNC → PLC</b>	Velocità di cambio gamma mandrino (asse #) raggiunta. Tale segnale indica che è in esecuzione un'istruzione del tipo G991, G992, G993 per il cambio gamma con mandrino in rotazione o in oscillazione e che la velocità settata in COST o COSTUSER è stata raggiunta. Il PLC deve testare tale segnale come sicurezza prima di inserire fisicamente la gamma voluta. Nel caso di mandrino senza trasduttore tale segnale può essere VERO anche con il mandrino che fisicamente non esegue ciò che il riferimento comanda; in questo caso si consiglia di testare, se presente, anche il segnale di mandrino a velocità bassa fornito dall'azionamento, o quello di mandrino fermo nel caso di oscillazione. Si consiglia di testare sempre tutti i segnali di sicurezza forniti dall'azionamento prima di eseguire fisicamente il cambio gamma. Se il PLC usa la routine <b>aggmnd()</b> , testare <b>VELCG VELCGS</b> (per mandrino Principale e Secondario). Per i particolari, vedere le note al paragrafo 5.2.4.2 .
<b>IGM1</b> <b>IGM2</b> <b>IGM4</b>	<b>PLC → CNC</b>	Codice della gamma mandrino attualmente inserita Per Mandrino Principale Per i particolari, vedere le note a fine tabella.
<b>IGM1S</b> <b>IGM2S</b> <b>IGM4S</b>	<b>PLC → CNC</b>	Codice della gamma mandrino attualmente inserita. Per Mandrino Secondario Per i particolari, vedere le note a fine tabella.
<b>IRME</b>	<b>PLC → CNC</b>	Mandrino "esterno" in rotazione. Il mandrino gestito dal PLC attraverso un convertitore D/A, oppure tramite hardware esterno alle carte assi del CNC, è in rotazione. Il PLC deve rendere VERO tale input se il mandrino è stato messo in rotazione e ruota.





		I segnali IRME e ISRME servono per verificare lo stato del mandrino, prima di eseguire un ciclo fisso. <i>NON utilizzare se il mandrino è un asse (con o senza encoder).</i>
<b>ISRME</b>	<b>PLC → CNC</b>	Senso di rotazione del Mandrino "esterno". ISRME= 0 : rotazione antioraria. ISRME= 1 : rotazione oraria. Valido solo se IRME= 1.
<b>IRMES</b>	<b>PLC → CNC</b>	Come IRME, ma per mandrino secondario
<b>ISMES</b>	<b>PLC → CNC</b>	Come ISRME, ma per mandrino secondario
<b>IRUR</b>	<b>PLC → CNC</b>	Come IRME, ma per utensile rotante
<b>ISRUR</b>	<b>PLC → CNC</b>	Come ISRME, ma per utensile rotante
<b>ISAM#</b> <b>RISAMP</b> <b>RISAMS</b>	<b>PLC → CNC</b>	Richiesta sospensione dell'allarme fuga mandrino asse #. Richiesta sospensione dell'allarme fuga mandrino principale. Richiesta sospensione dell'allarme fuga mandrino secondario. Per i particolari, vedere la routine <b>aggmnd()</b> ed il paragrafo "allarmi mandrino".

#### NOTA per asse mandrino:

Quando il mandrino è un asse (è definito in AXS.TAR con il campo AMN=Y o S o T) sono validi anche tutti gli altri segnali relativi appunto agli assi; per un approfondimento delle varie situazioni, si rimanda ai capitoli relativi.

#### NOTA su IMNDATT:

Il PLC, dopo aver eseguito eventuali predisposizioni necessarie per il nuovo mandrino da rendere attivo, deve scrivere in IMNDATT il valore opportuno e settare FATSMN.

Il PLC può NON modificare le sue predisposizioni e lo stato di IMNDATT in modo Lettura, mettendo comunque FATSMN=1; il CNC riemetterà il comando di selezione mandrino al primo passaggio in Esecuzione (se IMNDATT diverso da OCODSMN)

In modo Esecuzione, quando il PLC rende il fatto FATSMN=1 il CNC controlla la congruenza fra il codice presente su IMNDATT ed il comando dato OCODSMN; se tale verifica non è OK, il CNC emette un allarme specifico.

All'accensione e durante la fase di RESET, il CNC memorizza il mandrino attualmente inserito sulla macchina in funzione del codice presente su IMNDATT.

Qualora sulla macchina non esista nessun organo elettromeccanico da attuare per il cambio mandrino, nel PLC è sufficiente scrivere l'equazione IMNDATT=OCODSMN.

E' compito del PLC, in quelle macchine ove è richiesto, impedire la rotazione contemporanea di più mandrini (e segnalarlo con un allarme specifico).

#### NOTA su IGMn e IGMnS

Con la programmazione di M40÷M45, quando il PLC rende il fatto alla F.A., il CNC controlla la congruenza fra i segnali IGMn o IGMnS e la Mxx di cambio gamma emessa; se tale verifica non è OK, il CNC emette un allarme specifico.



I codici debbono corrispondere alla tabella seguente:

GAMMA	IGM4 – IGM4S	IGM2 – IGM2S	IGM1 - IGM1S
M40	0	0	0
M41	0	0	1
M42	0	1	0
M43	0	1	1
M44	1	0	0
M45	1	0	1

## 5.2 MANDRINO ASSE DEL CNC

Deve essere definito come asse in AXS.TAR (campo AMN=Y o AMN=S o AMN=T); per i parametri necessari ed il relativo significato, si rimanda al "Manuale Tarature" 720P385.

Il mandrino può assumere tre stati:

**FRV** = Controllato in velocità (M03 / M04).

**FRE** = Asse indipendente, per l'orientamento su una posizione fissa (M19).

**CNC** = Asse interpolante, per il posizionamento ad un angolo desiderato o per la maschiatura "rigida".

Infatti è l'unico asse che definito "FRE" in taratura può divenire CNC (interpolante) da programma attraverso l'istruzione <CNC:sigla\_asse\_mandrino>.

Il mandrino viene pilotato in VELOCITA', quando è nello stato di Asse Indipendente (FRE: agganciato in punto-a-punto e con OBVV#=1); in questa situazione, il CNC elabora il valore del segnale di riferimento tenendo conto di:

- Gamma attualmente inserita
- Velocità massima del motore mandrino (numero di GIRI MAX)
- Valore S programmato
- Override della speed

### 5.2.1 Aggancio mandrino nelle varie modalità CNC

L'asse-mandrino è normalmente agganciato in punto-a-punto (0BAG#=1) e velocità (OBVV#=1) di modo tale che se venisse ruotato a mano non intervenga l'allarme fuga asse. Il mandrino è agganciato in modo DIVERSO dal punto-a-punto e velocità in questi casi:

- con CNC in modo JOG, è in contouring; se all'ingresso nel modo JOG il mandrino è in rotazione, questo rimane agganciato in punto-a-punto e velocità ed in rotazione.
- in modo JOG il mandrino si può muovere solo in JOG di velocità o spazio alla FEED programmata e non in JOG rapido (sia questo di velocità o di spazio).
- quando il mandrino è agganciato in contouring (modo JOG), non è possibile muoverlo con mezzi esterni, altrimenti il CNC dà allarme fuga asse. Se occorre muoverlo, il PLC deve sganciarlo dalla condizione CNC.



- con comando di orientamento mandrino, il mandrino viene orientato rimanendo in modo velocità, a posizione raggiunta viene tolto il modo velocità e l'asse rimane in punto-a-punto.
- con l'istruzione <CNC:...>, il mandrino viene agganciato in contouring. Con una successiva istruzione <FRE:...> viene agganciato in punto-a-punto e velocità.
  - \* con l'istruzione M05, se il mandrino è in punto-a-punto e non velocità viene messo in punto-a-punto e velocità.
  - \* si ricorda che con mandrino agganciato in punto-a-punto e velocità l'asse può muoversi per deriva se rimane sempre abilitato. E` quindi compito del PLC abilitare l'azionamento mandrino solo quando c'è M03 o M04.

## 5.2.2 ROUTINES A DISPOSIZIONE DEL PLC

### 5.2.2.1 Routine "AGGMND"

Chiamata: **aggmnd()**;

Scopo: Gestisce l'abilitazione / disabilitazione dell'asse-mandrino.

Parametri in Ingresso :

**RABMND = 1** Richiesta Abilitazione Mandrino

Tale richiesta sarà resa vera dal PLC in presenza di tutte le condizioni (ausiliari inseriti, azionamento OK, ecc.) necessarie per abilitare il mandrino.

**BLCMND = 1** Mandrino Bloccato

**RISAMP = 1** Richiesta Sospensione Allarme Fuga Mandrino

Sono presenti anche i seguenti parametri per il mandrino secondario:

**RABMNS = 1** Richiesta Abilitazione Mandrino Secondario

**BLCMNS = 1** Mandrino Secondario Bloccato

**RISAMS = 1** Richiesta Sospensione Allarme Fuga Mandrino Secondario

I segnali **BLCMND** e **BLCMNS** devono rimanere ad "1" fintanto che il relativo mandrino non può accettare comandi di rotazione o posizionamento ( ad es. appena inserita la potenza e/o appena abilitato). Quando esso va a zero, il CNC può dare riferimento all'asse-mandrino.

Per un corretto funzionamento della routine, il mandrino deve essere definito bloccabile ( vedi <BLC:<%V40xx>> in COST).

Avendo definito l'asse-mandrino bloccabile, è consigliabile impostare un ritardo nel bloccare l'asse dopo che esso è "arrivato" in posizione. Se tale ritardo non è definito, il mandrino sarà immediatamente disabilitato appena arrivato in posizione.

Per far questo occorre caricare, con la funzione **lotime(...)**, il ritardo desiderato nei temporizzatori appositamente previsti:

```
lotime( #, (float)xx );
```

# = numero del temporizzatore, che corrisponde al numero dell'asse-mandrino

xx = tempo di ritardo espresso in secondi (tipico = 0.3 sec.)



---

Durante la fase di accelerazione/decelerazione, la routine sospende automaticamente l'allarme fuga mandrino (mettendo ISAM# = 1 quando OSAMAC# = 1). Se non si desidera tale gestione, non attivare la richiesta tramite i segnali RISAMP e RISAMS e gestire autonomamente i segnali ISAM# dopo la chiamata ad aggmnd().

La routine, una volta ricevuto la richiesta di abilitazione, verifica, per ciascun asse, se esso è in gestione al CNC (OBAG#=1) e se non è in allarme (OBOK#=1; OBALAX#=0); in caso affermativo fornisce in uscita al PLC il consenso ad abilitare l'azionamento mandrino appena il CNC emette un comando di movimento (M03, M04, M19, cambio gamma, posizionamento). Il riferimento di velocità sarà dato se **BLCMND = 0** (o **BLCMNDS = 0**).

Parametri di uscita:

#### **ABMND = 1** Comando di Abilitazione Mandrino

Quando si ha l'uscita ABMND = 1, il PLC deve abilitare l'azionamento dell'asse-mandrino.

All'arrivo in quota dell'asse-mandrino "bloccabile" il segnale **ABMND** ha uno dei seguenti comportamenti:

- ABMND = va a 0 istantaneamente (se non definito un ritardo)
- ABMND = va a 0 dopo il ritardo impostato attraverso la funzione **lotime(...)**

Il secondo approccio è più corretto in quanto lascia un po' di tempo all'asse-mandrino per recuperare un eventuale errore di posizionamento prima di venire disabilitato.

Con un comando M19, la routine mantiene ABMND=1 fino ad un successivo comando M03, M04, M05 o RESET.

#### **OSOK = 1** Velocità Mandrino Raggiunta

Quando OSOK=0, il mandrino è in fase di adeguamento della velocità a quella desiderata

Quando OSOK=1, non ci sono variazioni in corso (rampe di accelerazione / frenata).  
Quando è presente il trasduttore del mandrino, la velocità effettiva è dentro la soglia percentuale definita col parametro SOK in AXS.TAR (se diversa da zero).

#### **ACG = 1** Cambio Gamma in Corso

Quando ACG=1, il mandrino è in fase di cambio gamma (sta eseguendo una F.A. subroutine M40:-M45).

#### **VELCG = 1** Velocità di Cambio Gamma Raggiunta

Quando VELCG=1, il mandrino è in fase di cambio gamma e la velocità ha raggiunto quella desiderata. Può essere usato per abilitare l'esecuzione effettiva del cambio gamma meccanico. VELCG permane a 1 anche quando il mandrino sta oscillando e viene rimesso a 0 a cambio gamma terminato.

Ci sono anche i seguenti parametri per il mandrino secondario:

#### **ABMNDS = 1** Comando di Abilitazione Mandrino Secondario

#### **OSOKS = 1** Velocità Mandrino Secondario Raggiunta

#### **ACGS = 1** Cambio Gamma in Corso sul Mandrino Secondario

#### **VELCGS = 1** Velocità di Cambio Gamma Raggiunta



I parametri di ingresso e uscita della routine sono tutti di tipo bit; i relativi sinonimi sono definiti in DF\_SYN.INC e sono allocati nell'area dati B (512 word riservate ad appoggi per le routines di libreria PLC fornite da ECS).

```
/*-----  
ESEMPIO DI GESTIONE MANDRINO CON USO DELLA ROUTINE AGGMND  
-----*/  
if (pon)  
{  
  /* SETTA TEMPORIZZATI AL pon (Power ON) */  
  lotime(3,(float)2.0);  
  /* SETTA RITARDO DI 2 SEC PER ASSE_MANDRINO DI ORDINE 3 */  
}  
RISAMP= !RBMND || OHLD || ..... ;  
/* RICHIESTA SOSPENSIONE FUGA ALLARME MANDRINO */  
RBMND= MUON && POTMND && AZMNDOK && ... ;  
/* CONSENSO ABILITAZIONE MANDRINO */  
aggmnd();  
/* CHIAMATA ROUTINE AGGANCIO MANDRINO */  
ABILMND= ABMND && (ABILMND && (AM03 || AM04) || ACGOKR);  
/* ABILITAZIONE MANDRINO */
```

### 5.2.2.2 Routine "SetMovMnd"

Chiamata: **SetMovMnd(mandr; senso);**

Scopo: predisporre il senso di movimento del mandrino principale/secondario con comando di orientamento M19, partendo da mandrino fermo.

Parametri in Ingresso :

<b>mandr</b> = (unsigned short)	Mandrino da predisporre. <b>0</b> : mandrino principale <b>1</b> : mandrino secondario
<b>senso</b> = (unsigned short)	Senso di moto del mandrino per orientarsi. <b>0</b> = Movimento per la via più breve <b>1</b> = Movimento per quote positive (crescenti) <b>2</b> = Movimento per quote negative (decescenti)

Parametro di uscita:

**esito** = (short ) valore di ritorno della routine . Può assumere uno dei seguenti valori:

<b>0</b>	Successo – predisposizione OK
<b>-1</b>	Insuccesso – Mandrino NON definito

### 5.2.2.3 Routine "TstMovMnd"

Chiamata: **TstMovMnd(mandr);**

Scopo: Chiede al CNC il tipo di movimento predisposto con la funzione SetMovMnd().

Parametro in Ingresso :

<b>mandr</b> = (unsigned short)	Mandrino da interrogare. <b>0</b> : mandrino principale <b>1</b> : mandrino secondario
---------------------------------	--

Parametro di uscita:



**esito** = (short ) valore di ritorno della routine . Può assumere uno dei seguenti valori:

- <0    Insuccesso – Mandrino NON definito
- 0     Successo – movimento predisposto per via più breve
- 1     Successo – movimento predisposto per quote crescenti
- 2     Successo – movimento predisposto per quote decrescenti

### 5.2.3 Gestione cambio scala mandrino

Un azionamento mandrino "moderno", permette un'ampia regolazione di velocità pur con un ingresso di riferimento analogico di valore massimo  $\pm 10$  Volt. In queste situazioni, a velocità relativamente basse corrispondono tensioni di comando molto basse che possono essere pesantemente influenzate da disturbi o differenze di masse. Se l'azionamento permette di definire due diverse scale di rapporto riferimento/velocità, il CNC ECS permette di gestirne la funzione.

La libreria del PLC mette a disposizione le funzioni necessarie allo scopo.

#### 5.2.3.1 on\_refmnd : attiva la funzione di cambio scala riferimento

Chiamata: **esito = on\_refmnd(fatt\_scala, n\_word\_t, bit\_word\_t, ritardo, vel1, vel2 )**

Scopo: attiva la funzione di cambio scala riferimento e ne definisce i parametri di funzionamento.

Parametri in Ingresso :

**fatt\_scala** = (unsigned short) Coefficiente moltiplicativo del riferimento da usare a basse velocità. Deve essere un valore  $> 0$  (tipico = 10)

**n\_word\_t** = (unsigned short) Numero della word di output **T** dove è presente l'uscita (collegamento CNC $\Rightarrow$ azionamento) usata per attivare il cambio scala.

**bit\_word\_t** = (unsigned short) Numero del bit, nella word T suddetta, corrispondente all'uscita (collegamento CNC $\Rightarrow$ azionamento) usata per attivare il cambio scala. Indica, assieme a n\_word\_t, l'output fisico collegato con l'azionamento. Deve essere un valore valido (0 :- 15).

**ritardo** = (unsigned short) Numero (in **TCYC** Servo Loop Cycle Time; unità di tempo del SW/ASSI) che definisce il ritardo tra la commutazione del segnale di output e l'adeguamento del segnale di riferimento. Considerando ad esempio un tempo di ritardo di 5ms nell'azionamento + 2ms per la commutazione del relè ripetitore, è necessario considerare un ritardo di circa 7ms. **Con un TCYC=4ms, il valore più vicino di "ritardo" quindi diventa 7/4 = 2.**

**vel1** = (float) Prima soglia di velocità (in giri/minuto) che il CNC deve controllare.

**vel2** = (float) Seconda soglia di velocità (in giri/minuto) che il CNC deve controllare.

Parametro di uscita:

**esito** = (short ) valore di ritorno della routine. Può assumere uno dei seguenti valori:

- 2    Asse-mandrino senza trasduttore (non è possibile leggere la velocità)
- 1    Non esiste l'asse-mandrino (non è definito in AXS.TAR)
- 0     Successo – funzione attivata. Il controllo sulle soglie di velocità ed il calcolo del fattore di scala è stato attivato. Il riferimento sarà moltiplicato per il fattore di scala solo quando il bit della word di output T sarà ON.
- | 1    Fattore di scala errato ( negativo o =0)



- 
- 
- 2 Numero di word T errato ( non è definita in SWCNF.TAR)
  - 3 Numero di bit T errato ( maggiore di 15)
  - 5 Prima soglia errata ( minore o uguale a 0; oppure maggiore della seconda soglia)
  - 6 Seconda soglia errata ( minore o uguale a 0; oppure il prodotto vel2 \* fatt\_scala è maggiore della velocità massima del mandrino e quindi non raggiungibile)

### 5.2.3.2 off\_refmnd : disattiva la funzione di cambio scala riferimento

Chiamata: **off\_refmnd( )**

Scopo: disattiva la funzione di cambio scala riferimento precedentemente attivata con on\_refmnd(...); normalmente non usata.

### 5.2.3.3 tst\_refmnd : testa la velocità mandrino attuale

Chiamata: **esito = tst\_refmnd( )**

Scopo: legge dinamicamente in quale fascia di velocità sta girando attualmente il mandrino.

Parametro di uscita:

**esito** = (unsigned short ) valore di ritorno della routine . Può assumere uno dei seguenti valori:

- 0 La velocità mandrino è superiore sia alla soglia vel1 sia alla soglia vel2
- 1 La velocità mandrino è intermedia tra la soglia vel1 e la soglia vel2
- 2 La velocità mandrino è inferiore sia alla soglia vel1, sia alla soglia vel2

#### Nota:

Per il mandrino Secondario esistono le corrispondenti funzioni:

**on\_refmnds(), off\_refmnds(),tst\_refmnds()**

#### Uso:

Al pon (power-on) l'uscita T deve essere messa a zero e deve essere chiamata la funzione on\_refmnd(...) con i parametri desiderati.

Durante il giro della logica deve essere chiamata la funzione tst\_refmnd():

- se la funzione ritorna 2 l'uscita T deve essere messa ad 1.
- se la funzione ritorna 0 l'uscita T deve essere messa a 0.
- se la funzione ritorna 1 l'uscita T non deve cambiare stato (effetto isteresi).

```
/*-----  
ESEMPIO DI GESTIONE CAMBIO SCALA MANDRINO  
CON USO DELLE ROUTINE on_refmnd() E tst_refmnd()  
-----*/  
if (pon || ORST && ATT00)  
{  
ret= on_refmnd(10,0,9,2,200,700);  
/* ATTIVA GESTIONE CAMBIO SCALA */  
if (ret != 0)  
{  
ATT00= 1;  
/* Errore in fase di attivazione della */
```



```
        /* funzione cambio scala mandrino */
    }
    else
    {
        ATT00= 0;
    }
}
ret= tst_refmnd();
/* TESTA SOGLIE DI VELOCITA' MANDRINO */
switch (ret)
{
    case 0:
        CSCALA= 0;
        /* RESET CAMBIO SCALA */
        break;

    case 2:
        CSCALA= 1;
        /* RICHIESTA CAMBIO SCALA */
        break;
}
/*-----*/
```

#### 5.2.4 Gestione gamme mandrino

Si definiscono:

- GAMMA DIRETTA** quella gamma che consente la maggiore velocità di rotazione del mandrino a pari velocità del suo motore (in AXS.TAR, nel suo campo GMn, avrà valore 1).
- GAMMA FOLLE** quella o quelle gamme che comportano mandrino fermo, anche se gira il suo motore (in AXS.TAR avrà GMn=0).
- GAMMA** quelle ulteriori gamme presenti sulla catena cinematica del mandrino. Il valore da inserire in taratura deve essere calcolato come segue: Giri/min. massimi in "GAMMA" / Giri/min. massimi del mandrino in "GAMMA DIRETTA"; è quindi un valore minore di 1. Il CNC gestisce fino a 6 gamme diverse.

Il cambio gamma è risolto attraverso i sottoprogrammi indicati al punto 5.1.1 .

Il costruttore della M.U. può adeguarlo alla propria macchina attraverso il file COSTUSER. Il CNC è in grado di gestire il cambio gamma sia facendo ruotare il mandrino ad una velocità e senso di rotazione prefissata, sia mettendolo in oscillazione con un delta quota impostato sia mantenendolo fermo.

Per i mandrini di fresatura (Fresa o Tornio), è possibile scegliere se il cambio gamma debba essere fatto automaticamente in base alla Speed programmata, oppure solo in base alla  $M40 \div M45$ . La scelta del tipo di cambio gamma voluto viene fatta definendo nel file AXS.TAR il campo AMN: (vedi MANUALE TARATURE)

Al PLC non interessa quale tipo di cambio gamma sia stato definito in taratura (S o M), in quanto il CNC, attraverso i sottoprogrammi suddetti, emette sempre le F.A.  $M40 \div M45$ .





Le istruzioni speciali riservate al mandrino ed utilizzate nelle subroutine suddette, sono riportate nel manuale di Programmazione Avanzata, al quale si rimanda per un'eventuale approfondimento, normalmente non necessario.

In particolare, per testare se attualmente è attivo il mandrino Principale o Secondario, può essere usato:

```
N20 <IFF<%V2304 & '00FF'>:N2000;N100;N2000>  
{Selezionato Mandrino Secondario? Si-> N2000 No-> N100}
```

Per testare se attualmente è attivo l'utensile rotante, può essere usato:

```
N20 <IFF<%V2304 & 'FF00'>:N2000;N100;N2000>  
{Selezionato Utensile Rotante? Si-> N2000 No-> N100}
```

#### 5.2.4.1 Mandrino Monogamma

Qualora l'asse-mandrino non abbia cambio gamma è sufficiente da PLC scrivere nei segnali IGM1, IGM2, IGM3 la gamma che nel File AXS.TAR si è definita con GMn=1.

#### 5.2.4.2 Tarature contenute nel file COST

I parametri necessari per il cambio gamma sono:

##### #293 = tipo di cambio gamma desiderato

Questa variabile definisce cosa deve fare il mandrino durante il cambio gamma fisico:

-1	=	oscillazione
0	=	fermo
10	=	rotazione oraria
100	=	rotazione antioraria

##### #294 = numero giri/min. di cambio gamma in gamma diretta

Questo parametro è valido se è stato scelto il cambio gamma con mandrino in oscillazione o in rotazione oraria/antioraria. Il programma CGAMMA farà ruotare il MOTORE MANDRINO a giri costanti in qualsiasi gamma si trovi il CNC. Ciò è molto utile quando il PLC deve testare il segnale di velocità bassa dell'asservimento e questo potrebbe "TRILLARE". Garantendo un riferimento costante in tutte le gamme e ricavando sperimentalmente la SPEED in giri/ in gamma alta per la quale il segnale è OK, si è risolto tale problema in qualsiasi gamma sia il mandrino alla richiesta del cambio.

Se definito cambio gamma con mandrino fermo, tale parametro è ininfluente.

N.B.: il valore del parametro #294 deve essere maggiore del valore di MFR (soglia di mandrino fermo) definita in AXS.TAR, altrimenti i segnali OBCG# e VELCG non vanno mai a 1.

Il segnale di velocità di cambio gamma raggiunta OBCG# viene gestito, per default, in base alla velocità calcolata. Se si desidera che la gestione avvenga sulla velocità effettiva, occorre definire la soglia di SOK\_CG (espressa in percentuale della velocità di cambio gamma). Tale parametro viene definito nel file di taratura AXS.TAR:

```
Record "MND" - campo "SOK" - parametro "n4"          SOK=n1;n2;n3;n4  
"n4" = Percentuale della velocità per il calcolo di velocità raggiunta in Cambio Gamma.
```

Se = 0 o parametro non definito si ha funzionamento di default.

Naturalmente, OBCG# e VELCG lavorano con velocità reale solo per mandrino con encoder. Per un mandrino senza encoder vale sempre la velocità calcolata.



---

---

**#295 = delta quota oscillazione in gradi.**

Determina il DELTA QUOTA di oscillazione mandrino in gradi durante il cambio gamma. 30 significa, ad es., che il mandrino oscillerà di  $\pm 15$  gradi. Adattando tale valore al valore della variabile #294 si può variare la frequenza di oscillazione.

Questo parametro è significativo solo se si è definito un cambio gamma con mandrino in oscillazione, per gli altri tipi è ininfluente.

**Soglie di velocità per cambio gamma automatico (F.A. S)**

**#278** = SOGLIA DI VELOCITA' FRA GM0 E GM1 IN GIRI/MIN.

**#279** = SOGLIA DI VELOCITA' FRA GM1 E GM2 IN GIRI/MIN

**#280** = SOGLIA DI VELOCITA' FRA GM2 E GM3 IN GIRI/MIN

**#281** = SOGLIA DI VELOCITA' FRA GM3 E GM4 IN GIRI/MIN

**#282** = SOGLIA DI VELOCITA' FRA GM4 E GM5 IN GIRI/MIN.

Queste dichiarazioni sono valide solo per mandrini di fresatura e sono necessarie solo se il cambio gamma è risolto come programmazione della F.A. "S" (AMN=...;...;S), altrimenti tali variabili sono ininfluenti.

Sono presenti anche i seguenti parametri per il mandrino secondario:

**#292 = tipo di cambio gamma desiderato**

**#296 = numero giri/min. di cambio gamma in gamma diretta**

**#297 = delta quota oscillazione in gradi.**

**#252** = SOGLIA DI VELOCITA' FRA GM0 E GM1 IN GIRI/MIN.

**#253** = SOGLIA DI VELOCITA' FRA GM1 E GM2 IN GIRI/MIN

**#254** = SOGLIA DI VELOCITA' FRA GM2 E GM3 IN GIRI/MIN

**#255** = SOGLIA DI VELOCITA' FRA GM3 E GM4 IN GIRI/MIN

**#256** = SOGLIA DI VELOCITA' FRA GM4 E GM5 IN GIRI/MIN.

**N.B.:** si consiglia di non modificare direttamente i records del file COST ma di copiarli e modificarli nel file COSTUSER esplicitamente previsto per le personalizzazioni legate ad una macchina specifica. In questo modo si facilita il "trasporto" delle personalizzazioni da una release di software ad una successiva

#### **5.2.4.3 Sottoprogramma SPEED**

Il programma SPEED è preposto a risolvere il cambio gamma quando in AXS.TAR si è definito AMN=...;...;S ed è programmata una SPEED tale da richiedere un cambio. In funzione della SPEED programmata e delle soglie definite, attiva il programma CGAMMA per inserire la gamma giusta. Nel caso non siano state definite soglie in COST o COSTUSER, alla prima "S" programmata interviene l'allarme 29998. {T}La routine SPEED non fa niente, ovviamente, per il mandrino principale del tornio che, come già detto, esegue il cambio gamma solo se espressamente richiesto con M40÷45.

Normalmente, il sottoprogramma non deve essere modificato.

#### **5.2.4.4 Sottoprogramma CGAMMA**

Questa subroutine viene richiamata quando sia stata programmata una M40÷45 relativa al cambio gamma. CGAMMA, in funzione delle tarature eseguite, mette il mandrino nella



condizione scelta (oscillazione, rotazione o fermo), garantendo il riferimento opportuno e quindi emette la M40÷M45 di cambio verso il PLC.

Il PLC, con tali F.A. "M", deve inserire fisicamente la gamma (eccitare elettrovalvole, pistoni, frizioni, ecc.) tenendo conto delle opportune sicurezze, mettendo FATTOM=0 fino a cambio eseguito. Normalmente, il sottoprogramma non deve essere modificato.

### 5.2.5 Esempio di gestione di un cambio gamma

L'esempio si riferisce ad una Fresa ma vale, con le opportune modifiche, anche per il mandrino principale o secondario di un Tornio. L'esempio è per una M.U. avente le seguenti caratteristiche:

- \* 1ª GAMMA M41 da 1÷500 giri/
- \* 2ª GAMMA M42 da 501÷1000 giri/
- \* 3ª GAMMA M43 da 1001÷2000 giri/
- \* TIPO DI CAMBIO GAMMA: mandrino in oscillazione
- \* SELEZIONE GAMMA: attraverso le "S" con possibilità di forzare le M40÷M45 e ripristinare il cambio attraverso le "S" tramite la F.A. M15.
- \* ASSE-MANDRINO: sigla asse C con trasduttore presente.

#### 5.2.5.1 Tarature in AXS.TAR

- \* Record TIP: campo AMN=Y;P;S
- \* Record CIN: campo RAP=2000
- \* Record GAM: campi GM0=0 GM1=0,25 GM2=0,5 GM3=1 GM4=0 GM5=0

#### 5.2.5.2 Tarature in COSTUSER

- \* #293=-1 mandrino in oscillazione
- \* #294=20 velocità desiderata in gamma diretta, in giri/
- \* #295=30 delta quota in oscillazione = 15°
- \* #279=500 Soglia di velocità tra GM1 e GM2; con S>500, il CNC chiede l'inserimento della gamma M42 (2ª gamma).
- \* #280=1000 Soglia di velocità tra GM2 e GM3 con S>1000, il CNC chiede l'inserimento della gamma M43 (3ª gamma).

#### 5.2.5.3 Equazioni nel PLC

- \* Il PLC deve comunicare su IGM1, IGM2, IGM4 in quale gamma fisica è la M.U, prima di restituire FATTOM alle F.A di cambio gamma.
- \* Il PLC deve prevedere la logica d'inserimento delle gamme tenendo conto dello stato del segnale OBCG ed eventuali altre sicurezze.

### 5.2.6 Mandrino in velocità

Al comando delle velocità mandrino sono destinate le F.A. d'indirizzo **S** (speed in giri/min). L'emissione del codice segue le stesse regole descritte nel capitolo "FUNZIONI AUSILIARIE". Per un Tornio (e per la Fresa in modalità tornitura), la **S** può rappresentare la



velocità di taglio in metri/min (con G96 attiva); in tal caso la velocità effettiva del mandrino varia automaticamente al variare del diametro lavorato.

### 5.2.6.1 Rotazione

I comandi di rotazione ed arresto sono comandati tramite le seguenti F.A. d'indirizzo "M":

- M03:** Rotazione oraria (CW) (mandrino avvita).
- M04:** Rotazione antioraria (CCW).
- M13:** Rotazione oraria + Erogazione refrigerante (M08).
- M14:** Rotazione antioraria + Erogazione refrigerante (M08).
- M05:** Stop mandrino.
- M19:** Stop mandrino orientato (descritto in seguito).

Le F.A. M03, M04, M05 e M40 -:- M45 non debbono essere definite organi da riposizionare nel file FA.TAB; è il CNC che automaticamente "riposiziona" il mandrino (gamma, rotazione, speed) attraverso dialogo con l'operatore.

Il CNC, oltre ad emettere i suddetti codici secondo le regole delle F.A., dedica ai comandi di rotazione i segnali logici già visti all'inizio di questo capitolo.

Il riferimento analogico è presente sull'uscita della scheda asse dedicata al mandrino, solo dopo avere restituito il FATTOM alle F.A. di rotazione (M03/M04).

Se si vuol ritardare l'emissione del riferimento, occorre gestire il segnale FATTOM, restituendolo solo al momento opportuno (tipicamente dopo che il PLC ha comandato l'inserzione del teleruttore di potenza dell'azionamento e questi ha terminato il suo transitorio di attivazione).

Per il mandrino utensile rotante sono presenti in COST anche i seguenti parametri :

```
N2120 {---- Variabili per UTENSILE RUOTANTE non gestito da CNC ----}  
N2125  
N2130 <##106=-1>  
N2135 {Numero " M " per rotazione oraria - M33}  
N2140 <##107=-1>  
N2145 {Numero " M " per rotazione antioraria - M34}  
N2150 <##108=-1>  
N2155 {Numero " M " per arresto rotazione - M35}  
N2160 <##110=-1>  
N2165 {Speed tramite Seconda o Terza F.A. codice definito in GEN.TAR }  
N2170 { se 2 speed tramite Seconda F.A. }  
N2175 { se 3 speed tramite Terza F.A. }
```

È possibile usare altri valori, diversi da quelli proposti (M33, M34, M35), in accordo a quanto previsto, nel PLC, per il mandrino utensile rotante.

**N.B.:** si consiglia di non modificare direttamente i records del file COST ma di copiarli e modificarli nel file COSTUSER esplicitamente previsto per le personalizzazioni legate ad una macchina specifica. In questo modo si facilita il "trasporto" delle personalizzazioni da una release di software ad una successiva

### 5.2.7 SET automatico di asse-mandrino con trasduttore.

Si ricorda che il trasduttore mandrino deve sempre essere montato con **rapporto 1/1** con il mandrino, altrimenti non si può gestire in modo corretto l'orientamento.



---

---

L'asse-mandrino è considerato un asse "tondo" e come tale un suo giro completo corrisponde a 360 gradi; in AXS.TAR i parametri NIP e RSM debbono essere correttamente impostati:

$$\text{NIP} = \text{impulsi/giro-encoder} \times 8$$

$$\text{RSM} = 360 / \text{NIP}$$

Ciò premesso, il CNC esegue la presa di zero dell'asse-mandrino nel modo seguente:

- All'accensione il SET evidentemente non è eseguito (OBSE=0), ma non viene segnalato sul video.
- Al primo comando di rotazione (M03, M04), il SET viene fatto automaticamente.
- Al primo comando di orientamento (M19), se non già fatto, l'asse-mandrino ricerca automaticamente la marca elettrica alla velocità impostata nel file COST o COSTUSER tramite il parametro **#272**, eseguendo uno spostamento al massimo di un giro.
- Appena viene trovata la marca, il CNC esegue su di essa lo zero (OBSE=1), dopodiché posiziona il mandrino sull'angolo impostato in COST o COSTUSER sul parametro **#299**. Se in un giro non viene trovata la marca, l'asse-mandrino va in allarme. Lo zero viene considerato da rifare quando la velocità del mandrino supera il valore **10\*SGV** (definito in AXS.TAR). In tal caso lo zero verrà rieseguito automaticamente nei due casi seguenti:
  - a) Uscendo dal modo velocità con un comando di orientamento (M19): durante la rampa di decelerazione, raggiunta la velocità 10\*SGV definita in AXS.TAR la si mantiene fino a trovare la marca (se in un giro non viene trovata l'asse-mandrino va in allarme), si esegue su di essa lo zero dopodiché si posiziona il mandrino sull'angolo impostato.
  - b) Rimanendo agganciato in modo velocità ma riducendo la velocità stessa sotto a quella definita da 10\*SGV: appena raggiunta questa velocità la si mantiene fino a trovare la marca, si esegue lo zero su di essa, dopodiché si raggiunge la velocità desiderata.

### 5.2.8 Asse-mandrino senza trasduttore

Se in AXS.TAR è stato definito  $\text{AMN}=\dots;\mathbf{N};\dots$  significa che manca il trasduttore mandrino. Il mandrino non può essere agganciato in contouring e non si può eseguire l'orientamento. Si possono comunque eseguire i cambi gamma come più sopra specificati.

### 5.2.9 Abilitazione azionamento mandrino

La routine di libreria PLC **aggmnd()** risolve in maniera facile e sicura la gestione dell'abilitazione dell'azionamento nelle varie modalità di funzionamento del mandrino. L'uso della routine suddetta è già stato trattato in questo capitolo.

Qui di seguito vengono date alcune precisazioni nel caso si voglia approfondire l'argomento o non si voglia o possa usare la routine **aggmnd()**.

Il PLC deve abilitare l'azionamento dell'asse-mandrino quando:

- L'asse-mandrino è agganciato in contouring

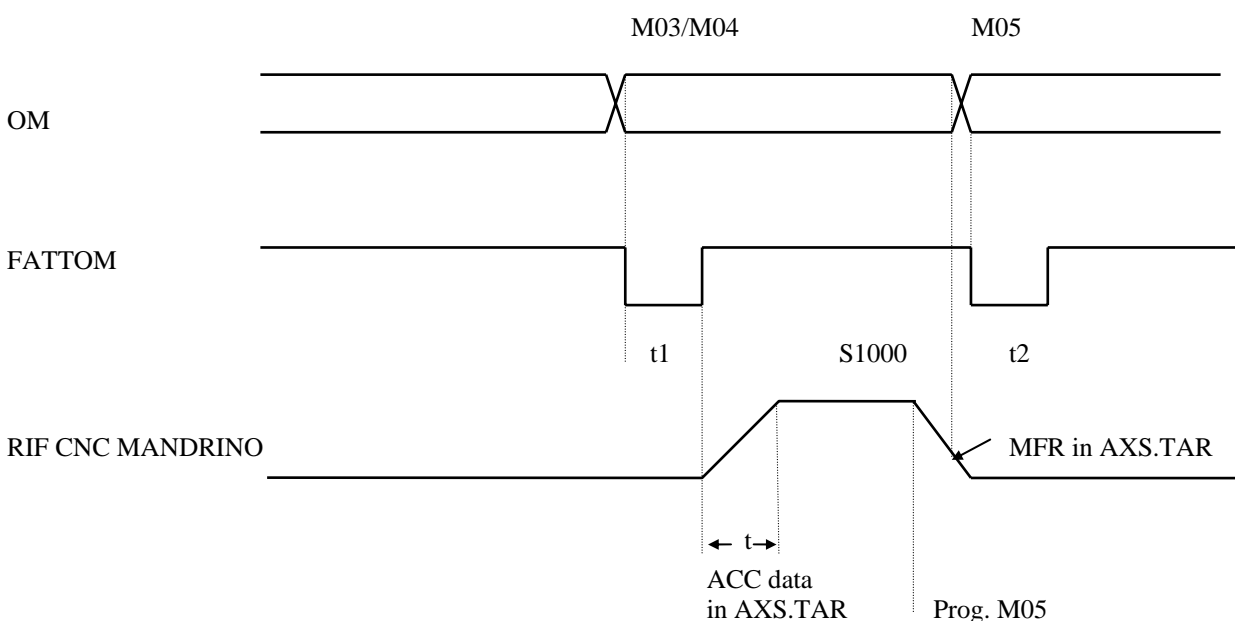
- L'asse-mandrino è agganciato in punto-a-punto (OBAG=1) e NON velocità (OBVV=0)
- L'asse-mandrino è agganciato in punto a punto e velocità ed è stata memorizzata, con CNC in modo esecuzione, l'emissione di M03 o M04. La memorizzazione di M03 M04 deve rimanere fino a che il CNC non emetta M05 o M19, sempre in modo esecuzione ( o fino a RESET).
- Viene comandata. con CNC in modo esecuzione, una G991-:-G994 di cambio gamma. Deve rimanere abilitato fino a che non viene programmata G990 (vedere anche bloccaggio mandrino). Utilizzando per cambiare le gamme della M.U. i file SPEED e CGAMMA forniti da ECS si possono utilizzare i valori di <SET:40> e <SET:140> di inizio e fine cambio gamma per abilitare l'azionamento mandrino.
- Viene comandato un orientamento mandrino (vedere anche bloccaggio mandrino)

Il CNC, con la programmazione M03/M04, si comporta nel modo seguente:

Appena viene ricevuto il FATTOM alle F.A. di rotazione M03/M04, il CNC emette il riferimento per la "S" desiderata con la rampa di accelerazione dichiarata in AXS.TAR. Il FATTOM alle funzioni ausiliarie sopra citate deve essere restituito solo quando il PLC è sicuro che l'azionamento mandrino è abilitato e pronto a seguire il riferimento con la rampa fornita dal CNC (è sottinteso che si deve disabilitare la rampa interna all'azionamento mandrino o, se impossibile, aver una rampa sul CNC che sia sicuramente più lenta di quella dell'azionamento) e non attendere che il mandrino ruoti dato che il riferimento analogico proporzionale alla "S" e alla gamma uscirà dal CNC solo dopo che tale FATTOM è stato restituito.

Programmando la funzione ausiliari M05 di stop rotazione mandrino, accade:

- Il CNC non emette subito la F.A. M05 ,ma abbassa il riferimento analogico seguendo la legge di accelerazione data da AXS.TAR per l'asse mandrino.
- Raggiunta la velocità dichiarata sempre in AXS.TAR al campo MFR, emette la F.A. M05 verso l'interfaccia; con tale F.A. il PLC deve disabilitare l'azionamento mandrino.





### 5.2.10 Bloccaggio asse mandrino

Con l'istruzione <BLC:C> attiva (dove C = sigla asse-mandrino), il funzionamento del CNC è il seguente:

- Asse mandrino FERMO agganciato in punto-a-punto (OBAG=1) e velocità (OBVV=1) oppure comandato in rotazione con M03 o M04: il CNC mantiene OBLC=1 e non considera IBLC per fare partire il mandrino.
- Comando di orientamento sia con mandrino in rotazione che con mandrino fermo. Il CNC si comporta come per un asse indipendente e cioè: prima di iniziare l'esecuzione del comando mette 0⇒OBLC e attende che il PLC metta IBLC=0. A mandrino orientato il CNC rimette 1⇒OBLC. (questa temporizzazione è utile per abilitare il riferimento mandrino).
- Con le G di cambio gamma (G990÷G994) il CNC si comporta come al primo punto.

In particolare, il segnale di OBLC è utilizzato per abilitare l'azionamento mandrino in orientamento (M19) e quando il mandrino è agganciato CNC (es. in modo JOG o in maschiatura RIGIDA)

Le istruzioni <BLC:...> per il mandrino principale e per l'eventuale secondario sono fatte automaticamente nel file COST standard.

### 5.2.11 Mandrino pilotato in spazio

Questa prestazione permette di comandare il posizionamento del MANDRINO di una M.U. direttamente dal CNC, in una posizione angolare desiderata, senza necessità di prevedere appositi dispositivi esterni, escluso il trasduttore mandrino.

Per questa prestazione è necessario avere installato il trasduttore sul mandrino della M.U. con rapporto 1/1 e definito in AXS.TAR i necessari parametri. Inoltre il KV deve essere diverso da 0 per mantenere il ciclo di spazio vivo dopo il posizionamento.

Per eseguire un posizionamento angolare da Programma o in modo MANUALE, il mandrino dovrà essere programmato con la relativa sigla definita in AXS.TAR; ad es.:

#### C quota

dove :

**C** è la sigla (definibile a piacere tra quelle ammesse per gli assi) relativa all'asse mandrino in AXS.TAR.

Quando il mandrino è messo nello stato CNC (interpolante), il controllo attende che questo sia posizionato sulla quota comandata prima di proseguire nell'elaborazione; non è possibile farlo ruotare attraverso una M03 o M04 prima di averlo ripristinato, attraverso l'istruzione <FRE:...>, indipendente.

#### 5.2.11.1 Caratteristiche necessarie alla M.U.

Per ciò che riguarda la catena cinematica, vale quanto detto per gli assi, ossia la "bontà" del posizionamento ed il suo eventuale pendolamento, dipenderanno dalla "pochezza" di giochi ed elasticità fra motore mandrino e trasduttore. Inoltre si dovrà avere cura nello scegliere un



---

idoneo azionamento mandrino che consenta, fra l'altro, una sufficiente regolazione a "velocità zero".

#### 5.2.11.2 Orientamento mandrino tramite M19

L'orientamento mandrino attraverso la F.A. M19 viene risolto mediante il sottoprogramma M19 parametrizzato con le istruzioni <MVM:...> (cioè indipendentemente dalla sigla associata al mandrino). Il posizionamento alla quota desiderata avviene dando all'asse-mandrino il valore della quota, la velocità ed il relativo comando. Anche in questo caso l'asse-mandrino è agganciato in punto-a-punto. I parametri relativi alla velocità ed alla posizione angolare sono contenuti nel file COST o COSTUSER.

#### 5.2.11.3 Orientamento mandrino libero.

Il mandrino può essere orientato anche durante la rotazione (asse mandrino indipendente), semplicemente programmandone la quota a cui si desidera l'orientamento associata alla relativa sigla definita in AXS.TAR.

#### ESEMPIO:

```
N10 M03 S1000 {Rotazione mandrino}
N30 C30 {Comando di orientamento mandrino}
N34 ...
N35 ...
```

Il CNC non attende che il mandrino sia orientato, per proseguire nell'elaborazione dei blocchi di programma successivi; è a cura del PLC, se il Costruttore M.U. lo reputa necessario, fermare il CNC fintanto che il mandrino non è orientato alla quota voluta. Nel caso si voglia aspettare che il mandrino sia arrivato sulla quota programmata C30 si può utilizzare l'istruzione <WAT:MND> inserita alla N34.

#### 5.2.11.4 Velocità di orientamento

La velocità con la quale viene orientato il Mandrino è data dai seguenti tre casi:

- Nel caso che il mandrino sia FERMO la velocità di orientamento è data tramite l'istruzione: <FED: SiglaAsseMandrino;ValoreVelocità> dove la velocità è espressa in gradi/' .
- Nel caso di mandrino in ROTAZIONE la velocità di orientamento è quella dichiarata nel campo SGV= al record MND: di AXS.TAR (solo se velocità di rotazione > SGV).
- Nel caso di Mandrino FERMO o in ROTAZIONE a velocità inferiore a SGV, se non programmata alcuna velocità con <FED:...;ValoreVelocità>, il Mandrino si orienta con una velocità di default pari a 360 gradi/' (1giro/').

Se si vuole comandare l'orientamento anche con il comando "NomeAsse quota", si consiglia di definire una velocità relativa all'asse mandrino nel file COSTUSER, la quale viene assunta come velocità di default all'accensione del CNC. Utilizzare l'istruzione:

```
<FED:nome asse;velocità in gradi/min.>
```





---

### 5.2.12 Mandrino come asse tondo

L'asse-mandrino deve essere definito "tondo" (TYP=INT in AXS.TAR). Il posizionamento angolare dell'asse-mandrino varia in funzione dello stato di asse Indipendente oppure asse CNC (interpolante) nel seguente modo:

- Asse CNC:  
Esegue sempre il posizionamento alla quota comandata. Il CNC attende che il posizionamento sia avvenuto prima di proseguire nell'elaborazione del Part-Program.
- Asse indipendente:  
Esegue il posizionamento sempre per la via più breve, il CNC non attende che il posizionamento sia avvenuto per proseguire nell'elaborazione del Part-Program.

Ad esempio, avendo un mandrino posto nello stato di asse indipendente (FRE) posizionato sulla quota +10, programmando un movimento a **360** gradi, il mandrino si porterà a quota **0** raggiungendola in senso negativo. In pratica effettuerà uno spostamento di 10 gradi in senso negativo. Lo stesso mandrino, posto nello stato CNC con un'istruzione <CNC:...>, eseguirà invece uno spostamento di 350 gradi in senso positivo.

Ancora:

asse-mandrino C , "FRE" e posizionato a 0 gradi; programmando:

**C 360 :** rimane fermo su 0 gradi

**C 410 :** si sposta di 50 gradi

mentre se fosse nello stato "CNC", programmando:

**C 360 :** raggiungerà 360° muovendosi in senso positivo (facendo un giro)

**C 410 :** eseguirà un giro più 50°.

### 5.3 MANDRINO GESTITO DAL PLC

Non esiste nessun asse definito mandrino in AXS.TAR.

Il CNC non attiva la sua gestione delle F.A. M03, M04, M05, M40 -:- M45.

Queste sono F.A. qualsiasi, definibili nel file FA.TAB.

Non è possibile eseguire né orientamento né maschiatura rigida.

Se il riferimento per la rotazione deve essere fornito dal CNC, occorre predisporre una delle uscite D/A dalle schede assi dichiarandola disponibile per il PLC nel file SWCNF.TAR; ad esempio:

```
;Analog D/A Outputs Configuration
DA00=RFR(AN,0,3)
;DA01=RFR(AN,1,3)
;DA02=RFR(AN,2,3)
```

In questo modo, la gestione del convertitore relativo (sinonimi VDA0, VDA1, ...) è a carico del PLC. E' a cura del PLC gestire l'override agendo sul convertitore di uscita utilizzato.

Se il riferimento è generato da un dispositivo elettronico esterno al CNC, vale tutto quanto riportato sopra tranne:

- non serve nessuna taratura nel file SWCNF.TAR

---



- il dato relativo alla speed (disponibile su OTDS oppure OTDSL ed eventualmente OTDSH) va ricopiato sulle uscite di una scheda I/O per essere elaborato dal dispositivo elettronico.

## 5.4 FILETTATURA

Il Tornio permette di eseguire un'ampia gamma di lavorazioni di filettatura del pezzo montato sul mandrino di tornitura. Anche sulle frese è possibile definire un mandrino secondario di tornitura (potrebbe essere la tavola rotante dotata di opportuni motore, trasduttore e catena cinematica). Per questo tipo di lavorazioni, sono presenti alcuni parametri specifici sia in AXS.TAR sia in COST; per i particolari vedi MANUALE TARATURE 720P385. Per il PLC è presente il segnale:

<b>OG33</b>	<b>CNC → PLC</b>	Movimento di Filettatura (G33, G34, G35) in esecuzione.
-------------	------------------	---

Quando si esegue una filettatura, il CNC ignora l'effetto dei tasti HOLD (assi e mandrino) e AXES H. (solo assi) e dei segnali di Input IBL, IBLAX e IFHD posticipando il loro effetto alla fine della G33 (il CNC risponde subito ACK al codice tasto premuto).

Il tasto RESET ha efficacia anche in G33 facendo fermare sia gli assi sia il mandrino.

Il CNC sia eseguendo sia leggendo l'istruzione G33, G34, G35 mette **OG33=1** altrimenti **OG33=0**.

Il CNC, premendo i tasti di cambio modo durante la G33, segnala un errore di tasto rifiutato; per i tasti di cambio sottomodo, posticipa il loro effetto al termine della G33 pur cambiando subito l'output verso il PLC e rispondendo ACK al tasto premuto.

Alla partenza della G33, il CNC attende che l'asse mandrino raggiunga la velocità desiderata (soglia e numero di volte definite in AXS.TAR) e che passi dal suo zero elettrico prima di iniziare ad agganciare l'asse o gli assi di filettatura al mandrino.

La sincronizzazione alla partenza dell'asse è fatta anche per:

- 1) assi bloccati meccanicamente; dopo la partenza degli assi il CNC non sente più gli input IBLC# per tutta la durata della/e G33.
- 2) rotazione mandrino OK cioè uguale a quella programmata; per tutta la durata della G33 il CNC forza override Speed, Feed e Rapido = 100%.

### 5.4.1 KEY durante la filettatura (G33, G34, G35)

Durante l'esecuzione della G33 (OG33=1) il CNC non verifica se vi sono delle richieste di KEY (immediato, sincrono, asincrono) attive. Le eventuali richieste di KEY vengono onorate solo al termine della G33. Se arriva una richiesta di KEY mentre il mandrino, dopo una G33, è in attesa del passaggio dallo zero, interviene uno specifico ALLARME; è quindi cura del PLC fare in modo che quando c'è OG33=1 non vi siano richieste di esecuzione KEY.

## 5.5 STOP MANDRINO

Nel file di taratura AXS.TAR, è presente il campo:

**MFR=C1;C2** Taratura per velocità mandrino fermo e stop mandrino

**C1** = Soglia velocità minima [giri/min] al disotto della quale si segnala mandrino fermo (segnale OBMT#)

**C2** = Soglia velocità minima [giri/min] al di sotto della quale si segnala stop mandrino (segnale OSTM#); utile per azionamenti con abilitazione automantenuta in frenatura .



## Funzionamento

Con mandrino in rotazione, facendo HOLD o programmando M05 il CNC mette OSTM#=1 quando la velocità attuale del mandrino è minore o uguale a quella definita in C2 di AXS.TAR

Il segnale OSTM# viene tenuto a "1" fino a che:

- con M05 il PLC non risponde fatto alla FA.
- con HOLD mandrino non si risblocca il CNC.

Qualunque sia il valore di soglia stop mandrino definita in AXS.TAR, la rampa di decelerazione sul riferimento del mandrino non cambia. Agendo sull'override del mandrino OSTM# non cambia stato.

## 5.6 ALLARMI MANDRINO

Nel file di taratura AXS.TAR è presente un campo per definire i parametri relativi al controllo della "fuga" mandrino.

FGV=c;n1;n2;n3;n4

Per il maggiori informazioni ed il relativo significato, si rimanda al "Manuale Tarature" 720P385.

Nel PLC è possibile sospendere l'intervento dell'allarme tramite:

ISAM#	PLC → CNC	Sospensione dell'allarme fuga mandrino asse #.
-------	-----------	--

Nel caso si desideri girare il mandrino a mano, occorre togliere l'abilitazione all'azionamento e mettere ISAM# = 1, altrimenti nel momento in cui viene riabilitato il mandrino, il Software Assi (SA) fa recuperare al mandrino tutto lo spazio percorso a mano con probabile intervento dell'allarme fuga.

Durante l'esecuzione della filettatura G33, il CNC non invia alla SA la sospensione dell'allarme fuga anche se il PLC ha messo ISAM# = 1.

Il segnale ISAM# va usato anche durante la fase di accelerazione se il mandrino segue con difficoltà la rampa di accelerazione impostata in taratura.

Se viene usata la routine standard di libreria **aggmnd()** già descritta in questo capitolo, non occorre agire direttamente su ISAM# se non in casi particolari.

## 5.7 AGGANCIAMENTO MANDRINO FRV / CNC

Il Part-program può comandare la commutazione dello stato di funzionamento dei mandrini, se sono assi del CNC, tramite le normali istruzioni di LIP "logico"(ad.es.:mandrino=asse C):

<FRE:C> Attiva il controllo in modo velocità (FRV) oppure in ciclo di spazio indipendente dagli altri assi (punto-a-punto ossia FRE) per il mandrino C.

<CNC:C> Attiva il controllo in ciclo di spazio assieme agli altri assi del CNC (contouring ossia CNC); il mandrino C diventa un asse di contornatura a tutti gli effetti.

Anche per il PLC sono disponibili alcune funzioni che permettono di commutare il mandrino (principale o secondario), da FRV (free, velocità) a CNC (interpolante) e viceversa, quando il CNC è in modo JOG.



Quando il CN non è in modo JOG, le stesse funzioni permettono di predisporre lo stato d'aggancio del mandrino FRV o CNC quando verrà attivato il modo JOG.

### 5.7.1 Gestione tramite Funzioni da PLC

Le funzioni previste sono le seguenti:

<b>AggMndFrv()</b>	Esegue comando di aggancio "FRV" sul mandrino richiesto .
<b>AggMndCnc()</b>	Esegue comando di aggancio "CNC" sul mandrino richiesto
<b>TstFrvCnc()</b>	Esegue test per verificare se comando terminato.
<b>TstStsMnd()</b>	Esegue test su tipo attuale di aggancio del mandrino richiesto.

#### 5.7.1.1 AggMndFrv: comando di aggancio Free/velocità

Chiamata: **esito = AggMndFrv(Mnd)**

Descrizione: Richiede l'aggancio in modo Free/velocità per il mandrino indicato.

#### 5.7.1.2 AggMndCnc: comando di aggancio CNC

Chiamata: **esito = AggMndCnc(Mnd)**

Descrizione: Richiede l'aggancio in modo CNC (contouring) per il mandrino indicato.

### Parametri di ingresso / uscita delle due funzioni :

Parametro di ingresso:

**Mnd** = (unsigned short) Numero per la scelta del mandrino; può essere:

- 0 Mandrino principale
- 1 Mandrino secondario

Parametro di uscita:

**esito** = (short) valore di ritorno delle funzioni . Può assumere uno dei seguenti valori:

- 6 richiesto aggancio in modo FRV ma il mandrino è attualmente agganciato in countoring e non arrivato in posizione
- 5 agganciato in punto-a-punto (FRE) e non in velocità
- 4 richiesto aggancio in modo CNC ma il mandrino è attualmente agganciato in velocità con comando di rotazione attivo
- 3 agganciato in velocità e non fermo
- 2 mandrino non OK; cause possibili:
  - l'asse è rotto (non disponibile)
  - non ha il BOK=1
  - non è agganciato
  - non ha il trasduttore
  - è gestito e movimentato da PLC
- 1 il mandrino non è un asse (non è definito in AXS.TAR).
- 0 Successo: comando già eseguito (stato richiesto = stato attuale)
- 1 Successo: comando di cambio stato inoltrato.
- 2 CNC in modo diverso da JOG; si effettua solo la predisposizione di aggancio del mandrino per quando si entrerà in JOG
- 3 stato del CNC non OK:
  - ha un comando in corso di esecuzione
  - è in allarme



- è in fase di RESET
- 4 c'è già un comando in corso sull'altro mandrino

### 5.7.1.3 TstFrvCnc: test comando di aggancio

Chiamata: **esito = TstFrvCnc(Mnd)**

Descrizione: Esegue test per verificare se c'è ancora in corso un comando inviato tramite le funzioni AggMndFrv() e AggMndCnc().

Parametro di ingresso:

**Mnd** = (unsigned short) Numero per la scelta del mandrino; può essere:

- 0 Mandrino principale
- 1 Mandrino secondario

Parametro di uscita:

**esito** = (short) valore di ritorno delle funzioni . Può assumere uno dei seguenti valori:

- 0 nessun comando in corso
- 1 comando in corso per questo mandrino
- 2 CNC non in modo JOG; è possibile effettuare solo la predisposizione di aggancio del mandrino per quando si entrerà in JOG
- 3 stato del CNC non OK:
  - ha un comando in corso di esecuzione
  - è in allarme
  - è in fase di RESET
- 4 c'è un comando in corso sull'altro mandrino

### 5.7.1.4 TstStsMnd: test stato di aggancio

Chiamata: **esito = TstStsMnd(Mnd)**

Descrizione: Esegue Test sullo stato di aggancio del mandrino.

Parametro di ingresso:

**Mnd** = (unsigned short) Numero per la scelta del mandrino; può essere:

- 0 Mandrino principale
- 1 Mandrino secondario

Parametro di uscita:

**esito** = (short) valore di ritorno delle funzioni . Può assumere uno dei seguenti valori:

- 0 Comando in corso o Mandrino non definito come asse in AXS.TAR .
- 1 Mandrino agganciato FRV (punto-a-punto ed in velocità)
- 2 Mandrino agganciato CNC (contouring)
- 3 Mandrino agganciato FRE (punto-a-punto e non in velocità)

## 5.8 MANDRINO ABBINATO A MATRICI DINAMICHE

Per la programmazione delle matrici dinamiche ed il loro funzionamento, occorre tenere presenti le seguenti considerazioni:

- 10. In AXS.TAR l'asse mandrino deve avere DRZ=2.



11. Non interviene l'allarme quando viene attivata la matrice dinamica su un'asse definito in AXS.TAR Tondo e/o Mandrino purché questo sia stato preventivamente agganciato in CNC.
12. La matrice dinamica viene disattivata con RESET, istruzioni %, :N00. Con cambi modo AUTO, MAN, JOG, SET la matrice dinamica rimane attivata.
13. La quota programmata dell'asse X è interpretata dal CNC come diametrale se l'asse X è definito diametrale.

### 5.8.1 Funzionamento G345

Per le matrici dinamiche è stata implementata una nuova istruzione: **G345**.

La **G345** esegue lo STOP del mandrino (se questo è in rotazione) anche con CNC in sottomodo lettura.

Il CNC con G345 (anche se lettura) porta la velocità di rotazione del mandrino a zero, dopodiché il CNC emette verso il PLC la **F.A. M05** e la G345 (*codice generico*).

Il PLC con la nuova G345 deve togliere la memorizzazione di M03 M04 e l'abilitazione all'asse mandrino anche se siamo in LETTURA.

Non è consentito programmare nella stessa N.. una qualsiasi F.A. intermedia "M"

```
N.. G345 M54
```

altrimenti interviene un allarme.

La G345 è necessaria perché, quando veniva letto un part-program dove c'è l'attivazione di una matrice su assi X Z C (mandrino), con il mandrino in rotazione, interverrebbe un allarme. Poiché l'asse C è agganciato in p/to p/to, lo stesso p.program letto con mandrino fermo va bene.

La **G345** agisce sul mandrino selezionato, quindi se precedentemente c'è **G634** la **G345** agisce sul mandrino secondario.

Facciamo un esempio tipico per il tornio "classico", C sia l'asse mandrino, X l'asse diametrale e Z l'asse degli spallamenti; di conseguenza occorre scrivere:

```
N10 G345 {ferma il mandrino se ruota}  
N20 M19 {fa SET del mandrino se non fatto}  
N30 <CNC:C> {mette il mandrino CNC}  
N40 G16 ZCX+ {Cambia assi del PC}  
N50 <DMX:Z;X;C>
```

### 5.9 MODIFICA DELLA SPEED

La gestione della modifica dei valori di default, dei valori limite e dei valori attuali della Speed Mandrino (Principale e Secondario) è realizzata tramite apposite funzioni di libreria PLC, riassunte nelle tabelle seguenti e già descritte al capitolo 3.11 .

<b>SetDefSpeed()</b>	Imposta la velocità di rotazione (Speed) di Default Mandrino
<b>SetLimSpeed()</b>	Limita la velocità massima Mandrino
<b>SetTarSpeed()</b>	Ripristina la velocità di taratura del Mandrino
<b>SetSpeed()</b>	Setta il valore di velocità di rotazione (Speed) del Mandrino
<b>ValueDefSpeed()</b>	Legge il valore della Speed di default
<b>ValueLimSpeed()</b>	Legge il valore limite attuale per la Speed
<b>ValueTarSpeed()</b>	Legge il valore di taratura per la Speed
<b>ValueSpeed()</b>	Legge il valore di velocità di rotazione (Speed) del Mandrino

Le seguenti funzioni riguardano il mandrino secondario ed hanno lo stesso significato di quelle viste sopra per il mandrino principale.



<b>SetDefSpeedSec()</b>	Imposta la velocità di rotazione (Speed) di Default Mand.Sec.
<b>SetLimSpeedSec()</b>	Limita la velocità massima Mand.Sec.
<b>SetTarSpeedSec()</b>	Ripristina la velocità di taratura del Mand.Sec.
<b>SetSpeedSec()</b>	Setta il valore di velocità di rotazione (Speed) del Mand.Sec.
<b>ValueDefSpeedSec()</b>	Legge il valore della Speed di default del Mand.Sec.
<b>ValueLimSpeedSec()</b>	Legge il valore limite attuale per la Speed del Mand.Sec.
<b>ValueTarSpeedSec()</b>	Legge il valore di taratura per la Speed del Mand.Sec.
<b>ValueSpeedSec()</b>	Legge il valore di velocità di rotazione (Speed) del Mand.Sec.

## 5.10 M05 E ATTESA MANDRINO FERMO

Normalmente, la programmazione di M05 (stop mandrino) provoca l'arresto "controllato" del mandrino stesso; il riferimento viene portato a zero con la rampa definita in taratura (parametro ACC= di AXS.TAR) e poi viene emessa la F.A. M05 che provoca la caduta di ABMND (vedi routine standard aggmnd() ). Il CNC non passa alla N.. successiva fino a quando tutta la sequenza non è terminata.

Se, per esigenze di Macchina Utensile, occorre proseguire al più presto con il Part-Program senza attendere l'arresto del mandrino, occorre predisporre il funzionamento "M05 senza attesa mandrino fermo" e ripristinare poi, se necessario, il funzionamento standard.

La libreria del PLC mette a disposizione le funzioni necessarie allo scopo.

### 5.10.1 Gestione M05 tramite Funzioni da PLC

Le funzioni previste sono le seguenti:

<b>TstMfrM05()</b>	Testa la predisposizione attuale
<b>OnMfrM05()</b>	Predisporre emissione M05 <u>con</u> attesa mandrino fermo.
<b>OffMfrM05()</b>	Predisporre emissione M05 <u>senza</u> attesa mandrino fermo.

#### 5.10.1.1 OnMfrM05: Predisporre emissione M05 con attesa mandrino fermo.

Chiamata: **esito = OnMfrM05 (Mnd)**

Descrizione: Richiede l'emissione di M05 con attesa mandrino fermo, per il mandrino indicato.

#### 5.10.1.2 OffMfrM05: Predisporre emissione M05 senza attesa mandrino fermo.

Chiamata: **esito = OffMfrM05 (Mnd)**

Descrizione: Richiede l'emissione di M05 senza attesa mandrino fermo, per il mandrino indicato.

#### Parametri di ingresso / uscita delle due funzioni:

Parametro di ingresso:



---

---

**Mnd** = (unsigned short) Numero per la scelta del mandrino; può essere:

- 0 Mandrino principale
- 1 Mandrino secondario

Parametro di uscita:

**esito** = (short) valore di ritorno delle funzioni . Può assumere uno dei seguenti valori:

- 0 predisposizione effettuata
- 1 mandrino non definito in AXS.TAR

### 5.10.1.3 TstMfrM05: test predisposizione attuale

Chiamata: **esito = TstMfrM05(Mnd)**

Descrizione: Esegue Test sulla predisposizione attuale del mandrino specificato.

Parametro di ingresso:

**Mnd** = (unsigned short) Numero per la scelta del mandrino; può essere:

- 0 Mandrino principale
- 1 Mandrino secondario

Parametro di uscita:

**esito** = (short) valore di ritorno delle funzioni . Può assumere uno dei seguenti valori:

- 1 Predisposta l'emissione M05 con attesa mandrino fermo
- 0 Predisposta l'emissione M05 senza attesa mandrino fermo
- 1 Mandrino non definito in AXS.TAR

**Note:**

- 1) All'accensione, il CNC predispose l'emissione M05 con attesa mandrino fermo sia per il mandrino principale sia per quello secondario. Qualora sia richiesto il funzionamento opposto, è necessario che il PLC chiami, almeno una volta, la funzione **OffMfrM05()** per il(i) mandrino(i) desiderato(i).
- 2) La routine di libreria PLC **aggmnd()** provvede a mantenere abilitato l'azionamento mandrino (ABMND e/o ABMNDS) fino all'arresto dello stesso, utilizzando la TstMfrM05(), anche quando è predisposta l'emissione M05 senza attesa mandrino fermo.





## CAPITOLO 6

### 6 MISCELLANEA

In questo capitolo sono raccolte una serie di segnali e di funzioni di uso generico che non riguardano un aspetto particolare del CNC (Assi, Mandrino, Tastiera, ...) e che quindi non trovano posto nei capitoli specifici.

#### 6.1 Sinonimi miscellanei

Per la gestione del CNC tramite il PLC, sono presenti alcuni segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

<b>TAUCNC</b> (double)	<b>CNC → PLC</b>	Tempo di ciclo del CNC. Valore espresso in secondi (TCYC * KT di SWCNF.TAR)
<b>TAUPLC</b> (double)	<b>CNC → PLC</b>	Tempo di ciclo del PLC. Valore espresso in secondi (TCYC * KT * TAUPLC di SWCNF.TAR)
<b>NORESET</b>	<b>PLC → CNC</b>	Blocca la prosecuzione della fase di <b>RESET</b> (MAX 15 sec) dalla premuta del tasto RESET. Esso é da utilizzare quando é necessario comunque terminare una certa attività indipendentemente dalla premuta del tasto di RESET. Se si supera il tempo max (15 secondi) il CNC da allarme.
<b>CK1SEC</b>	<b>CNC → PLC</b>	CK di 1 Secondo Tipicamente usato per incrementare contatori per vari utilizzi. CK1SEC rimane a 1 per un giro intero del PLC, ogni secondo.
<b>TARAZ</b>	<b>CNC → PLC</b>	Utility di <b>Taratura Analogica Azionamenti</b> attiva
<b>JOGPAZ</b>	<b>CNC → PLC</b>	JOG Positivo dall'utility Taratura Azionamenti
<b>JOGNAZ</b>	<b>CNC → PLC</b>	JOG Negativo dall'utility Taratura Azionamenti
<b>ILSAZ</b>	<b>CNC → PLC</b>	Richiesta Libera-Servo dall'utility Taratura Azionamenti
<b>COMAZ</b> (char)	<b>CNC → PLC</b>	Numero d'ordine dell'asse da muovere richiesto dall'utility Taratura Azionamenti
<b>IOPLCSTS</b> (short)	<b>CNC → PLC</b>	<u>Stato dell'ultimo comando inoltrato dal PLC al CNC.</u> <u>Usato da alcune funzioni che non possono essere eseguite</u> <u>direttamente dal PLC ma richiedono l'intervento di un altro</u> <u>Task (tipicamente WINVIS).</u> Le funzioni sono dettagliatamente descritte più avanti in questo stesso capitolo.



<p><b>IOPLCRET</b> (short)</p>	<p><b>CNC → PLC</b></p>	<p><u>Esito</u> dell'ultimo comando inoltrato dal PLC al CNC. Usato da alcune funzioni che non possono essere eseguite direttamente dal PLC ma richiedono l'intervento di un altro Task (tipicamente WINVIS). Le funzioni sono dettagliatamente descritte più avanti in questo stesso capitolo.</p>
------------------------------------	-------------------------	---

## 6.2 POWER-ON

Il "primo giro" del PLC, sia **all'accensione** del CNC sia al passaggio **ALT ⇒ EXE**, è indicato dal segnale **pon** (power-on). Questo segnale è utile per inizializzare variabili, settare il tempo dei ritardi, attivare funzioni, ecc..

```

/*-----*/
/*  INIZIALIZZAZIONI AL POWER-ON  */
/*-----*/
if (pon)
{
  /* SETTA TEMPORIZZATI AL pon (Power ON) */
  lotime(3,2.0);
  /* SETTA RITARDO DI 2 SEC PER ASSE_MANDRINO DI ORDINE 3 */
  lotime(1,0.3) ;
  /* SETTA RITARDO DI 0.3 SEC PER ASSE Y DI ORDINE 1 */
  A0.word= iniz_drf(3,4,1,4,1,4);
  /* INIZIALIZZA VOLANTINI DRF */
  /* VOL1 SU TASTIERA; ALTRI SU SCHEDE ASSI */
  if(A0.word)
  { /* INSUCCESSO */
    INIDRFOK= 0;
  }
  else
  { /* INIZIALIZZAZIONE OK */
    INIDRFOK= 1;
  }
  start_corr_ext(1,1);
  /* Attiva correzione termica assi Y e Z con filtro */
  read_var(VL_FLOT,10000,600,-1, &CTEMPY);
  /* Coeff. dilatazione termica Y da #300 scritto in COSTUSER */
}

```

## 6.3 MESSAGGI PARAMETRIZZATI

Il PLC ha la possibilità di comunicare all'operatore situazioni operative particolari o pericolose oppure guasti intervenuti sui vari componenti della Macchina Utensile, tramite messaggi "in chiaro" che vengono visualizzati in un'apposita finestra del video.

I messaggi del PLC sono suddivisi in due categorie che differiscono solo per come vengono visualizzati:

**ALLARMI** : visualizzati dal CNC su sfondo rosso

**ATTENZIONI** : visualizzate dal CNC su sfondo azzurro



I testi dei messaggi del PLC sono definiti negli appositi files ALLARxxx.MU (uno per ogni lingua selezionabile), con le particolarità e le regole indicate nella monografia 720P394 "AMBIENTE e PROCEDURE di SVILUPPO e DEBUG PLC".

Ciascun ALLARME o ATTENZIONE del PLC è associato ad una variabile predefinita che il PLC setta, resetta e testa come una qualunque altra variabile booleana ( 0, 1) trattata in questa monografia.

Per la gestione dei messaggi PLC, sono presenti le variabili associate ai sinonimi qui di seguito elencati.

<b>AALRnn</b> (short)	<b>PLC → CNC</b>	Words di appoggio per gli ALLARMI PLC. <b>nn vale 00 . . . 12</b> Ogni word raccoglie lo stato di 16 ALLARMI, che sono anche definiti singolarmente; ad es. AALR00 corrisponde, dal bit più leggero al più pesante, agli allarmi da ALR00 ad ALR15; AALR06 corrisponde ad ALR96...ALR111 e così via fino ad AALR12 che raccoglie gli ultimi 8 allarmi ALR192 ... ALR199 per un totale di 200. Il raggruppamento in words può essere utile per suddividere gli Allarmi in diverse categorie che possono essere quindi trattate come insieme per le azioni da intraprendere.
<b>ALRnn</b>	<b>PLC → CNC</b>	ALRnn= 1: il CNC deve visualizzare il testo definito nel file ALLARxxx.MU (secondo la lingua attualmente selezionata) nelle righe seguenti la definizione #define ALRnn 19... corrispondente . <b>nn vale 00 . . . 199</b> per un totale di 200 Allarmi.
<b>AATTnn</b> (short)	<b>PLC → CNC</b>	Words di appoggio per le ATTENZIONI PLC. <b>nn vale 00 . . . 12</b> Ogni word raccoglie lo stato di 16 ATTENZIONI, che sono anche definite singolarmente; ad es. AATT00 corrisponde, dal bit più leggero al più pesante, alle attenzioni da ATT00 ad ATT15; AATT06 corrisponde ad ATT96...ATT111 e così via fino ad AATT12 che raccoglie le ultime 8 attenzioni ATT192 ... ATT199 per un totale di 200. Il raggruppamento in words può essere utile per suddividere le Attenzioni in diverse categorie che possono essere quindi trattate come insieme per le azioni da intraprendere.
<b>ATTnn</b>	<b>PLC → CNC</b>	ATTnn= 1: il CNC deve visualizzare il testo definito nel file ALLARxxx.MU (secondo la lingua attualmente selezionata) nelle righe seguenti la definizione #define ATTnn 19... corrispondente . <b>nn vale 00 . . . 199</b> per un totale di 200 Attenzioni.
<b>ACTALAT</b> (short)	<b>CNC → PLC</b>	<b>ACTALAT</b> contiene il numero di allarme od attenzione visualizzato dal CNC in quel momento. Con ACTALAT=0 non c'è nessun messaggio visualizzato. Codici validi per il PLC: 19400 (MECS00) ÷ 19431 (MECS31) 19600 (ATT00) ÷ 19799 (ATT199) 19800 (ALR00) ÷ 19999 (ALR199)



Il testo degli messaggi (ALLARMI, ATTENZIONI) PLC può essere parametrizzato.

Per fare questo è necessario agire in due direzioni distinte:

- Nel testo del messaggio è necessario inserire nel punto desiderato una delle seguenti sequenze di caratteri:

@u (o @U)

@d (o @D)

@x (o @X)

@l (o @L)

@s (o @S)

- Nella logica del PLC, prima di attivare l'allarme o attenzione (o anche quando è già attivo), chiamare una delle seguenti funzioni per passare il parametro desiderato:

`addint(short n);`

`addlong(long n);`

`addstr(char *stringa);`

All'interno del testo del messaggio, nel punto in cui è presente una delle sequenze predefinite verrà inserita la visualizzazione del parametro corrispondente nella seguente modalità:

@u - numero intero decimale senza segno (0-65535)

@d - numero intero decimale segnato (-32768...+32767)

@x - numero intero esadecimale (0000... FFFF)

@l - numero long segnato (-2.147.483.648...+2.147.483.647)

@s - stringa (massimo 40 caratteri)

**ATTENZIONE:** la locazione in cui le tre funzioni **addint()**, **addlong()**, **addstr()** appoggiano il parametro è unica per tutti gli Allarmi, Attenzioni e per i vari tipi di parametro.

La lunghezza massima della stringa è di 40 caratteri, una stringa più lunga viene pertanto scartata. Inoltre, se sostituendo ad una sequenza il suo valore corrispondente, la lunghezza totale del record del messaggio viene a superare i 40 caratteri, i caratteri in eccesso a questo valore vengono scartati.

La modalità di visualizzazione del parametro dipende dal tipo di sequenza presente all'interno del testo dell'Allarme e non dal modo in cui il parametro è passato, cioè un parametro passato come intero verrà visualizzato come numero intero segnato, senza-segno, esadecimale o come long-segnato a seconda che nel testo dell'allarme sia presente @d, @u, @x, @l.

All'interno del testo dell'allarme vi possono essere più occorrenze della stessa sequenza oppure anche sequenze diverse tra di loro, ad esempio:

#### File ALLARITA.MU

```
#define ATT22 19622 /* */
/*VALORE VARIABILE APPO01: */
/* intero decimale con segno: @d */
/* intero decimale senza segno: @u */
/* intero esadecimale: @x */
```

il parametro visualizzato sarà comunque sempre lo stesso.



---

---

La richiesta di un allarme può essere tenuta sempre vera ed è possibile cambiare in modo dinamico il testo in esso contenuto chiamando una delle tre funzioni in modo continuativo senza avere aggravii del tempo di visualizzazione in quanto l'aggiornamento viene eseguito solamente se il testo da visualizzare è diverso da quello attualmente visualizzato.

### 6.3.1 Esempio di utilizzo con stringa :

#### File ALLARITA.MU

```
#define ALR00 19800 /* */
/* Avaria alla @s idraulica */
```

#### File QAL.C

```
if(POMPAKO || MOTOREKO)
{
    ALR00=1;
    if (POMPAKO && (ACTALAT == 19800))
    {
        addstr("pompa")
    }
    if (MOTOREKO && (ACTALAT == 19800))
    {
        addstr("centralina")
    }
}
else
{
    ALR00=0;
}
```

### 6.3.2 Esempio di utilizzo con numero :

#### File ALLARITA.MU

```
#define ATT00 19600 /* */
/* Il numero é: @d */
/* */
```

#### File QAL.C

```
if(pon)
    NUMERO=0;
ATT00=PIU || MENO;
if(PIU)
    NUMERO+=1;
if(MENO)
    NUMERO-=1;
if(ACTALAT == 19600)
    addint(NUMERO);
```

### 6.3.3 Gestione Storico Allarmi ed Eventi del CNC.

Dalla release software V2.04 è possibile generare un file ASCII in cui vengono registrati allarmi ed eventi del CNC. Il file si chiama **ECSCLOG.TXT** e risiede nella Directory **C:\ECS.CNC\LA\WORK**. Esso viene scritto "appendendo" gli eventi ed ha indicativamente una dimensione massima di 1 MB. Se questo limite viene superato, il file è ridotto al 50% eliminando le registrazioni più vecchie.



Al momento della installazione del CNC, la scrittura del file è disabilitata. E' possibile attivarla tramite il tasto [EcsCncLog], sottomenu Comandi [F7] del menu principale di WINVIS.

All'attivazione è possibile decidere se si desidera siano registrati gli ALLARMI, ATTENZIONI, MESSAGGI del PLC; è inoltre possibile escludere le registrazioni dei seguenti Eventi:

- Eventi registrati alla pressione di un tasto della tastiera CNC ( escluso il tasto RESET che viene sempre registrato)
- Eventi registrati alla richiesta di cambio modo del CNC
- Eventi registrati al cambio modo del CNC
- Eventi registrati alla variazione dei segnali OHLD, OESE, IFHD, FATTOM del CNC.

In ogni registrazione (sia evento sia allarme) viene incluso la data e l'ora.

Di seguito viene riportato l' elenco degli Eventi e degli Allarmi che sono registrati nel file.

### 6.3.3.1 Eventi registrati al BOOT, ALT, EXE del CNC

Evento	viene effettuata la seguente registrazione
CNC BOOT	CNC BOOT : data - ora
CNC passa nello stato di ALT	EXE -> ALT : data - ora
CNC passa nello stato di EXE	ALT -> EXE : data - ora

### 6.3.3.2 Eventi registrati alla pressione di un tasto della tastiera CNC:

Evento	viene effettuata la seguente registrazione
Pressione tasto [TASTO]	KEY : TASTO data - ora

dove TASTO può essere: START, HOLD, HOLD AXES, REL, RESET.

### 6.3.3.3 Eventi registrati alla richiesta di cambio modo del CNC

Evento	viene effettuata la seguente registrazione
Richiesta di <u>modo</u>	REQ MODE : MODREQ data - ora

dove modo e MODREQ possono essere:

Automatico	AUTO
Manuale	MDI
Jog	JOG
Esecuzione Continua	EXE CNT
Esecuzione Singola	EXE SNG
Lettura Continua	TST CNT
Lettura Singola	TST SNG
Riposizionamento	REP

### 6.3.3.4 Eventi registrati al cambio modo del CNC

Questi eventi vengono monitorati continuamente.

I modi (MODE) registrati possono essere SET, AUTO, MDI, JOG, REP

I sottomoduli (SUBMODE) registrati possono essere EXE/TST e CNT/SNG

Evento	viene effettuata la seguente registrazione
--------	--



Passaggio in modo MODE	CNC MODE : MODE - SUBMODE data - ora
------------------------	---

### 6.3.3.5 Eventi registrati alla variazione dei segnali OHLD, OESE, IFHD, FATTOM

Questi eventi vengono monitorati continuamente.

Evento	viene effettuata la seguente registrazione
Variazione di OHLD o OESE o IFHD o FATTOM	STATUS :OHLD=x OESE=x IFHD=x FATTOM=x data - ora

dove "x" può essere "0" o "1"

### 6.3.3.6 Eventi registrati allo START e FINE Part-Program

Le registrazioni di questi eventi sono sempre abilitate e hanno il seguente formato:

Evento	viene effettuata la seguente registrazione
Start Part-Program con CNC in modo Automatico	START PP :NomePartProgram data - ora
Fine esecuzione del Part-Program di cui sopra	END PP : NomePartProgram data - ora

### 6.3.3.7 Allarmi registrati

Questi eventi vengono monitorati continuamente. Per allarmi e attenzioni PLC, viene registrato anche quando l'allarme viene rimosso (con la nota: \*\*\* REMOVED). Per gli allarmi nel cui testo è presente "@a" viene specificata la sigla dell'asse sul quale l'allarme è avvenuto.

Evento	viene effettuata la seguente registrazione
Allarme Software Assi	SA ALARM :ALARNUM - Axis X data - ora
Attenzione Software Assi	SA WARNING:ALARNUM - Axis X data - ora
Allarme Tool Controller	TC ALARM :ALARNUM data - ora
Allarme Part-Program	PP ALARM :ALARNUM data - ora
Allarme PLC	PLC ALARM :ALARNUM data - ora
Attenzione PLC	PLC WARNING:ALARNUM data - ora
Messaggio PLC	PLC MESSAGE:ALARNUM data - ora

A titolo d'esempio, vediamo come sarebbe compilato il file **ECSLOG.TXT** allorché il CNC, in modo Manuale, desse allarme per fuga asse e l'operatore premesse il tasto Reset.

```
CNC MODE : MDI - EXE CNT
05/11/2002 - 08:10:01
```

```
SA ALARM : 32716 - Axis X
05/11/2002 - 08:12:34
```

```
STATUS : OHLD=1 OESE=0 IFHD=0 FATTOM=1
05/11/2002 - 08:12:34
```



---

KEY: RESET  
05/11/2002 - 08:12:50

STATUS: OHLD=0 OESE=0 IFHD=0 FATTOM=1  
05/11/2002 - 08:12:50

CNC MODE :AUTO - EXE CNT  
05/11/2002 - 08:12:50

CNC MODE :MDI - EXE CNT  
05/11/2002 - 08:12:51

### 6.3.4 Gestione Invio Messaggi tramite SMS.

Dalla release software V4.0 è possibile inviare uno o più messaggi del PLC ad uno o più telefoni abilitati alla ricezione di SMS. La funzione richiede un **modem GSM** collegato sulla linea seriale COM1 del CNC.

La configurazione delle comunicazioni è fatta tramite un apposito applicativo ECSP LCSMS.EXE, presente in C:\ECS.CNC\WINWIS\BIN, che permette di verificare la copertura segnale ricevuto dal Modem GSM (in modo da poter regolare opportunamente la posizione del Modem stesso e la direzione dell'antenna), di definire una rubrica di destinatari e selezionare quali "attivare".

Il livello del segnale ricevuto dal Modem, il numero del Centro Servizi e i numeri dei destinatari degli SMS sono visualizzati nella finestra dell'applicazione.

L'applicativo può essere associato ad uno dei tasti funzione del menu orizzontale previsti per il lancio di applicativi esterni ([F8] Tools, [F10] dal menu base) o attivato tramite la funzione IOPLCRunFileExe () descritta più avanti al paragrafo 6.13.1 su comando di un tasto funzione del menu verticale (vedi capitolo 7).

La gestione dell'invio messaggi è realizzata dal PLC tramite la seguente funzione di libreria:

#### 6.3.4.1 IOPLCSendSMS: invia messaggio tramite SMS

Chiamata: **esito = IOPLCSendSMS (PLCAalarmCode)**

Parametro di ingresso:

**PLCAalarmCode** = (unsigned short) Numero di allarme o attenzione da inviare.

Parametro di uscita:

**esito**        **1** : canale occupato  
(short)        **0** : comando inoltrato

I numeri validi per **PLCAalarmCode** sono quelli corrispondenti agli allarmi e attenzioni del PLC:

19600 (ATT00) ÷ 19799 (ATT199)

19800 (ALR00) ÷ 19999 (ALR199)

Il testo che viene inviato è quello definito negli appositi files ALLARxxx.MU (uno per ogni lingua selezionabile), visti al paragrafo 6.3 .

Per verificare la corretta gestione del comando, il PLC può testare i valori assunti dalle variabili associate **IOPLCSTS** e **IOPLCRET** che possono assumere i valori:





---

---

<b>IOPLCSTS</b> (short)	= -1 : Comando inoltrato ad esecutore (task CNHANDLE) = 1 : Operazione in corso di esecuzione = 0 : Operazione terminata
<b>IOPLCRET</b> (short)	>= 0 : OK Comando terminato correttamente; numero di messaggi trasmessi = -1 : Errore apertura porta seriale = -2 : Errore di comunicazione sulla porta seriale = -3 : Livello segnale ricevuto dal modem insufficiente = -11 : Scheda SIM non pronta = -12 : Modem non registrato alla rete = -13 : Modem non registrato al Centro Servizi = -14 : Numero del Centro Servizi non presente o non valido = -15 : Modem non in formato testo = -12 : Modem non registrato alla rete = -101 : Errore accesso a file destinatari SMS (RUBRICA.ECS) = -102 : Errore lettura testo allarme corrispondente a "PCAlarmCode"

Il PLC deve quindi testare, nell' ordine:

1. **esito** : se = 1 chiamare nuovamente la routine; se =0 passare alla fase successiva
2. **IOPLCSTS** : se =0 passare alla fase successiva altrimenti attendere
3. **IOPLCRET**: se =0 successo altrimenti segnalare l'errore all'operatore

## 6.4 GESTIONE CONVERTITORI A/D E D/A

### 6.4.1 Convertitori A/D

Sulle schede I/O dei CNC ECS sono presenti alcuni convertitori Analogici/Digitali (A/D) per acquisire delle tensioni esterne e per la successiva elaborazione da parte del PLC. Altri convertitori A/D sono presenti sul modulo IKR (Interfaccia Keyboard Remota) e sull'unità centrale BASIC. Sono tipicamente usati per input di sonde termiche o per i potenziometri di override quando non si usano (solo) quelli presenti sulla consolle del CNC.

Il manuale "CONNESSIONI HARDWARE" ne specifica la quantità ed il tipo presente sui vari tipi di apparato o scheda.

Nei files PLC\_SYN.INC, DF\_IOLOC.H, DF\_IOEXP.H, DF\_IOIKR.H, DF\_IOCAN.H, DF\_IOBAS.H sono definiti i sinonimi standard proposti da ECS.

```
/*
*****
/*      MODULO BASIC 080H259/269/459      */
/*      4 convertitori A/D ( 12 bit )      */
/*      2 fast-input digitali optoisolati */
/******
/*      INPUT Morsettiera AN              */
/* ----- */
/* Convertitori A/D                       */
/* Ingressi: Pin 1-2-3-4                  */
/* Comune:   Pin 5 (0V Analogico)         */
/* Definiz:  12 Bits                      */
/* Range:    -10 Volt(-2048)...+10V(+2047) */
/*                                               */
/* Non ridefinire e lasciare commentati   */
/* Definiti in df_syn.inc                 */
/* ----- */
```



```

/* #define    IAD0      pn_iosign...Digital[0] */
/* INGRESSO PRIMO CONVERTITORE A/D          */
/*                                           */
/* #define    IAD1      pn_iosign...Digital[1] */
/* INGRESSO SECONDO CONVERTITORE A/D        */
/*                                           */
/* #define    IAD2      pn_iosign...Digital[2] */
/* INGRESSO TERZO CONVERTITORE A/D          */
/*                                           */
/* #define    IAD3      pn_iosign...Digital[3] */
/* INGRESSO QUARTO CONVERTITORE A/D         */
/*                                           */
/* ----- */

/*=====*/
/* DOCK STATION    080H229/249              */
/* MODULO ESPANSIONE I/O BASIC              */
/* PER MAX QUATTRO SCHEDE I/O 040F621      */
/*=====*/
.....
/******/
/* PRIMA SCHEDA I/O 040F621                */
/* (Jumper I/O = 0)                        */
/*                                           */
/* 48 input digitali optoisolati           */
/* 8 convertitori A/D ( 8 bit )            */
/* 32 output tipo High-side (0.5A)         */
/******/
.....
/*----- */
/* Connettore X2 (Convertitori A/D)        */
/* Ingressi: Pin 25 ... 32                  */
/* Comuni:   Pin 33-34-36 (0V Analogico)    */
/* Definiz:  8 Bits                          */
/* Range:    0 Volt (0) ... +10V (+255)     */
/* ----- */
/* WORD R3...R6 (Jumper I/O = 0)           */
/* ----- */
#define      AD0      R3.byte.l
/* X2.25 - Libero */
#define      AD1      R3.byte.h
/* X2.26 - Libero */
#define      AD2      R4.byte.l
/* X2.27 - Libero */
#define      AD3      R4.byte.h
/* X2.28 - Libero */
#define      AD4      R5.byte.l
/* X2.29 - Libero */
#define      AD5      R5.byte.h
/* X2.30 - Libero */
#define      AD6      R6.byte.l
/* X2.31 - Libero */
#define      AD7      R6.byte.h
/* X2.32 - Libero */
/* ----- */

```



```
/*=====*/
/* MODULO IKR: INTERFACCIA KEYBOARD REMOTA */
/*=====*/
.....
/*****/
/* Connettore CN7      (Convertitori A/D) */
/*  uso tipico potenziometri */
/* Ingressi: Pin 1 ... 8 */
/* Comuni:   Pin  9-10-11 (0V Analogico) */
/* Comuni:   Pin 13-14-15 (5 Volt dc) */
/* Definiz:   8 Bits */
/* Range:    0 Volt (0) ... +5V (+255) */
/*****/
#define          ANALOG1          RS7.byte.l
/* CN7.1 - Libero */
#define          ANALOG2          RS7.byte.h
/* CN7.2 - Libero */
#define          ANALOG3          RS8.byte.l
/* CN7.3 - Libero */
#define          ANALOG4          RS8.byte.h
/* CN7.4 - Libero */
#define          ANALOG5          RS9.byte.l
/* CN7.5 - Libero */
#define          ANALOG6          RS9.byte.h
/* CN7.6 - Libero */
#define          ANALOG7          RS10.byte.l
/* CN7.7 - Libero */
#define          ANALOG8          RS10.byte.h
/* CN7.8 - Libero */
/*****/
```

## 6.4.2 Convertitori D/A

Sulle schede assi sono disponibili fino a 12 convertitori Digitali/Analogici (D/A); tramite opportune tarature definite nel file SWCNF.TAR (vedi MANUALE TARATURE 720P385), si può decidere l'appartenenza del singolo convertitore al CNC od al PLC. Se utilizzati dal PLC, sono indirizzati come uscite **VDA0, VDA1 ... VDA11**.

Possono essere usati per alimentare semplicemente un potenziometro o per comandare un azionamento mandrino gestito completamente dal PLC.

Ogni convertitore può fornire una tensione continua tra -10 e +10 Volt in corrispondenza dei valori numerici compresi tra -32768 e +32767.

```
/*-----
ESEMPIO DI GESTIONE MANDRINO "DA PLC"
-----*/
if(NEWS && OTDS <= 30000)
{
  SPEED= OTDS * 1.092;
  /* Calcolo speed per convertitore D/A */
  /* 1.092 = 32767/30000 (giri max.) */
}
ASPEED= SPEED / 100 * IOVS;
/* Calcola override */
```



```

if(ASPEED > 32700)
  ASPEED = 32700;
  /* Forza la max. velocita */
if(OEXE && M03)
  VDA1= (int)ASPEED;
  /* Emette riferimento positivo su convertitore D/A */
if(OEXE && M04)
  VDA1= (int)-ASPEED;
  /* Emette riferimento negativo su convertitore D/A */
if(OHLD)
  VDA1=0;
  /* Azzerà riferimento in Hold */
if((NEWM && (OM == 5)) || ORST || ALR81 || ALR82 || ALR86)
{
  VDA1=0;
  ASPEED = 0;
  SPEED = 0;
  /* Azzerà riferimento su uscita e appoggi */
}
/*-----*/

```

### 6.4.3 Bufferizzazione Convertitori A/D

Tramite apposite funzioni della libreria PLC, descritte di seguito, il PLC ha la possibilità di abilitare una bufferizzazione, ossia un campionamento, su di **un canale A/D specificato**.

Per queste funzioni, per **numero di canale** si intende il numero sequenziale, a partire da 1 ed a partire dalla prima scheda I/O, del convertitore a 16 bit, oppure della coppia di convertitori ad 8 bit, su cui operare.

La scheda CPU486 possiede 4 convertitori A/D a 12 bits:

range:-10 Volt(-2048)...+10V(+2047)

La scheda I/O 040F421/621 possiede 8 convertitori A/D a 8 bits:

range: 0 Volt (0) ... +10V (+255)

Il modulo BASIC possiede 4 convertitori A/D a 12 bits:

range:-10 Volt(-2048)...+10V(+2047)

Ad esempio, con una scheda I/O 040F421/621, oltre alla scheda base CPU486-I/O 040F481/483, si ha:

scheda CPU486-I/O A/D 12 bit		scheda I/O 040F421/621 A/D 8 bit	
Can.	sinonimo PLC	Can.	sinonimi PLC
1	ADW0	5	AD0 - AD1
2	ADW2	6	AD2 - AD3
3	ADW4	7	AD4 - AD5
4	ADW6	8	AD6 - AD7

Il campionamento dei dati avverrà internamente al CNC ad una frequenza multipla di quella della logica PLC; più precisamente ad una frequenza equivalente al loop di chiusura del ciclo di spazio sugli assi.

Sui CNC ECS serie WIN, tutte le basi dei tempi sono funzione di una variabile "TCYC" (Servo Loop Cycle Time, tempo di campionamento del CNC), espressa in secondi, impostata nel file di taratura SWCNF.TAR. La logica PLC viene invece elaborata con una cadenza multipla del "TCYC", definita tramite la variabile TAUPC (e da KT), anch'essa presente in SWCNF.TAR.

Con TCYC = 0.003 (3 msec.), KT= 2 e con TAUPC= 3 si ha:

Servo Loop Cycle Time (controllo assi)      3 msec



---

---

TAU del CNC (interpolatore)	TAU= TCYC * KT (6 msec)
PLC Cicle Time (aggiornamento I/O PLC)	TCYC * KT * TAUPLC (18 msec)
<b>campionamento del convertitore A/D</b>	<b>3 msec</b>

Con questa configurazione vengono quindi campionati 6 valori per ogni loop del PLC.

#### 6.4.4 Funzioni per bufferizzazione convertitori A/D

##### 6.4.4.1 start\_buf\_ad: attiva la bufferizzazione

Chiamata: **start\_buf\_ad (ch\_ad)**

Parametro di ingresso:

**ch\_ad** = (unsigned short) numero del canale A/D sul quale attivare la bufferizzazione.

##### 6.4.4.2 stop\_buf\_ad: disattiva la bufferizzazione

Chiamata: **stop\_buf\_ad ()**

Non richiede parametri.

##### 6.4.4.3 read\_buf\_ad: legge i valori bufferizzati dal CNC

Chiamata: **pbufad = read\_buf\_ad ( &typead, &nsample)**

Parametri di ingresso:

**typead** = (unsigned short \*) Indirizzo della parola nella quale la routine ritorna il tipo di campionamento effettuato:

- 0**      Campionamento 8 bit
- 1**      Campionamento 12 bit

**nsample** = (unsigned short \*) Indirizzo della parola nella quale la routine ritorna il numero di campionamenti effettuati dalla precedente chiamata (o dall'attivazione). Un valore > 32 significa buffer overflow, ossia dati persi.

Parametro di uscita:

**pbufad** = (void \*) Puntatore al buffer dati acquisiti.

La funzione ritorna in **pbufad** l'indirizzo del buffer contenente gli **nsample** campionamenti effettuati dalla precedente chiamata (o dall'attivazione). Se **nsample** > MAX\_AD\_SAMPLE (32), il buffer conterrà solamente MAX\_AD\_SAMPLE dati campionati, in quanto gli altri sono andati persi a causa di overflow.

Con **typead** = 1 il ritorno é un puntatore ad un buffer di **nsample** dati short (conv. A/D 12 bit)

Con **typead** = 0 il ritorno é un puntatore ad un buffer di **nsample** word, di cui il byte leggero contiene il valore campionato del convertitore A/D 8 bit "pari", ed il byte pesante contiene il valore campionato del convertitore A/D 8 bit "dispari".

Un ritorno = 0 (NULL) significa nessun campionamento attivo; in questo caso viene posto anche typead = 0 e nsample = 0.

La funzione **start\_buf\_ad()** va chiamata una sola volta (ad esempio al **pon**).

L' eventuale **stop\_buf\_ad()**, può essere chiamata quando necessario.

La funzione **read\_buf\_ad()** va invece chiamata ad ogni loop del PLC (nsample sarà un valore costante dipendente da TAUPC) pena la perdita di dati; il CNC riparte infatti a memorizzare i nuovi dati da inizio buffer ad ogni tic del PLC.



**ATTENZIONE:** l'attivazione della bufferizzazione di un canale A/D rende non disponibile la lettura diretta di tutti i convertitori della scheda relativa sugli indirizzi standard (ADWn o ADn).

## 6.5 Funzioni per il Part-Program

### 6.5.1 sel\_pp: selezione Part-Program

Scopo: controlla il nome del Part-Program desiderato ed eventualmente lo seleziona per l'esecuzione sul CNC.

Chiamata: **esito = sel\_pp (\*indbuf, selflag)**

Parametri di ingresso:

**\*indbuf** = (char \*) nome del Part-Program da selezionare

**selflag** = (short) tipo di selezione

= 0 controlla se il nome del Part-Program desiderato è corretto

= 1 seleziona il Part-Program

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

**0**      Successo

**!=0**     Errore

Dalla V3.01, il nome del Part-Program può contenere anche il path completo; es.:

```
esito = sel_pp("C:\\ECS.CNC\\PROGRAM\\ARCHIVIO_0001\\PROGR01", 1)
/* selezione programma PROGR01 di PROGRAM/ARCHIVIO_0001 */
```

**ATTENZIONE:** la funzione non controlla se il Part-Program esiste o meno, semplicemente verifica se la stringa è sintatticamente valida e se lo stato attuale del CNC permette di effettuare un cambio di selezione. Se il file non esiste, interverrà l'apposito allarme al momento dello START.

```
/*-----
ESEMPIO DI UTILIZZO SEL_PP()
-----*/
if( espressione )
{
esito = sel_pp("PROGR1", 1)
/* selezione programma */
}
if (esito == 0)
{
if (!OHLDD && !STBCOD)
{
CODICE=0xB2;
STBCOD=1;
}
/* si provvede a dare START al CNC */
}
STBCOD= STBCOD && !PASEXT ;
```



---

```
/* Strobe tasto esterno fino a risposta CNC */  
tastcnc();  
/* routine standard gestione tastiera */  
/*-----*/
```

### 6.5.2 GetPPSel: leggi il nome del Part-Program selezionato

Scopo: legge il nome del Part-Program attualmente selezionato sul CNC.

Chiamata: **esito = GetPPSel(\*nomfile)**

Parametro di ingresso:

**\*nomfil** = (char \*) nome dell'array di char dove "depositare" il nome del Part-Program attualmente selezionato. La dimensione dell'array deve essere tale da poter contenere il nome di Part-Program più lungo (dalla V3.01 compreso il path; per la dimensione si può usare la costante MAX\_PATH).

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

<b>0</b>	Successo
<b>!=0</b>	Errore

### 6.5.3 GetExecutionTime: legge il tempo di esecuzione

Scopo: Legge il tempo di esecuzione del Part-Program. Lo start al tempo è dato con la istruzione <TIM:ON> scritta nel Part-Program stesso.

Chiamata: **timepp = GetExecutionTime( )**

Parametro di uscita:

**timepp** = (double) tempo, in secondi, di esecuzione del Part-Program

### 6.5.4 IOPLCBlockPPSel: blocco/sblocco del Part-Program

Scopo: Blocca/Sblocca la possibilità di selezione un nuovo Part-Program. Inoltre Blocca/Sblocca l'editing del file del Part-Program attualmente selezionato, anche se non è in esecuzione. Può essere usata in congiunzione con sel\_pp() per forzare la scelta del Part-Program da eseguire ed impedirne la modifica.

Chiamata: **esito = IOPLCBlockPPSel(blclflag)**

Parametro di ingresso:

**blclflag** = (short) tipo di comando

= 1 blocca il Part-Program attualmente selezionato.

= 0 sblocca il file del Part-Program precedentemente bloccato

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

<b>0</b>	Successo: comando inoltrato
<b>1</b>	Errore: altro comando già in corso

Per una corretta gestione del comando, il PLC deve testare anche i valori assunti dalle variabili associate **IOPLCSTS** e **IOPLCRET** che possono assumere i valori:

**IOPLCSTS** = -1 : Comando inoltrato ad esecutore (task WINVIS)



(short) = 0 : Operazione terminata  
= 1 : Operazione in corso di esecuzione

**IOPLCRET** = 1 : OK Comando terminato correttamente  
(short) < 0 : KO : Errore

Il PLC deve quindi testare, nell'ordine:

1. **esito** : se = 1 chiamare nuovamente la routine; se =0 passare alla fase successiva
2. **IOPLCSTS** : se =0 passare alla fase successiva altrimenti attendere
3. **IOPLCRET**: se = 1 successo altrimenti segnalare l'errore all'operatore

### 6.5.5 IOPLCSelectGraphPP: selezione del Part-Program di Grafica

Scopo: Seleziona sul CNC, l'ultimo Part-Program selezionato in ambiente grafica o code&show.

Chiamata: **esito = IOPLCSelectGraphPP()**

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

**0** Successo: comando inoltrato  
**1** Errore: altro comando già in corso

Per una corretta gestione del comando, il PLC deve testare anche i valori assunti dalle variabili associate **IOPLCSTS** e **IOPLCRET** che possono assumere i valori:

**IOPLCSTS** = -1 : Comando inoltrato ad esecutore (task WINVIS)  
(short) = 1 : Operazione in corso di esecuzione  
= 0 : Operazione terminata

**IOPLCRET** = 1 : OK Comando terminato correttamente  
(short) < 0 : KO : Errore  
= -1 Non esiste Part-Program selezionato in grafica o code&show  
= -2 Errore nella lettura del nome file  
= -11 Il file Part-Program è bloccato dal PLC  
= -12 Nome del Part-Program non corretto

NOTE:

Il Part-Program viene selezionato anche se nel frattempo il file è stato cancellato dal direttorio LAV

Il Part-Program viene selezionato anche se sul CNC è in esecuzione un altro programma.

### 6.5.6 IsMtXOn: Test matrice dinamica

Scopo: Verifica se c'è una matrice dinamica attiva sul CNC.

Chiamata: **esito = IsMTXOn()**

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

**0** Nessuna matrice dinamica attiva  
**1** Matrice dinamica attiva





### 6.5.7 Cambio o modifica Part-Program

Dalla release software V4.0 è presente un nuovo segnale per avvisare il PLC quando viene selezionato un nuovo Part-Program oppure quando quello selezionato viene modificato. Questo segnale può essere utile per lanciare la lettura del part-program attuale e vedere il percorso utensile nel formato RTG (Real Time Graphic).

<b>PARTPSTS</b> (short)	<b>PLC → CNC</b>	<p>=1 Selezionato nuovo part-program =99 Il part-program attuale è stato modificato con CNC in esecuzione e quindi è stato generato un file temporaneo. Ora, essendo terminata l'esecuzione del part-program, il CNC rende attive le modifiche (il file temporaneo nuovo è reso attuale).</p> <p>I seguenti codici sono emessi durante la sessione di editing del p.p. selezionato, solo quando questo non è in esecuzione/lettura sul CNC.</p> <p>=11 Modifica generica sul file. Emesso durante l'esecuzione della modifica =12 Eseguito salva file [CTRL-S], durante la sessione di editor. =13 Uscita da editor salvando modifiche =14 Uscita da editor senza salvare le modifiche fatte. =15 Uscita da editor senza aver fatto nessuna modifica.</p>
----------------------------	------------------	---

**Note:**

1. Il CNC, ad ogni nuovo evento, sovrascrive la word PARTPSTS.
2. Il valore di PARTPSTS permane anche con OFF/ON CNC.
3. E' compito del PLC, memorizzare lo stato della word PARTPSTS, ed **azzerarla**, dopo tale operazione.

### 6.6 KEY

La funzione KEY permette il lancio di sottoprogrammi ( di nome **KEYnn** ) scritti con i normali linguaggi di programmazione ISO e LIP. I KEY impostabili sono 100 ( da KEY00 a KEY99 ). I sottoprogrammi KEYnn debbono essere messi nella directory **C:\ECS.CNC\LA\MACRO** per evitare che vengano modificati dall'utilizzatore.

I KEY sono funzionalmente distinti in tre categorie o tipi:

- IMMEDIATO
- ASINCRONO
- SINCRONO

La distinzione di tipo avviene attraverso tre distinti segnali di richiesta.

<b>RKEYIM</b>	<b>PLC → CNC</b>	Il PLC richiede consenso all'inoltro di un KEY IMMEDIATO. Tale richiesta deve essere fatta quando non è in corso nessun key di tipo immediato (!KEYIM && !OKEYIM).
<b>RKEYSN</b>	<b>PLC → CNC</b>	Il PLC richiede consenso all'inoltro di un KEY SINCRONO. Tale richiesta deve essere fatta quando non è in corso nessun key di tipo sincrono (!KEYSN && !OKEYSN).



<b>RKEYAS</b>	<b>PLC → CNC</b>	Il PLC richiede consenso all'inoltro di un KEY ASINCRONO. Tale richiesta deve essere fatta quando non é in corso nessun key di tipo asincrono (!IKEYAS && !OKEYAS).
---------------	------------------	---

Il CNC segnala il consenso ad eseguire il KEY richiesto tramite i tre segnali di risposta:

<b>OKEYIM</b>	<b>CNC → PLC</b>	Il CNC dà il consenso all'inoltro di un KEY IMMEDIATO. Tale consenso rimane a 1 fino a quando permane la richiesta. OKEYIM permane =1 fino a che il key NON è stato totalmente eseguito dal CNC.
<b>OKEYSN</b>	<b>CNC → PLC</b>	Il CNC dà il consenso all'inoltro di un KEY SINCRONO. Tale consenso rimane a 1 fino a quando permane la richiesta. OKEYSN permane =1 fino a che il key NON è stato totalmente eseguito dal CNC.
<b>OKEYAS</b>	<b>CNC → PLC</b>	Il CNC dà il consenso all'inoltro di un KEY ASINCRONO. Tale consenso rimane a 1 fino a quando permane la richiesta. OKEYAS permane =1 fino a che il key NON è stato totalmente eseguito dal CNC.

Il PLC rende esecutivo il KEY tramite uno dei 100 segnali di attivazione:

<b>KEY00</b> -:- <b>KEY99</b>	<b>PLC → CNC</b>	Il PLC richiede l'esecuzione di un KEY. Tale richiesta deve essere fatta quando sono attive sia la richiesta RKEYxx sia la risposta OKEYxx corrispondente.
-------------------------------------	------------------	--

Il CNC mette in esecuzione, a questo punto, il sottoprogramma KEYnn selezionato.

### 6.6.1 Regole per la scrittura del sottoprogramma KEY:

- Nome file: **KEYnn** dove nn=numero del KEY (0 ÷ 99).
- Omettere nel nome lo 0 non significativo (es. scrivere KEY1 e non KEY01).
- Nel programma KEY è facoltativa la numerazione dei blocchi.
- Omettere i caratteri inizio e fine sottoprogramma ( %, M02, M17, <RET> ).
- Usare un'istruzione <SET:nnn> (1000 < nnn ≤ 29999, i valori da 0 a 1000 sono riservati a ECS) per comunicare al PLC il termine dell'esecuzione del sottoprogramma.

### 6.6.2 KEY Immediati

Possono essere eseguiti in qualsiasi momento, indipendentemente dal fatto che un programma di lavoro sia o meno in esecuzione, e non hanno alcun sincronismo.

L'esecuzione della subroutine KEY è istantanea, e non lascia traccia sul programma di lavoro.

Il programma KEY immediato **può contenere** istruzioni del tipo:

- Assegnazioni e calcoli con variabili **# , R , V**.
- Richiamo di sottoprogrammi **L<....>**.
- Istruzioni e comandi generici (anche di movimento) riferiti ad assi indipendenti

**Non può invece contenere:**

- Emissione di funzioni ausiliarie di qualsiasi tipo
- Movimenti di assi CNC (sono comunque accettate istruzioni SAX)



- Istruzioni G, <WAT:...>, <FRE:...>, <RPT:...>, <TRU:...>, <FLS:...>, <IFF:...>, <JMP:...>.

Per l'esecuzione di un KEY Immediato è richiesto l'input specifico :

**RKEYIM = "1"** : PLC richiede l'esecuzione del key

Il CNC onorerà tale richiesta rispondendo in output :

**OKEYIM = "1"** : key immediato pronto o in corso

Esempio :

- Su PLC

```
RKEYIM= !IKEYIM && !OKEYIM && !RKEYIM && logica
      || RKEYIM && !(PSET && (OVGENL== 2010));
/* RICHIESTA KEY IMMEDIATO */
KEY10= OKEYIM && RKEYIM ;
/* LANCIATA ESECUZIONE KEY */
```

Testo KEY10 sotto C:\ECS.CNC\LAV\MACRO

```
N0 { MEMORIZZA VALORE FC- ASSE X }
N10
N20 <LCK:OFF>
N30 <#340=V5075/0>
N40 <LCK:ON>
N50 <SET:2010>
```

### 6.6.3 KEY Sincroni

Non esistono limitazioni circa le istruzioni programmabili. L'esecuzione di questo KEY avviene normalmente sincronizzandosi con una F.A. scritta nel programma di lavoro, definita **WAIT** nel file di taratura **FA.TAB**

Per facilitare la comprensione esaminiamo il seguente esempio supponendo di eseguire il sottoprogramma KEY1 attraverso la F.A. M122:

Programma principale	KEY1
<pre>... N100 X100 Y50 F200 N110 M122 {ATTIVA KEY1}  N120 X200 Y300 ...</pre>	<pre>N10 X30 Y30 M70 ... N50 X50 Y10 F50</pre>

Al ritorno nel programma principale gli assi si trovano a X50 Y10 e la Feed vale 50.

Si precisa inoltre che se la prima parte del programma (fino alla N110) viene eseguito in ricerca o lettura, il CNC richiederà il riposizionamento alle quote X50, Y10 (richieste dal KEY). Se la F.A. M70 è stata definita riposizionabile ( nel file FA.TAB), il CNC richiederà il riposizionamento anche di quest'ultima.

Per l'esecuzione di un KEY sincrono è richiesto l'input specifico :

**RKEYSN = "1"** : Il PLC richiede esecuzione del KEY

Il CNC onorerà tale richiesta rispondendo in output :



**OKEYSN = "1"** : KEY Sincrono pronto o in corso

#### 6.6.4 KEY Asincroni

L'esecuzione di questo KEY non avviene in modo sincrono al programma eventualmente in esecuzione. Con riferimento all'esempio sotto riportato, se la richiesta di esecuzione viene inviata dal PLC durante l'esecuzione della N100, può darsi che il CNC esegua la subroutine quando si è arrivati alla N120, per effetto del meccanismo di lettura in avanti dei blocchi di programma.

L'esecuzione del KEY avviene tenendo conto delle funzioni modali programmate precedentemente. Quando si ritorna ad eseguire il programma principale, non vengono tenute in considerazione le funzioni modali definite nel sottoprogramma KEYnn.

Se durante il KEY sono stati mossi gli assi e non sono stati riportati nel punto in cui si trovavano, il CNC dà allarme di riposizionamento.

Permane solo l'effetto delle istruzioni <CNC:...>, <FRE:...>, M02 e M30.

Il KEY ASINCRONO forza automaticamente G40 e ripristina in modo automatico alla fine della sua esecuzione G41 o G42.

Programma principale	KEY1
...	
N100 X100 Y50 F200	
N110 M03 S1000	N10 X30 Y30 M70
...	...
...	...
...	N50 X50 Y10 F50
N120 X200 Y300	
...	

Per l'esecuzione di un KEY Asincrono è richiesto l'input specifico :

**RKEYAS = "1"** : Il PLC richiede esecuzione del KEY

Il CNC onorerà tale richiesta rispondendo in output :

**OKEYAS = "1"** : KEY Asincrono pronto o in corso

Dopo che il PLC ha fatto la richiesta di un KEYxx una apposita routine di sistema provvede a passare tale richiesta al CNC attraverso gli inputs :

**IKEYIM = "1"** Richiesta esecuzione key immediato

**IKEYSN = "1"** Richiesta esecuzione key sincrono

**IKEYAS = "1"** Richiesta esecuzione key asincrono

#### 6.6.5 Note importanti

- Tutti i tipi di KEY non possono eseguire CAMBI MODO.
- Quando vengono ricevuti dal CNC più richieste di KEY dello stesso tipo, viene presa in considerazione la prima; se ne vengono inoltrate di tipi diversi contemporaneamente, vale la seguente priorità:

**IMMEDIATO, SINCRONO, ASINCRONO.**



- Con [RESET] il CNC si inizializza e annulla la propria memorizzazione di sottoprogramma KEY in corso, pertanto è opportuno che il PLC annulli la propria richiesta.

### 6.6.6 Precisazioni sull'uso di KEY

- Il CNC prende in considerazione un KEY con la transizione da 0 a 1 di una delle tre richieste (RKEYIM, RKEYAS, RKEYSN).
- Il CNC interrompe l'esecuzione del KEY quando sente la transizione da 1 a 0 della richiesta, terminando comunque l'esecuzione del record in corso.
- Il PLC deve rendere vera una sola richiesta alla volta.
- Il testo del sottoprogramma KEY viene sempre eseguito dal CNC in ESECUZIONE CONTINUA (EC).
- Con una delle richieste di KEY attiva, il CNC accetta dalla tastiera o dal PLC solo i seguenti tasti: [REL], [HOLD], [AXESH], [RESET], mentre, solo dal PLC, anche i seguenti tasti : [FAST], [MADIG]
- Il sottoprogramma KEY98 è utilizzato da E.C.S. per uso interno. Si consiglia di usare i KEY a partire da KEY0.

## 6.7 CONTROLLO DELLO STATO DEL CNC

### 6.7.1 is\_cnc\_work: testa se il CNC sta lavorando

Chiamata: **esito = is\_cnc\_work ()**

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

<b>1</b>	Il CNC sta lavorando
<b>0</b>	Il CNC è a riposo

Testa se il CNC sta lavorando.

Questa condizione è vera quando il CNC è in esecuzione, oppure c'è un part-program sospeso a causa di esecuzione singola o di cambi modo, oppure il CNC non ha ancora svuotato i buffers interni di interscambio oppure è in esecuzione un KEY.

### 6.7.2 cnc\_all: mette il CNC in Allarme

Chiamata: **cnc\_all (numall)**

Parametro di ingresso:

**numall** = (short) numero di allarme.

Normalmente il PLC usa gli allarmi (ALR00 ÷ ALR199) ed attenzioni (ATT00 ÷ ATT199) a lui riservati e che saranno visualizzati nella finestra dedicata al PLC; questi allarmi e attenzioni non mettono automaticamente in Hold il CNC e non richiedono RESET. Usando invece la **cnc\_all()**, il testo sarà visualizzato nella finestra normalmente dedicata agli allarmi gestiti dal CNC, verrà automaticamente forzato l'Hold e sarà necessario effettuare un RESET. I codici **numall** da usare sono comunque quelli riservati al PLC



per evitare conflitti con quelli del CNC; i testi da visualizzare debbono essere scritti, come al solito, nei files ALLARxxx.MU presenti sul direttorio di sviluppo del PLC (vedere il manuale 720P394 "AMBIENTE e PROCEDURE di SVILUPPO e DEBUG PLC).

### 6.7.3 IOPLCReadCNCLanguage: legge la Lingua attuale del CNC

Scopo: Conoscere nel PLC qual'è la lingua attiva sul CNC.

Chiamata: **esito = IOPLCReadCNCLanguage ();**

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |                                    |
|----------|------------------------------------|
| <b>0</b> | Successo: comando inoltrato        |
| <b>1</b> | Errore: altro comando già in corso |

Per una corretta gestione del comando, il PLC deve testare anche i valori assunti dalle variabili associate **IOPLCSTS** e **IOPLCRET** che possono assumere i valori:

<b>IOPLCSTS</b>	= -1 : Comando inoltrato ad esecutore (task WINVIS)
(short)	= 0 : Operazione terminata
	= 1 : Operazione in corso di esecuzione

<b>IOPLCRET</b>	= 0 : Selezionato Italiano
(short)	= 1 : Selezionato Inglese
	= 2 : Selezionato Francese
	= 3 : Selezionato Spagnolo
	= 4 : Selezionato Tedesco

Il PLC deve testare nell' ordine:

1. **esito** : Se = 1 chiamare nuovamente la routine . Se =0 passare a fase successiva
2. **IOPLCSTS**: Se =0 passare a fase successiva
3. **IOPLCRET**: Codice lingua selezionata

### 6.7.4 IOPLCShutDown: "spegne" il CNC

Scopo: Comandare da PLC lo shut-down del CNC (che deve, normalmente, essere comandato dall'operatore tramite l'apposito tasto funzione del menu).

Chiamata: **esito = IOPLCShutDown ();**

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- |          |                                       |
|----------|---------------------------------------|
| <b>0</b> | Successo: comando inoltrato           |
| <b>1</b> | Errore: altro comando ancora in corso |

La funzione può essere usata per arrestare tutto alla fine del Part-Program in corso. L'operatore sceglie questa opzione tramite, ad esempio, un tasto funzione del menu verticale; il PLC testa quindi l'emissione della M02 di fine Part-Program, spegne gli ausiliari della Macchina e poi chiama questa funzione che arresta il CNC. Se occorre spegnere tutto il quadro elettrico della Macchina, è necessario disporre di un timer hardware che agisca, dopo un po' di tempo, sull'interruttore generale.



## 6.8 SINCRONIZZAZIONI CNC↔PLC senza arresto sul blocco

Vengono messe a disposizione **10 parole (160 bit)** di sincronizzazione.

Le parole vengono azzerate dal Controllo Numerico all'accensione ed al passaggio ALT⇒EXE.

I bit vengono controllati da Part-Program tramite le seguenti istruzioni LIP:

```
<OUT: Numero parola; Numero bit; Numero bit; ...>
```

Setta ad 1, ad **inizio blocco interpolato**, i bit specificati nei campi "Numero bit" della parola di output specificata nel campo "Numero parola".

Non è consentito programmare più parole in uno stesso blocco di programma, ma è consentito programmare più bit per ogni parola.

Numero parola può variare da 0 a 9.

Numero bit può variare da 0 a 15.

```
<TOU: Numero parola; Numero bit; Variabile Esito>
```

Testa il bit specificato nel campo "Numero bit" della parola di output specificata nel campo "Numero parola" e ne ritorna l'esito (0-1) nella "Variabile Esito".

Numero parola può variare da 0 a 9.

Numero bit può variare da 0 a 15.

La "Variabile Esito" può essere una qualsiasi tra le variabili a disposizione del Controllo Numerico: #, %#, %%#, R.

Se la variabile è di tipo "#" occorre che il CNC sia predisposto con lo sbloccaggio, ossia senza il write-protect (<LCK:OFF>).

I bit vengono controllati da PLC tramite le seguenti funzioni di libreria:

### 6.8.1 tst\_out: testa i bit delle parole di sincronizzazione CNC↔PLC

Chiamata: **esito = tst\_out (num\_word, num\_bit)**

Parametri di ingresso:

**num\_word** = (unsigned short) Numero di parola da testare (da 0 a 9).

**num\_bit** = (unsigned short) Numero di bit da testare nella parola (da 0 a 15).

Parametro di uscita:

**esito** = (short) valore di ritorno della routine . Può assumere uno dei seguenti valori:

0	Successo – stato del bit = 0
1	Successo – stato del bit = 1
-1	Errore sul primo parametro (numero di word errato)
-2	Errore sul secondo parametro (numero di bit errato)

### 6.8.2 zero\_out: azzerare i bit delle parole di sincronizzazione CNC↔PLC

Chiamata: **esito = zero\_out (num\_word, num\_bit)**

Parametri di ingresso:

**num\_word** = (unsigned short) Numero di parola da azzerare (da 0 a 9).

**num\_bit** = (unsigned short) Numero di bit da azzerare nella parola (da 0 a 15).

Parametro di uscita:



**esito** = (short ) valore di ritorno della routine . Può assumere uno dei seguenti valori:

- 0**            Successo – azzerato il bit specificato
- 1**          Errore sul primo parametro (numero di word errato)
- 2**          Errore sul secondo parametro (numero di bit errato)

In conclusione, i bit delle parole vengono posti ad "1" dal CNC ad inizio blocco interpolato <OUT:...> ed azzerati solamente dal PLC, che perciò può effettuare una corretta sincronizzazione sull'evento desiderato senza arrestare il lavoro del CNC, come farebbero le normali Funzioni Ausiliarie . È altresì possibile "sentire" lo stato dei bit sia da Part-Program, tramite istruzioni LIP <TOU:...>, sia da PLC, tramite `tst_out(...)`.

## 6.9 OPZIONI SOFTWARE

Alcune funzioni dei CNC ECS serie WIN sono disponibili solo se è presente ed abilitata la corrispondente OPZIONE. Il PLC può controllare quali opzioni sono abilitate, per eseguire o saltare l'esecuzione di quella parte di logica legata alle opzioni suddette. In questo modo è possibile scrivere la logica del PLC per la macchina più completa ma eseguire run-time solo quella che serve.

### 6.9.1 rd\_opz: legge le word OPZIONI presenti sul CNC

Chiamata: **esito = rd\_opz (num\_word)**

Parametro di ingresso:

**num\_word** = (unsigned short) Numero della parola OPZIONI da testare (da 0 a 9).

Parametro di uscita:

**esito** = (unsigned short ) ritorno della routine: valore della parola desiderata.

La corrispondenza dei valori di ritorno con le opzioni presenti può essere vista tramite la procedura lanciata dal tasto [Carica] presente nel menù di service del CNC.

Per il PLC, sono presenti alcuni segnali già "decodificati" che sono stati associati ai sinonimi qui di seguito elencati:

WOPZCN#	CNC → PLC	Parola opzioni n. # (0 . . . 3) Uguali a quanto letto con <b>rd_opz()</b>
<b>OP_TWT{F}</b>	<b>CNC → PLC</b>	Opzione RTCP (TWIST) 4 o 5 assi
<b>OP_DGT{F}</b>	<b>CNC → PLC</b>	Opzione COPIA (Digitalizzazione).
<b>OP_GAN</b>	<b>CNC → PLC</b>	Opzione ASSI GANTRY
<b>OP_DRF</b>	<b>CNC → PLC</b>	Opzione volantini DRF (Differential Resolver Function). Vedi Capitolo 2 ASSI, paragrafo VOLANTINI
<b>OP_03</b>	<b>CNC → PLC</b>	Opzione Compensazione errori mutui Assi
<b>OP_AX#</b>	<b>CNC → PLC</b>	Opzione gestione sino a # assi. (# = 4 . . . 16)
<b>OP_V3D{F}</b>	<b>CNC → PLC</b>	Opzione Simulazione Grafica 3D





---

---

OP_SRL	CNC → PLC	Opzione Comunicazioni su linea seriale
OP_TORFR{T}	CNC → PLC	Opzione Fresatura su Tornio
OP_NET	CNC → PLC	Opzione funzionamento da Rete
OP_G31{F}	CNC → PLC	Opzione modalità galleggiamento G31
OP_TRN	CNC → PLC	Opzione Transmit G37
OP_TAS	CNC → PLC	Opzione cicli di misura con Tastatore ON/OFF
OP_LASER{F}	CNC → PLC	Opzione verifica utensili con
OP_VAX{F}	CNC → PLC	Opzione Assi Virtuali
OP_TOP{F}	CNC → PLC	Opzione macro per TOP lavandini
OP_FRETR{F}	CNC → PLC	Opzione Tornitura su Fresa
OP_CAMTR{T}	CNC → PLC	Opzione CamTorni
OP_GVUT	CNC → PLC	Opzione gestione vita ed usura Utensili
OP_WOD	CNC → PLC	Opzione Wood (per lavorazione pannelli)
OP_MARMO{T}	CNC → PLC	Opzione sgrossatura colonne in marmo
OP_I3D{T}	CNC → PLC	Opzione interpolazione Cilindrica

## 6.10 LETTURA/SCRITTURA DI VARIABILI

Le funzioni qui descritte eseguono una lettura od una scrittura immediata sulla variabile di sistema specificata. Immediata significa che l'esito dell'operazione è presente già al termine della routine stessa senza stati di attesa quali "comando non finito".

### 6.10.1 read\_var: lettura di una variabile di sistema

Chiamata: **esito = read\_var (tipovar, numvar, offset, nasse, &indvar)**

### 6.10.2 write\_var: scrittura di una variabile di sistema

Chiamata: **esito = write\_var (tipovar, numvar, offset, nasse, &indvar)**

### 6.10.3 rw\_var: lettura o scrittura di una variabile di sistema

Chiamata: **esito = rw\_var (flgatt, &pdsc, &indvar)**

### 6.10.4 Parametri in ingresso ed in uscita

Parametri di ingresso:



---

**tipovar** = (unsigned short) Tipo della variabile a cui accedere. Sono accettati i seguenti tipi:

- VL\_DOUBLE** Variabile Float in doppia precisione (double)  
**VL\_FLOT**
- VL\_LONG** Variabile Long
- VL\_INTG** Variabile Intera: Operazione di Move Word
- VL\_CHARL** Variabile Intera: Operazione di Move Byte Leggero  
L'indirizzo di Input/Output **indvar** è trattato come byte:  
In caso di READ il byte leggero della variabile viene scritto in indvar.  
In caso di WRITE il byte indvar viene scritto nel byte leggero della variabile.
- VL\_CHARH** Variabile Intera: Operazione di Move Byte Pesante  
L'indirizzo di Input/Output **indvar** è trattato come byte:  
In caso di READ il byte pesante della variabile viene scritto in indvar.  
In caso di WRITE il byte indvar viene scritto nel byte pesante della variabile.
- VL\_AND** Variabile Intera: Operazione di And Logico tra variabile ed **indvar**.  
In caso di READ il risultato viene scritto in indvar.  
In caso di WRITE il risultato viene scritto nella variabile.
- VL\_OR** Variabile Intera: Operazione di Or Inclusivo tra variabile ed **indvar**.  
In caso di READ il risultato viene scritto in indvar.  
In caso di WRITE il risultato viene scritto nella variabile.
- VL\_XOR** Variabile Intera: Operazione di Or Esclusivo tra variabile ed **indvar**.  
In caso di READ il risultato viene scritto in indvar.  
In caso di WRITE il risultato viene scritto nella variabile.

**numvar** = (unsigned short) numero della variabile da leggere (read\_var) o dove scrivere (write\_var).

Tutte le variabili di sistema accessibili sono elencate e descritte nel file VARCNC.DEF consultabile attraverso gli appositi tasti funzione presenti nel menu del CNC.

**offset** = (short) Offset (in word) della variabile desiderata appartenente ad un array, rispetto l'inizio dell'array stesso.

Se < 0 (-1) Offset non presente.

Se >= 0 Valore di offset.

**nasse** = (short) Numero di asse per variabili relative ad assi.

Se < 0 (-1) Variabile non per assi;

Se >= 0 Numero dell'asse desiderato.

**&indvar** = (void \*) Indirizzo di Input/Output (WRITE/READ)

Indica dove scrivere la variabile da LEGGERE (attività di READ) oppure da dove prendere il valore della variabile da SCRIVERE (attività di WRITE).

**flgatt** = (short) Tipo di attività per la funzione rw\_var(...).

**RQVAR\_READ** = Attività di READ



---

---

**RQVAR\_WRITE** = Attività di WRITE

**&pdsc** = (DSCVAR \*) Puntatore al Descrittore Variabile da Leggere o Scrivere (vedi DSCVAR.INC)

Parametro di uscita:

**esito** = (short ) valore di ritorno della routine . Può assumere uno dei seguenti valori:

- 0**            Successo – lettura/scrittura effettuata
- > 0**          Insuccesso – il numero rappresenta l'allarme il cui significato può essere ricercato con l'apposito tasto presente nel menu del CNC.

```
/*-----  
ESEMPIO D'USO FUNZIONI  READ_VAR() WRITE_VAR()  
-----*/  
read_var(VL_FLOT,6003,-1,3,&QUOTAW);  
/* Legge la quota assoluta (in mm.) dell'asse W di ordine 3 */  
write_var(VL_FLOT,10000,996,-1,&QUOTAW);  
/* Salva la quota assoluta dell'asse W in #498 */  
.....  
read_var(VL_FLOT,10000,996,-1,&CANC498);  
/* Legge dal #498 la quota salvata di W */  
/*-----*/
```

## 6.11 ACCESSO AD AREE U

Per **aree U** s'intendono 512 words dedicate al colloquio tra il PLC e le applicazioni scritte da un costruttore di M.U. in ambiente Windows (tipicamente per avere un'interfaccia uomo/macchina specifica per una particolare applicazione del CNC).

La libreria PLC mette a disposizione una serie di funzioni per leggere/scrivere nella suddetta area. L'elenco completo delle funzioni è riportato nel file WINTERF.H presente in C:\ECS.CNC\SWPLC\bininc\WINTERF.H; qui sono riportati i prototipi di quelle d'uso più comune, (**pos** indica l'offset in bytes o word o long o double dall'inizio dell'area).

Le funzioni di accesso all'area U richiedono che sia attivo il software d'interfaccia; l'attivazione è comandata dalla funzione IniSWint() eseguita automaticamente al pon. Per verifica sono presenti i segnali:

**SWINTAT =1** Il software d'interfaccia è attivo

**MECS07 = 1** Il software d'interfaccia non è stato attivato

### 6.11.1 Funzioni per accesso all'area U

**char u\_rdc( unsigned short pos)**

Legge un byte alla posizione **pos** dell'area U.

**short u\_rdw( unsigned short pos)**

Legge una word alla posizione **pos** dell'area U.

**long u\_rdl( unsigned short pos)**

Legge un long alla posizione **pos** dell'area U.

**double u\_rdf( unsigned short posi)**



---

Legge un double alla posizione **pos** dell'area U.

NOTA: le funzioni di lettura ritornano il byte o word o long o double richiesto.

**short u\_wrc( unsigned short pos, char val)**

Scrive **val** in un byte alla posizione **pos** dell'area U.

**short u\_rwr( unsigned short pos, short val)**

Scrive **val** in una word alla posizione **pos** dell'area U.

**short u\_wrl( unsigned short pos, long val)**

Scrive **val** in un long alla posizione **pos** dell'area U.

**short u\_wrf( unsigned short pos, double val)**

Scrive **val** in un double alla posizione **pos** dell'area U.

NOTA: le funzioni di scrittura ritornano

0 = successo

-1 = errore ( **pos** superiore al massimo valore ammesso)

**short u\_setb( unsigned short pos, unsigned short posbit)**

Setta il bit **posbit** nella word di posizione **pos** dell'area U.

**short u\_resb( unsigned short pos, unsigned short posbit)**

Resetta il bit **posbit** nella word di posizione **pos** dell'area U.

NOTA: le funzioni di set/reset bit ritornano

0 = successo

-1 = errore ( **pos** o **posbit** superiore al massimo valore ammesso)

**short u\_tstb( unsigned short pos, unsigned short posbit)**

Testa lo stato del il bit **posbit** nella word di posizione **pos** dell'area U.

NOTA: la funzione di test bit ritorna

-1 = errore ( **pos** o **posbit** superiore al massimo valore ammesso)

0 = il bit è a zero

1 = il bit è a uno

## 6.12 COMANDI MANUALI DA PLC

### 6.12.1 qalcmd: invio comandi MDI al CNC

Scopo: Invia stringhe di comando al CNC come se l'operatore le scrivesse nell'apposita finestra MDI.



---

Chiamata: **esito = qalcmd (\*indbuf)**

Parametro di ingresso:

**\*indbuf** = (char \*) stringa da inviare

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 1      Successo
- 1     Comando rifiutato (altro comando in corso o Part-Program in esecuzione).
- 2     Stringa NULLA o TROPPO LUNGA
- 3     Il CNC è in ALLARME
- 4     Il CNC non è nei modi JOG o MANUALE

## 6.13 ESEGUIBILE ESTERNO AL CNC

### 6.13.1 IOPLCRunFileExe: lancio di eseguibile esterno al CNC

Scopo: Inoltra il comando di partenza di un eseguibile esterno, sia questo di tipo **.BAT**, **.EXE** o **.COM**.

Chiamata: **esito = IOPLCRunFileExe(FileName);**

Parametro di ingresso:

**FileName**            Nome del file sul PC, completo di path. Esempio:  
(const char \*)        C : \ \ PROGRAMMI \ \ MYAPP \ \ MYAPP . EXE  
Oppure:  
                          C : / PROGRAMMI / MYAPP / MYAPP . EXE

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 0      Successo: comando inoltrato
- 1      Errore: altro comando già in corso

Per una completa gestione del comando, il PLC può testare anche il valore assunto dalla variabile associata **IOPLCSTS** che assume i valori:

**IOPLCSTS**            = -1 : Comando inoltrato ad esecutore (task WINVIS)  
(short)                = 1 : Operazione in corso di esecuzione  
                          = 0 : Operazione terminata; il task è stato lanciato

## 6.14 AUTOAPPRENDIMENTO {F}

Questa funzionalità permette di registrare in un file le quote assi della Macchina Utensile mentre l'operatore, in JOG, posiziona gli assi stessi sui vari punti significativi di un pezzo.

L'autoapprendimento si attiva tramite un tasto del menù principale del CNC. Il CNC chiede il nome del file in cui si vogliono memorizzare le quote, dopodiché si predispose in un formato di lavoro che include una finestra Editor sul file richiesto ed una serie di tasti funzione per la memorizzazione dei punti.

Il CNC deve essere in modo JOG.

L'autoapprendimento è possibile fino ad un massimo di 5 assi, compresi quelli di una eventuale testa RTCP; i tasti funzione del menù orizzontale permettono di definire il gruppo di assi interessati e quali memorizzare con la premuta del tasto STORE.



### 6.14.1 Sinonimi utilizzati per l'autoapprendimento

Per la gestione dell'autoapprendimento tramite il PLC, sono presenti alcuni segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati.

<b>OAPP</b>	<b>CNC → PLC</b>	Il CNC è in modalità Autoapprendimento
<b>OMPUNTA</b>	<b>CNC → PLC</b>	OMPUNTA = 1: E' attiva la memorizzazione quote rispetto alla Punta (centro utensile). OMPUNTA = 0: E' attiva la memorizzazione quote rispetto al Pivot (centro di rotazione della testa twist in funzionamento RTCP).
<b>OSTORE0</b>	<b>CNC → PLC</b>	Primo asse selezionato abilitato alla memorizzazione.
<b>OSTORE1</b>	<b>CNC → PLC</b>	Secondo asse selezionato abilitato alla memorizzazione.
<b>OSTORE2</b>	<b>CNC → PLC</b>	Terzo asse selezionato abilitato alla memorizzazione.
<b>OSTORE3</b>	<b>CNC → PLC</b>	Quarto asse selezionato abilitato alla memorizzazione.
<b>OSTORE4</b>	<b>CNC → PLC</b>	Quinto asse selezionato abilitato alla memorizzazione.

La libreria PLC mette a disposizione una serie di funzioni per gestire la scrittura del file, in modo da poterlo fare con un tastierino remoto senza dover ricorrere alla tastiera del CNC.

### 6.14.2 set\_marker: definisce un marker

Chiamata: **set\_marker (\*p\_marker )**

Parametri di ingresso:

- \* **p\_marker** = (char \*) stringa di caratteri da scrivere nel file di autoapprendimento come **Marker**; sono accettati, al massimo, 20 caratteri che verranno scritti sul file tra parentesi quadre [...], prima delle quote assi, con la funzione **StoAxValMarker ()**.

### 6.14.3 set\_text: definisce un testo

Chiamata: **set\_text (\*p\_text )**

Parametri di ingresso:

- \* **p\_text** = (char \*) stringa di caratteri da scrivere nel file di autoapprendimento come testo; sono accettati, al massimo, 80 caratteri che verranno scritti sul file , in un record a sé stante, esattamente come definiti; il record deve soddisfare la sintassi ammessa dal CNC o dal Post Processor che elaborerà il file autoappreso.



---

---

#### 6.14.4 StoreAxisValue: registra le quote assi

Scopo: Esegue la richiesta di scrittura delle quote degli assi abilitati, nel file di autoapprendimento. Il tipo di quota, se il Twist è attivo, dipende dallo stato di **OMPUNTA**.

Chiamata: **esito = StoreAxisValue ();**

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

<b>0</b>	Successo
<b>!=0</b>	Errore

#### 6.14.5 StoAxValMark: registra le quote assi con Marker

Scopo: Esegue la richiesta di scrittura delle quote degli assi abilitati, nel file di autoapprendimento, con il marker settato con **set\_marker()**. Il tipo di quota, se il Twist è attivo, dipende dallo stato di **OMPUNTA**.

Chiamata: **esito = StoAxValMark ();**

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

<b>0</b>	Successo
<b>!=0</b>	Errore

#### 6.14.6 StoreText: registra un testo

Scopo: Esegue la richiesta di scrittura di una stringa di testo nel file di autoapprendimento. Il testo è quello definito con la **set\_text(...)**.

Chiamata: **esito = StoreText ();**

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

<b>0</b>	Successo
<b>!=0</b>	Errore

#### 6.14.7 wTestApp: stato della richiesta di memorizzazione attuale

Scopo: Verifica lo stato della richiesta di memorizzazione fatta tramite le funzioni StoreAxisValue(), StoAxValMark(),StoreText() .

Chiamata: **esito = wTestApp();**

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

<b>0</b>	La richiesta è in corso
<b>1</b>	La richiesta è stata ricevuta dal task esecutore
<b>2</b>	La richiesta è stata eseguita OK
<b>-1</b>	Non è attivo l'editor del file di autoapprendimento
<b>-2</b>	Errore nella scrittura sul file



```
/*-----  
ESEMPIO DI GESTIONE AUTOAPPRENDIMENTO "DA PLC"  
-----*/  
/* Memorizza punto con flag attivo */  
if(!FLAGTG && !FLAGLI && !FLAGCU && !FLAGPI && !FLAGCER && !FLAGRET)  
{  
    ESITO = StoreAxisValue ();  
}  
else if (FLAGTG)  
{  
    set_marker ("PERDITA_TG");  
    ESITO = StoAxValMark ();  
}  
else if (FLAGLI)  
{  
    set_marker ("LINEARE");  
    ESITO = StoAxValMark ();  
}  
else if (FLAGCU)  
{  
    set_marker ("CURVA");  
    ESITO = StoAxValMark ();  
}  
else if (FLAGPI)  
{  
    set_marker ("PIANO");  
    ESITO = StoAxValMark ();  
}  
else if (FLAGCER)  
{  
    set_marker ("CERCHIO");  
    ESITO = StoAxValMark ();  
}  
else if (FLAGRET)  
{  
    set_marker ("RETTANGOLO");  
    ESITO = StoAxValMark ();  
}  
/*-----*/
```

## 6.15 FUNZIONI DI SCRITTURA/LETTURA FILES

### 6.15.1 IOPLCWriteFile: scrittura file

Scopo: Inoltra al PC un comando di scrittura file.

Chiamata: **esito = IOPLCWriteFile(FileName, OpenMode, &Record, LenRec, FileOffset)**

Parametri in ingresso:

<b>FileName</b> (const char *)	Nome del file sul PC, completo di path. Esempio: C:\\ECS.CNC\\LAV\\MIOFILE.TXT Oppure: C:/ECS.CNC/LAV/MIOFILE.TXT
<b>OpenMode</b> (short)	Modo di scrittura file; può assumere i valori: <b>1: scrittura nuovo file</b>





Viene creato un nuovo file di nome **FileName**.

Nel file viene scritto il record **Record** di lunghezza **LenRec**.

Il parametro **FileOffset** non viene considerato (mettere 0).

### 2: scrittura file binario

Il record viene scritto nel file alla posizione **FileOffset** che rappresenta un offset (in bytes,  $\geq 0$  e valido) all'interno del file.

Se **FileOffset** vale  $-1$ , il record viene appeso al file.

### 3: scrittura file ASCII

Il **Record** viene scritto sul record numero **FileOffset** del file (valido solo per file ASCII, in cui ogni record termina con CR,LF).

Se il record da scrivere contiene CR,LF il record di numero **FileOffset** viene riscritto, altrimenti il nuovo record sovrascrive il precedente per un numero di bytes non maggiore di quelli del record precedente stesso.

Se **FileOffset** vale  $-1$  il record viene appeso al file.

<b>&amp;Record</b> (const char *)	Record da scrivere sul file. Stringa di caratteri per file ASCII, oppure puntatore ad una struttura di dati per file binario.
<b>LenRec</b> (unsigned short)	Numero di caratteri (bytes) da scrivere.
<b>FileOffset</b> (long)	Offset, in bytes, per file binario o numero di record per file ASCII. Se <b>FileOffset</b> vale $-1$ , il record viene appeso al file.

Parametro in uscita:

<b>esito</b> (short)	<b>1</b> : canale occupato <b>0</b> : comando inoltrato
-------------------------	--

Per una corretta gestione del comando, il PLC deve testare anche i valori assunti dalle variabili associate **IOPLCSTS** e **IOPLCRET** che possono assumere i valori:

<b>IOPLCSTS</b> (short)	= <b>-1</b> : Comando inoltrato ad esecutore (task WINVIS) = <b>0</b> : Operazione terminata = <b>1</b> : Operazione in corso di esecuzione
----------------------------	---

<b>IOPLCRET</b> (short)	= <b>0</b> : OK Comando terminato correttamente = <b>-1</b> : Errore apertura file = <b>-2</b> : Errore scrittura record nel file = <b>-3</b> : Errore lettura file = <b>-4</b> : Errore posizionamento sul file = <b>-5</b> : Errore remove vecchio file = <b>-6</b> : Errore copy/rename dopo remove vecchio file = <b>-7</b> : Numero di record NON trovato
----------------------------	---

Il PLC deve quindi testare, nell'ordine:

4. **esito** : se = 1 chiamare nuovamente la routine; se =0 passare alla fase successiva
5. **IOPLCSTS** : se =0 passare alla fase successiva altrimenti attendere
6. **IOPLCRET**: se =0 successo altrimenti segnalare l'errore all'operatore

#### 6.15.1.1 Esempio di scrittura File

Qui di seguito è riportato un frammento di PLC che effettua la registrazione, su un file ASCII, degli Allarmi del CNC, con ora e data.



Per ogni evento vengono scritti due records del tipo:

```
Registrazione dell'evento N: 68 Allarme Nr: 2001  
->Data e tempo: Thu Nov 29 19:59:39 2001
```

Vengono registrati, al massimo, 20 eventi (vedi case 20); quando tutti e venti gli allarmi sono stati registrati, si riparte da inizio file, sovrascrivendo i records più vecchi; il contatore di eventi NUMERO si incrementa comunque e permette di vedere quali sono i più recenti.

L'esempio fornisce anche alcune informazioni su come costruire un record e su come leggere/formattare data e ora.

#### In PLC\_SYN.INC :

```
. . . . .  
#define      ESITOWF          A6.word  
/* Esito della funzione IOPLCWriteFile() */  
#define      NUMREC          A7.word  
/* Numero di record da scrivere */  
#define      NUMERO          A8.word  
/* Numero di evento da registrare */  
#define      NUMALL          D_A10  
/* Numero dell'allarme del CNC */  
#define      NEWEVENT        A12.bit._00  
/* Nuovo evento da registrare */  
  
. . . . .  
/* ----- */  
/*          WORD M0          */  
/* ----- */  
  
#define      FILEOPEN        MM0.bit._00  
/* Memorizza se il File e' gia' stato aperto */
```

#### In QAL.C :

```
. . . . .  
#include <string.h> /* per funzioni su stringhe */  
#include <stdlib.h> /* per conversioni da/in ASCII */  
#include <math.h> /* per funzioni matematiche */  
#include <time.h> /* per funzioni tempo */  
  
#define MAXRECL 100  
/* Massima lunghezza di un record */  
static char NomeFile[] = {"C:\\\\ECS.CNC\\LAV\\ALLARMI.TXT"};  
/* Nome del File da aprire */  
static char Titolo[] = {"File di registrazione degli eventi\r\n"};  
/* Stringa da mettere come primo record */  
static char Evento[] = {"Registrazione dell'evento N: "};  
/* Parte fissa di ogni stringa */  
  
char Record[MAXRECL + 2];  
/* Appoggio per costruire il record */  
char AppBuff[50];  
/* Buffer di appoggio */  
  
struct tm *time_of_day;  
time_t ltime;  
char *pstrtime;  
/* Appoggi per elaborazione data/ora */  
  
void gal( void )
```



```
{
. . . . .
/***** SCRITTURA FILE *****/
if(pon || ORST)
{
SEQ00= 0 ;
ESITOWF= 0 ;
}
if(NEWM && (OM == 199))
FILEOPEN= 0;
/* Reset per nuovo File con Funzione Ausiliaria M199 */
NEWEVENT= !ORST && (OALL && (ESITOWF == 0) || NEWEVENT) ;
/* Nuovo evento da registrare: Allarme CNC */

switch(SEQ00)
{
case 0:
if(NEWEVENT)
/* Nuovo evento; apro il file e/o registro l'evento */
{
if(!FILEOPEN)
SEQ00= 10 ;
else
SEQ00= 20 ;
}
break;

/*****/
case 10: /* Apertura File */
ESITOWF= IOPLCWriteFile (NomeFile,1,Titolo,strlen(Titolo),0);
/* Apre nuovo file con titolo */
if(ESITOWF == 0)
SEQ00= 12;
break;

/*****/
case 12: /* Test se comando inoltrato */
if(IOPLCSTS==0)
SEQ00= 15;
break;

/*****/
case 15: /* Attesa risultato */
if(IOPLCRET==0)
{
NUMREC= -1 ;
NUMERO= 0 ;
FILEOPEN = 1 ;
SEQ00= 0;
}
break;

/*****/
case 20:
NUMERO++ ;
if(NUMERO > 20) /* Massimo numero di eventi nel file */
{
if(NUMREC <= 0 || NUMREC > 40)
```



```
        NUMREC= 2 ; /* Primo record per sovrascrittura evento */
    else
        NUMREC += 2 ; /* Records successivi */
    }
else
    {
        NUMREC= -1 ; /* Prime scritture in appendi */
    }
strcpy(Record,Evento);
itoa(NUMERO,AppBuff,10);
strcat(Record, AppBuff);
strcat(Record," Allarme Nr:  ");
if(OCGEN < 99)
    NUMALL= OVGEN ;
else
    NUMALL= -1 ;

ltoa(NUMALL,AppBuff,10);
strcat(Record, AppBuff);
strcat(Record, "\r\n");
/* Costruisce il primo record da scrivere */
SEQ00= 21;
break;

/*****/
case 21:
ESITOWF= IOPLCWriteFile (NomeFile,3,Record,strlen(Record),NUMREC);
if(ESITOWF == 0)
    SEQ00= 22;
break;
/* Se il canale e' occupato, riprova al prossimo loop PLC */

/*****/
case 22: /* Test se comando inoltrato */
if(IOPLCSTS==0)
    SEQ00= 23;
break;

/*****/
case 23: /* Attesa risultato */
if(IOPLCRET==0)
    SEQ00= 24;
break;

/*****/
case 24:
strcpy(Record," ->Data e tempo: ");
time(&lttime);
time_of_day= localtime(&lttime);
pstrtime= asctime(time_of_day);
memcpy(AppBuff,pstrtime,24);
AppBuff[24]= '\0';
strcat(Record, AppBuff);
strcat(Record, "\r\n");
/* Costruisce il secondo record con data e ora */
SEQ00= 26;
break;
```



```

/*****
case 26:
if(NUMREC > 0)
    ESITOWF= IOPLCWriteFile (NomeFile,3,Record,strlen(Record),NUMREC+1);
    /* Sovrascrittura secondo record */
else
    ESITOWF= IOPLCWriteFile (NomeFile,3,Record,strlen(Record),NUMREC);
    /* In append quando NUMREC = -1 */

if(ESITOWF == 0)
    SEQ00= 30 ;
break;
/* Se il canale e' occupato, riprova al prossimo loop PLC */

/*****
case 30: /* Test se comando inoltrato */
if(IOPLCSTS==0)
    SEQ00= 35;
break;

/*****
case 35: /* Attesa risultato */
if(IOPLCRET==0)
    SEQ00= 100;
break;

/*****
case 100:
if(!NEWEVENT)
    SEQ00= 0 ;
break;

/*****
default:
break;
}
/*****FINE SCRITTURA FILE *****/
}

```

Il file generato è come qui sotto riportato:

```

File di registrazione degli eventi
Registrazione dell'evento N: 63 Allarme Nr: 5148
->Data e tempo: Thu Nov 29 19:59:28 2002
Registrazione dell'evento N: 64 Allarme Nr: 3700
->Data e tempo: Thu Nov 29 19:59:30 2002
Registrazione dell'evento N: 65 Allarme Nr: 3700
->Data e tempo: Thu Nov 29 19:59:32 2002
. . . .

```

### 6.15.2 IOPLCReadFile: lettura file

Scopo: Inoltra al PC un comando di lettura file.

Chiamata: **esito = IOPLCReadFile(FileName, LenRec, FileOffset)**

Parametri in ingresso:



---

---

<b>FileName</b> (const char *)	Nome del file sul PC, completo di path. Esempio: C:\\ECS.CNC\\LAV\\MIOFILE.TXT Oppure: C:/ECS.CNC/LAV/MIOFILE.TXT
<b>LenRec</b> (unsigned short)	Numero di caratteri da leggere: - se =0: lettura file ASCII - se >0: bytes da leggere da un file binario
<b>FileOffset</b> (long)	Offset sul file: - in bytes, per file binario, da cui leggere LenRec bytes. - numero di record per file ASCII.

Parametro in uscita:

<b>esito</b> (short)	<b>1</b> : canale occupato <b>0</b> : comando inoltrato
-------------------------	--

Per una corretta gestione del comando, il PLC deve testare anche i valori assunti dalle variabili associate **IOPLCSTS** e **IOPLCRET** che possono assumere i valori:

<b>IOPLCSTS</b> (short)	= -1 : Comando inoltrato ad esecutore (task WINVIS) = 0 : Operazione terminata = 1 : Operazione in corso di esecuzione
----------------------------	--

<b>IOPLCRET</b> (short)	= 0 : OK Comando terminato correttamente = -1 : Errore apertura file = -3 : Errore lettura file = -4 : Errore posizionamento sul file = -7 : Numero di record NON trovato
----------------------------	---

Il PLC deve quindi testare, nell'ordine:

1. **esito** : se = 1 chiamare nuovamente la routine; se =0 passare alla fase successiva
2. **IOPLCSTS** : se =0 passare alla fase successiva altrimenti attendere
3. **IOPLCRET**: se =0 successo altrimenti segnalare l'errore all'operatore

Questa funzione NON restituisce, come risultato, il record desiderato, ma inoltra la richiesta di lettura; i servizi messi a disposizione dal sistema operativo provvederanno alla effettiva lettura del record dal file e lo renderanno disponibile per essere letto dal PLC con la funzione `IOPLCGetRecord()` del paragrafo seguente.

### 6.15.3 IOPLCGetRecord: lettura del record

Scopo: Legge il record precedentemente richiesto con la funzione `IOPLCReadFile()`.

Chiamata: **esito = IOPLCGetRecord (&Record, MaxRec, &LenRec)**

Parametri in ingresso:

<b>&amp;Record</b> (char *)	Indirizzo dell'array su cui la funzione deve scrivere il record ASCII o i bytes binari letti. La stringa ASCII NON include i caratteri CR/LF.
--------------------------------	--



---

---

<b>MaxRec</b> (unsigned short)	Numero massimo di caratteri o bytes richiesti
<b>&amp;LenRec</b> (unsigned short*)	Indirizzo (nome) della variabile su cui scrivere il numero di caratteri del record ASCII o dei bytes binari letti. In caso di lettura da un file ASCII, il numero può anche essere maggiore di <b>MaxRec</b> , ma i caratteri copiati in <b>Record</b> sono comunque limitati da MaxRec e quelli in eccesso sono "persi".

Parametro in uscita:

<b>esito</b> (short)	<b>0</b> : successo; su <b>Record</b> ci sono i bytes <b>LenRec</b> (o <b>MaxRec</b> ) letti dal file.
	<b>1</b> : canale occupato
	<b>2</b> : errore nel comando
	<b>3</b> : errore (l'ultimo comando inoltrato non è una lettura da file)

## 6.16 FUNZIONI MISCELLANEE

### 6.16.1 GetSerialNumber: lettura numero di commessa

Scopo: Legge il numero di commessa (Serial Number) assegnato da ECS a questo CNC. Il numero di commessa è lo stesso che si può leggere nella finestra "Software Info" ( [F8] Tools, [F6] Info, [F1] Software Info).

Chiamata: **esito= GetSerialNumber();**

Parametro in uscita:

<b>esito</b> (unsigned short)	Numero di commessa ECS
----------------------------------	------------------------

### 6.16.2 s\_man: salva modo manuale

Scopo: salva lo stato attuale del modo manuale (MDI). Può essere utilizzata dal PLC per salvare le modalità del MODO MANUALE, passare in modo JOG e, dopo aver riattivato il modo MANUALE, poter ripristinare con **r\_man()** le modalità precedentemente attive.

Chiamata: **esito= s\_man();**

Parametro in uscita:

<b>esito</b> (unsigned short)	<b>0</b> : successo; è stato eseguito il salvataggio.
	<b>7075</b> : errore; il CNC deve essere in modo MANUALE per accettare il comando.
	<b>7076</b> : errore; comando non consentito se il CNC è in RUN.
	<b>7077</b> : errore; comando non consentito se il CNC sta eseguendo un Part-Program.
	<b>7078</b> : errore; comando non consentito se il CNC ha in carico istruzioni di programma.

### 6.16.3 r\_man: ripristina modo manuale

Scopo: ripristina lo stato del modo manuale (MDI), salvato con la funzione **s\_man()** vista sopra.

Chiamata: **esito= r\_man();**

---

---



Parametro in uscita:

**esito**

(unsigned short)

**0** : successo; è stato eseguito il ripristino.

**7075** : errore; il CNC deve essere in modo MANUALE per accettare il comando.

**7076** : errore; comando non consentito se il CNC è in RUN.

**7077** : errore; comando non consentito se il CNC sta eseguendo un Part-Program.

**7078** : errore; comando non consentito se il CNC ha in carico istruzioni di programma.

**7079** : errore; per eseguire il ripristino occorre prima eseguire il salvataggio.





---

---

## CAPITOLO 7

### 7 TASTIERA

In questo capitolo sono raccolte le informazioni relative alla gestione della tastiera da parte del PLC.

I CNC ECS serie WIN possono essere dotati di due tipi di tastiera (pannelli operatore):

- tastiera con tasti a corsa lunga.
- tastiera con tasti a membrana.

In entrambe le tastiere si possono identificare due parti differenti:

- parte P.C. (tastiera QWERTY + tasti F1 ÷ F10)
- parte CNC (tasti operativi + tasti P1 ÷ P8)

La parte P.C. è gestita dal sistema operativo e dai vari task che realizzano l'interfaccia uomo-macchina; la parte CNC è gestita dal PLC.

I tasti funzione F1 ÷ F10 e P1 ÷ P8 assumono dinamicamente il significato indicato via via da icone diverse sul video in base allo stato attuale del CNC.

La tastiera comprende anche i comandi per l'override assi e mandrino, che sono trattati nei corrispondenti capitoli di questa monografia.

ECS fornisce una gestione completa per la parte CNC attraverso le routine di libreria PLC di cui le principali sono:

**tastcnc()** che elabora i tasti operativi del CNC

**tastfuecs()** che si prende cura dei tasti funzione del PLC (P1 ÷ P8) nella configurazione standard ECS definita in BASMENU.MEN.

ECS fornisce uno strumento SW, integrato nel CNC, che consente la completa caratterizzazione ed interconnessione logica dei tasti funzione P1 ÷ P8. La descrizione di questo strumento è ampiamente documentata nel capitolo "TOOL DEFINIZIONE TASTI PLC" della monografia **720P394** "AMBIENTE e PROCEDURE di SVILUPPO e DEBUG PLC".

ECS può fornire uno strumento SW (MacroMenuDev), normalmente non integrato nel CNC, per la caratterizzazione del Menu MACRO dell'editor di Part-Program. La descrizione di questo strumento è ampiamente documentata nella monografia 720P405 alla quale si rimanda per ogni approfondimento.

I CNC ECS serie WIN possono funzionare anche senza tastiera ECS; le relative funzioni possono essere realizzate tramite Software di Interfaccia realizzato dal Costruttore della Macchina Utensile utilizzando le funzioni di dialogo appositamente previste (INTERF.DLL) o, al limite, tramite pulsanti dedicati collegati agli Input delle schede I/O. Per la parte P.C. (tastiera QWERTY + tasti F1 ÷ F10) può essere usata una normale tastiera standard AT collegata all'apposito connettore del CNC.



## 7.1 Gestione della tastiera

La gestione dei tasti operativi del CNC è risolta in maniera sicura dalle routines di libreria PLC descritte più avanti e quindi non è normalmente necessario conoscere nei dettagli le sincronizzazioni esistenti tra TASTIERA, PLC e CNC. Qui sono date alcune informazioni supplementari, utili nel caso non si possano o vogliano usare le routines standard.

Il colloquio tra tastiera e CNC avviene nel seguente modo:

- In seguito alla premuta di un tasto, la tastiera invia al PLC il segnale di strobe nuovo codice (STBKCN) ed il codice relativo al tasto premuto (CODTAS).
- Il PLC deve rispondere alla tastiera complementando di stato il segnale di acknowledge ACKKCN, trasferire il codice tasto al CNC (ICTS) con relativo segnale di strobe (INCTS), complementando anch'esso.
- Il CNC risponde tasto acquisito complementando lo stato del segnale OAKCT.
- Il tasto successivo sarà inviato dalla tastiera solo se ACKKCN=STBKCN e interpretato dal CNC solo se OAKCT=INCTS.

L'invio dei tasti con l'opportuna sincronizzazione qui riportata, è realizzato dalla routine PLC **tastcnc()**.

La gestione degli overrides è realizzata dalle routine **GestOvr3()**, **gestovr()** e **gestovrc()**.

### 7.1.1 Sinonimi utilizzati per la gestione tastiera

Per la gestione della tastiera tramite il PLC, sono presenti alcuni segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

La maggior parte di questi segnali è sentita, elaborata o generata dalla routine **tastcnc()**.

<b>TAS2700</b>	<b>CNC → PLC</b>	<b>TAS2700 = 1:</b> è presente e collegata una tastiera con tasti a membrana standard ECS con 2 override a potenziometro.
<b>TASBASIC</b>	<b>CNC → PLC</b>	<b>TASBASIC = 1:</b> è presente e collegata una tastiera con tasti a corsa lunga standard ECS con 3 override ad encoder .
<b>CODTAS</b> (word)	<b>CNC → PLC</b>	Codice del tasto premuto E' il codice <u> fisso </u> dei <u> tasti "fisici" </u> (START, REL, HOLD, AXES H., RESET) oppure quello <u> programmato </u> dei <u> tasti funzione "software" </u> realizzati su P1 ÷ P8.
<b>NCTS</b>	<b>CNC → PLC</b>	Nuovo codice tasto da tastiera. NCTS = 1 per un giro PLC ogni volta che viene premuto un tasto operativo della consolle del CNC. <i>Gestito dalla routine tastcnc()</i>
<b>PUCNC</b>	<b>CNC → PLC</b>	Premuto tasto con codice riservato al CNC (CODTAS<255). Stessa temporizzazione di NCTS. <i>Gestito dalla routine tastcnc()</i>



<b>PUECS</b>	<b>CNC → PLC</b>	Premuto un tasto con un codice "ECS" ( $255 \leq \text{CODTAS} < 1000$ ). Stessa temporizzazione di NCTS. <i>Gestito dalla routine <code>tastcnc()</code></i>
<b>PUSER</b>	<b>CNC → PLC</b>	Premuto un tasto con un codice "USER" ( $\text{CODTAS} \geq 1000$ ). Stessa temporizzazione di NCTS. <i>Gestito dalla routine <code>tastcnc()</code></i>
<b>CKCN</b>	<b>PLC → CNC</b>	<b>CKCN = 0</b> : Tasti operativi CNC <b>SBLOCCATI</b> . <b>CKCN = 1</b> : Tasti operativi CNC <b>BLOCCATI</b> completamente escluso i tasti REL - HOLD - AXES H. - RESET Per tasti operativi del CNC s'intendono sia quelli "fisici" (START, REL, HOLD, AXES H., RESET) sia quelli "software" realizzati tramite P1 ÷ P8.
<b>CNOK</b>	<b>CNC → PLC</b>	Il CNC è pronto per accettare tasti. Con CNOK = 0, nessun nuovo codice tasto sarà passato al CNC. <i>Gestito dalla routine <code>tastcnc()</code></i> Questo segnale è utilizzata da altre routine, ad esempio <code>aggvol()</code> , e non deve essere quindi modificato
<b>CODICE</b> (char)	<b>PLC → CNC</b>	Contiene il codice del tasto operativo da passare a CNC. I codici previsti sono elencati più avanti. <i>Gestito dalla routine <code>tastcnc()</code></i> .
<b>STBCOD</b>	<b>PLC → CNC</b>	Strobe per abilitare l'invio del codice tasto presente su <b>CODICE</b> . <b>STBCOD</b> va tenuto a 1 fino alla risposta <b>PASEXT</b> <i>Gestito dalla routine <code>tastcnc()</code></i>
<b>PASEXT</b>	<b>CNC → PLC</b>	Il codice tasto inviato su <b>CODICE</b> è stato passato al CNC Rimane ad 1 finché <b>STBCOD= 1</b> . <i>Gestito dalla routine <code>tastcnc()</code></i>
<b>RSTMUON</b>	<b>PLC → CNC</b>	Comanda il RESET del CNC. Tipicamente usato per effettuare un RESET all'accensione della Macchina Utensile ( a CNC già acceso) per annullare eventuali Allarmi e/o comandi non terminati. Il RESET viene eseguito sul fronte di salita di RSTMUON. <i>Gestito dalla routine <code>tastcnc()</code></i>
<b>RETG84</b>	<b>PLC → CNC</b>	Richiesta retract assi con G84 in corso. Da usare quando, durante l'esecuzione della macro G84 (maschiatura con compensatore), fosse necessario interrompere il ciclo per ritornare al punto di partenza. Setta l'input ITF03 del CNC (True/False bit 03) sentito dalla macro G84 Il RETRACT viene eseguito sul fronte di salita di RETG84. <i>Gestito dalla routine <code>tastcnc()</code></i>
<b>AB_SHIFT</b>	<b>PLC → CNC</b>	<b>AB_SHIFT= 1</b> : abilita la simulazione dei tasti [P1]...[P8] e [GoUp] tramite [Shift]+[F1]...[F8] ed [Esc]. Questa funzione è necessaria quando non è montata una tastiera standard ECS e quindi i tasti [P1]...[P8] e [GoUp] non sono presenti.



## 7.1.2 Codici dei tasti operativi del CNC

I codici presenti su CODTAS o da scrivere su CODICE sono raccolti nella seguente tabella.

Simbolo	codice ottale	codice decimale	codice esadec.	Funzione
CT_JOG	0221	145	0x91	Richiesta di modo JOG
CT_MAN	0222	146	0x92	Richiesta di modo MDI
CT_SET	0223	147	0x93	Richiesta di modo SET
CT_AUT	0224	148	0x94	Richiesta di modo AUTO
CT_FND	0225	149	0x95	Richiesta di sottomodo RICERCA
CT_RIP	0226	150	0x96	Richiesta di modo RIPOSIZIONAMENTO
CT_LS	0227	151	0x97	Richiesta di sottomodo LETT SINGOLA
CT_LC	0230	152	0x98	Richiesta di sottomodo LETT CONTINUA
CT_ES	0231	153	0x99	Richiesta di sottomodo ESEC SINGOLA
CT_EC	0232	154	0x9A	Richiesta di sottomodo ESEC CONTINUA
CT_RST	0233	155	0x9B	Tasto [RESET]
CT_REL	0235	157	0x9D	Tasto [REL]
CT_FST	0237	159	0x9F	Tasto [FAST]
CT_JPP	0243	163	0xA3	Tasto JOG + premuto
CT_JNP	0244	164	0xA4	Tasto JOG - premuto
CT_REC	0246	166	0xA6	Copiatura con registrazione (ON/OFF)
CT_MDG	0251	169	0xA9	Richiesta di scansione manuale.
CT_STO	0252	170	0xAA	Richiesta di memorizz. punti.
CT_HLA	0261	177	0xB1	Tasto [AXES.H] solo assi
CT_STA	0262	178	0xB2	Tasto [START]
CT_HLD	0263	179	0xB3	Tasto [HOLD] anche mandrino
CT_JNR	0264	180	0xB4	Tasto JOG - rilascio
CT_JPR	0265	181	0xB5	Tasto JOG + rilascio
CT_DVL	0300	192	0xC0	Richiesta disinserzione volantini
CT_V1	0301	193	0xC1	inserzione volante 1
CT_V2	0302	194	0xC2	inserzione volante 2
CT_V12	0303	195	0xC3	inserzione volantini 1 e 2
CT_V3	0304	196	0xC4	inserzione volante 3
CT_V13	0305	197	0xC5	inserzione volantini 1 e 3
CT_V23	0306	198	0xC6	inserzione volantini 2 e 3
C_V123	0307	199	0xC7	inserzione volantini 1 e 2 e 3
CT_DDRF	0310	200	0xC8	Richiesta disinserz. volantini DRF
CT_IDRF	0311	201	0xC9	Richiesta inserzione volantini DRF

## 7.1.3 Routines a disposizione del PLC

### 7.1.3.1 Routine "TASTCNC"

Chiamata: **tastcnc()**;

Scopo: Gestisce i tasti operativi del CNC, sia quelli "fisici" (START, REL, HOLD, AXES H., RESET) sia quelli "software" realizzati tramite P1 ÷ P8.

Segnali in ingresso:

**CKCN, CODICE, STBCOD, RSTMUON, RETG84**

Segnali di uscita:

**CNOK, PASEXT, NCTS, PUSER, PUECS, PUCNC**



Il significato dei segnali suddetti è stato già spiegato al paragrafo "Sinonimi utilizzati per la gestione tastiera".

Si ricorda che quando è attivo l'input **CKCN**, la routine scarta tutti i codici CNC diversi da RESET, HOLD, AXES H. e REL che non sono quindi passati al CNC.

Quando viene usata la IKR per collegare dei pulsanti che debbono fare le funzioni dei tasti operativi del CNC, su **CODICE** va messo il codice del tasto da inviare al CNC e poi va settato **STBCOD**:

```
if (!OHLD && KEY2M68 && !STBCOD)
{
  CODICE=0xB2;
  STBCOD=1;
} /* START da pulsante IKR */
STBCOD= STBIKR || STBCOD && !PASEXT;
/*STROBE DEL TASTO DA IKR FINO A RISPOSTA DA CNC */
```

Il CNC rende automaticamente ACKIKR alla IKR.

### 7.1.3.2 Routine "TASFUECS"

Chiamata: **tasfuecs()**;

Scopo: La routine gestisce la premuta dei tasti funzione del PLC ( P1 ÷ P8 ) in associazione alla configurazione standard ECS definita in BASMENU.MEN.

Desiderando modificare la gestione dei tasti P1÷P8, si consiglia di scrivere una diversa routine {ad es. PLCtasfu()} da abbinare ad un PLCMENU.MEN opportunamente scritto.

Segnali in Ingresso:

**NAXJOG** = (unsigned char) numero assi da selezionare in JOG e per il volante;  
devono essere tutti contigui a partire dall'asse di ordine 0; se non definito,  
la routine permette la selezione di tutti gli assi presenti sul CNC.

La routine usa diversi segnali generati dalla tastcnc() e quindi deve essere sempre abbinata ad essa.

I segnali di uscita della routine tasfuecs() sono quelli necessari per la corretta gestione delle altre routines standard della libreria PLC.

```
/*-----
ESEMPIO D'USO DELLE ROUTINE TASTCNC E TASFUECS
-----*/
tastcnc();
/* routine che gestisce la tastiera CNC: tasti P1...P8 */
/* e tasti dedicati (START, REL, HOLD, AXES H., RESET) */
if(pon)
{
  NAXJOG= 3;
}
tasfuecs();
/* routine che gestisce la premuta dei tasti funzione */
/* del PLC (P1-P8) in associazione alla configurazione */
/* standard ECS definita in BASMENU.MEN. */
modijog();
/* passa al CNC il numero d'ordine dell'asse da muovere */
/* in modo JOG e le modalita' di movimento */
aggvol();
```



```

/* gestisce input Volantini da SCHEDE ASSI, da TASTIERA */
/* o da IKR (Interfaccia Keyboard Remota). */
if( pon )
{
    MAXOVF = 120;
    MAXOVR = 100;
    MAXOVS = 150;
    MINOVS = 10;
    /* inizializza i valori MAX e MIN degli override */
}
if ( TASBASIC )
{
    GestOvr3();          /* tastiera BASIC con tre encoders */
}
else
{
    gestovr();          /* tastiera 2700 con due potenziometri */
}
/* gestisce gli override della tastiera */
/*-----*/

```

#### 7.1.4 Tasti funzione CUSTOM

Il progettista del PLC può aggiungere ai tasti del menu verticale (quello a destra del video) presenti nella configurazione standard ECS definita in BASMENU.MEN, i propri tasti funzione a partire dal menu aperto con il tasto P8 (tasto CUSTOM) del menu base.

ECS fornisce uno strumento SW, integrato nel CNC, che consente la completa caratterizzazione ed interconnessione logica dei tasti funzione P1 ÷ P8. La descrizione di questo strumento è ampiamente documentata nel capitolo "TOOL DEFINIZIONE TASTI PLC" della monografia **720P394** "AMBIENTE e PROCEDURE di SVILUPPO e DEBUG PLC".

Come già detto, si consiglia di non modificare il menu standard ECS ma di costruirne uno proprio (ad es.: PLCMENU.MEN) per rendere evidente questa personalizzazione. Se non vengono modificate le funzioni realizzate da BASMENU.MEN, può essere comunque usata la routine standard di gestione **tstfuecs()**; lo stesso si può dire se vengono tolti dei tasti predefiniti o ne vengono aggiunti altri con codice tasto diverso da quelli riservati al CNC o ad ECS. Al paragrafo 7.1.1 sono stati indicati i codici ammessi ed i segnali legati alla pressione del tasto di ciascuna "categoria". Come si può vedere, i codici disponibili sono compresi tra 1000 e 32767; il tool di personalizzazione dei tasti funzione (vedi 720P394) segnala un'attenzione se vengono usati codici inferiori a 1000 ma permette comunque di introdurli per una più libera progettazione del menu.

Con il CNC in EXE, premendo un tasto funzione, il codice relativo viene scritto nella word **CODTAS** con il relativo strobe **NCTS** (e **PUECS** o **PUSER** o **PUCNC** secondo la categoria di appartenenza); al rilascio del tasto viene scritto lo stesso codice cambiato di segno con i soliti strobe (es.: 1000 ⇒ -1000, 1234 ⇒ -1234, ecc.):

```

if( PUSER && (CODTAS == 1002) )
{ /* Tasto premuto */
    RUNMOTOR=1;
}
else if( PUSER && (CODTAS == -1002) )
{ /* Tasto rilasciato */
    RUNMOTOR=0;
}

```



Facendo riferimento alla configurazione base data in BASMENU.MEN, i tasti funzione CUSTOM sono inseriti nel menu 10, al quale si salta premendo il tasto P8 del menu 1.

Da questo menu si può generare una struttura di successivi menu "ad albero" per realizzare le più disparate funzioni di macchina, anche legate alle FINESTRE VIDEO PERSONALIZZATE che saranno trattate più avanti.

Ad ogni tasto funzione, oltre al codice da passare al PLC, può essere associato un bit "lampada" che ne determina lo stato di "rilasciato" o "premuto". A questo scopo sono dedicate alcune words dell'area E ( words 21 ÷ 30) da scrivere con la funzione:

```
e_u_wrw(20, MLAMP10);
```

Ogni tasto può inoltre essere condizionato dinamicamente da otto segnali "Flag" che determinano, se messi a 1, lo stato di "tasto disattivato". Questi Flag sono posti ad 1 od a 0 dal PLC in base allo stato operativo della macchina o a logiche di sicurezza per segnalare all'operatore che una o più funzioni sono attualmente bloccate, in maniera simile ai tasti del menu orizzontale gestiti direttamente dal CNC.

I Flag suddetti sono definiti con i sinonimi:

<b>FLGMENW</b> (short)	<b>PLC → CNC</b>	Word flags bloccaggio menu PLC I primi 8 bit della word (di peso 1÷8) sono sinonimati singolarmente come indicato qui sotto.
<b>FLGMEN#</b>	<b>PLC → CNC</b>	8 flags bloccaggio menu PLC bit #. (con # = 1÷8) Sono i primi 8 bit della word <b>FLGMENW</b> . Corrispondono a <b>Flag 1 ÷ Flag 8</b> definiti nella finestra "Condizioni di blocco menu" del "TOOL DEFINIZIONE TASTI PLC".

I tasti funzione così definiti possono essere quindi facilmente gestiti dalla logica del PLC.

Il PLC può conoscere quale menu è attivo in un dato istante tramite:

<b>ACTQMENU</b> (short)	<b>CNC → PLC</b>	Numero del menu PLC attivo.
----------------------------	------------------	-----------------------------

Il PLC può comandare quale menu rendere attivo in un dato istante tramite:

<b>ASKQMENU</b> (char)	<b>PLC → CNC</b>	Numero del menu PLC da attivare.
<b>STBQMENU</b>	<b>PLC → CNC</b>	<b>STBQMENU = 1</b> : richiesta di attivare il menu il cui numero è definito in ASKQMENU; il CNC attiva, se possibile, il menu richiesto e rimette, comunque, <b>STBQMENU = 0</b> .

#### 7.1.4.1 Esempio di gestione Tasti funzione CUSTOM

Nel seguente esempio, si suppone di aver definito i menu 10 e 11 come da tabella:

Menu-Tasto	Testo da scrivere sul tasto		Bitmap xxxxxx.BMP	Codice Tasto	Word lamp.	Bit
	RILASCIATO	PREMUTO				
10-P1	Assi	Assi	OIL_PUMP	1000	21	0



10-P2	Mandrino	Mandrino	OIL_PUMP	1001	21	1
10-P3	Attiva Override Assi c.ut.	Disattiva Override Assi c.ut.		1002	21	2
10-P4			HELP	1003	21	3
10-P5	Temperat. Cuscinetti Mandrino	Temperat. Cuscinetti Mandrino		1004	21	4
10-P6	Spegni macchina a fine lavoro	Spegni macchina a fine lavoro		1005	21	5
10-P7	Abilita Sblocco Utensile	Abilita Sblocco Utensile		1006	21	6
10-P8	Comandi manuali cambio ut.	Comandi manuali cambio ut.		1007	21	7
11-P1			MANO	1100	21	8
11-P2	Ribalta Portaut.	Ribalta Portaut.	ARROW_DN ARROV_DN	1101	21	9
11-P3	Portaut. a Riposo	Portaut. a Riposo	ARROW_UP ARROV_UP	1102	21	10
11-P3	Sblocco Utensile	Sblocco Utensile	UT_BLC UT_BLCB	1103	21	11
11-P5	Blocco Utensile	Blocco Utensile	UT_SBL UT_SBLS	1104	21	12
11-P6	Slitta	Slitta	VOLANT_0 VOLANT_1	1105	21	13
11-P7	Catena	Catena	VOLANT_0 VOLANT_1	1106	21	14
11-P8	Braccio	Braccio	VOLANT_0 VOLANT_1	1107	21	15

Esempio di utilizzo:

14. in **PLC\_SYN.INC**:

```

/* ----- */
/*          WORD A2          */
/* ----- */
#define      MLAMP10          A2.word
/* APPOGGIO PER LAMPADE MENU 10 e 11 */
#define      LMP1000          A2.bit._00
/* PULSANTE P1 MENU 10 */
#define      LMP1001          A2.bit._01
/* PULSANTE P2 MENU 10 */
#define      LMP1002          A2.bit._02
/* PULSANTE P3 MENU 10 */
#define      LMP1003          A2.bit._03
/* PULSANTE P4 MENU 10 */
: . . . .
: . . . .

```

15. in **QAL.C**:





```
/*-----
GESTIONE DEI TASTI FUNZIONE CUSTOM
-----*/
if( PUSER && (CODTAS == 1002) && !ROVERCU )
{
    ROVERCU=1;
}
else if( PUSER && (CODTAS == 1002) && ROVERCU )
{
    ROVERCU=0;
    OVRFRE0=100;
    OVRFRE1=100;
    OVRFRE2=100;
    OVRFRE3=100;
    OVRFRE4=100;
    OVRFRE5=100;
}
LMP1002 = ROVERCU ;
/* ON-OFF OVERRIDE ASSI FREE */
if(ROVERCU)
{
    OVRFRE0= OVRFRE1= OVRFRE2= OVRFRE3= OVRFRE4= OVRFRE5= IOVR ;
}
. . . . .
. . . . .
FCRIUT= ABMAN && PUSER && (CODTAS == 1101);
/* RIBALTA PORTAUTENSILE DA TASTO CUSTOM */
LMP1009= INTERM && EVRIUT || UTERIBA ;
/* RIBALTAMENTO IN CORSO, ESEGUITO */
FCUTRI= ABMAN && PUSER && (CODTAS == 1102);
/* PORTAUTENSILE A RIPOSO DA TASTO CUSTOM */
LMP1010= INTERM && EVUTRIP || UTERIPO ;
/* P.UT. A RIPOSO IN CORSO, ESEGUITO */
FCSBUT= ABMAN && (PUSER && (CODTAS == 1104) || FCSBUT && !FCBLUT) ;
/* SBLOCCO UTENSILE DA TASTO CUSTOM */
LMP1012= INTERM && APPIMA || PAPERT ;
/* SBL. UT. IN CORSO, ESEGUITO */
FCBLUT= ABMAN && (PUSER && (CODTAS == 1103) || FCBLUT && !FCSBUT) ;
/* BLOCCO UTENSILE DA TASTO CUSTOM */
LMP1011= INTERM && CHPIMA || PCHIUS ;
/* BL. UT. IN CORSO, ESEGUITO */
. . . . .
. . . . .
e_u_wrw(21,MLAMP10);
/* SCRIVE SULLA WORD PER "LAMPAD" */
```

## 7.2 I/O su tastiera

Sulla tastiera può essere presente un modulo di I/O che comprende:

- 16 ingressi per pulsanti o selettori
- 8 uscite per leds e lampade

I dati relativi sono trasmessi dal/al CNC attraverso il normale cavo Tastiera ↔ CNC. Le caratteristiche elettriche sono descritte nella monografia CONNESSIONI HARDWARE.



Il PLC vede questi I/O attraverso i sinonimi:

<b>INPOUT</b>	<b>I/O → PLC</b>	INPOUT=1: è presente la scheda input/output sulla tastiera
<b>KCNOUT</b> (unsigned char)	<b>PLC → I/O</b>	Otto uscite per leds o lampade
<b>KCNOUTnn</b>	<b>PLC → I/O</b>	8 uscite per leds o lampade. <b>nn</b> vale <b>00 . . . 07</b> Sono le singole uscite di <b>KCNOUT</b>
<b>KCNINP</b> (unsigned short)	<b>I/O → PLC</b>	Sedici ingressi da pulsanti o selettori
<b>KCNINPnn</b>	<b>I/O → PLC</b>	16 ingressi da pulsanti o selettori. <b>nn</b> vale <b>00 . . . 15</b> Sono i singoli ingressi presenti su <b>KCNINP</b>

### 7.3 Modulo IKR (Interfaccia Keyboard Remoto)

Il modulo IKR è un dispositivo remotabile dal CNC fino a 50m che permette di collegare facilmente una serie di input ed output (pulsanti, selettori, lampade, volantini) che normalmente richiederebbero un cablaggio complesso.

Il modulo IKR comprende:

- 63 ingressi TTL per pulsanti e/o selettori (47 per tasti a matrice, 16 per selettori).
- 31 uscite open-collector per leds e lampade.
- 8 ingressi A/D a 8 bits (utilizzo tipico per potenziometri assi e mandrino).
- 3 contatori veloci d'impulsi per encoder (utilizzo tipico per volantini).

Tutti i dati sono inviati al CNC attraverso una linea seriale RS-422.

La connessione al CNC e le caratteristiche meccaniche ed elettriche sono descritte nella monografia NORME DI MONTAGGIO, INSTALLAZIONE E MANUTENZIONE (720P406).

#### 7.3.1 Sinonimi utilizzati per la gestione IKR

Per la gestione del modulo IKR tramite il PLC, sono presenti alcuni segnali di scambio tra IKR e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

<b>IKR</b>	<b>CNC → PLC</b>	Presenza, in configurazione (SWCNF.TAR), del modulo IKR. <i>Usato dalle routines standard della libreria PLC; NON modificare.</i>
<b>STBIKR</b>	<b>IKR → PLC</b>	Strobe nuovo tasto; segnala che è cambiata la situazione dei tasti collegati sulla matrice <i>Usato dalle routines standard della libreria PLC; NON modificare.</i>



<b>ACKIKR</b>	<b>PLC → IKR</b>	Ack del nuovo tasto. <b>Deve</b> ricopiare lo stato di <b>STBIKR</b> perché possa essere inviata una nuova situazione tasti <i>Usato dalle routines standard della libreria PLC; NON modificare.</i>
<b>KEY#M68</b>	<b>IKR → PLC</b>	15 segnali relativi ai tasti collegati sulla prima matrice. # vale <b>2 . . . 16</b> ; ( es.: KEY2M68, KEY14M68, ecc.) <i>Vedi nota a fine tabella</i>
<b>KEY#M69</b>	<b>IKR → PLC</b>	16 segnali relativi ai tasti collegati sulla seconda matrice. # vale <b>1 . . . 16</b> ; ( es.: KEY1M69, KEY16M69, ecc.)
<b>KEY#M70</b>	<b>IKR → PLC</b>	16 segnali relativi ai tasti collegati sulla terza matrice. # vale <b>1 . . . 16</b> ; ( es.: KEY1M70, KEY16M70, ecc.)
<b>KT116M67</b> (short)	<b>IKR → PLC</b>	16 ingressi per commutatori Possono fare riferimento ad un commutatore a 16 posizioni oppure a 4 commutatori 4 bit in codice binario <i>Vedi nota a fine tabella</i>
<b>VOL1IKR</b> (short)	<b>IKR → PLC</b>	Primo contatore veloce per encoder (Volantino 1) <i>Usato da aggvol() se ABILIKR = 1. NON modificare</i>
<b>VOL2IKR</b> (short)	<b>IKR → PLC</b>	Secondo contatore veloce per encoder (Volantino 2) <i>Usato da aggvol() se ABILIKR = 1. NON modificare</i>
<b>VOL3IKR</b> (short)	<b>IKR → PLC</b>	Secondo contatore veloce per encoder (Volantino 3) <i>Usato da aggvol() se ABILIKR = 1. NON modificare</i>
<b>ANALOG#</b> (unsigned char)	<b>IKR → PLC</b>	8 convertitori A/D 8 bit (0÷255) per segnali 0÷5Vdc # vale <b>1 . . . 8</b>
<b>LAMP#</b>	<b>PLC → IKR</b>	31 uscite per leds o lampade. # vale <b>1 . . . 31</b>

### Nota sui sinonimi

I sinonimi indicati in tabella sono definiti nel file DF\_IOIKR.H, presente sulla directory di progetto del PLC, che deve essere inserito in PLC\_SYN.INC dove consigliato:

```
/*=====      IKR      =====*/  
/* Inserire il file DF_IOIKR.H      */  
/* se presente il modulo IKR 040F061      */  
/*=====*/
```

Come indicato sul file stesso, tutti i sinonimi "predefiniti" possono essere modificati dal progettista del PLC, a sua esclusiva discrezione. Sono esclusi da questa possibilità solo quelli espressamente indicati come "non ridefinire".



## 7.4 Commutazione di due Tastiere

La presente prestazione è possibile con pannelli operatore del tipo con tasti a corsa lunga e con CNC di tipo BASIC.

I collegamenti hardware delle due tastiere sono descritti nella monografia NORME DI MONTAGGIO, INSTALLAZIONE E MANUTENZIONE (720P406) al capitolo 4.

Per la commutazione da una tastiera all'altra, occorre premere, sulla tastiera attiva, i tasti **[CTRL] [GOUP]** ([GOUP] è il tasto di ritorno al menu precedente P1-P8).

La commutazione viene eseguita dal CNC al momento del rilascio dei tasti sopra citati. Qualora la tastiera attiva non venga commutata, occorre controllare che il cablaggio fra T0 e R0 delle due tastiere sia eseguito correttamente: infatti il CNC attende per 0,5 secondi che la nuova tastiera diventi attiva e risponda correttamente, se ciò non avviene la commutazione è annullata .

La commutazione della tastiera viene eseguita anche con CNC in ALT.

La tastiera NON attiva ha il led [CAPS LOCK] che lampeggia.

Il funzionamento dei segnali di input/output dalle due tastiere è il seguente :

Tastiera ATTIVA	Uscita T0 della tastiera		Simbolici del PLC			
	MASTER	SLAVE	KCNINP00	KCNOUT00	KCNINP01 -- KCNINP15	KCNOUT01 -- KCNOUT15 (Uscite T1-T15)
Master	0	0	0	0	Sono attivi gli input della tastiera MASTER	Sono attivi gli output della tastiera MASTER. T1-T15 della slave tutti = 0
Slave	1	1	1	1	Sono attivi gli input della tastiera SLAVE	Sono attivi gli output della tastiera SLAVE. T1-T15 della master tutti = 0

Per il PLC, che deve conoscere lo stato attuale di commutazione, è disponibile il sinonimo:

KCNINP00	CNC → PLC	
		=0: E' attiva la tastiera MASTER (identificata con dip-switch 1=OFF) =1: E' attiva la tastiera SLAVE (identificata con dip-switch 1=ON)

Il volantino da pannello può essere collegato alla sola tastiera master, ed è attivo solo quando la tastiera master è attiva. Nel PLC occorre mettere RICVOL=0, quando è attiva la tastiera slave, quindi l'equazione di richiesta volantino deve essere:

```
RICVOL = !KCNINP00 && ( altre condizioni.. );
```

## 7.5 TASTIERA PORTATILE

I CNC ECS serie WIN possono essere dotati di una tastiera portatile (chiamata anche Console Remota o Teach Pendant) con le principali funzioni della tastiera CNC ed



eventualmente un Volantino o un Joystick tridimensionale. Per le caratteristiche e le modalità operative si rimanda alla documentazione apposita.

La Console Remota dialoga con il P.C. del CNC su linea seriale; un apposito task in ambiente Windows si occupa del dialogo con la Console e dell'interfaccia con il CNC ed il PLC. Sul PLC la routine di libreria **coremecs()** provvede al dialogo con il task di gestione, all'esecuzione dei comandi di competenza del PLC, a passare e ricevere al/dal PLC della Macchina i comandi e dati previsti.

La routine **coremecs()** va inserita nel PLC nella sequenza qui indicata:

```
/*-----  
      USO DELLA ROUTINE coremecs() IN QAL.C  
      (questa sequenza e' OBBLIGATORIA per il corretto funzionamento!)  
-----*/  
  
LMP1000= MU_ON && (!CR_START && !OESE && PUSER && (CODTAS == 1000)  
        || LMP1000 && CR_START ;  
/* Richiesta Attivazione Consol Remota */  
/* Standard */  
LMP1001= MU_ON && (!CR_START && !OESE && PUSER && (CODTAS == 1001)  
        || LMP1001 && CR_START ;  
/* Richiesta Attivazione Consol Remota */  
/* Con richiesta Start File di inizializzazione */  
if (LMP1001)  
{  
  CR_OPTYP= 1;  
/* Opzione 1= Con richiesta Start File di inizializzazione */  
}  
else  
{  
  CR_OPTYP= 0;  
/* Opzione 0= Standard */  
}  
CR_START= LMP1000 || LMP1001;  
/* Richiesta Attivazione Consol Remota */  
/* Viene resettato da coremecs() */  
FLGMEN6= CR_ACTIV;  
/* blocca alcuni tasti P1 - P8 con Console attiva */  
coremecs();  
/* GESTIONE CONSOLE REMOTA ECS */  
tastcnc();  
/* routine che gestisce la tastiera CNC: tasti P1...P8 */  
/* e tasti dedicati (START, REL, HOLD, AXES H., RESET) */  
tasfuecs();  
/* routine che gestisce la premuta dei tasti funzione */  
/* del PLC (P1-P8) in associazione alla configurazione */  
/* standard ECS definita in BASMENU.MEN. */  
modijog();  
/* passa al CNC il numero d'ordine dell'asse da muovere */  
/* in modo JOG e le modalita' di movimento */  
aggvol();  
/* gestisce input Volantini da SCHEDE ASSI, da TASTIERA */  
/* o da IKR (Interfaccia Keyboard Remota). */  
GestOvr3();  
/* routine che gestisce i tre override incrementali */  
/* ad encoder della tastiera BASIC e gli override incrementali*/
```



```

/* della Consol remota ECS */

Oppure, per Tastiere con Override a Potenzimetri:

if(CR_ACTIV)
{
gestovrc();
/* routine che gestisce gli override incrementali */
/* della Consol remota ECS */
}
else
{
gestovr();
/* gestisce gli override a potenziometri della tastiera */
}
/*-----*/

```

Per la gestione della console remota tramite il PLC, sono presenti alcuni segnali di scambio tra la Console stessa (**CR**) e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

Questi segnali sono sentiti, elaborati o generati dalla routine **coremecs()**.

<b>CR_START</b>	<b>PLC → CR</b>	Richiesta di attivazione della Console
<b>CR_OPTYP</b> (unsigned char)	<b>PLC → CR</b>	Tipo di attivazione 0= Standard 1= Con richiesta Start File di inizializzazione
<b>CR_ABORT</b>	<b>PLC → CR</b>	Richiesta di disattivazione della Console
<b>CR_FATTO</b>	<b>PLC → CR</b>	Fatto del Nuovo comando arrivato dalla Console Posto a 1 comunica alla CR che il comando è stato acquisito o eseguito.
<b>CR_ACTIV</b>	<b>CR → PLC</b>	La Console Remota è attiva. Posto a 1 in risposta alla richiesta <b>CR_START</b>
<b>CR_NEWC</b>	<b>CR → PLC</b>	Nuovo comando dalla Console
<b>CR_CODCO</b> (short)	<b>CR → PLC</b>	Codice del nuovo comando dalla Console al PLC. Sulla word CR_CODCO sono presenti tutti i comandi scambiati tra Console e CNC; il PLC, se necessario, deve eseguire quelli di sua competenza, indicati da <b>CR_NEWC= 1</b> .
<b>CR_VVOL1</b> (short)	<b>CR → PLC</b>	Valore del volantino della Console

Il Joystick (quando è presente in sostituzione del Volantino) permette di muovere la Macchina in maniera "intuitiva" per, ad esempio, eseguire l'autoapprendimento di punti significativi del pezzo montato in macchina. In questo caso vengono attivati i volantini in "modo feed" (vedi Par. 2.9.4). Per le caratteristiche e le modalità operative del Joystick si rimanda alla documentazione apposita della Console.



Per questa funzione sono presenti alcuni altri segnali di scambio tra la Console stessa (CR) ed il PLC che sono stati associati ai sinonimi qui di seguito elencati

<b>CR_JOYON</b>	<b>CR → PLC</b>	Richiesta di attivazione del modo Joystick Il PLC deve resettare CR_JOYON quando eseguita la funzione.
<b>CR_JOYOF</b>	<b>CR → PLC</b>	Richiesta di disattivazione del modo Joystick Il PLC deve resettare CR_JOYOF quando eseguita la funzione.
<b>CR_JOY10</b>	<b>CR → PLC</b>	Selezionato fattore di moltiplicazione per i movimenti da Joystick Può essere usato per attivare movimenti lenti o rapidi degli assi.
<b>CR_VIEW0</b>	<b>CR → PLC</b>	Selezionata vista 0 per il Joystick
<b>CR_VIEW1</b>	<b>CR → PLC</b>	Selezionata vista 1 per il Joystick
<b>CR_VIEW2</b>	<b>CR → PLC</b>	Selezionata vista 2 per il Joystick
<b>CR_VIEW3</b>	<b>CR → PLC</b>	Selezionata vista 3 per il Joystick
<b>CR_UHAL</b>	<b>CR → PLC</b>	Sblocco Joystick in tutte le direzioni
<b>CR_UHJX</b>	<b>CR → PLC</b>	Sblocco Joystick nella sola direzione X
<b>CR_UHJY</b>	<b>CR → PLC</b>	Sblocco Joystick nella sola direzione Y
<b>CR_UHJZ</b>	<b>CR → PLC</b>	Sblocco Joystick nella sola direzione Z
<b>CR_RTCP</b>	<b>CR → PLC</b>	Richiesta inserzione / disinserzione RTCP Il PLC, verificando lo stato attuale dell'RTCP (segnale OTWT), provvede all'attivazione / disattivazione tramite KEY sincrono. Il PLC deve resettare CR_RTCP quando ha eseguito la funzione.
<b>CR_VAX</b>	<b>CR → PLC</b>	Richiesta attivazione / disattivazione assi virtuali per piano inclinato (per la funzionalità Assi Virtuali vedi Par. 3.16) Il PLC deve resettare CR_VAX quando ha eseguito la funzione.
<b>CR_PLANE</b>	<b>CR → PLC</b>	Matrice dinamica attiva (ad es. dopo una richiesta attivazione del piano inclinato). Può essere usato per applicare una "rotazione" dei comandi di movimento provenienti dal Joystick in base alla posizione attuale degli assi rotanti della testa birotativa, per ottenere movimenti riferiti al mandrino come attualmente orientato.

Con Autoapprendimento attivo, la coremeccs() provvede a memorizzare le quote assi dei punti scelti dall'operatore secondo quanto indicato nella documentazione della CR. Se il PLC vuole realizzare qualcosa di diverso da quanto fatto dalla routine, deve sincronizzarsi con i segnali qui sotto indicati e resettarli subito (nello stesso loop di PLC) per impedire il "trattamento standard".

<b>CR_STORE</b>	<b>CR → PLC</b>	<b>Memorizzazione quote senza flag</b> La coremeccs(), se non impedita, esegue la funzione: <code>StoreAxisValue();</code>
-----------------	-----------------	--



---

---

CR_STOR0	CR → PLC	<b>Memorizzazione quote con flag 0 (interpolazione lineare)</b> La coremeccs(), se non impedita, esegue le funzioni: <code>set_marker("0");</code> <code>StoAxValMark();</code>
CR_STOR2	CR → PLC	<b>Memorizzazione quote con flag 2 (Interpolazione Spline)</b> La coremeccs(), se non impedita, esegue le funzioni: <code>set_marker("2");</code> <code>StoAxValMark();</code>
CR_STOR4	CR → PLC	<b>Memorizzazione quote con flag 4 (Interpolazione Spline con perdita di tangenza)</b> La coremeccs(), se non impedita, esegue le funzioni: <code>set_marker("4");</code> <code>StoAxValMark();</code>
CR_STOR6	CR → PLC	<b>Memorizzazione quote con flag 6 (Interpolazione Circolare)</b> La coremeccs(), se non impedita, esegue le funzioni: <code>set_marker("6");</code> <code>StoAxValMark();</code>





## CAPITOLO 8

### 8 FINESTRE VIDEO PERSONALIZZATE

I CNC ECS serie WIN permettono al progettista del PLC di realizzare delle FINESTRE VIDEO contenenti immagini di tipo Bitmap, campi di testo fisso o prelevato da file o variabile, campi di input o di visualizzazione di dati numerici, liste di selezione anche di tipo a discesa, strumenti di misura analogici o digitali e semplici lampade ON/OFF tipo LEDS di varia forma. Per ogni elemento è possibile definire specifici attributi quali colore, size e tipo di font carattere, ecc.

La costruzione delle finestre video suddette è realizzata tramite un apposito tool software "Resource Builder" (costruttore di risorse) la cui descrizione e modo d'uso sono ampiamente descritte nella monografia 720P361 "RESOURCE BUILDER".

Nella monografia suddetta sono riportate anche le specifiche funzioni di libreria PLC che permettono di gestire le finestre (apertura, chiusura, scrittura, test).

Qui vengono descritte alcune subroutines messe a disposizione del progettista per un facile uso della funzione FINESTRE VIDEO.

#### 8.1 Sinonimi utilizzati per la gestione FINESTRE VIDEO

Per la gestione delle FINESTRE VIDEO tramite il PLC, sono presenti alcuni segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

Questi segnali sono sentiti, elaborati o generati dalla routine **gestwplc()**

<b>OPE_WIN</b>	<b>PLC → CNC</b>	Richiesta di apertura Window PLC
<b>WIN_PLC</b>	<b>CNC → PLC</b>	Comando di apertura Window PLC inoltrato
<b>WIN_OPN</b>	<b>CNC → PLC</b>	Window PLC aperta
<b>WIN_VIS</b>	<b>CNC → PLC</b>	Window PLC visibile
<b>WIN_ENT</b>	<b>CNC → PLC</b>	Premuto ENTER per Window PLC
<b>WIN_ESC</b>	<b>CNC → PLC</b>	Premuto ESC per Window PLC
<b>WPLCTST</b> (short)	<b>CNC → PLC</b>	Stato della Window (letto da <b>wTestWindow()</b> ). -1 : è stato premuto ESC 0 : la Window non è aperta 1 : la Window è aperta ma non visibile 2 : la Window è aperta e visibile 3 : è stato premuto ENTER
<b>PARPLCnn</b> (float)	<b>CNC ↔ PLC</b>	25 variabili per lo scambio dei dati dal PLC alla Window e viceversa. <b>nn</b> vale <b>00 . . . 24</b>



---

## 8.2 Routines a disposizione del PLC

### 8.2.1 Routine "WINPLC"

Chiamata: **winplc(\*NomeMde);**

Scopo: Definisce il nome della Window Custom PLC da visualizzare.

Parametro in ingresso:

- \* **NomeMde** = (char \*) NomeMde è una stringa di max 12 caratteri contenente il nome del file (con o senza estensione) che descrive la Window (generata con l'utility ECSFI.EXE) da aprire con la wOpenWindow(). Se l'estensione manca, è considerata = .MDE .

### 8.2.2 Routine "GESTWPLC"

Chiamata: **gestwplc();**

Scopo: Gestisce l'apertura, chiusura, visualizzazione e test delle Windows Custom PLC.

Segnali in ingresso:

**OPE\_WIN**

Segnali di uscita:

**WIN\_PLC, WIN\_OPN, WIN\_VIS, WIN\_ENT, WIN\_ESC, WPLCTST**

Il significato dei segnali suddetti è stato già spiegato al paragrafo 8.1 "Sinonimi utilizzati per la gestione FINESTRE VIDEO".

### 8.2.3 Routine "WWRITEWINDOW"

Chiamata: **esito = wWriteWindow(idField, \*text);**

Scopo: Esegue la richiesta di scrittura di una stringa di caratteri in un determinato campo "dynamic text" della finestra custom attualmente aperta..

Parametri in ingresso:

**idField** = (short) identificatore (0÷255) del campo "dynamic text" (questo valore viene assegnato durante la fase di progetto della finestra)

\* **text** = (char \*) è una stringa di max 100 caratteri da scrivere nel campo **idField** "dynamic text" della finestra custom attualmente aperta.

Le stringhe che possono essere contemporaneamente scritte in una finestra custom sono al massimo 20; la routine si preoccupa di "trasmetterle" alla window con le necessarie sincronizzazioni. Le stringhe che superano i 100 caratteri vengono troncate a tale lunghezza.

Parametro di uscita:

**esito** = (short) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

- 1 Richiesta accettata
- 1 Superato il numero massimo di stringhe ammesse



### 8.3 Esempi di gestione WINDOWS PLC

Nel seguente esempio si fa riferimento alla definizione dei tasti custom vista al paragrafo "Tasti funzione CUSTOM" del Capitolo 7.

```
/*-----  
    ESEMPIO D'USO ROUTINE GESTWPLC IN QAL.C  
-----*/  
  
LMP1000= !WIN_OPN && !LMP1000 && PUSER && (CODTAS == 1000) ||  
          LMP1000 && !(PUSER && (CODTAS == 1000)) && !WIN_ESC;  
/* Chiede visualizzazione e set tempi di lubrificazione assi */  
  
if (LMP1000 && !WIN_OPN)  
{  
    PARPLC00= (float)(TLUBAS / 60) ;  
    PARPLC01= (float)TLAVAS ;  
    winplc("LUBRIF");  
    OPE_WIN= 1;  
    /* Chiede apertura Window */  
}  
/* APRE LA WINDOW PLC */  
  
if (LMP1000 && WIN_ENT)  
{  
    TLUBAS= (int)(PARPLC00 * 60) ;  
    TLAVAS= (int)(PARPLC01) ;  
    lotime(15,(float)TLAVAS);  
    /* Setta nuovo valore in secondi */  
}  
LMP1003= !WIN_OPN && !LMP1003 && PUSER && (CODTAS == 1003) ||  
          LMP1003 && !(PUSER && (CODTAS == 1003)) || WIN_ESC );  
/* Chiede visualizzazione Help */  
if(LMP1003 && !WIN_OPN)  
{  
    winplc("HELP");  
    OPE_WIN= 1;  
    /* Chiede apertura Window */  
}  
/* APRE LA WINDOW PLC */  
.  
.  
.  
LMP1004= !WIN_OPN && !LMP1004 && PUSER && (CODTAS == 1004) ||  
          LMP1004 && !(PUSER && (CODTAS == 1004)) || WIN_ESC );  
/* Chiede visualizzazione temperatura cuscinetti mandrino */  
if(LMP1004 && !WIN_OPN)  
{  
    winplc("TEMPMND");  
    OPE_WIN= 1;  
    /* Chiede apertura Window */  
}  
/* APRE LA WINDOW PLC */  
if(LMP1004 && WIN_OPN)  
{  
    itoa(VISUALA, str1, 10);  
    wWriteWindow(41,str1);  
    itoa(VISUALB, str2, 10);  
    wWriteWindow(42,str2);  
}
```



```

}
/* SCRIVE NELLA WINDOW PLC */
if( OPE_WIN && !LMP1000 && !LMP1003 && !LMP1004 )
{
    OPE_WIN= 0;
}
/* Reset richiesta apertura Window PLC */
gestwplc();
/* Routine standard per gestione Windows PLC */
e_u_wrw(21,MLAMP10);
/* SCRIVE SULLA WORD PER "LAMPADE" */
/*-----*/

```

Nel seguente esempio si suppone di avere creato una Window MUSTATUS.MDE per la visualizzazione, tramite "leds", dello stato della Macchina Utensile (ripari, bloccaggi, cambio utensile, etc.). Per i leds è stata prevista la word PARPLC01. La window viene aperta su richiesta dell'operatore tramite un tasto funzione CUSTOM.

in **PLC\_SYN.INC**:

```

/* ----- */
/*                WORD A12                */
/* ----- */
#define      MLEDS          A12.uword
/* Ausiliario per i leds di stato della M.U. */
#define      LEDS1          A12.bit._00
#define      LEDS2          A12.bit._01
#define      LEDS3          A12.bit._02
#define      LEDS4          A12.bit._03
#define      LEDS5          A12.bit._04
. . .
. . .
#define      LEDS16         A12.bit._15
/* ----- */

```

in **QAL.C**:

```

/*-----*/
    GESTIONE DELLA WINDOW DI STATO M.U.
-----*/
AWINDOW= !WIN_OPN && PUSER && (CODTAS== 1005) ||
        AWINDOW && !(PUSER && (CODTAS == 1005) || WIN_ESC || ORST);
LMP1005=AWINDOW;
/* Ausiliario apertura window */

if(AWINDOW && !WIN_OPN)
{
    winplc("MUSTATUS");
    OPE_WIN=1;
}
/* Apre la window PLC */

if(AWINDOW && WIN_OPN )
{
    LEDS1= PORTAP;
    /* Porte aperte */
}

```



```
LEDS2= SBLTAV;
/* Tavola sbloccata */
LEDS3= SBLUTM;
/* Utensile sul Mandrino sbloccato */
LEDS4= SBLUTMAG;
/* Utensile sul Magazzino sbloccato */
LEDS5= BRCUAV;
/* Braccio C.Ut. avanti */
LEDS6= BRCUIN;
/* Braccio C.Ut. indietro */
LEDS7= BRCUOR;
/* Braccio C.Ut. orario */
LEDS8= BRCUAN;
/* Braccio C.Ut. antiorario */
. . .
. . .
LEDS14= GAMMA1;
/* Gamma 1 Mandrino inserita */
LEDS15= GAMMA2;
/* Gamma 2 Mandrino inserita */
LEDS16= !GAMMA1 && !GAMMA2;
/* Mandrino in folle */

PARPLC01=MLEDS;
/* Passa il valore di stato alla Window PLC */
}

if(OPE_WIN && !AWINDOW)
{
OPE_WIN= 0 ;
}
/* Chiusura Window PLC */
gestwplc();
/* Routine standard gestione Windows PLC */
e_u_wrw(21,MLAMP10);
/* SCRIVE SULLA WORD PER "LAMPADA" */
/*=====*/
```

Il software "Resource Builder" (costruttore di risorse) permette di definire strumenti di misura di tipo analogico o digitale variamente configurabili; il seguente esempio mostra come costruire uno strumento personalizzato del tipo a barra di leds.

Si suppone di avere creato una Window LEDMETER.MDE per la visualizzazione della corrente assorbita dal mandrino.

Per lo strumento è stata prevista la word PARPLC01. Sono stati inseriti 16 leds rettangolari affiancati abbinando ad ognuno un bit diverso della PARPLC01; il bit 0 è stato associato al led di inizio scala e via via gli altri fino al led 15 associato al fondo scala. I leds sono stati colorati con coppie di colori scura/chiaro per simulare lo stato di spento/acceso e si sono usati colori diversi (ad esempio verde, giallo e rosso) per evidenziare un assorbimento normale, al limite, eccessivo.

La window viene aperta su richiesta dell'operatore tramite un tasto funzione CUSTOM.

in **QAL.C**:

```
.
.
#include "df_syn.inc"
#include "plc_syn.inc"
```

```
/* Conversione da Numero a bits */
short CNumBits(short number)
{
  short index, result;
  result= 1;
  for (index=1; index <= number; ++index)
  {
    result = result * 2;
  }
  return (result-1);
}
/*----- INIZIO LOGICA -----*/

void gal( void )
{
  LMP1008= !WIN_OPN && !LMP1008 && PUSER &&(CODTAS==1008) ||
          LMP1008 && !(PUSER &&(CODTAS==1008) || WIN_ESC );
  /* Chiede visualizzazione corrente mandrino */
  if(LMP1008)
  {
    PARPLC01= (float)(CNumBits((int)(ADW0*16/2047)));
    /* 16= numero massimo di leds da accendere */
    /* 2047= valore massimo del convertitore A/D con 10 Volts */
    /*      in ingresso, corrispondenti al fondo scala      */
    if(!WIN_OPN)
    {
      winplc("LEDMETER");
      OPE_WIN= 1;
      /* Chiede apertura Window */
    }
  }

  if(OPE_WIN && !LMP1008)
  {
    OPE_WIN= 0 ;
  }
  /* Chiusura Window PLC */
  gestwplc();
  /* Routine standard gestione Windows PLC */
  e_u_wrw(21,MLAMP10);
  /* SCRIVE SULLA WORD PER "LAMPADE" */
}

```

## 8.4 Logo personalizzato

I CNC ECS serie WIN permettono al costruttore della Macchina Utensile di personalizzare il logo che appare sul video, sotto i tasti del menù verticale (quello sulla destra).

A questo scopo occorre sostituire il logo standard ECS, presente come LOGO.BMP in C:\ECS.CNC\WINVIS\ICONE, con un file di ugual nome e caratteristiche (64 x 32 pixels x 16 colori).

## 8.5 Gestione formato attuale del CNC

I CNC ECS serie WIN permettono al progettista del PLC di leggere quale formato di visualizzazione del CNC sia attualmente attivo e di forzarne uno diverso.



---

### 8.5.1 IOPLCWinvisActualForm: gestisce formato attuale CNC

Scopo: Legge e/o modifica il formato di visualizzazione attuale del CNC.

Chiamata: **esito = IOPLCWinvisActualForm(flag, form)**

Parametri di ingresso:

**flag** = (short) tipo di comando

= 0 Si vuole leggere il formato di visualizzazione attuale.

= 1 Si vuole attivare il formato di visualizzazione desiderato.

**form** = (short) Numero del formato che si vuole attivare. Con **flag= 0** il parametro form non viene considerato.

Parametro di uscita:

**esito** = (short ) valore di ritorno delle routine . Può assumere uno dei seguenti valori:

**0** Successo: comando inoltrato

**1** Errore: altro comando già in corso

Per una completa gestione del comando, il PLC deve testare anche i valori assunti dalle variabili associate **IOPLCSTS** e **IOPLCRET** che possono assumere i valori:

**IOPLCSTS** = -1 : Comando inoltrato ad esecutore (task WINVIS)

(short) = 1 : Operazione in corso di esecuzione

= 0 : Operazione terminata

**IOPLCRET** = -1 : KO; task Winvis non attivo; comando non eseguito

(short) = -2 : KO; numero di formato non valido (con flag= 1)

= 1: OK; comando di aggiornamento formato accettato (con flag= 1)

> 0 : OK; numero del formato attuale del CNC (con flag= 0)

I numeri di formato previsti sono:

- 1 Formato base (quote grandi)
- 2 Formato completo (posizioni + distanze)
- 3 Formato quota errore (posizioni + quote errore)
- 4 Formato quota reale (quote reali + conti sa)
- 5 Formato quota DRF (posizioni + quote DRF)
- 6 Formato digitizer (posizioni + deflessioni)
- 7 Formato info stato degli assi
- 8 Formato fine corsa
- 9 Formato Grafica Real Time
- 10 Formato Grafica Real Time espanso



---

---

**Note:**





---

---

## CAPITOLO 9

### 9 CAMBIO UTENSILI

#### 9.1 GENERALITÀ

I CNC ECS della serie WIN controllano gli **utensili** della Macchina Utensile (M.U.) attraverso uno specifico Software, sempre presente, denominato **TCI (Tool Controller Interno)** che permette di gestire fino a 150 utensili con un massimo di 255 "taglienti" (ovvero correttori) e svariati tipi di cambi utensili.

**{T}** Per il Torno (CNC ECS 1802 / 4802) la gestione del cambio utensili è più semplice, data la ridotta tipologia che si riscontra nella pratica; per il tornio si rimanda quindi al paragrafo 9.12 CAMBIO UTENSILI NEL TORNIO e successivi.

In un apposito campo del file di taratura TYPUCU.TAB (vedi MANUALE TARATURE 720P385) va precisato se il cambio è MANUALE o AUTOMATICO; in caso di Cambio Utensili Automatico (CUA), nel file di taratura CUMAG.TAB va descritta la configurazione del magazzino utensili.

Se la M.U. non ha un cambio utensili automatico, il CNC emette ugualmente il codice "T" relativo all'utensile desiderato, anche se questo dato non è utilizzato dal PLC. La Funzione Ausiliaria (F.A.) "**M06**" (definita FASUB) fornita di base da E.C.S. sui propri CNC, richiama la subroutine **CUM** che arresta la prosecuzione del Part-program e visualizza sul monitor un messaggio del tipo:

**Cambiare utensile  
poi premere [START].**

Il CNC attende che l'operatore prema il pulsante [START], dopo aver cambiato l'utensile manualmente, per procedere nell'esecuzione del Part-program.

Se la M.U. ha un cambio utensili automatico (CUA), il TCI del CNC è in grado di gestire le varie fasi di "RICERCA" e "CAMBIO" oltre che quelle di "CARICO-SCARICO" per diversi tipi di cambi, sia nel caso di codifica diretta degli utensili che nel caso di codifica dei posti del magazzino utensili, come illustrato nei paragrafi seguenti.

La Funzione Ausiliaria (F.A.) "**T**" (definita da E.C.S. FASUB), richiama la subroutine TOOL (fornita di base sui CNC ECS) che contiene l'istruzione \$T<%%V7005>. Questa istruzione, oltre ad essere emessa verso il PLC, attiva il TCI che, in base alle tarature presenti:

- se necessario, deposita un utensile già prelevato,
- comanda la ricerca dell'utensile desiderato,
- comanda il prelievo dello stesso in attesa del cambio,
- oppure non fa niente e memorizza il T desiderato fino alla prossima M06.

La Funzione Ausiliaria (F.A.) "**M06**" (definita da E.C.S. FASUB e fornita di base sui propri CNC), esegue tutta una serie di verifiche e poi emette semplicemente la F.A. \$M06 oppure richiama la subroutine CUM (per il cambio utensili manuale, ad esempio, di un utensile "a terra") oppure la subroutine CUA (per il cambio utensile automatico). Queste predispongono il tipo di cambio utensile da eseguire (da subroutine oppure da PLC) ed attivano il TCI tramite l'istruzione \$M06; il TCI, in base alla situazione utensili attuale, alle predisposizioni fatte ed alle tarature presenti, emette una serie di comandi verso il PLC o verso la



Subroutine CUA per eseguire le operazioni (prelievo, deposito, ricerca, cambio) necessarie per completare il cambio utensile programmato.

Come precedentemente accennato, il TCI del CNC permette di realizzare il Cambio Utensili tramite due metodologie:

#### a) CUA da PLC

La sequenza dei movimenti sulla Macchina Utensile e lo scambio di segnali con il CNC ed il Tool Controller sono realizzati tramite equazioni scritte nel PLC. Normalmente vengono usati uno o più sequenziatori ("macchine" logiche a più stati con specifiche azioni legate ai singoli stati e condizioni prefissate che controllano il passaggio da stato a stato; il linguaggio "C" permette di realizzare tali "macchine" con la funzione `switch(...)` ed i `case nnn:` associati).

#### b) CUA da Subroutine

La sequenza dei movimenti sulla Macchina Utensile e lo scambio di segnali con il CNC ed il Tool Controller sono realizzati tramite un Sottoprogramma scritto con istruzioni ISO e LIP2.

I comandi per gli organi meccanici del cambio ( bracci, pinze, ecc.) sono normalmente comandati tramite Funzioni Ausiliarie M e possono essere usate le apposite aree ILIP1÷ILIP5 (%V4178÷%V4182) per il controllo di pressostati, microswitch, ecc. .

I movimenti degli assi (CNC e/o Indipendenti) sono comandati con le normali istruzioni previste dalla programmazione.

Il CUA da Subroutine è particolarmente utile ed usato quando ci sono da controllare diversi Assi e relativi movimenti, come nel caso di Magazzino Utensili a rastrelliera o sono usati motori in Corrente Continua anziché attuatori idraulici per i vari organi mobili.

### 9.1.1 Parametri di taratura del Cambio Utensili

Le caratteristiche del Cambio Utensili sono definite negli appositi files di taratura ed in particolare in TYPUCU.TAB.

Qui di seguito, sono riassunti i vari files che contengono tarature e/o procedure riguardanti il Cambio Utensili.

TYPUCU.TAB	Tipo di cambio utensile
CUMAG.TAB	Configurazione del magazzino
COST	Parametri per magazzino utensili (se asse CNC)
COSTUSER	Personalizzazione dei parametri di COST
CUM	Ciclo di cambio utensili manuale
CUA	Ciclo di cambio utensili automatico
CUA.SUB	Ciclo CUA per magazzini a rastrelliera o simili
CUM.SUB	Ciclo c. ut. manuale se usato CUA.SUB
TOOL	F.A. subroutine "T".
DCOR	F.A. subroutine "D" (per correttore utensile)

La funzione di questi files è dettagliatamente spiegata nel MANUALE TARATURE 720P385.



I files CUM e CUA, presenti sul CNC, sono adatti per un Cambio Utensili "da PLC" e normalmente non richiedono modifiche; essi possono comunque essere ritoccati per adattarli alle esigenze particolari di una determinata Macchina Utensile.

I files CUM.SUB e CUA.SUB forniti da ECS sono le basi di partenza per la realizzazione di un Cambio Utensili "da Subroutine" e debbono sostituire i corrispondenti CUM e CUA quando si vuole usarli. Naturalmente, vanno completati con le istruzioni necessarie per la Macchina Utensile particolare che si sta progettando.

## 9.2 GESTIONE DEL MAGAZZINO UTENSILI

Il TOOL CONTROLLER INTERNO (TCI) è il software responsabile della gestione per tutto quanto riguarda gli utensili a bordo macchina per cui deve conglobare molteplici funzioni; fra queste le principali sono:

- Fornire al CNC il giusto utensile da montare sul mandrino per la lavorazione successiva.
- Permettere il monitoraggio e la modifica dello stato del magazzino e dello stato d'ogni singolo utensile.
- Gestire l'occupazione del magazzino in modo da ottimizzarla.
- Gestire tutte le operazioni di carico/scarico che possono essere eseguite automaticamente.
- Gestire la vita e l'usura d'ogni singolo tagliente permettendo di ottimizzarne l'utilizzo, aggiornando le tabelle opportune.

La configurazione del magazzino utensili viene definita dal costruttore M.U. in fase d'installazione del CNC compilando il file **CUMAG.TAB**. Questo file permette sia il dimensionamento del magazzino (numero di posti ed eventuali posti dedicati) sia la definizione dell'ingombro d'ogni tipo d'utensile che potrà essere montato su di esso (piccolo, medio, grande, ecc.) sia il tipo di gestione dei posti (posti dedicati, random, random fisso).

### 9.2.1 Gestione a POSTI DEDICATI

Questo tipo di gestione permette di assegnare a priori la posizione d'ogni singolo utensile sul magazzino (corrispondenza tra codice utensile e suo posto); questa disposizione rimane poi inalterata durante il funzionamento. Questo comporta che non ci saranno scambi di posto fra gli utensili ma, una volta definita la corrispondenza utensile/posto, questa sarà sempre mantenuta.

### 9.2.2 Gestione RANDOM

Questo tipo di gestione permette di inserire gli utensili in magazzino in qualunque posto libero. La ricerca del posto migliore (secondo una strategia definita che sarà trattata più avanti) sarà attuata nella fase di spostamento dell'utensile da mandrino a magazzino. In questo caso può succedere che un utensile cambi continuamente posto sul magazzino tutte le volte che viene scaricato dal mandrino.



---

### 9.2.3 Gestione RANDOM FISSO

Questo tipo di gestione è analoga a quella di tipo RANDOM, ma solo in fase d'introduzione di un nuovo utensile sul magazzino.

In caso di carico automatico (occorre ci sia una stazione di carico/scarico: MID=0 in CUMAG.TAB) la decisione di dove inserire ogni nuovo utensile viene presa dal TCI; una volta caricati gli utensili ed associati ad un posto specifico, non possono più cambiarlo. Chiaramente il posto prescelto dal TCI è quello che ottimizza l'occupazione del magazzino.

In caso di carico fatto "a mano" (perché non è presente una stazione di carico/scarico: MID=1 in CUMAG.TAB), l'ottimizzazione del magazzino sarà fatta la prima volta che ciascun utensile, una volta montato sul mandrino, viene da questo scaricato. Da questo momento, ciascun utensile non cambierà più posto.

### 9.3 DIMENSIONE DELL'UTENSILE

Sono previsti quattro diversi tipi d'utensili, con dimensioni definibili tramite tarature:

#### **PICCOLA, MEDIA, GRANDE, EXTRA**

Ad eccezione del primo tipo (PICCOLO), che occupa un solo posto magazzino, le dimensioni degli altri sono definite dal costruttore della Macchina Utensile nel file di taratura CUMAG.TAB. C'è da notare che la definizione dell'occupazione sul magazzino permette di distinguere anche la metà del posto-utensile, nell'una e nell'altra direzione. Per maggiore chiarezza faremo un esempio.

Consideriamo un magazzino circolare (catena o ruota). Chiamiamo C la circonferenza e indichiamo con C+ e C- il verso di percorrenza positivo (posti utensili crescenti) e negativo sulla circonferenza. Supponiamo che l'utensile A occupi mezzo posto, oltre al suo (Pa), nel verso C+. Ciò vuol dire che non è possibile inserire alcun utensile nel posto Pa+1, ma si potrà inserire in Pa+2 un utensile B che occupa nel verso C- mezzo posto. In tal modo il posto Pa+1 risulterà occupato per metà dall'utensile A e per metà dall'utensile B. In questo modo siamo riusciti a "compattare" due utensili di misura MAGGIORE del PICCOLO in tre posti adiacenti di magazzino.

La definizione e la gestione dei diversi tipi di utensili permette di costruire, ad esempio, un magazzino a catena con passo maglia ridotto e tale da poterlo riempire tutto con utensili piccoli; permette altresì, quando necessario, di montare utensili più ingombranti (che occupano parzialmente o totalmente i posti-catena vicini) senza preoccuparsi delle interferenze che saranno gestite automaticamente dal Tool Controller

### 9.4 TT.BIN (file della tabella utensili: Tool Table)

Tutti i parametri di tutti gli utensili (T) e correttori (D), inseriti in tabella manualmente o con la procedura di carico/scarico da PLC o con la procedura di carico dati da file (funzioni G79x, vedi Norme di programmazione), sono scritti in un file di nome TT.BIN. È un file "data-base" gestito dal CNC, non è visibile o modificabile con un normale editor ASCII poiché i dati sono registrati in formato binario.

Il file TT.BIN contiene 255 blocchi-dati. Ogni blocco-dati contiene tutte le informazioni relative ad un singolo tagliente (la gestione utensili prevede anche utensili multitagliente con un



codice fisico per il complesso e tanti codici tipo quanti sono i taglienti). La lettura/scrittura dei dati viene eseguita automaticamente dal CNC nelle varie fasi di lavoro della Macchina Utensile.

L'operatore può inserire nuovi dati, vederli e modificarli utilizzando le apposite "maschere" accessibili tramite i tasti funzione specifici del menù orizzontale "Utensili" (vedi monografia NORME D'USO).

Per l'accesso (in lettura e/o ricerca) al file TT.BIN da parte del PLC, sono disponibili le funzioni **IOPLCnread\_tt(...)** e **IOPLCGet\_nread\_tt(..)** che saranno descritte più avanti in questo capitolo, assieme alla struttura dati della Tool Table.

## 9.5 Routines a disposizione del PLC

### 9.5.1 IOPLC\_nread\_tt: lettura dei dati di un'utensile

Chiamata: esito= IOPLC\_nread\_tt( Indi, Dead, Codtpc, Codttc, Codpos )

Scopo: Richiede la lettura di tutti i dati presenti nella tabella utensili (TT.BIN), relativi ad un dato utensile selezionato in base a vari criteri. Le chiavi di lettura per prelevare i dati sono le seguenti, in ordine di priorità:

- 1) per indice (Indi > 0)
- 2) per status (Dead > 0)
- 3) per codice tool e posto in base alla combinazione dei tre parametri : Codtpc, Codttc, Codpos.

Parametri in ingresso:

<b>Indi</b> (unsigned short)	> 0: lettura dei dati dal record di indice specificato (1....256). Dalla release SW V4.00.06 i record sono 1024 (1...1024) 0: record non specificato (valgono i parametri seguenti)
<b>Dead</b> (unsigned short)	1: ricerca di un utensile morto per usura 2: ricerca di un utensile morto per vita 0: nessuna ricerca di utensile morto
<b>Codtpc</b> (long)	> 0: lettura dei dati dell'utensile di codice fisico specificato 0: ricerca non in base al codice fisico
<b>Codttc</b> (long)	> 0: lettura dei dati dell'utensile di codice tipo specificato 0: ricerca non in base al codice tipologico
<b>Codpos</b> (unsigned short)	> 0: lettura dei dati dell'utensile presente nel posto dato 0: ricerca non in base al posto

Parametro in uscita:

<b>esito</b> (short)	<b>1</b> : canale occupato <b>0</b> : comando inoltrato
-------------------------	--

Per una corretta gestione del comando, il PLC deve testare anche i valori assunti dalle variabili associate **IOPLCSTS** e **IOPLCRET** che possono assumere i valori:

<b>IOPLCSTS</b> (short)	= -1 : Comando inoltrato ad esecutore (task WINVIS) = 0 : Operazione terminata = 1 : Operazione in corso di esecuzione
----------------------------	--



**IOPLCRET** < 0 : KO : Errore  
(short) > 0 : OK. IOPLCRET è il numero del record letto nel file TT.BIN.  
= 0 : Nessun record letto o record file TT.BIN vuoto

Il PLC deve quindi testare, nell'ordine:

4. **esito** : se= 1 chiamare nuovamente la routine; se=0 passare alla fase successiva
5. **IOPLCSTS** : se=0 passare alla fase successiva altrimenti attendere
6. **IOPLCRET**: se > 0 successo altrimenti segnalare l'errore all'operatore

Questa funzione NON restituisce, come risultato, il record desiderato, ma inoltra la richiesta di lettura; i servizi messi a disposizione dal sistema operativo provvederanno alla effettiva lettura del record dal file e lo renderanno disponibile per essere letto dal PLC con la funzione **IOPLCGet\_nread\_tt()** del paragrafo seguente.

### 9.5.2 IOPLCGet\_nread\_tt: lettura dei dati di un utensile

Chiamata: **esito**= IOPLCGet\_nread\_tt( BIN\_TTC \* table)

Scopo: Legge il record precedentemente richiesto con la funzione **IOPLC\_nread\_tt()**.

Parametro in ingresso:

**&table** indirizzo della struttura dati su cui scrivere i risultati  
(BIN\_TTC \*)

Parametro in uscita:

**esito** 0 : successo; i dati sono in **table** .  
(short) 1 : canale occupato  
2 : errore nel comando (vedere **IOPLCRET**)  
3 : errore (l'ultimo comando inoltrato non è una lettura di file TT.BIN)

I dati presenti nella tabella, relativi ad un particolare utensile (o ad un tagliente), sono contenuti in un "record" dei **1024** che compongono il file TT.BIN, ovvero in una struttura di dati binari organizzati secondo la seguente definizione del linguaggio C:

```
/*-----
      STRUTTURA FILE TT.BIN TOOL CONTROLLER
-----*/
typedef struct {
    long      tpc;      /* codice fisico dell'utensile      */
    long      ttc;      /* codice del tipo utensile         */
    unsigned short atl; /* vita attuale                    */
    unsigned short wtl; /* soglia di allarme di vita utens. */
    double    lxdata;   /* lunghezza utensile in mm        */
    double    lxoff;   /* offset di lunghezza in mm       */
    double    rdata;   /* raggio utensile in mm           */
    double    roff;    /* offset di raggio in mm          */
    double    lzdata;  /* lunghezza secondaria utensile   */
                    /* definita in mm (per tornio)     */
    double    lzoff;   /* offset lungh. seconda in mm     */
    double    maxlxo;  /* massimo offset sulla lunghezza  */
    double    maxrw;  /* massimo offset sul raggio       */
    double    maxlzo;  /* massimo offset su lungh. second.*/
    double    maxp;    /* max errore ammesso su correzioni*/
                    /* prima volta                     */
    double    maxu;    /* max errore unitario su correzioni*/
}
```



CAMBIO UTENSILI

```
unsigned short pos; /* posizione dell'utensile */
unsigned short sts; /* stato utensile */
unsigned short cdr; /* Codice Quadrante (per Torno) */
unsigned short com; /* descrizione utens.(forma grafica)*/
long color; /* colore utensile (grafica) */
unsigned short m_atl; /* memoria vita utensile definita */
unsigned short m_wtl; /* memoria soglia di allarme vita
/*utensile.E' la differenza fra vita
/* utensile e soglia, definite
/* dall'operatore (atl - wtl)

/*----- Dati utilizzati da CAMTORNI -----*/
char comment[40];/* Descrizione
double lung_plac; /* Lunghezza placchetta
double alfa_plac; /* Angolo Alfa della placchetta
double beta_plac; /* Angolo Beta della placchetta
double passo; /* Passo
double speed; /* Velocita' mandrino desiderata
double feed; /* Avanzamento assi desiderato
double alt; /* Altezza utensile
double lar; /* Larghezza utensile
double rag_tgl; /* Raggio Tagliente

/*----- DATI PRESENTI DA RELEASE SOFTWARE V3.00 ----- */
double d_camtl; /* Disponibile per appoggi vari
unsigned short tip_ft; /* Ambiente di utilizzo utensile
/* 0:Utensile per fresa. Non si
/* considerano i parametri
/* relativi alla seconda
/* lunghezza ed al quadrante
/* 1:Utensile per tornio. Si
/* considerano i parametri
/* relativi alla seconda
/* lunghezza e al quadrante
unsigned short u_tcil; /* Disponibile per appoggi vari
/*----- Dati disponibili per costruttore -----*/
unsigned long l_cost1; /* Disponibile per appoggi vari
unsigned long l_cost2; /* Disponibile per appoggi vari
double d_cost1; /* Disponibile per appoggi vari
double d_cost2; /* Disponibile per appoggi vari
} BIN_TTC;
```

**N.B.** Per Release Software precedenti la V3.00, i dati sopra indicati come double sono invece del tipo float e non sono presenti quelli seguenti rag\_tgl.

Il campo pos (posto utensile) può valere (in decimale):

0	utensile non ancora montato sul magazz.
1...150	posto di magazzino o torretta
1000	utensile sul mandrino principale
1010	posto intermedio per cambio utensile mandrino principale
1011	posto intermedio per deposito mandrino principale
2000	utensile sul mandrino secondario
2010	posto intermedio per cambio utensile mandrino secondario
2011	posto intermedio per deposito mandrino secondario
3000	posto intermedio carico/scarico
7000	posto fittizio per utensili a terra
8000	posto fittizio per correttori
9000	appoggio per cambio doppio

Il campo `sts` (stato utensile) è codificato a bits:

Bit	valore decimale	nome ECS	descrizione
0	1	DISP	indica se il tool è disponibile per il Tool Controller. = 0 se definito ma ancora non montato sul magazzino
1	2	NEWCUT	utensile nuovo, che non ha mai lavorato o ha eseguito una sola lavorazione
2	4	IN_USE	utensile usato almeno una volta
3	8	TEXP	utensile morto per fine vita
4	10	LEXP	ut. morto per usura o accidentalmente
5	20	MCUT	utensile multitagliente
6	40	MAG	magazzino dove risiede l'utensile;
7	80		attualmente sempre= 1.
8	100	SIZE	misura dell'utensile:
9	200		0= piccolo, 1= medio, 2= grande, 3= extra
10	400	TYPE	tipo di utensile:
11	800		0= normale, 1= a sfacciare, 2= speciale, 3= probe
12	1000	BAD	bad dati
13	2000	TPC_USE	utensile attualmente in uso
14	4000	UTER	utensile a terra
15	8000	DUSE	con questo bit si identifica un correttore di un utensile e non l'utensile. Con DUSE= 1 tutti gli altri bits devono essere 0

Il campo `com` (forma utensile) può valere (in decimale):

Valore decimale	nome ECS	descrizione
1	T_SPHERICAL_SHAPE	Fresa sferica
2	T_CYLINDER_SHAPE	Fresa cilindrica
3	T_TOROIDAL_SHAPE	Fresa toroidale
4	T_BIT_SHAPE	Fresa elicoidale
5	T_TAP_SHAPE	Maschio
6	T_PROBE_SHAPE	Probe
20	F_ROMBO	Romboidale
21	F_TRONC	Troncatore
22	F_TONDO	Tondo
23	F_GODRO	Godronatore
24	F_QUADR	Quadrato
25	F_FILET	Filettatore

Nel PLC, per poter operare sulla Tabella Utensili, occorre definire la struttura dati locale **table**, di tipo **BIN\_TTC**, su cui verranno scritti i dati letti dalla **IOPLCGet\_nread\_tt()** ed una serie di variabili di appoggio su cui operare; vediamo un esempio di ricerca di utensile morto , cancellazione, oppure ripristino utensile:

```
#include "plc_syn.inc"
#include "df_syn.inc"
#include "ibmkeys.inc"
#include <stdlib.h>
#include <string.h>
```





## CAMBIO UTENSILI

---

```
#include <math.h>

char str1[60];
char *pntstr;

void ttlive (void)
{
  BIN_TTC tabella_ut;
  static unsigned short indi;
  static unsigned short indice;
  static unsigned short dead;
  static unsigned short codpos;
  static long codtpc;
  static long codttc;

  if(pon || ORST || !LMP1501)
  {
    SQZ_TT= 0;
    RKEYSN=KEY01=0;
  }
  ATT09= AUXON && ODEA; /* Ut morti magazzino */
  ATT10= AUXON && OTAWAR; /* Ut mandrino fine vita */
  ATT11= AUXON && OWAR && !ATT10 && !ATT09; /*UT fine vita magazzino */

  switch (SQZ_TT)
  {
  case 00: /* attesa */
    indi= 0;
    dead= 0;
    codtpc= 0;
    codttc= 0;
    codpos= 0;
    PARPLC00= 0;
    ATT35= 0;
    if(LMP1301 && WIN_OPN)
      SQZ_TT= 5;
    break;

  case 5: /* preparatoria chiamata di nread_tt */
    indi++;
    SQZ_TT= 10;
    break;

  case 10: /* chiamata di nread_tt */
    indice=indi;
    IND_TT= indice;
    RETNREAD=IOPLC_nread_tt(indice,dead,codtpc,codttc,codpos);
    if(RETNREAD==0)
      SQZ_TT= 12;
    break;

  case 12: /* verifica chiamata nread_tt */
    if(IOPLCSTS==0) /* Eseguito correttamente */
      SQZ_TT= 13;
    break;

  case 13: /* verifica chiamata nread_tt */
    if(IOPLCRET==0) /* nessun record letto o TT.BIN vuota */
```

---



```

    SQZ_TT=1010; /*TT.BIN vuota */
if(IOPLCRET>0) /* TT.BIN ok */
{
    if(indi >= 1024)
        SQZ_TT= 1000; /* Fine tabella */
    else
        SQZ_TT= 14;
}
break;
case 14: /* Chiamata lettura dati da TT.BIN */
    RETGETTT=IOPLCGet_nread_tt(&tabella_ut);
    if(RETGETTT==0) /* Eseguito correttamente */
    {
        STATOUT= tabella_ut.sts; /* stato utensile */
        POSTOUTM= tabella_ut.pos; /* posto utensile */
        TOOLM= (int)(tabella_ut.tpc); /* tool utensile */
        SQZ_TT= 20;
        TIME_23=0;
    }
break;

case 20: /* case di verifica */
    TIME_23=1;
    if( (STATOUT & 0x0008)==0x0008)
    {
        SQZ_TT= 30; /*Utensile morto per vita */
    }
    else
    {
        if(!LMP1502 && RITP_23 || LMP1502)
        {
            if(PARPLC00==4)
                SQZ_TT= 1000; /* FINE RICERCA */
            else
                SQZ_TT= 10;
        }
    }
break;

case 30: /* Attesa comandi da operatore */
    write_var (VL_INTG,10002,55,-1, & POSTOUTM) ; /* %#55=n.Posto */
    if(LMP1501 && WIN_ENT)
    {
        if(PARPLC00==1)
            SQZ_TT= 10; /* Proseguì ricerca */
        if(PARPLC00==2)
            SQZ_TT = 100; /* Cancella Utensile */
        if(PARPLC00==3)
            SQZ_TT = 200; /* Ripristina Utensile */
        if(PARPLC00==4)
        {
            SQZ_TT= 0;
            if(LMP1501)
            {
                LMP1501=0; /* Stop ricerca */
                OPE_WIN= 0; /* chiede chiusura window */
            }
        }
    }
}

```



## CAMBIO UTENSILI

```
    }
break;

case 100: /* case di cancellazione utensile */
    RKEYSN=!OKEYSN && !IKEYSN || RKEYSN;
    if(OKEYSN && IKEYSN)
    {
        KEY01=1;
        SQZ_TT=110;
    }
break;

case 110: /* attesa fine KEY */
    if(PSET && OVGENL==2000)
    {
        KEY01=0;
        RKEYSN=0;
        SQZ_TT=10;
    }
break;

case 200: /* ripristino vita utensile */
    if(INTC && OATCK || !INTC && !OATCK)
    {
        ITCOD= 25; /*RIPUTM*/
        IDTCH= IND_TT;
        IDTCL= 0;
        INTC= !INTC; /* Comando da PLC a TCI */
        SQZ_TT= 210;
    }
break;

case 210: /* Attesa fine ripristino utensile */
    if(INTC && OATCK || !INTC && !OATCK)
    {
        if(OESCOM== 1)
            SQZ_TT= 10;
    }
break;

case 1000: /* fine analisi tabella */
    if(!LMP1501)
        SQZ_TT= 0;
break;

case 1010: /* TT.BIN vuota */
    ATT35=1;
    if(!LMP1301)
        SQZ_TT= 0;
break;
} /* fine switch SQZ_TT */

if(LMP1501 && !WIN_OPN && ACTQMENU==15)
{
    LMP1502=1; /*TEST VELOCE */
    winplc("TTLIVE");
    OPE_WIN= 1; /* chiede apertura window */
}
```



```
if(LMP1501)
{
  if(ACTQMENU!=15 || WIN_ESC || PUSER && CODTAS==1501)
  {
    LMP1501= 0;
    OPE_WIN= 0;      /* chiede chiusura window */
  }
}

LMP1501= AUXON && (!LMP1501 && PUSER && CODTAS==1501 || LMP1501);
/* Verifica utensili morti */
LMP1502= LMP1501 && (!LMP1502 && PUSER && CODTAS==1502 ||
  LMP1502 && !(PUSER && CODTAS==1502));
/*Veloce SENZA ATTESA FRA UTENSILI */

if(LMP1501 && WIN_OPN)
{
  if(SQZ_TT== 30)
  {
    strcpy(&str1[0],"Utensile MORTO POSTO=          ");
    pntstr= itoa(POSTOUTM,&str1[22],10);
    wWriteWindow(1,str1);

    strcpy(&str1[0],"TOOL=          ");
    pntstr= itoa(TOOLM,&str1[6],10);
    wWriteWindow(2,str1);
    strcpy(&str1[0],"STS=0x          ");
    pntstr= itoa(STATOUT,&str1[7],16);
    wWriteWindow(3,str1);
  }
  if(SQZ_TT >= 10 && SQZ_TT<=29)
  {
    strcpy(&str1[0],"Ricerca in corso          ");
    wWriteWindow(1,str1);
    strcpy(&str1[0],"INDICE=          ");
    pntstr= itoa(IND_TT,&str1[7],10);
    wWriteWindow(2,str1);

    strcpy(&str1[0],"STS=0x          ");
    pntstr= itoa(STATOUT,&str1[7],16);
    wWriteWindow(3,str1);
  }
  if(SQZ_TT>=200 && SQZ_TT<=999)
  {
    strcpy(&str1[0],"Ripristino vita in corso          ");
    wWriteWindow(1,str1);
  }
  if(SQZ_TT== 1000)
  {
    strcpy(&str1[0],"Finito tutta la tabella utensili");
    wWriteWindow(1,str1);
    strcpy(&str1[0],"Premi ESC  ");
    wWriteWindow(2,str1);
    strcpy(&str1[0],"          ");
    wWriteWindow(3,str1);
  }
}
} /* Fine File */
```



Il testo del **KEY01** è il seguente:

```
N10 {File "KEY01" - CANCELLAZIONE POSTO TABELLA}  
N20 L<DELPOS>  
N30 <SET:2000>
```

La routine **DELPOS** è:

```
N10 {File "DELPOS" - COMANDI PER CANCELLAZIONE POSTO TABELLA}  
N20 <LCK:OFF>  
N25 G795  
N30 <POS=ITF(%#55)> G793  
N40 <WAT:CNC>  
N50 <RET>
```



## 9.6 CLASSIFICAZIONE DEI CAMBI UTENSILI AUTOMATICI

I CNC ECS 1801 / 2801 / 4801 per frese gestiscono tre tipi di cambio utensile:

- a) **TORRETTA**
- b) **CICLICO**
- c) **ACICLICO**

Questi cambi utensili sono di tipo "sincrono", poiché la gestione prevede che dopo un CAMBIO, l'utensile che è stato appena tolto dal mandrino rientri nel magazzino prima che inizi la RICERCA dell'utensile successivo. La gestione prevede anche cambi utensili con un utensile intermedio temporaneo. È il caso in cui può esistere un altro utensile fuori magazzino, oltre a quello montato sul mandrino, durante la rotazione del magazzino stesso per la ricerca del posto di deposito.

### a) CUA di tipo TORRETTA

- Gli utensili occupano sempre lo stesso posto nel magazzino.
- Posto Magazzino e T coincidono (fisicamente e logicamente); in altre parole il Tool Attuale TA lavora rimanendo nel magazzino oppure ne è estratto, ma RIENTRA nel suo posto magazzino PRIMA che sia possibile eseguire un'altra ricerca.
- L'istruzione **T** non provoca la ricerca dell'utensile nel magazzino; la ricerca è rimandata a quando sarà emessa l'istruzione **M06**, che provoca allora questa sequenza di eventi:

DEPOSITO: TA torna in Magazzino (eventuale)

RICERCA: viene cercato T nel Magazzino

PRELIEVO: T diventa TA (ed eventualmente viene estratto dal magazzino)

Tale tipo di cambio utensile è selezionato ponendo **TYP='T'** nel file TYPKU.TAB (vedi Manuale Tarature 720P385 al Cap.6).

### b) CUA di tipo CICLICO

Sono quei cambi utensili che non possono eseguire due o più RICERCHE consecutive nel magazzino, ma devono completare un CAMBIO dopo ogni RICERCA. In altre parole, essi possono eseguire solo cicli completi di RICERCA-CAMBIO, RICERCA-CAMBIO, ecc. .

Tale tipo di cambio utensile è selezionato ponendo **TYP='C'** nel file TYPKU.TAB .

### c) CUA tipo "ACICLICO"

Sono quei cambi utensili che possono eseguire due o più RICERCHE consecutive nel magazzino; non è necessario eseguire un CAMBIO dopo ogni RICERCA. In altre parole, possono eseguire cicli di tipo RICERCA-RICERCA-CAMBIO oppure CAMBIO-CAMBIO.

Tale tipo di cambio utensile è selezionato ponendo **TYP=' '** (spazio) nel file TYPKU.TAB .



## 9.7 SEGNALI LOGICI

Per la gestione del Cambio Utensili tramite il PLC, sono presenti diversi segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi qui di seguito elencati. Dove necessario è indicato il tipo di variabile secondo le regole del linguaggio C (usato per il progetto del PLC).

Alcuni di questi segnali sono già stati elencati al Cap.1, paragrafo 1.2.5 "Sinonimi utilizzati per le F.A.". Le modalità di sincronizzazione **CNC $\leftrightarrow$ PLC** sono descritte più avanti nello stesso capitolo.

Molti dei segnali elencati sono di uso raro, necessari per cambi utensili molto particolari oppure utili solo per una gestione più sofisticata; alcuni sono specifici per il CUA da Subroutine e saranno indicati più avanti.

{W} Nella prima colonna, il sinonimo scritto in **GRASSETTO** è quello da preferire; l'eventuale altro sinonimo è comunque valido ed è stato mantenuto per compatibilità con la serie D.

<b>NEWT</b>	<b>CNC → PLC</b>	<b>Nuova Funzione Ausiliaria T. (Tool)</b>
<b>NEWD</b>	<b>CNC → PLC</b>	<b>Nuova F.A. D. (correttore tool)</b>
<b>OTDS</b> (long) <b>OTDSL</b> (short) <b>OTDSH</b> (short)	<b>CNC → PLC</b>	<b>Valore della F.A. T o D o S (0 ÷ 4294967295).</b> Con NEWT= 1, il valore riguarda la T (Tool) Con NEWD= 1, il valore riguarda la D (correttore tool) Con NEWS= 1, il valore riguarda la S (Speed) OTDSL è la word leggera di OTDS OTDSH è la word pesante di OTDS
<b>FATTOT</b>	<b>PLC → CNC</b>	<b>Fatto della Nuova F.A. T.</b> Posto a 1 comunica al CNC che la T è stata acquisita. Il CNC può proseguire con la prossima istruzione.
<b>FATTOD</b>	<b>PLC → CNC</b>	<b>Fatto della Nuova F.A. D.</b> Posto a 1 comunica al CNC che la D è stata acquisita. Il CNC può proseguire con la prossima istruzione.
<b>OTCAUT</b> OAUTTC	<b>TCI → PLC</b>	<b>Tool Controller</b> in modo <b>AUTomatico</b> , cioè gestito attraverso specifiche richieste del CNC.
<b>OTCMCU</b> OMCU	<b>TCI → PLC</b>	<b>Cambio Utensile Manuale.</b> Non viene inviato dal TCI alcun comando verso il PLC, né ricevuto.
<b>OTCBMM</b> OBMM	<b>TCI → PLC</b>	<b>Blocco Movimentazione Magazzino.</b> Il Tool Controller non è in grado di accettare dal PLC comandi di movimentazione. Testare tale segnale prima di inviare al TCI comandi di carico/scarico, altrimenti interviene l'ALLARME 31810.
<b>OTCPNT</b> OPNT	<b>TCI → PLC</b>	<b>Programmato Nuovo Tool.</b> Questo segnale viene resettato con RESET o M06 eseguito.
<b>OTCRTP</b> ORTP	<b>TCI → PLC</b>	<b>Ricerca Tool Prenotata.</b> Tale segnale è reso vero quando viene emessa una F.A. "T" e la ricerca fisica del Tool richiesto (segnale ORFC=1) non è ancora iniziata. Testare il segnale OTCRTP quando il PLC desidera anticipare movimenti di predisposizione del cambio utensili.



<b>OTCRFC</b> ORFC	<b>TCI → PLC</b>	<b>Ricerca Fisica in Corso.</b> Rimane vero da quando viene emessa la T fino a che il movimento di ricerca ed eventualmente quello di prelievo sono terminati.
<b>OTCCUC</b> OCUC	<b>TCI → PLC</b>	<b>Cambio Utensile in Corso.</b> Viene reso vero all'emissione della M06 in modo Esecuzione e viene resettato quando è terminato tutto il ciclo di cambio utensile, eventualmente comprese le fasi di prelievo e deposito dell'utensile.
<b>CNTCOK</b>	<b>TCI → PLC</b>	Il TCI comunica di aver completato tutte le azioni relative al comando M06. Può essere usato per dare il fatto (FATTOM) alla M06.
<b>OTCDEA</b> ODEA	<b>TCI → PLC</b>	Uno o più utensili morti ( <b>DEAd</b> ) nel magazzino.
<b>OTCWAR</b> OWAR	<b>TCI → PLC</b>	Uno o più utensili in fine vita ( <b>WARning</b> ) nel magazzino.
<b>OTCTAWAR</b> OTAWAR	<b>TCI → PLC</b>	Utensile sul mandrino <b>TA</b> in fine vita ( <b>WARning</b> )
<b>ITCBT</b> IBT	<b>PLC → TCI</b>	<b>Blocca T.</b> Porre a 1 fintanto che la M.U. non è <u>meccanicamente</u> in grado di effettuare una ricerca utensili. Con ITCBT=1, il TCI non emette nessun comando di ricerca utensile, ma lo memorizza e lo emetterà alla M06 successiva.
<b>ITCBICU</b> IBICU	<b>PLC → TCI</b>	<b>Blocca Cambio Utensile.</b> Porre a 1 fintanto che la M.U. non è <u>meccanicamente</u> in grado di effettuare un cambio utensili. Con ITCBICU=1, il TCI non emette nessun comando di cambio utensile.
<b>ITCCRCU</b> ICRCU	<b>PLC → TCI</b>	<b>Cambio o Ricerca in Corso.</b> Con ITCCRCU=1 il TCI non emette nessun comando di ricerca o cambio utensile. Con RESET tale segnale deve essere rimesso a zero (se viene interrotta la sequenza di ricerca o cambio).
<b>ITCMPOS</b> IMPOS	<b>PLC → TCI</b>	<b>Magazzino in posizione.</b> Il PLC deve mettere ITCMPOS=1 quando il magazzino è posizionato sul posto richiesto. Il PLC deve mettere ITCMPOS=0 quando il magazzino viene mosso (anche in assenza di una ricerca comandata dal TCI) ed anche in situazioni di allarme dell'eventuale asse relativo alla movimentazione del magazzino utensili. Con ITCMPOS=0 e TRIC (utensile ricercato) uguale a TA (utensile attuale) oppure TI (utensile intermedio), il TCI, prima di emettere uno dei comandi PREUTE o DEPUTE o PREUCS o DEPUCS o CUTE (vedi più avanti), emetterà di nuovo la richiesta di posizionamento magazzino con MOVCU o MOVCS. Se TI o TA sono diversi da TRIC il TCI emette in ogni caso il comando di ricerca.





## CAMBIO UTENSILI

<b>OTCCUTER</b>	<b>TCI → PLC</b>	<b>Cambio Utensile “a terra”</b> Segnala che l'utensile da cambiare non è nel magazzino ma è dichiarato “a terra” e deve quindi essere cambiato manualmente
<b>OTCSECU</b>	<b>TCI → PLC</b>	<b>Secondo cambio Utensile</b> Segnala che viene comandato un secondo Cambio perchè con solo il primo M06 l'utensile desiderato non è andato sul mandrino (ad es. in CUA a posti dedicati).
<b>TCCURAS</b>	<b>PLC → TCI</b>	<b>Cambio utensile a rastrelliera</b> Segnala al TCI che occorrono due fasi di cambio; una per deposito ed una per prelievo.
<b>TCCUMAG</b>	<b>TCI → PLC</b>	<b>Cambio utensile da magazzino</b> Segnala che il cambio riguarda un utensile di magazzino
<b>TCCUREV</b>	<b>TCI → PLC</b>	<b>Cambio utensile da revolver</b> Segnala che il cambio riguarda un utensile sulla torretta (revolver).

**NOTA:**

*I segnali **ITCBT** e **ITCBICU** debbono identificare la posizione dei vari bracci, leve, rotismi, ecc.. del cambio in modo che non si modifichi il loro stato spegnendo e riaccendendo la Macchina Utensile. Sostanzialmente, assieme ad **ITCCRCU**, servono per i cambi di tipo **CICLICO**, proprio per informare il Tool Controller TCI quando la situazione meccanica del cambio impedisce la ricerca di un utensile ovvero un cambio.*

*Con **ITCBT=ITCBICU=1** ed **ITCCRCU=0**, il PLC indica al TCI che il cambio utensile automatico è meccanicamente fuori fase. In questo caso il TCI, con M06, emette l'allarme 31403, segnalando quindi all'operatore la necessità di effettuare manualmente il rifasamento della meccanica. Tale operazione deve essere eseguita ripristinando la situazione di riposo del C.Ut. attraverso i mezzi previsti dal costruttore della M.U. e la situazione degli utensili in base a quella memorizzata dal TCI (visibile nell'apposita finestra "Situazione Utensili in Magazzino" accessibile tramite i tasti-funzione del menu orizzontale, a partire dal tasto "Utensili"). Per la temporizzazione dei segnali suddetti si veda la **Figura 9/1**.*

*Con **ITCBT** o **ITCBICU=1** e **ITCCRCU=0** quando deve essere eseguita una ricerca o un cambio, il TCI emette il warning 31900 per segnalare tale situazione anomala.*

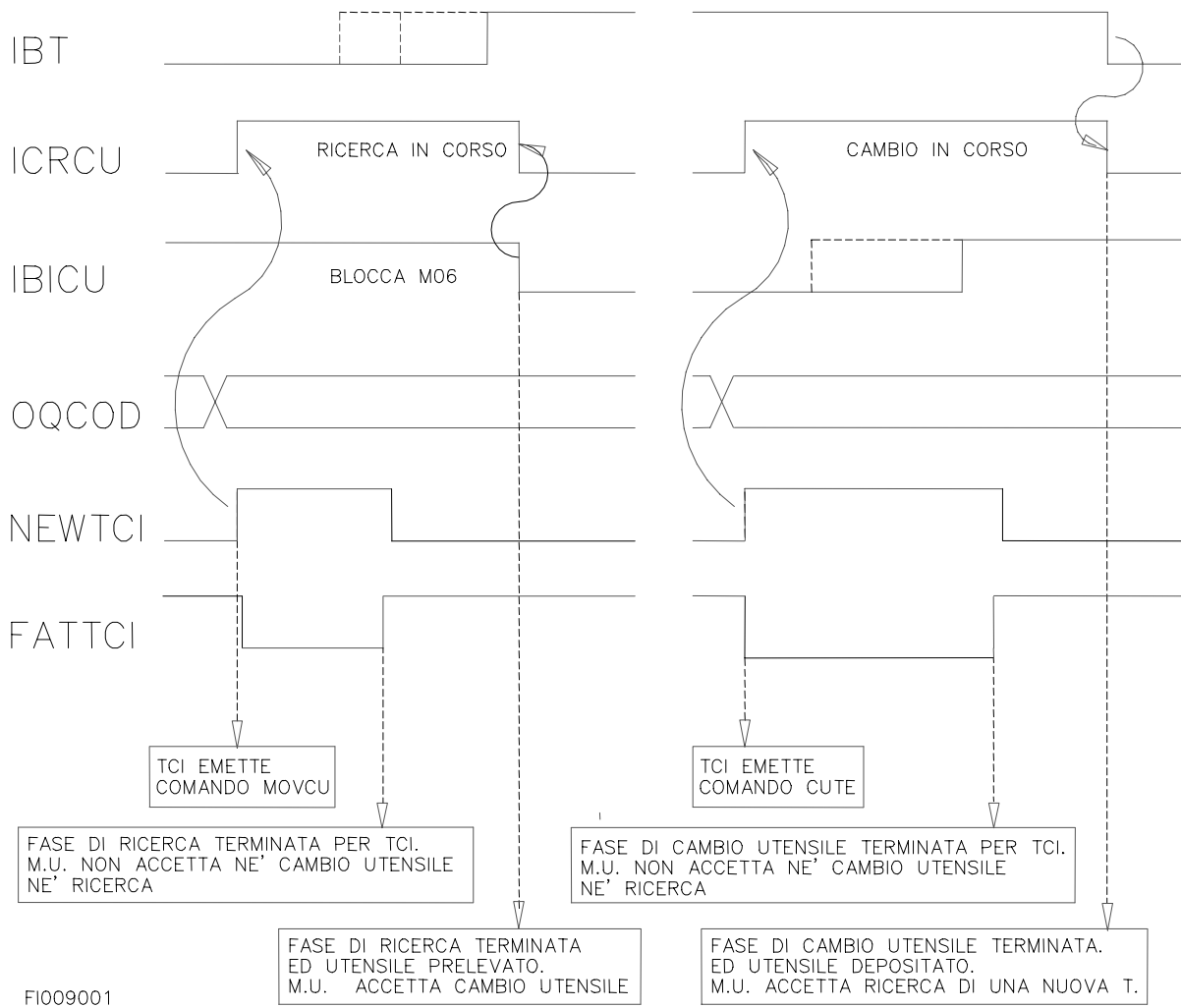


FIG. 9-1



I seguenti sinonimi si riferiscono ai segnali associati ai COMANDI da TCI al PLC.

{W} Nella prima colonna, il sinonimo scritto in **GRASSETTO** è quello da preferire; l'eventuale altro sinonimo è comunque valido ed è stato mantenuto per compatibilità con la serie D.

<b>NEWTCI</b>	<b>TCI → PLC</b>	<b>Nuovo comando da TCI a PLC.</b>
<b>OTCQCOD</b> OQCOD (unsigned char)	<b>TCI → PLC</b>	<b>Codice del comando inviato dal TCI al PLC.</b> Il valore è valido con NEWTCI= 1. I codici previsti sono descritti più avanti alla voce <b>COMANDI TCI ⇒ PLC</b> .
<b>FATTCI</b>	<b>PLC → TCI</b>	<b>Fatto del Nuovo comando da TCI.</b> Posto a 1 comunica al TCI che il comando è stata acquisito e/o eseguito. Il TCI può proseguire con il prossimo comando.
<b>OTCDATI</b> ODATI (long) <b>OTCDATIL</b> ODATIL (short) <b>OTCDATIH</b> ODATIH (short)	<b>TCI → PLC</b>	<b>Dati da TCI a PLC.</b> Sono dati associati al Nuovo comando da TCI a PLC. Su OTCQCOD c'è il codice che specifica il comando e quindi il tipo di dato. OTCDATI è una doppia word OTCDATIL è la word leggera di OTCDATI OTCDATIH è la word pesante di OTCDATI
<b>OTCSTDAT</b> ONDAT	<b>TCI → PLC</b>	<b>Nuovi dati da TCI a PLC.</b> I dati emessi sono riferiti al comando presente su OTCQCOD. <b>OTCSTDAT</b> è un segnale bistabile; ciò significa che tale segnale cambia di stato (0⇒1; 1⇒0) ogni volta che ci sono nuovi dati presenti su OTCDATI; il PLC deve dare la risposta di "dati acquisiti" ponendo il segnale <b>ITCAKDAT=OTCSTDAT</b> .
<b>ITCAKDAT</b> IAKDAT	<b>PLC → TCI</b>	<b>Acknowledge ai dati da TCI a PLC.</b> Risposta di "dati acquisiti" del PLC al TCI. Quando <b>OTCSTDAT</b> diventa diverso da ITCAKDAT, il PLC è avvisato che esistono nuovi dati su OTCDATI; una volta che i dati sono acquisiti, il PLC rimette <b>ITCAKDAT=OTCSTDAT</b> .

I seguenti sinonimi si riferiscono ai segnali associati ai COMANDI dal PLC al TCI.

<p><b>ITCSTTCI</b> INTC</p>	<p><b>PLC → TCI</b></p>	<p><b>Nuovo comando da PLC a TCI.</b> ITCSTTCI è un segnale bistabile; ciò significa che tale segnale deve cambiare di stato (0⇒1; 1⇒0) ogni volta che il PLC vuole inviare un nuovo comando al TCI. Il codice del comando deve essere stato preventivamente scritto su <b>ITCOD</b> ed i dati associati posti su <b>IDTC</b>. Il TCI darà la risposta di "dati acquisiti" ponendo il segnale <b>OTCAKTCI=ITCSTTCI</b> .</p>
<p><b>ITCOD</b> (char)</p>	<p><b>PLC → TCI</b></p>	<p><b>Codice del comando inviato dal PLC al TCI.</b> Il valore è considerato valido dal TCI con <b>ITCSTTCI</b> ≠ <b>OTCAKTCI</b>. I codici previsti sono descritti più avanti alla voce <b>COMANDI PLC → TCI</b>.</p>
<p><b>IDTCD</b> (double) <b>IDTC</b> (long) <b>IDTCL</b> (short) <b>IDTCH</b> (short)</p>	<p><b>PLC → TCI</b></p>	<p><b>Dati da PLC a TCI.</b> Sono dati relativi al Nuovo comando da PLC a TCI. Su <b>ITCOD</b> c'è il codice che specifica il comando e quindi il tipo degli eventuali dati. <b>IDTCD</b> è un float in doppia precisione <b>IDTC</b> è una doppia word <b>IDTCL</b> è la word leggera di <b>IDTC</b>. <b>IDTCH</b> è la word pesante di <b>IDTC</b>.</p>
<p><b>IDTC1D</b> (double) <b>IDTC1</b> (long) <b>IDTC1L</b> (short) <b>IDTC1H</b> (short)</p>	<p><b>PLC → TCI</b></p>	<p><b>Altri Dati da PLC a TCI.</b> Sono dati relativi al Nuovo comando da PLC a TCI. Su <b>ITCOD</b> c'è il codice che specifica il comando e quindi il tipo degli eventuali dati. <b>IDTC1D</b> è un float in doppia precisione <b>IDTC1</b> è una doppia word <b>IDTC1L</b> è la word leggera di <b>IDTC1</b>. <b>IDTC1H</b> è la word pesante di <b>IDTC1</b>.</p>
<p><b>OTCAKTCI</b> OATCK</p>	<p><b>TCI → PLC</b></p>	<p><b>Acknowledge al comando da PLC a TCI.</b> Risposta di "comando acquisito" del TCI al PLC. Quando <b>ITCSTTCI</b> diventa diverso da <b>OTCAKTCI</b>, il TCI considera che esiste un nuovo comando su <b>ITCOD</b>; una volta che il comando è stato acquisito e/o eseguito, il TCI rimette <b>OTCAKTCI=ITCSTTCI</b> .</p>
<p><b>OTCESCOM</b> OESCOM (short)</p>	<p><b>TCI → PLC</b></p>	<p><b>Esito del comando da PLC a TCI.</b> Su questa word è presente l'esito del comando richiesto dal PLC su <b>ITCOD</b>. <b>OTCESCOM=1</b>: esito OK; <b>OTCESCOM=-1</b>: ERRORE. In caso di errore, su <b>OTCDATQ</b> è presente il codice di errore.</p>



## CAMBIO UTENSILI

<b>OTCDATQD</b> (double) <b>OTCDATQ</b> ODATQ (long) <b>OTCDATQL</b> ODATQL (short) <b>OTCDATQH</b> ODATQH (short)	<b>TCI → PLC</b>	<b>Dati da TCI a PLC.</b> Sono dati passati dal TCI al PLC in risposta alla richiesta fatta dal PLC su ITCOD. OTCDATQD è un float in doppia precisione OTCDATQ è una doppia word OTCDATQL è la word leggera di OTCDATQ. OTCDATQH è la word pesante di OTCDATQ. In caso di esito errato, OTCDATQ è il codice di errore.
<b>ODATQ1D</b> (double) <b>ODATQ1</b> (long) <b>ODATQ1L</b> (short) <b>ODATQ1H</b> (short)	<b>TCI → PLC</b>	<b>Altri dati da TCI a PLC.</b> Sono dati passati dal TCI al PLC in risposta alla richiesta fatta dal PLC su ITCOD. ODATQ1D è un float in doppia precisione ODATQ1 è una doppia word ODATQ1L è la word leggera di ODATQ1. ODATQ1H è la word pesante di ODATQ1. In caso di esito errato, ODATQ1 è il codice di errore.
<b>ITCRANN</b> IRANN	<b>PLC → TCI</b>	<b>Annulla il comando richiesto dal PLC al TCI.</b> Il PLC, se desidera interrompere il comando di movimentazione da lui richiesto, deve verificare che ci sia nuovo comando TCI→PLC (NEWTCI) e che il comando su OTCQCOD sia uno dei seguenti: MOVCS, DEPUCS, PREUCS. Se queste condizioni sono verificate, il PLC mette ITCRANN=1 fintanto che NEWTCI=1.
<b>OTCCQIC</b> OCQIC	<b>TCI → PLC</b>	<b>Comando richiesto dal PLC In Corso.</b> Il TCI mette a 1 questo segnale quando riceve un comando da PLC a TCI (variazione di ITCSTTCI ) e lo rimette a 0 appena ha eseguito quanto richiesto.

I seguenti sinonimi si riferiscono ai segnali utilizzabili quando il Cambio Utensili è realizzato con la metodologia "**CUA da Subroutine**". Alcuni di questi segnali sono già stati visti più sopra e sono segnalati con **(V)**; qui vengono ripetuti per evidenziarne l'uso nei Sottoprogrammi. In generale, questi segnali non vanno gestiti dal PLC perché sono le routines M06, CUA e CUM che li testano o settano e resettano; sono stati messi comunque a disposizione anche del PLC nel caso possa essere utile. Per le routines, viene data la corretta sintassi di test, set e o reset se e quando applicabili. Ricordiamo che la Funzione Ausiliaria "**T**" è, di base, implementata come FASUB e realizzata tramite la subroutine TOOL. Anche la F.A. **M06** è, di base, definita come FASUB e realizzata tramite la subroutine M06 che non richiede, normalmente, modifiche. La M06 richiama a sua volta le subroutines CUM o CUA a seconda della situazione del Cambio Utensili e degli utensili da montare/smontare. Il progettista della specifica applicazione per scrivere le subroutines **CUM** e **CUA** può usare le tracce date in **CUM.SUB** e **CUA.SUB** fornite di base da ECS.



<b>TCCUTES</b> CUTES	<b>PLC → TCI</b>	<p><b>Comando CUTE da Subroutine</b> Permette di scegliere se il ciclo di C. Ut. Automatico deve essere eseguito da PLC o da Subroutine. La scelta deve essere effettuata dalla Subroutine CUA prima di emettere \$M06. Il TCI svilupperà la corretta sequenza di comandi/risposte verso il PLC o verso la subroutine in base allo stato di questo segnale. CUA deve settare/resettare questo segnale con: N1100 &lt;%V1202.16=1&gt; N1110 {Abilita CUTE da subroutine} N2100 &lt;%V1202.16=0&gt; N2110 {Disabilita CUTE da subroutine}</p>
<b>TCCUMANS</b> CUMANS	<b>PLC → TCI</b>	<p><b>Comando CUMAN da Subroutine</b> Permette di scegliere se il C. Ut. Manuale deve essere eseguito da PLC o da Subroutine. La scelta deve essere effettuata dalla Subroutine CUM prima di emettere \$M06. Il TCI svilupperà la corretta sequenza di comandi/risposte verso il PLC o verso la subroutine in base allo stato di questo segnale. CUM deve settare/resettare questo segnale con: N10 &lt;%V1202.16=1&gt; N11 {Abilita CUMAN da subroutine} N5 &lt;%V1202.16=0&gt;&lt;%V1202.17=0&gt;&lt;%V1202.18=0&gt; N6 {Disabilita CUMAN da subroutine }</p>
<b>OTCCUAT</b> CUAT	<b>TCI → PLC</b>	<p><b>Cambio Utensili Attivo (non finito).</b> Tale segnale è settato da TCI con \$M06 e resettato a fine cambio, cioè quando sul mandrino è montato l'utensile desiderato. Un Sottoprogramma può testare questo segnale con: N6600 &lt;IFF&lt;%V1202.14&gt;:N6800;N7100;N6800&gt; N6700 {FINE CAMBIO UTENSILE ? NO -&gt; N6800}</p>
<b>TM</b>	<b>TCI → PLC</b>	<p><b>Presenza Utensile sul Mandrino.</b> Settato da TCI con \$M06 se c'è un utensile sul mandrino, resettato alla fine della M06 stessa. Word valida solo durante la fase di c. utens. – OQCOD==CUTE Un Sottoprogramma può testare questo segnale con: N50&lt;IFF&lt;%V4157.0 &amp; '0001'&gt;:N60;N99;N60&gt; {N60 utensile su mandrino, N99 mandrino vuoto}</p>
<b>TIPTM</b>	<b>TCI → PLC</b>	<p><b>Tipo Utensile sul Mandrino.</b> Scritto da TCI quando setta TM. 0 = Utensile da Fresa; 1 = Utensile da Tornio Word valida solo durante la fase di c. utens. – OQCOD==CUTE Un Sottoprogramma può testare questo segnale con: N50&lt;IFF&lt;%V4157.0 &amp; '0020'&gt;:N60;N99;N60&gt; {N60 utensile da tornio, N99 utensile da fresa}</p>
<b>TRIC</b>	<b>TCI → PLC</b>	<p><b>Utensile già ricercato o presenza Ut. su posto intermedio.</b> Settato da TCI con \$M06 se utensile presente sul posto intermedio, resettato a fine M06 quando viene effettuato il deposito o alla prossima ricerca T se non c'è deposito da eseguire. Un Sottoprogramma può testare questo segnale con: N50&lt;IFF&lt;%V4157.0 &amp; '0002'&gt;:N60;N99;N60&gt; {N60 utensile già ricercato, N99 TI vuoto} Word valida solo durante la fase di c. utens. – OQCOD==CUTE</p>



## CAMBIO UTENSILI

<b>TIPTRIC</b>	<b>TCI → PLC</b>	<b>Tipo Utensile già ricercato o su posto intermedio.</b> Scritto da TCI quando setta TRIC. 0 = Utensile da Fresa; 1 = Utensile da Tornio Un Sottoprogramma può testare questo segnale con: N50<IFF<%V4157.0 & '0040':N60;N99;N60> {N60 utensile da tornio, N99 utensile da fresa} Word valida solo durante la fase di c. utens. – OQCOD==CUTE
<b>TMEM</b>	<b>TCI → PLC</b>	<b>Presenza utensile memorizzato ancora da ricercare.</b> Settato da TCI con nuova T se: 1- Cambio Utensile escluso. 2- PLC blocca ricerca TOOL 3- PLC non ha terminato fase di ricerca o c.ut. 4- L'utensile che si vuole montare è un utensile a terra 5- C.Ut. a torretta Un Sottoprogramma può testare questo segnale con: N50<IFF<%V4157.0 & '0004':N60;N99;N60> {vero->N60, falso->N99} Word valida solo durante la fase di c. utens. – OQCOD==CUTE
<b>TIPTMEM</b>	<b>TCI → PLC</b>	<b>Tipo Utensile memorizzato ancora da ricercare.</b> Scritto da TCI quando setta TMEM. 0 = Utensile da Fresa; 1 = Utensile da Tornio Un Sottoprogramma può testare questo segnale con: N50<IFF<%V4157.0 & '0080':N60;N99;N60> {N60 utensile da tornio, N99 utensile da fresa} Word valida solo durante la fase di c. utens. – OQCOD==CUTE
<b>OTCPNT</b> <b>OPNT</b> <b>(V)</b>	<b>TCI → PLC</b>	<b>Programmato Nuovo Tool.</b> Un Sottoprogramma può testare questo segnale con: N140 <IFF<%V1202.0>;N395;N395;N150> {T programmata? SI->N150; NO-> N395}
<b>OTCMCU</b> <b>OMCU</b> <b>(V)</b>	<b>TCI → PLC</b>	<b>Cambio Utensile Manuale.</b> Un Sottoprogramma può testare questo segnale con: N50 <IFF<%V1202.2>;N140;N60;N140> {CU Definito Manuale? Si-> N140}
<b>OTCRTP</b> <b>ORTP</b> <b>(V)</b>	<b>TCI → PLC</b>	<b>Ricerca Tool prenotata.</b> Un Sottoprogramma può testare questo segnale con: N70 <IFF<%V1202.7>;N70;N80;N70> {Ricerca tool prenotata ? SI -> N70 }
<b>TCSTRNCU</b> <b>STNCU</b>	<b>TCI → PLC</b>	<b>STROBE da TCI a Subroutine per comando CUTE.</b> Una Sub. può testare questo segnale su %V1203.0 Per l'uso vedere esempi (1) a fine tabella
<b>TCACKNCU</b> <b>AKNCU</b>	<b>TCI → PLC</b>	<b>ACK (Acknowledge) da Sub. al comando CUTE da TCI.</b> Una Sub. può settare/resettare questo segnale su %V1203.1 Per l'uso vedere esempi (1) a fine tabella
<b>TCSTRFIN</b> <b>STFIN</b>	<b>TCI → PLC</b>	<b>STROBE da TCI a Subroutine per comando FINE</b> Una Sub. può testare questo segnale su %V1203.2 Per l'uso vedere esempi (1) a fine tabella
<b>TCACKFIN</b> <b>ACKFIN</b>	<b>TCI → PLC</b>	<b>ACK da Subroutine al comando FINE da TCI</b> Una Sub. può settare/resettare questo segnale su %V1203.3 Per l'uso vedere esempi (1) a fine tabella



<b>TCSTRAGG</b> STAGG	<b>TCI → PLC</b>	<b>STROBE da Sub a TCI per Aggiornamento Parametri Tool</b> Una Sub. può settare/resettare questo segnale su %V1203.4 Per l'uso vedere esempi (1) a fine tabella
<b>TCACKAGG</b> AKAGG	<b>TCI → PLC</b>	<b>ACK da TCI a Sub. al comando Aggiorna Parametri</b> Una Sub. può testare questo segnale su %V1203.5 Per l'uso vedere esempi (1) a fine tabella

## 1) Esempi di uso degli Strobe/Ack tra TCI e PLC

```
N400<IFF<(%V1203.0^%V1203.1)|(%V1203.2^%V1203.3)|(%V1203.4^%V1203.5)>:N7700
;N600;N7700>
N500 {VERIFICA SE STROBE E ACK SONO FASATI}
{ ->N600 Strobe e Ack in fase }
{ ->N7700 Strobe e Ack sfasati }
N600 {Proseguì}
```

-----

```
N1500 <IFF<(%V1203.0 ^ %V1203.1)>:N1700;N1500;N1700>
N1600 {ATTESA COMANDO CUTE}
N1700 <IFF<%V1203.1>:N2100;N1900;N2100>
N1800 {RENDE ACK A COMANDO CUTE}
N1900 <%V1203.1= 1>
N2000 <JMP:N2200>
N2100 <%V1203.1= 0>
N2200 {Proseguì}
```

-----

```
N3700 <IFF<%V1203.4>:N4100;N3900;N4100>
N3800 {RICHIESTA AGGIORNAMENTO PARAMETRI TOOL}
N3900 <%V1203.4= 1>
N4000 <JMP:N4200>
N4100 <%V1203.4= 0>
N4200 <IFF<(%V1203.4 ^ %V1203.5)>:N4200;N4400;N4200>
N4300 {ATTESA ACK DA TCI PER AGGIORNAMENTO PARAMETRI TOOL}
N4400 $M06
N4500 {QUESTO M06 SERVE PER POTER FARE AGGIORNAMENTO}
N4600 {Proseguì}
```

-----

```
N5200 <IFF<(%V1203.2 ^ %V1203.3)>:N5400;N5200;N5400>
N5300 {ATTESA DA TCI COMANDO FINE FASE DI CUTE}
N5400 <IFF<%V1203.3>:N5800;N5600;N5800>
N5500 {ACK A COMANDO FINE FASE DI CUTE}
N5600 <%V1203.3= 1>
N5700 <JMP:N6100>
N5800 <%V1203.3= 0>
N5900 {FINE FASE DI CUTE}
N6000 {Proseguì}
```

## 2) Altri esempi di test

```
N80 <IFF<%V6116+%V6117>:N380;N90;N380>
{TCI in allarme? Si->N380; NO->N90}
{%V6116: Allarmi PLC}
```





## CAMBIO UTENSILI

```
{%V6117: Allarmi globali}
```

```
-----
```

```
N100 <IFF<%%V6146>:N140;N110;N110>
{Programmata "T" in automatico ? SI->N110}
{%V6146: Ultimo T letto in auto; -1 all'inizio e con RESET, M02}
N110 <IFF<%%V6153-%%V6146>:N120;N140;N120>
{Ultima "T" letta= "T" programmata in automatico ? SI -> N140}
{%V6153: Ultimo T emesso. All'inizio e con RESET, M02 è uguale}
{ all'utensile montato sul mandrino }
```

```
-----
```

Per gli utenti più esperti che vogliono approfondire l'argomento, diamo qui di seguito la corrispondenza tra gli indirizzi di alcuni segnali visti dal PLC (come definiti in DF\_SYN.INC) e le variabili V viste dalle subroutines. Per altri segnali, vedi VARCNC.DEF .

DF_SYN.INC	Variabile	Note
vari	%V1202.n	Stato del TCI: vedi VARCNC.DEF
OTCPNT	%V1202.0	Programmato nuovo tool
OTCOUT	%V1202.1	Il Cambio Utensili è automatico
OTCMCU	%V1202.2	Il Cambio Utensili è manuale
OTCBMM	%V1202.3	Blocco movimentazione magazzino
OTCRFC	%V1202.4	Ricerca fisica in corso
OTCCUC	%V1202.5	Cambio Utensile in corso
OTCCUAT	%V1202.14	Cambio Utensile attivo (non finito)
TCCUTES	%V1202.16	Abilita CUTE da Subroutine
TCCUMANS	%V1202.17	Abilita CUMAN da Subroutine
TCSTRNCU	%V1203.0	Strobe da TCI a Subroutine per comando CUTE
TCACKNCU	%V1203.1	Ack da Subroutine a TCI per comando CUTE
TCSTRFIN	%V1203.2	Strobe da TCI a Subroutine. per comando FINE
TCACKFIN	%V1203.3	Ack da Subroutine a TCI per comando FINE
TCSTRAGG	%V1203.4	Strobe da TCI a Subroutine per Aggiornamento parametri
TCACKAGG	%V1203.5	Ack da Subroutine a TCI per comando Aggiorna parametri
OTCDATI	%%V4157	Valore del parametro generico
OTCDATIL	%V4157.0	Può essere un numero long od un intero
OTCDATIH	%V4157.1	Se è un numero intero, il dato è in %V4157.1 (OTCDATIH) %V4157.0 è usato a bits
OLIP1	%V4166	Word usata per colloquio tra programmi LIP e PLC Vedi (3) per un esempio d'uso nella Sub. CUA
OLIP2	%V4167	Word usata per colloquio tra programmi LIP e PLC
ILIP1	%V4178	Word usata per colloquio tra PLC e programmi LIP
ILIP2	%V4179	Word usata per colloquio tra PLC e programmi LIP
ILIP3	%V4180	Word usata per colloquio tra PLC e programmi LIP
ILIP4	%V4181	Word usata per colloquio tra PLC e programmi LIP
ILIP5	%V4182	Word usata per colloquio tra PLC e programmi LIP

### 3) Esempi di sincronizzazioni tra subroutines e PLC

#### 3a) con uso di OLIPx e ILIPx

```
. . .
N30 <%V4166=0>
{ reset comandi verso PLC su OLIP1 }
. . .
N250 <%V4166=1>
{ FASE 1 Apertura pinza: OLIP1=1 }
N260 <IFF<%V4179 & '0002':N260;N270;N260>
{ attesa pinza aperta: ILIP201 }
```



---

```
N270 <%V4166=2>
{ FASE 2 avvicinamento magazzino : OLIP1=2}
N280 <IFF<%V4179 & '0200':N285;N280;N285>
{ attesa magazzino posizionato: ILIP209 }
. . .
N310 <%V4166=3>
{ FASE 3 Chiusura pinza: OLIP1=3}
N320 <IFF<%V4179 & '0002':N330;N320;N330>
{ attesa pinza chiusa: ILIP201 }
N330 <%V4166=4>
{ FASE 4 sbloccaggio utensile: OLIP1=4}
N340 <IFF<%V4179 & '0008':N355;N340;N355>
{ attesa ut. sbloccato: ILIP203 }
. . .
```

**3a) con uso di Funzioni Ausiliarie Mxx e ILIPx**

```
. . .
N2150 M54
{ RIBALTA PORTAUTENSILE }
N2400 <IFF<%V4157.0 & '0001':N2550;N4250;N2550>
{ UTENSILE SU MANDRINO ? SI -> N2550 }
N2550 G53 Z<QCU>
{ QUOTA PER DEPOSITO TOOL }
N2700 <IFF<%V4178 & '0400':N2700;N2700;N2750>
{ PORTAUTENSILI RIBALTATO ? SI -> N2750 }
N2750 <WAT:MND>
{ Verifica M19 eseguito }
N2760 <WAT:FRE;CNC>
N2800 U<#381>
{ BRACCIO IN POSIZIONE DI AGGANCIAMENTO }
N2850 <WAT:FRE>
N2900 M50
{ SBLOCCA UTENSILE SU MANDRINO }
N3050 <IFF<%V4178 & '0002':N3050;N3050;N3150>
{ UTENSILE SBLOCCATO ? SI -> N3150 }
N3150 U<#382>
{ RUOTA BRACCIO DI 180 GRADI }
N3250 <WAT:FRE>
. . .
```

**3b) con uso di Funzioni SETxx e ILIPx**

```
. . .
N270 <%%#50=%%V6102>
{ Legge utensile su mandrino}
N280 <IFF<%%#50>:N290;N390;N290>
{ Test se utensile 0 su mandrino}
N290 <IFF<%%#50-%%#53>:N320;N320;N1570>
{ Test numero utensile > Max.ut.}
N320 <IFF<%%#50-%%#53/2>:N330;N330;N350>
{ Scelta magazzino 0 o 180 gradi}
N330 <SET:1300>
{ Comanda magazzino a 0 gradi}
N340 <JMP:N370>
N350 <SET:1301>
{ Comanda magazzino a 180 gradi}
N360 <JMP:N380>
N370 <IFF<%V4178&'0100':N370;N370;N390>
{ Test per Magazzino a 0}
```

---



```
N380 <IFF<%V4178&'0200'>:N380;N380;N390>  
{ Test per Magazzino a 180}  
N390  
. . .
```

## 9.8 DIALOGO TCI ↔ PLC

Per l'esecuzione delle varie sequenze relative alla ricerca utensili, prelievo, deposito e cambio utensile, è previsto un dialogo tra il Tool Controller TCI e PLC. Il colloquio avviene attraverso dei codici comando inviati dal TCI al PLC e viceversa; la sincronizzazione è realizzata tramite Strobe/Acknowledge utilizzando i segnali **NEWTCI/FATTCI** oppure **ITCSTTCI /OTCAKTCI** visti più sopra.

### 9.8.1 Comandi inviati da TCI a PLC.

Tali comandi sono inviati dal TCI (per sua iniziativa o per richiesta del CNC o, infine, per richiesta del PLC fatta come indicato al successivo paragrafo 9.7.2) su **OTCQCOD** dove è presente il codice del comando mentre su OTCDATI o OTCDATIH e/o OTCDATIL sono presenti gli eventuali dati relativi al comando. I comandi vanno interpretati ed eseguiti quando **NEWTCI=1** e, al termine dell'esecuzione, il PLC deve mettere **FATTCI=1** per permettere al TCI di passare alla fase successiva. Più oltre saranno dati esempi d'uso di questi comandi.

Per i comandi si consiglia di usare il nome anziché il codice per una miglior lettura della logica PLC.

Nome del Comando	Codice su OTCQCOD (unsigned char)	Descrizione	Dati su OTCDATI (long) OTCDATIH (short)
<b>COMANDI PER CAMBIO UTENSILI SUL MANDRINO</b>			
<b>MOVCU</b>	1	ricerca di un posto utensile in magazzino	POSTO desiderato [OTCDATIH]
<b>CUTE</b>	2	cambia utensile	Ultimo POSTO [OTCDATIH]
<b>PREUTE (*)</b>	3	preleva utensile	Ultimo POSTO [OTCDATIH]
<b>DEPUTE (*)</b>	4	deposita utensile	POSTO di deposito [OTCDATIH]
<b>CUMAN</b>	20	cambio utensile manuale per: - c.ut. autom. escluso (G309) - cambio utensile a terra	codice utensile "T" desiderato [OTCDATI]
<b>COMANDI PER CARICO/SCARICO UTENSILE DAL MAGAZZINO</b>			
<b>MOVCS (**)</b>	5	posizionare un posto del magazzino in posizione di carico/scarico utensile	POSTO desiderato [OTCDATIH]
<b>PREUCS (**)</b>	12	preleva utensile da magazzino per scaricarlo o metterlo nel posto intermedio di Carico/Scarico (***)	Ultimo POSTO [OTCDATIH]
<b>DEPUCS (**)</b>	13	deposita in magazzino un utensile da caricare o già caricato nel posto intermedio di Carico/Scarico (***)	Ultimo POSTO [OTCDATIH]



COMANDI PER MAGAZZINO UTENSILI			
INIMAG	19	Inizializzazione magazzino	

Note:

(\*) = Valido solo se nel file TYPUCU.TAB il campo UTI= 1 o 2 (esiste posto intermedio)

(\*\*) = Valido solo se nel file CUMAG.TAB il campo MID= 0 (carico/scarico da PLC)

(\*\*\*) = Valido solo se nel file CUMAG.TAB il campo PCS= 1 (esiste posto intermedio c/s)

### 9.8.2 Comandi inviati da PLC a TCI.

Tali comandi sono inviati dal PLC su **ITCOD** (char), dove deve essere scritto il codice del comando, mentre su **IDTC** (long) o **IDTCH** (short) e/o **IDTCL** (short) e/o **IDTC1D** (double) e/o **IDTC1**(long), debbono essere presenti gli eventuali dati relativi al comando. Il TCI legge questi comandi e dati quando il PLC pone **ITCSTTCI** ≠ **OTCAKTCI**. Il TCI comunicherà al PLC l'accettazione e/o l'avvenuta esecuzione del comando, riportando **OTCAKTCI=ITCSTTCI** e scrivendo su **OTCESCOM** l'esito del comando:

**OTCESCOM= 1**: esito OK

**OTCESCOM=-1**: ERRORE

Eventuali dati richiesti saranno disponibili dal momento della comunicazione **OTCAKTCI=ITCSTTCI** con **OTCESCOM= 1**. In caso di errore, su **OTCDATQ** è presente il codice di allarme il cui significato può essere visto tramite i tasti funzione [Tools] e poi [Allarmi].

Alle richieste fatte dal PLC, il Tool Controller risponderà inviando, se necessario, gli opportuni comandi **TCI → PLC** visti nella tabella del precedente paragrafo 9.7.1 . Più oltre saranno dati esempi d'uso di questi comandi.

Per i comandi si consiglia di usare il nome anziché il codice per una miglior lettura della logica PLC.

Nome del Comando e Costanti per PLC	Codice su ITCOD (char)	Descrizione	Dati inviati su IDTC (long) IDTCL (short) IDTCH (short) IDTC1D (double) IDTC1 (long )	Dati di risposta su OTCDATQ (long) OTCDATQL (short) OTCDATQH (short) OTCDATQD(double) OTCDATQ (long)
<b>COMANDI PER CARICO/SCARICO UTENSILE DAL MAGAZZINO</b>				
<b>MOVCS</b> (*)	5	richiesta di posizionare un posto del magazzino in posizione di Carico/Scarico	POSTO desiderato [IDTCH]	vedi note (1) e (ALR)
<b>PREUCS</b> (*)(**)	12	richiesta di prelevare utensile da magazzino per metterlo nel posto intermedio di Car./Scar.	POSTO [IDTCH] (****)	vedi note (2) e (ALR)
<b>DEPUCS</b> (*)(**)	13	richiesta di depositare in magazzino un utensile dal posto intermedio di Car./Scar.	POSTO [IDTCH] (****)	(ALR)
<b>CARUCS</b> (*)	14	richiesta di caricare un utensile in magazzino o nel posto intermedio di Car./Scar.	Codice fisico dell'utensile (TPC) [IDTC]	(ALR)
<b>SCAUCS</b> (*)	15	richiesta di scaricare un utensile dal magazzino o dal posto intermedio di Car./Scar.	Codice fisico dell'utensile (TPC) [IDTC] (****)	vedi note (5) e (ALR)



## CAMBIO UTENSILI

COMANDI PER RICHIESTA PARAMETRI DEL MAGAZZINO				
<b>CERPOV</b> (***)	11	richiesta ricerca di posto vuoto nel magazzino	Codice dimensione [IDTCH] 0=piccolo 1=medio 2=grande 3=extra	POSTO in OTCDATQH se trovato. Altrimenti: (ALR)
<b>CERTPC</b>	16	richiesta del codice fisico TPC di un utensile in base al posto in magazzino	POSTO di magazzino [IDTCH]	TPC in OTCDATQ se trovato. Altrimenti: (ALR)
<b>CERPOS</b>	17	richiesta del posto di magazzino di un utensile in base al codice fisico TPC	codice fisico TPC [IDTC]	POSTO in OTCDATQH dimens. in OTCDATQL se trovato. Altrimenti: (ALR)
<b>CERDEA</b>	18	richiesta ricerca in tabella del primo utensile morto per vita e/o usura	codice su IDTCH: 0=indifferente 1=morti per vita 2=morti per usura	POSTO in OTCDATQH indice tabella utensili in OTCDATQL se trovato. (ALR)
<b>RIPUTM</b>	25	richiesta di ripristino dell'utensile morto per vita e/o usura	indice tabella utensili ( ad es. è quello ricevuto in OTCDATQH con il com. CERDEA)	(ALR)
COMANDI PER PARAMETRI CUSTOM				
<b>WRCUST</b>	26	richiesta di scrittura dei parametri "custom" (non predefiniti da ECS e liberi per il progettista del PLC)	IDTCH=record IDTCL=parametro IDTC1D=valore IDTC1=valore	(ALR) vedi descrizione a fine tabella
<b>RDCUST</b>	27	richiesta di lettura dei parametri "custom" (non predefiniti da ECS e liberi per il progettista del PLC)	IDTCH=record IDTCL=parametro OTCDATQD=val. OTCDATQ=valore	(ALR) vedi descrizione a fine tabella
COMANDI RELATIVI AL MAGAZZINO				
<b>INIMAG</b>	19	richiesta di inizializzazione magazzino		

## Note:

(ALR) : Codice di allarme in OTCDATQL se OTCESCOM= -1; il significato dell'allarme può essere visto tramite i tasti funzione [Tools] e poi [Allarmi] del menu orizzontale del CNC oppure ricercato in ALLAR.INC .

(\*) : Valido solo se nel file CUMAG.TAB il campo MID= 0 (carico/scarico da PLC)

(\*\*) : Valido solo se nel file CUMAG.TAB il campo PCS= 1 (esiste posto intermedio c/s)

(\*\*\*) : Valido solo se nel file CUMAG.TAB il campo TGM= 1 o 2 (gestione random)

(\*\*\*\*) : Da definire se si vuole il controllo che sia lo stesso del comando MOVCS ultimo.

(1) : Il TCI risponde inviando il comando MOVCS con il numero di posto su OTCDATI.H.

(2) : Il TCI risponde inviando il comando PREUCS con il numero di posto su OTCDATI.H.

(5) : Il TCI non invia comandi ma aggiorna la situazione magazzino (posto di TPC= 0).

### 9.8.2.1 WRCUST e RDCUST: scrittura e lettura dei parametri Custom.

La tabella utensili TT.BIN (vedi par. 9.4) contiene, per ogni tagliente, quattro campi per dati non definiti da ECS e quindi liberamente usabili dal progettista del PLC per le sue necessità:



---

```
/*----- Dati disponibili per costruttore -----*/  
unsigned long   l_cost1; /* Disponibile per appoggi vari */  
unsigned long   l_cost2; /* Disponibile per appoggi vari */  
double float    d_cost1; /* Disponibile per appoggi vari */  
double float    d_cost2; /* Disponibile per appoggi vari */
```

Questi dati sono tipicamente importanti in fase di ricerca dell'utensile o quando l'utensile stesso è attivo (montato sul mandrino o in posizione di lavoro); con i comandi RDCUST e WRCUST è quindi facile leggere/scrivere i dati usando l'indice di tabella ricavabile con una `read_var()` descritta al capitolo 6 :

```
read_var (VL_INTG,6101,-1,-1, & INDEXATT) ;  
/* legge l'indice dell'utensile attivo */  
read_var (VL_INTG,6105,-1,-1, & INDEXRIC) ;  
/* legge l'indice dell'utensile ricercato */
```

Sono da osservare le seguenti regole:

#### **WRCUST** (codice 26 su ITCOD)

IDTCH= record della tabella utensili su cui scrivere (1 . . . 1024)

IDTCL= numero del parametro (1 . . . 4)

1= d\_cost1 (double)

2= d\_cost2 (double)

3= l\_cost1 (long)

4= l\_cost2 (long)

IDTC1D= valore (double) del parametro da scrivere (se IDTCL= 1 o 2)

IDTC1= valore (long) del parametro da scrivere (se IDTCL= 3 o 4)

#### **RDCUST** (codice 27 su ITCOD)

IDTCH= record della tabella utensili da leggere (1 . . . 1024)

IDTCL= numero del parametro (1 . . . 4)

1= d\_cost1 (double)

2= d\_cost2 (double)

3= l\_cost1 (long)

4= l\_cost2 (long)

OTCDATQD= valore (double) del parametro letto (se IDTCL= 1 o 2)

OTCDATQ= valore (long) del parametro letto (se IDTCL= 3 o 4)

NOTA:

Nelle routines scritte in linguaggio LIP, il numero di parametro custom [IDTCL] dei comandi RDCUST e WRCUST corrisponde a:

1= d\_cost1 (double) PGA

2= d\_cost2 (double) PGB

3= l\_cost1 (long) %%PGC

4= l\_cost2 (long) %%PGD



## 9.9 PROCEDURA CARICO/SCARICO UTENSILI DA MAGAZZINO

Il TCI prevede due diversi modi di effettuare il carico e scarico utensili dal magazzino; la scelta viene fatta in base al valore scritto nel campo MID nel file CUMAG.TAB :

- MID=1: PROCEDURA MANUALE (attraverso tastiera del CNC)
- MID=0: PROCEDURA AUTOMATICA (attraverso comandi da PLC)

### 9.9.1 PROCEDURA MANUALE

In questo caso gli utensili, per il TCI, risultano presenti o assenti in magazzino dal momento stesso del loro inserimento, o della loro cancellazione, nella TABELLA UTENSILI tramite comandi (tasti funzione e finestre di introduzione dati) impartiti dall'operatore sulla tastiera del CNC.

La responsabilità della corrispondenza tra presenza fisica o meno di un certo utensile in un dato posto di magazzino è quindi esclusivamente dell'operatore.

### 9.9.2 PROCEDURA AUTOMATICA

Questa procedura permette di caricare e scaricare utensili dal magazzino in modo automatico, tramite una sequenza di comandi impartiti dal PLC al TCI. In questo caso è possibile cancellare un utensile dalla tabella utensili, con comandi da tastiera, solo se precedentemente tale utensile è stato scaricato dal magazzino; inoltre un utensile può essere caricato in magazzino solo se precedentemente è stato inserito nella tabella utensili, tramite comandi da tastiera.

In questo caso, la cura della dislocazione fisica degli utensili in magazzino è a carico del TCI. Per il carico/scarico occorre esistano e siano definiti:

- la posizione del magazzino per eseguire un carico/scarico.
- la presenza o meno di un posto intermedio per l'utensile in fase di carico/scarico; campo PCS=1 nel file CUMAG.TAB .
- un posto fisico di carico/scarico esterno al magazzino.

La procedura di carico/scarico è molto utile nei magazzini con gestione di tipo RANDOM e RANDOM FISSO, per aiutare l'operatore ad inserire gli utensili in magazzino una volta definiti in Tabella Utensili con i dati geometrici, il codice fisico TPC e la dimensione.

Se il magazzino è a posti dedicati, occorre definire in Tabella Utensili anche il posto.

Per realizzare la procedura automatica di carico/scarico, il costruttore della Macchina Utensile deve montare una serie di pulsanti e/o commutatori vicino al magazzino per realizzare i comandi necessari.

Qui di seguito riportiamo un esempio di carico/scarico, per vedere quali comandi da PLC a TCI debbono essere usati e quali sono le "reazioni" del Tool Controller. Il magazzino in questione ha una gestione di tipo RANDOM, senza o con posizione per un utensile intermedio in carico/scarico (TGM=1, PCS=0 o 1 e MID=0 nel file CUMAG.TAB).

Prima di iniziare la procedura, l'operatore deve inserire i dati necessari in tabella utensili.

Tale manovra può essere realizzata tramite i comandi manuali di Inserimento Utensili con gli appositi tasti funzione del Menu orizzontale (e relative finestre di introduzione dati); per i dettagli si rimanda al manuale NORME D'USO.

---

---



In alternativa, può essere usato il caricamento utensili da file come spiegato nel capitolo relativo agli utensili delle NORME DI PROGRAMMAZIONE.

In ogni caso è indispensabile definire TPC e Dimensione.

### 9.9.2.1 Carico utensile

Per il **carico utensile**, l'operatore deve conoscere e introdurre il codice dimensione ed il codice fisico dell'utensile da montare, che deve essere già stato definito in tabella utensili (senza un posto di magazzino assegnato). Nella tabella seguente, la nota (**→n**) indica di passare direttamente alla fase **n**, in caso di sequenze alternative dovute alle varie situazioni di macchina. Su fondo grigio sono date le operazioni a carico del PLC.

La sequenza deve essere del tipo:

Fase	Com. PLC→TCI o azione PLC	Dati	Com. TCI→PLC o risposta TCI	Dati
1	<b>CERPOV</b>	Codice dimensione in IDTCH		
2			<b>OTCESCOM=1</b>	Posto trovato in OTCDATQH
3	<b>MOVCS</b>	Posto trovato in IDTCH		
4			<b>MOVCS</b>	con Posto in OTCDATIH.
5	Il PLC posiziona il Posto magazzino in posizione di carico/scarico			
6	L'operatore pone l'utensile nella postazione di carico/scarico ( <b>→7</b> ) o direttamente nel magazzino ( <b>→8</b> ) e conferma di aver eseguito l'operazione.			
7	Il PLC preleva l'utensile dalla postazione di carico/scarico e lo pone nel posto intermedio, se questo esiste, o direttamente nel magazzino.			
8	<b>CARUCS</b>	Codice fisico <b>TPC</b> in IDTC	NON c'è posto intermedio ( <b>→11</b> )	
9			Il TCI aggiorna la situazione magazzino (TPC in posto intermedio di carico/scarico).	
10 (*)	<b>DEPUCS</b>	Posto trovato in IDTCH		
11			<b>DEPUCS (**)</b>	con Posto in OTCDATIH.
12	Il PLC preleva l'utensile dal posto intermedio di carico/scarico, se questo esiste e lo pone nel magazzino.			
13			Il TCI aggiorna la situazione magazzino (TPC in magazzino nel posto definito).	

#### NOTE

(\*) Queste fasi non sono previste se non esiste il posto intermedio di carico/scarico.





- (\*\*) Se manca il segnale **ITCMPOS** (Mag. in posizione) TCI ridà prima un comando **MOVCS**.
- Come già detto, il PLC segnala di aver completato un comando dando il **FATTCI** in risposta a **NEWTCI**; a questo punto il TCI può passare alla fase successiva.
  - Allo stesso modo, il TCI segnala al PLC di aver accettato e/o eseguito il comando ponendo **OTCAKTCI= ITCSTTCI** ; a questo punto il PLC può passare alla fase successiva.

### 9.9.2.2 Scarico utensile

Per lo **scarico utensile**, l'operatore deve conoscere e introdurre il codice fisico dell'utensile da smontare, che deve essere presente in tabella utensili (in un posto di magazzino).

La sequenza deve essere del tipo:

Fase	Com. PLC→TCI o azione PLC	Dati	Com. TCI→PLC o risposta TCI	Dati
1	<b>CERPOS</b>	Codice fisico <b>TPC</b> in IDTC		
2			<b>OTCESCOM=1</b>	Posto trovato in OTCDATQH
3	<b>MOVCS</b>	Posto trovato in IDTCH		
4			<b>MOVCS</b>	con Posto in OTCDATIH.
5	Il PLC posiziona il Posto magazzino in posizione di carico/scarico. NON c'è posto intermedio (→n)			
6 (*)	<b>PREUCS</b>	Posto trovato in IDTCH		
7 (*)			<b>PREUCS</b>	con Posto in OTCDATIH.
8 (*)	Il PLC preleva l'utensile da magazzino e lo pone nel posto intermedio di carico/scarico.			
9 (*)			Il TCI aggiorna la situazione magazzino (TPC in posto intermedio di carico/scarico).	
10	L'operatore toglie l'utensile e conferma di aver eseguito l'operazione.			
11	<b>SCAUCS</b>	Codice fisico <b>TPC</b> in IDTC		
12			Il TCI aggiorna la situazione magazzino (posto di TPC= 0). L'utensile da questo momento è ancora presente in Tabella Utensili, ma non più in magazzino in quanto è stato reso NON DISPONIBILE.	

### NOTE

- (\*) Queste fasi non sono previste se non esiste il posto intermedio di carico/scarico.
- La movimentazione del magazzino in carico/scarico è possibile anche quando la M.U. sta lavorando purché non vi sia in atto una fase di ricerca Txx e/o un cambio utensile M06.
  - Il carico/scarico non è permesso se è stata eseguita una ricerca e non è stato ancora effettuato il cambio utensile. Per segnalare al PLC questa situazione, TCI emette il



segnale **OTCBMM** (Blocco Movimentazione Magazzino) che, se = 0, indica che è possibile movimentare il magazzino; quando è = 1, se viene dato uno dei comandi di movimento sopra riportati, il TCI emette verso il PLC l'allarme 31810.

- Al PLC viene data la possibilità di annullare un comando fra quelli sopra riportati, tramite il segnale **ITCRANN** (Annulla il comando richiesto dal PLC al TCI).
- Il PLC, se desidera interrompere il comando di movimentazione da lui richiesto, deve verificare che ci sia nuovo comando TCI⇒PLC in corso (NEWTCI) e che il comando su **OTCQCOD** sia uno dei seguenti: MOVCS, DEPUCS, PREUCS. Se queste condizioni sono verificate, il PLC mette ITCRANN=1 fino a quando il TCI non pone NEWTCI=0 per segnalare di aver accettato ed eseguito l'annullamento. Il PLC deve ovviamente arrestare l'azione in corso e rimettere la macchina in situazione di riposo.

### 9.10 COMPORTAMENTO TCI IN FASE DI RESET

Il PLC, durante la fase di RESET (ORST=1), nel caso sia in corso un comando da TCI (NEWTCI=1), può decidere se annullare o terminare il comando stesso tramite il segnale **NORESET**, già visto al Capitolo 1 al quale si rimanda per una più completa trattazione.

<b>NORESET</b>	<b>PLC → CNC</b>	Blocca la prosecuzione della fase di <b>RESET</b> (MAX 15 sec) dalla premuta del tasto RESET. Esso è da utilizzare quando è necessario comunque terminare una certa attività indipendentemente dal RESET. Se si supera il tempo massimo, il CNC dà allarme.
----------------	------------------	--

Se, ad esempio, è in corso un comando CUTE (cambio utensile), settando **NORESET** è possibile completare il cambio, restituire FATTCI e poi ritogliere **NORESET**; così facendo il comando viene considerato eseguito ed il nuovo Tool risulta montato sul mandrino.

In caso contrario, il comando in corso viene considerato annullato e la situazione utensili rimane quella precedente.

Attenzione: il tempo massimo tra l'arrivo di ORST e la rimessa a zero di **NORESET** non deve superare 15 secondi, altrimenti scatta l'allarme di timeout 31407.



### 9.11 VARI TIPI DI CAMBI UTENSILI AUTOMATICI

In questo paragrafo si intende spiegare il comportamento del TCI nei vari tipi di CUA. Si è pensato di utilizzare delle tabelle che rendono più comprensibile la sequenza.

Il progettista del PLC deve normalmente predisporre uno *switch* (vedi esempio più oltre) che intercetti i comandi **TCI→PLC** ed esegua le fasi della macchina che realizzano il comando. Infatti il PLC deve comportarsi da SLAVE (non prende iniziative) e deve esclusivamente stare in attesa dei comandi che arrivano dal TCI, eseguirli e rendere il FATTCI al comando senza dover ricordare niente dell'utensile.

Occorre tener presente che il TCI ricorda sempre (anche dopo un OFF/ON CNC) l'ultimo comando al quale ha ricevuto il "fatto" e che esegue le sue scelte basandosi su questo.

Solo nel caso che il PLC preveda di poter muovere il magazzino utensili in modo indipendente (senza richieste del TCI) occorre prendere qualche precauzione. In questo caso torna utile il segnale **ITCMPOS** che, messo a zero, segnala la necessità di riposizionare il magazzino stesso; il TCI provvederà a rimettere un comando MOVCU o MOVCS prima di passare ad una fase successiva che richieda il magazzino in posizione.



### 9.11.1 C.Ut. a TORRETTA

#### 9.11.1.1 TORRETTA(TYP='T') - Posti dedicati(TGM=0) - SENZA Ut. Intermedi (UTI=0)

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTCI	FATTOM	TOOL Mandrino	TOOL Intermed.	TOOL Ricercato
T20	1	CNC	Emette T20				T0		
	2	PLC		X					
M06	1	CNC	Emette M06			X			
	2	TCI	Emette MOVCU posizione 20						
	3	PLC	Posiziona la torretta sul posto 20; ITCMPOS=1		X				
	4	TCI	Emette CUTE posizione 20						
	5	PLC	Preleva l'utensile (se necessario) Memorizza T20 sul Mandrino		X				
	6	TCI	Aggiorna correttori				T20		
T30	1	CNC	Emette T30						
	2	PLC		X					
M06	1	CNC	Emette M06			X			
	2	TCI	Se ITCMPOS=1: →4 Se ITCMPOS=0: emette MOVCU pos. 20						
	3	PLC	Posiziona la torretta sul posto 20; ITCMPOS=1		X				
	4	TCI	Emette CUTE posizione 20 (T attuale)						
	5	PLC	Reinserisce T20 se estratto Resetta T20 sul Mandrino		X				
	6	TCI	Aggiorna correttori				T0		
	7	TCI	Emette MOVCU posizione 30						
	8	PLC	Posiziona la torretta sul posto 30; ITCMPOS=1		X				
	9	TCI	Emette CUTE posizione 30						
	10	PLC	Preleva l'utensile (se necessario) Memorizza T30 sul Mandrino		X				
	11	TCI	Aggiorna correttori				T30		
T0	1	CNC	Emette T0						
	2	PLC		X					
M06	1	CNC	Emette M06			X			
	2	TCI	Se ITCMPOS=1: →4 Se ITCMPOS=0: emette MOVCU pos. 30						
	3	PLC	Posiziona la torretta sul posto 30; ITCMPOS=1		X				
	4	TCI	Emette CUTE posizione 30 (T attuale)						
	5	PLC	Reinserisce T30 se estratto Resetta T30 sul Mandrino		X				
	6	TCI	Aggiorna correttori				T0		

#### Esempio PLC di Cambio utensile a torretta

```

/*****
COLLOQUIO TCI->PLC PER: MOVCU CUTE INIMAG
Comandi del TCI con cambio utensile a torretta con M06
1)CUTE con OTCDATIH= "T" utensile vecchio da togliere. PLC risponde FATTCI
2)MOVCU con OTCDATIH= "T" utensile nuovo. PLC risponde con FATTCI
3)CUTE con OTCDATIH= "T" utensile nuovo. PLC risponde con FATTCI. TCI
aggiorna i correttori
*****/

```



## CAMBIO UTENSILI

---

```
if(pon || ORST)
{
  SQZTCI= 0 ; /* Inizializzazione Sequenziatore */
}
ITCMPOS= ITCMPOS && (POSCAT== PCERC) && !UTDS70 && !UTSS77 && !SBUTS79;
/* Automantenimento flag MAGAZZINO IN POSIZIONE per TCI */
switch(SQZTCI) /* SEQUENZIATORE DI COLLOQUIO TCI->PLC */
{
  case 00:
    FATTCI= 0 ; /* Toglie Fatto a TCI */
    if(NEWTCI && (OTCQCOD== MOVCU))
    {
      SQZTCI= 20 ; /* COMANDO MOVCU esegui movimento magazzino */
      PCERC= OTCDATIH ; /* POSTO DA CERCARE */
    }
    if(NEWTCI && (OTCQCOD== CUTE))
    {
      SQZTCI= 100 ; /* COMANDO CUTE esegui cambio utensile */
      PCERC= OTCDATIH ; /* POSTO DA CERCARE */
    }
    if(NEWTCI && (OTCQCOD== INIMAG))
    {
      SQZTCI= 200 ; /* COMANDO INIMAG */
    }
    break;
  /*******/
  case 20: /* Esegue comando MOVCU */
    ITCMPOS=0 ; /* Reset Magazzino in Posizione */
    if( SQZMAG== 10 ) /* Attesa sequenza magazzino partita */
    {
      SQZTCI= 40; /* prossima fase */
    }
    break;
  /*******/
  case 40: /* Attesa utensile trovato */
    if((SQZMAG== 200)) /* Test sequenza magazzino */
    {
      ITCMPOS= (PCERC== POSCAT); /* Magazzino in posizione */
      SQZTCI= 200; /* Fatto per TCI */
    }
    break;
  /*******/
  case 100: /* ESEGUE COMANDO CUTE */
    . . .
    break;
  /*******/
  case 200: /* Dà "FATTO" a TCI */
    FATTCI= 1; /* Fatto per TCI */
    if(!NEWTCI)
    {
      SQZTCI= 0; /* ritorno ad attesa comandi */
    }
    break;
}
}
```

---

## 9.11.2 C.Ut. a CATENA ACICLICO

### 9.11.2.1 ACICLICO (TYP=' ') - RANDOM (TGM=1) - SENZA Ut. Intermedio (UTI=0)

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTOM	FATTCI	TOOL Mandrino	TOOL Intermed.	TOOL Ricercato
T10	1	CNC	Emette T10 (utensile <b>PICCOLO</b> )	X			T0		
	2	TCI	Emette MOVCU posizione T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
T20	1	CNC	Emette T20 (utensile <b>PICCOLO</b> )	X					
	2	TCI	Emette MOVCU posizione T20						
	3	PLC	ITCMPOS=0 (fare sempre prima di mov. cat.)						
	4	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			T20
M06	1	CNC	Emette M06		X				
	2	TCI	Se ITCMPOS=1: →4 Se ITCMPOS=0: riemette MOVCU pos. T20						
	3	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			T20
	4	TCI	Emette CUTE posizione T20						
	5	PLC	Carica l'utensile T20			X			
	6	TCI	Aggiorna correttori				T20		
T10	1	CNC	Emette T10 (utensile <b>PICCOLO</b> )	X					
	2	TCI	Emette MOVCU posizione T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			
M06	1	CNC	Emette M06		X				
	2	TCI	Se ITCMPOS=1: →4 Se ITCMPOS=0: riemette MOVCU pos. T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
	4	TCI	Emette CUTE posizione T10						
	5	PLC	Cambia l'utensile (dep. T20 nel posto di T10)			X			
	6	TCI	Aggiorna correttori				T10		
T40	1	CNC	Emette T40 (utensile <b>MEDIO</b> )	X					
	2	TCI	Emette MOVCU posto ottimizzato per T10						
	3	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
M06	1	CNC	Emette M06		X				
	2	TCI	Se ITCMPOS=1: →4 Se ITCMPOS=0: riemette MOVCU pos. per T10						
	3	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
	4	TCI	Emette CUTE posizione per T10 (attuale)						
	5	PLC	Smonta l'utensile (T10 in posto ottimizzato)			X			
	6	TCI	Aggiorna correttori				T0		
	7	TCI	Emette MOVCU posizione T40						T40
	8	PLC	Posiziona la catena sul posto T40; ITCMPOS=1			X			
	9	TCI	Emette CUTE posizione T40						
	10	PLC	Carica l'utensile T40			X			
	11	TCI	Aggiorna correttori				T40		
T60	1	CNC	Emette T60 (utensile <b>MEDIO</b> )	X					
	2	TCI	Emette MOVCU posto T60						
	3	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
M06	1	CNC	Emette M06		X				
	2	TCI	Se ITCMPOS=1: →4 Se ITCMPOS=0: riemette MOVCU pos. T60						



CAMBIO UTENSILI

	3	PLC	Posiziona la catena sul posto T60; ITCMPOS=1			X			T60
	4	TCI	Emette CUTE posizione T60						
	5	PLC	Cambia l'utensile (dep. T40 nel posto di T60)			X			
	6	TCI	Aggiorna correttori				T60		
T10	1	CNC	Emette T10 (utensile <b>PICCOLO</b> )	X					
	2	TCI	Emette MOVCU posto ottimizzato per T60						
	3	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
M06	1	CNC	Emette M06		X				
	2	TCI	Se ITCMPOS=1: →4 Se ITCMPOS=0: riemette MOVCU pos. per T60						
	3	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
	4	TCI	Emette CUTE posizione per T60 (attuale)						
	5	PLC	Smonta l'utensile (T60 in posto ottimizzato)			X			
	6	TCI	Aggiorna correttori				T0		
	7	TCI	Emette MOVCU posizione T10						T10
	8	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			
	9	TCI	Emette CUTE posizione T10						
	10	PLC	Carica l'utensile T10			X			
	11	TCI	Aggiorna correttori				T10		
T0	1	CNC	Emette T0 (si vuole scaricare T10)	X					
	2	TCI	Emette MOVCU posto ottimizzato per T10						
	3	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
M06	1	CNC	Emette M06		X				
	2	TCI	Emette CUTE posizione per T10 (attuale)						
	3	PLC	Smonta l'utensile (T10 in posto ottimizzato)			X			
	4	TCI	Aggiorna correttori				T0		

NOTE:

- Quando il cambio avviene tra due utensili della stessa dimensione, l'utensile che era sul mandrino va ad occupare il posto del nuovo utensile.
- Quando il cambio avviene tra due utensili di dimensioni diverse, viene prima forzato un T0 M06 e quindi l'utensile del mandrino viene depositato in catena (in un posto "ottimizzato" secondo alcune regole insite nel Tool Controller); viene poi ricercato e montato l'utensile desiderato.
- Quando viene eseguito un T0 M06 (programmato o forzato), il TCI cerca un posto "ottimizzato", solo se quel particolare utensile non era mai stato oggetto di "ottimizzazione".
- Come si può notare dalle varie sequenze della tabella, il PLC non deve preoccuparsi d'altro che eseguire i comandi del TCI. Deve comunque mettere **ITCMPOS=0** nel caso la catena venga mossa senza ordini del TCI, per garantirne il riposizionamento automatico prima del cambio.
- Il PLC deve mettere **FATTCl=1** quando ha terminato di eseguire il comando ricevuto con lo strobe **NEWTCl=1**; il TCI può così passare alla fase successiva o considerare finita tutta la sequenza attivata dal CNC con Txx ed M06.

9.11.2.2 ACICLICO (TYP=' ') - RANDOM (TGM=1) - CON Ut. Intermedio (UTI=1 o 2)

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTOM	FATTCI	TOOL Mandrino	TOOL Intermed.	TOOL Ricercato
T10	1	CNC	Emette T10 (utensile <b>PICCOLO</b> )	X			T0	vuoto	
	2	TCI	Emette MOVCU posizione T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
	4	TCI	UTI=1: Emette PREUTE posizione T10 UTI=2: sequenza finita.						
	5	PLC	Muove T10 da catena in posto intermedio			X		T10	
T20	1	CNC	Emette T20 (utensile <b>PICCOLO</b> )	X					
	2	TCI	UTI=1: Emette DEPUTE posizione T10 UTI=2: →4						
	3	PLC	Rimette T10 da posto intermedio in catena			X		vuoto	
	4	TCI	Emette MOVCU posizione T20						
	5	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			T20
	6	TCI	UTI=1: Emette PREUTE posizione T20 UTI=2: sequenza finita.						
	7	PLC	Muove T20 da catena in posto intermedio			X		T20	
M06	1	CNC	Emette M06		X				
	2	TCI	UTI=1: →4 UTI=2: Emette PREUTE posizione T20						
	3	PLC	Muove T20 da catena in posto intermedio			X		T20	
	4	TCI	Emette CUTE						
	5	PLC	Monta l'utensile T20 sul mandrino			X			
	6	TCI	Aggiorna correttori				T20	vuoto	
T10	1	CNC	Emette T10 (utensile <b>PICCOLO</b> )	X					
	2	TCI	Emette MOVCU posizione T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
	4	TCI	UTI=1: Emette PREUTE posizione T10 UTI=2: sequenza finita.						
	5	PLC	Muove T10 da catena in posto intermedio			X		T10	
M06	1	CNC	Emette M06		X				
	2	TCI	UTI=1: →4 UTI=2: Emette PREUTE posizione T10						
	3	PLC	Muove T10 da catena in posto intermedio			X		T10	
	4	TCI	Emette CUTE						
	5	PLC	Cambia l'utensile (monta T10; toglie T20)			X			
	6	TCI	Aggiorna correttori				T10	T20	
	7	TCI	Se ITCMPOS=1: →9 Se ITCMPOS=0: rimette MOVCU pos. T10						
	8	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			
	9	TCI	Emette DEPUTE posto T10						
	10	PLC	Deposita T20 nel posto di T10			X		vuoto	
T40	1	CNC	Emette T40 (utensile <b>MEDIO</b> )	X					
	2	TCI	Emette MOVCU posto T40						
	3	PLC	Posiziona la catena sul posto T40; ITCMPOS=1			X			T40
	4	TCI	UTI=1: Emette PREUTE posizione T40 UTI=2: sequenza finita.						
	5	PLC	Muove T40 da catena in posto intermedio			X		T40	





## CAMBIO UTENSILI

M06	1	CNC	Emette M06		X				
	2	TCI	UTI=1: →4 UTI=2: Emette PREUTE posizione T40						
	3	PLC	Muove T40 da catena in posto intermedio			X		T40	
	4	TCI	Emette CUTE						
	5	PLC	Cambia l'utensile (monta T40; toglie T10)			X			
	6	TCI	Aggiorna correttori				T40	T10	
	7	TCI	Emette MOVCU posto ottimizzato per T10						
	8	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
	9	TCI	Emette DEPUTE posto ottimizzato						
	10	PLC	Deposita T10 nel posto ottimizzato			X		vuoto	
T0	1	CNC	Emette T0 (si vuole scaricare T40)	X					
M06	1	TCI	Emette MOVCU posto ottimizzato per T40						
	2	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
	3	TCI	Emette CUTE						
	4	PLC	Smonta l'utensile (T40 in posto intermedio)			X			
	5	TCI	Aggiorna correttori				T0	T40	
	6	TCI	Emette DEPUTE posto ottimizzato						
	7	PLC	Deposita T40 nel posto ottimizzato			X		vuoto	

## NOTE:

- Quando il cambio avviene tra due utensili della stessa dimensione, l'utensile che era sul mandrino va ad occupare il posto del nuovo utensile.
- Quando il cambio avviene tra due utensili di dimensioni diverse, viene prima ricercato e montato l'utensile desiderato. Viene poi cercato un posto adatto per l'utensile del mandrino che viene infine depositato in catena (in un posto "ottimizzato" secondo alcune regole insite nel Tool Controller).
- Quando viene eseguito un T0 M06, il TCI cerca un posto "ottimizzato", solo se quel particolare utensile non era mai stato oggetto di "ottimizzazione".
- Come si può notare dalle varie sequenze della tabella, il PLC non deve preoccuparsi d'altro che eseguire i comandi del TCI. Deve comunque mettere **ITCMPOS=0** nel caso la catena venga mossa senza ordini del TCI, per garantirne il riposizionamento automatico prima del deposito a cambio utensile eseguito o per una successiva ricerca.
- Il PLC deve mettere **FATTCI=1** quando ha terminato di eseguire il comando ricevuto con lo strobe **NEWTCI=1**; il TCI può così passare alla fase successiva o considerare finita tutta la sequenza attivata dal CNC con Txx ed M06.
- Notare la differenza tra le due gestioni scelte tramite il parametro UTI:
  - **UTI=1**: il prelievo (PREUTE) viene comandato appena finita la ricerca (MOVCU)
  - **UTI=2**: il prelievo viene eseguito solo quando deve essere fatto il cambio (M06)



9.11.2.3 ACICLICO (TYP=' ') - Posti DEDICATI(TGM=0) - CON Ut. Interm. (UTI=1 o 2)

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTOM	FATTICI	TOOL Mandrino	TOOL Interm.	TOOL Ricercato
T10	1	CNC	Emette NEWT (T10 : utensile <b>PICCOLO</b> )	X			T0	vuoto	
	2	TCI	Emette MOVCU posizione T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
	4	TCI	UTI=1: Emette PREUTE posizione T10 UTI=2: sequenza finita.						
	5	PLC	Muove T10 da catena in posto intermedio			X		T10	
T20	1	CNC	Emette NEWT (T20 : utensile <b>PICCOLO</b> )	X					
	2	TCI	UTI=1: Emette DEPUTE posizione T10 UTI=2: →4						
	3	PLC	Rimette T10 da posto intermedio in catena			X		vuoto	
	4	TCI	Emette MOVCU posizione T20						
	5	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			T20
	6	TCI	UTI=1: Emette PREUTE posizione T20 UTI=2: sequenza finita.						
	7	PLC	Muove T20 da catena in posto intermedio			X		T20	
M06	1	CNC	Emette M06		X				
	4	TCI	Emette CUTE posizione T20						
	5	PLC	Carica l'utensile T20			X			
	6	TCI	Aggiorna correttori				T20		
T10	1	CNC	Emette NEWT (T10 : utensile <b>PICCOLO</b> )	X					
	2	TCI	Emette MOVCU posizione T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
	4	TCI	UTI=1: Emette PREUTE posizione T10 UTI=2: sequenza finita.						
	5	PLC	Muove T10 da catena in posto intermedio			X		T10	
M06	1	CNC	Emette M06		X				
	2	TCI	Emette CUTE						
	3	PLC	Cambia l'utensile (monta T10; toglie T20)			X			
	4	TCI	Aggiorna correttori				T10	T20	
	5	TCI	Emette MOVCU posizione dedicata T20						
	6	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			T20
	7	TCI	Emette DEPUTE posto T20						
	8	PLC	Deposita T20 nel posto dedicato			X		vuoto	
T40	1	CNC	Emette NEWT (T40 : utensile <b>MEDIO</b> )	X					
	2	TCI	Emette MOVCU posizione T40						
	3	PLC	Posiziona la catena sul posto T40; ITCMPOS=1			X			T40
	4	TCI	UTI=1: Emette PREUTE posizione T40 UTI=2: sequenza finita.						
	5	PLC	Muove T40 da catena in posto intermedio			X		T40	
M06	1	CNC	Emette M06		X				
	2	TCI	Emette CUTE						
	3	PLC	Cambia l'utensile (monta T40; toglie T10)			X			
	4	TCI	Aggiorna correttori				T40	T10	
	5	TCI	Emette MOVCU posizione dedicata T10						
	6	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
	7	TCI	Emette DEPUTE posto T10						
	8	PLC	Deposita T10 nel posto dedicato			X		vuoto	



## CAMBIO UTENSILI

---

---

T0	1	CNC	Emette NEWT (si vuole scaricare T40)	X					
M06	1	CNC	Emette M06		X				
	2	TCI	Emette MOVCU posto dedicato per T40						T40
	3	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
	4	TCI	Emette CUTE						
	5	PLC	Smonta l'utensile (T10 in posto intermedio)			X			
	6	TCI	Aggiorna correttori				T0	T40	
	7	TCI	Emette DEPUTE posto T40						
	8	PLC	Deposita T40 nel posto dedicato			X		vuoto	

## NOTE:

- Ogni utensile viene ridepositato nel posto originale (dedicato) e quindi non ci sono differenze tra piccoli, medi, etc. .
- Quando il cambio avviene tra due utensili qualunque, viene prima ricercato e montato l'utensile desiderato e quindi l'utensile del mandrino viene depositato in catena nel suo posto dedicato. Ogni cambio utensile comporta quindi una **doppia ricerca**: la prima per cercare l'utensile nuovo ed la seconda per cercare il posto dedicato a quello che era sul mandrino, per depositarlo.
- Quando viene programmato T0, non viene fatta nessuna ricerca. Con il successivo M06, il TCI comanda la ricerca del posto dedicato all'utensile sul mandrino; comanda poi un cambio ed un deposito dell'utensile sul magazzino, lasciando vuoti mandrino e posto intermedio.
- Per **ITCMPOS**, **NEWTCI** e **FATTCI** valgono le note date nelle tabelle precedenti.

9.11.2.4 ACICLICO (TYP=' ') - Posti DEDICATI (TGM=0) - SENZA Ut. Intermedio (UTI=0)

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTOM	FATTICI	TOOL Mandrino	TOOL Intermed.	TOOL Ricercato
T10	1	CNC	Emette NEWT	X			T0		
	2	TCI	Emette MOVCU posizione T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
T20	1	CNC	Emette NEWT	X					
	2	TCI	Emette MOVCU posizione T20						
	3	PLC	ITCMPOS=0 (fare sempre prima di mov. cat.)						
	4	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			T20
M06	1	CNC	Emette M06		X				
	2	TCI	Se ITCMPOS=1: →4 Se ITCMPOS=0: riemette MOVCU pos. T20						
	3	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			T20
	4	TCI	Emette CUTE posizione T20						
	5	PLC	Carica l'utensile T20			X			
	6	TCI	Aggiorna correttori				T20		
T10	1	CNC	Emette NEWT	X					
	2	TCI	Emette MOVCU posizione T20 (tool su mandr.)						
	3	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			
M06	1	CNC	Emette M06		X				
	2	TCI	Se ITCMPOS=1: →4 Se ITCMPOS=0: riemette MOVCU pos. T20						
	3	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			T20
	4	TCI	Emette CUTE posizione T20						
	5	PLC	Cambia l'utensile (depone T20 nel suo posto)			X			
	6	TCI	Aggiorna correttori				T0		
	7	TCI	Emette MOVCU posizione T10 (tool desiderato)						
	8	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
	9	TCI	Emette CUTE posizione T10						
	10	PLC	Carica l'utensile T10			X			
	11	TCI	Aggiorna correttori				T10		
T0	1	CNC	Emette T0 (si vuole scaricare T10)	X					
	2	TCI	Emette MOVCU posto dedicato per T10						
	3	PLC	Posiziona la catena sul posto; ITCMPOS=1			X			
M06	1	CNC	Emette M06		X				
	2	TCI	Emette CUTE posizione per T10						
	3	PLC	Smonta l'utensile (T10 in posto dedicato)			X			
	4	TCI	Aggiorna correttori				T0		



NOTE:

- Ogni utensile viene ridepositato nel posto originale (dedicato) e quindi non ci sono differenze tra piccoli, medi, etc. .
- Quando il cambio avviene tra due utensili qualunque, viene prima forzato un T0 M06 e quindi l'utensile del mandrino viene depositato in catena nel suo posto dedicato; viene poi ricercato e montato l'utensile desiderato. Ogni cambio utensile comporta quindi un **doppio cambio**: il primo per scaricare l'utensile dal mandrino ed il secondo per caricare quello nuovo.
- Per **ITCMPOS**, **NEWTCI** e **FATTCI** valgono le note date nelle tabelle precedenti.



### 9.11.3 C.Ut. a CATENA CICLICO

#### 9.11.3.1 CICLICO (TYP='C') - RANDOM (TGM=1) - CON Ut. Intermedio (UTI=1 o 2)

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTOM	FATTCI	TOOL Mandrino	TOOL Intermed.	TOOL Ricercato
T10	1	CNC	Emette T10 (utensile <b>PICCOLO</b> )	X			T0	vuoto	
	2	TCI	Emette MOVCU posizione T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
	4	TCI	UTI=1: Emette PREUTE posizione T10 UTI=2: sequenza finita.						
	5	PLC	Muove T10 da catena in posto intermedio			X		T10	
T20	1	CNC	Emette T20 (utensile <b>PICCOLO</b> )	X					
M06	1	CNC	Emette M06		X				
	2	TCI	Emette CUTE						
	3	PLC	Monta l'utensile T10 sul mandrino (il primo)			X			
	4	TCI	Aggiorna correttori				T10	vuoto	
	5	TCI	Emette MOVCU posizione T20						
	6	PLC	Posiziona la catena sul posto T20; ITCMPOS=1			X			T20
	7	TCI	Emette PREUTE posizione T20						
	8	PLC	Muove T20 da catena in posto intermedio			X		T20	
	9	TCI	Emette CUTE						
	10	PLC	Monta l'utensile T20 sul mandrino (il secondo)			X			
	11	TCI	Aggiorna correttori				T20	T10	
	12	TCI	Emette DEPUTE posizione T20						
	13	PLC	Rimette T10 in catena su posto di T20			X		vuoto	
T10	1	CNC	Emette T10 (utensile <b>PICCOLO</b> )	X					
	2	TCI	Emette MOVCU posizione T10						
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X			T10
	4	TCI	UTI=1: Emette PREUTE posizione T10 UTI=2: sequenza finita.						
	5	PLC	Muove T10 da catena in posto intermedio			X		T10	
M06	1	CNC	Emette M06		X				
	2	TCI	UTI=1: →4 UTI=2: Emette PREUTE posizione T10						
	3	PLC	Muove T10 da catena in posto intermedio			X		T10	
	4	TCI	Emette CUTE						
	5	PLC	Cambia l'utensile			X			
	6	TCI	Aggiorna correttori				T10	T20	
	7	TCI	Emette DEPUTE posizione T10						
	8	PLC	Rimette T20 in catena su posto di T10			X		vuoto	
T40	1	CNC	Emette T40 (utensile <b>MEDIO</b> )	X					
	2	TCI	Emette MOVCU posto T40						
	3	PLC	Posiziona la catena sul posto T40; ITCMPOS=1			X			T40
	4	TCI	UTI=1: Emette PREUTE posizione T40 UTI=2: sequenza finita.						
	5	PLC	Muove T40 da catena in posto intermedio			X		T40	
M06	1	CNC	Emette M06		X				
	2	TCI	UTI=1: →4 UTI=2: Emette PREUTE posizione T40						
	3	PLC	Muove T40 da catena in posto intermedio			X		T40	



CAMBIO UTENSILI

	4	TCI	Emette CUTE					
	5	PLC	Cambia l'utensile (monta T40; toglie T10)			X		
	6	TCI	Aggiorna correttori				T40	T10
	7	TCI	Emette MOVCU posto ottimizzato per T10					
	8	PLC	Posiziona la catena sul posto; ITCMPOS=1			X		
	9	TCI	Emette DEPUTE posto ottimizzato					
	10	PLC	Deposita T10 nel posto ottimizzato			X		vuoto
T60	1	CNC	Emette T60 (utensile <b>MEDIO</b> )	X				
	2	TCI	Emette MOVCU posto T60					
	3	PLC	Posiziona la catena sul posto T60; ITCMPOS=1			X		T60
	4	TCI	UTI=1: Emette PREUTE posizione T60 UTI=2: sequenza finita.					
	5	PLC	Muove T60 da catena in posto intermedio			X		T60
M06	1	CNC	Emette M06			X		
	2	TCI	UTI=1: →4 UTI=2: Emette PREUTE posizione T60					
	3	PLC	Muove T60 da catena in posto intermedio			X		T60
	4	TCI	Emette CUTE					
	5	PLC	Cambia l'utensile (monta T60; toglie T40)			X		
	6	TCI	Aggiorna correttori				T60	T40
	7	TCI	Emette DEPUTE posizione T60					
	8	PLC	Rimette T40 in catena su posto di T60			X		vuoto
T10	1	CNC	Emette T10 (utensile <b>PICCOLO</b> )	X				
	2	TCI	Emette MOVCU posizione T10					
	3	PLC	Posiziona la catena sul posto T10; ITCMPOS=1			X		T10
	4	TCI	UTI=1: Emette PREUTE posizione T10 UTI=2: sequenza finita.					
	5	PLC	Muove T10 da catena in posto intermedio			X		T10
M06	1	CNC	Emette M06		X			
	2	TCI	UTI=1: →4 UTI=2: Emette PREUTE posizione T10					
	3	PLC	Muove T10 da catena in posto intermedio			X		T10
	4	TCI	Emette CUTE					
	5	PLC	Cambia l'utensile (monta T10; toglie T60)			X		
	6	TCI	Aggiorna correttori				T10	T60
	7	TCI	Emette MOVCU posto ottimizzato per T60					
	8	PLC	Posiziona la catena sul posto; ITCMPOS=1			X		
	9	TCI	Emette DEPUTE posto ottimizzato					
	10	PLC	Deposita T60 nel posto ottimizzato			X		vuoto
T0	1	CNC	Emette T0 (si vuole scaricare T10)	X				
M06	1	TCI	Emette MOVCU posto ottimizzato per T10					
	2	PLC	Posiziona la catena sul posto; ITCMPOS=1			X		
	3	TCI	Emette CUTE					
	4	PLC	Smonta l'utensile (T10 in posto intermedio)			X		
	5	TCI	Aggiorna correttori				T0	T10
	6	TCI	Emette DEPUTE posto ottimizzato					
	7	PLC	Deposita T10 nel posto ottimizzato			X		vuoto

NOTE:

- Questo tipo di cambio è necessario nel caso di pinze del braccio di scambio differenziate e tali da non poter ridepositare in catena un utensile già ricercato. Diventa quindi necessario e obbligatorio effettuare comunque un cambio dopo una ricerca, anche se poi occorrerà riefettuare una seconda ricerca ed un nuovo cambio per mettere l'utensile desiderato sul mandrino.
- Valgono anche tutte le note già viste per il cambio utensili ACICLICO (TYP=' ') - RANDOM (TGM=1) - CON Ut. Intermedio (UTI=1 o 2).



---

Sono possibili anche altri tipi di cambi utensili, ad esempio:

CICLICO (TYP='C' ) - RANDOM (TGM=1) - SENZA Ut. Intermedio (UTI=0)

CICLICO (TYP='C') - Posti DEDICATI(TGM=0) - CON o SENZA Ut. Intermed. (UTI=0 o 1 o 2)  
ma non ne vengono date le relative tabelle in quanto molto rari; è comunque possibile ricavarne  
la logica di funzionamento facendo riferimento a quelle date.





## 9.12 CAMBIO UTENSILI NEL TORNIO {T}

I CNC ECS 1802 / 4802 controllano gli **utensili** del Tornio attraverso uno specifico Software, sempre presente, denominato **TCI (Tool Controller Interno)** che permette di gestire fino a 150 posti utensile con un massimo di 255 "taglienti" ed i diversi tipi di cambi utensili che solitamente s'incontrano sui torni orizzontali e verticali.

NOTA:

Per chi conosce le varie gestioni realizzate sulle "frese", si precisa che il tipo realizzato sul tornio è a "posti dedicati"; inoltre, tutti gli utensili sono in pratica multitaglienti (in ogni posto possono essere messi più utensili), non esiste TPC (codice fisico) ma solo TTC (codice tipo), non esiste distinzione per size (utile/necessaria solo per gestioni random), non esistono correttori Dxxx ed infine non è prevista la gestione del c.ut. da subroutine.

In un apposito campo del file di taratura TYPUCU.TAB (vedi MANUALE TARATURE 720P385) va precisato se il cambio è del tipo a torretta classica (revolver) o con un magazzino circolare (ruota o catena) con possibilità di far "migrare" l'utensile dal magazzino stesso alla posizione di lavoro. Per il secondo tipo è inoltre possibile definire se il cambio è del tipo ciclico o aciclico e se ci sono utensili intermedi.

Nel file di taratura CUMAG.TAB vengono definiti i posti utensili presenti e, nel caso di "revolver", se ci sono più torrette e quante posizioni hanno ciascuna.

La Funzione Ausiliaria (F.A.) "**T**" di ricerca utensile (definita da E.C.S. FASUB), richiama la subroutine TOOL (fornita di base sui CNC ECS) che contiene l'istruzione \$T<%%V7005>. Questa istruzione, oltre ad essere emessa verso il PLC, attiva il TCI che, in base alle tarature presenti:

- se necessario e possibile, deposita un utensile già prelevato,
- comanda la ricerca dell'utensile desiderato,
- comanda il prelievo dello stesso in attesa del cambio,
- oppure non fa niente e memorizza il T desiderato fino alla prossima M06.

La Funzione Ausiliaria (F.A.) "**M06**" di cambio utensile (definita FASUB) fornita di base da E.C.S. sui propri CNC, esegue tutta una serie di verifiche e poi emette semplicemente la F.A. \$M06 oppure richiama la subroutine CUM (per il cambio utensili manuale, ad esempio, di un utensile "a terra") oppure la subroutine CUA (per il cambio utensile automatico).

Queste attivano il TCI tramite l'istruzione \$M06; il TCI, in base alla situazione utensili attuale, alle predisposizioni fatte ed alle tarature presenti, emette una serie di comandi verso il PLC per eseguire le operazioni (prelievo, deposito, ricerca, cambio) necessarie per completare il cambio utensile programmato. La subroutine **CUA** può contenere movimenti assi per portare gli stessi in posizione di sicurezza o di cambio.

Con cambio utensile escluso (G309), la M06 richiama la **CUM** che arresta il mandrino e la prosecuzione del Part-program; visualizza poi sul monitor un messaggio del tipo:

**Cambiare utensile  
poi premere [START].**

Il CNC attende che l'operatore prema il pulsante [START], dopo aver cambiato l'utensile manualmente, per procedere nell'esecuzione del Part-program.

Il TCI del CNC aiuta a realizzare il Cambio Utensili tramite il metodo detto **CUA da PLC**.



La sequenza dei movimenti sul tornio e lo scambio di segnali con il CNC ed il Tool Controller sono realizzati tramite equazioni scritte nel PLC. Normalmente vengono usati uno o più sequenziatori ("macchine" logiche a più stati con specifiche azioni legate ai singoli stati e condizioni prefissate che controllano il passaggio da stato a stato; il linguaggio "C" permette di realizzare tali "macchine" con la funzione **switch(...)** ed i **case nnn:** associati).

### 9.12.1 Parametri di taratura del Cambio Utensili

Le caratteristiche del Cambio Utensili sono definite negli appositi files di taratura ed in particolare in TYPUCU.TAB.

Qui di seguito, sono riassunti i vari files che contengono tarature e/o procedure riguardanti il Cambio Utensili.

TYPUCU.TAB	Tipo di cambio utensile e vari parametri
CUMAG.TAB	Configurazione delle torrette e/o magazzino
CUM	Ciclo di cambio utensili manuale
CUA	Ciclo di cambio utensili automatico
TOOL	F.A. subroutine "T".

La funzione di questi files è dettagliatamente spiegata nel MANUALE TARATURE 720P385.

I files CUM e CUA, presenti sul CNC, normalmente non richiedono modifiche; essi possono comunque essere ritoccati per adattarli alle esigenze particolari di un determinato tornio.

### 9.12.2 GESTIONE DEGLI UTENSILI

Il TOOL CONTROLLER INTERNO (TCI) è il software responsabile della gestione per tutto quanto riguarda gli utensili a bordo macchina per cui deve conglobare molteplici funzioni; fra queste le principali sono:

- Fornire al CNC il giusto utensile da usare per la lavorazione successiva.
- Permettere la controllo e la modifica dello stato della o delle torrette e/o del magazzino e dello stato d'ogni singolo utensile.
- Gestire l'eventuale trasferimento dell'utensile tra magazzino e posto di lavoro.
- Gestire la vita e l'usura d'ogni singolo tagliente permettendo di ottimizzarne l'utilizzo, aggiornando le tabelle opportune.

La configurazione della o delle torrette e/o del magazzino utensili viene definita dal costruttore M.U. in fase d'installazione del CNC compilando il file **CUMAG.TAB**.

#### 9.12.2.1 Gestione a POSTI DEDICATI

Per chi conosce le varie gestioni realizzate sulle "frese", si ripete qui che il tipo di gestione realizzato sul tornio è sempre a "posti dedicati".

Questo tipo di gestione permette di assegnare a priori la posizione d'ogni singolo utensile sul magazzino e/o torretta (corrispondenza tra codice utensile e suo posto); questa disposizione rimane poi inalterata durante il funzionamento. Questo comporta che non ci saranno scambi di posto fra gli utensili ma, una volta definita la corrispondenza utensile/posto, questa sarà sempre mantenuta.



---

### 9.13 TT.BIN (file della tabella utensili: Tool Table) {T}

Tutti i parametri di tutti gli utensili inseriti in tabella manualmente o con la procedura di carico dati da file (funzioni G79x, vedi Norme di programmazione) sono scritti in un file di nome TT.BIN . E' un file "data-base" gestito dal CNC, non è visibile o modificabile con un normale editor ASCII poiché i dati sono registrati in formato binario.

Il file TT.BIN contiene 255 blocchi-dati. Ogni blocco-dati contiene tutte le informazioni relative ad un singolo tagliente. La lettura/scrittura dei dati viene eseguita automaticamente dal CNC nelle varie fasi di lavoro del tornio.

L'operatore può inserire nuovi dati, vederli e modificarli utilizzando le apposite "maschere" accessibili tramite i tasti funzione specifici del menù orizzontale "Utensili" (vedi monografia NORME D'USO).

Per l'accesso (in lettura e/o ricerca) al file TT.BIN da parte del PLC, sono disponibili le funzioni **IOPLCnread\_tt(...)** e **IOPLCGet\_nread\_tt(..)** che sono già state descritte al capitolo 9.5, assieme alla struttura dati della Tool Table.

### 9.14 CLASSIFICAZIONE DEI CAMBI UTENSILI AUTOMATICI {T}

I CNC ECS 1802 / 4802 per torni gestiscono quattro tipi di cambio utensile:

- a) **REVOLVER**
- b) **TORRETTA**
- c) **CICLICO**
- d) **ACICLICO**

Questi cambi utensili sono di tipo "sincrono", poiché la gestione prevede che dopo un CAMBIO, l'utensile che è stato appena tolto dal mandrino rientri nel magazzino prima che inizi la RICERCA dell'utensile successivo. La gestione prevede anche cambi utensili con un utensile intermedio temporaneo. È il caso in cui può esistere un altro utensile fuori magazzino, oltre a quello montato sul mandrino, durante la rotazione del magazzino stesso per la ricerca del posto di deposito.

#### a) CUA di tipo REVOLVER (torretta tipica dei torni orizzontali)

- Gli utensili occupano sempre lo stesso posto nella torretta a revolver.
- Ogni posto Torretta/revolver può contenere più taglienti; il Tool Attuale TA lavora rimanendo nel suo posto sulla torretta/revolver.
- L'istruzione **T** non provoca movimenti sul tornio; la rotazione torretta/revolver è comandata dall'istruzione **M06**.

Tale tipo di cambio utensile è selezionato ponendo **TYP='R'** nel file TYPKU.TAB (vedi Manuale Tarature 720P385).

#### b) CUA di tipo TORRETTA

- Gli utensili occupano sempre lo stesso posto nel magazzino.
- Ogni posto Magazzino può contenere più taglienti; il Tool Attuale TA ne è estratto, ma **RIENTRA** nel suo posto magazzino **PRIMA** che sia possibile eseguire un'altra ricerca.



- L'istruzione **T** non provoca la ricerca dell'utensile nel magazzino; la ricerca è rimandata a quando sarà emessa l'istruzione **M06**, che provoca allora questa sequenza di eventi:

DEPOSITO: TA torna in Magazzino  
RICERCA: viene cercato T nel Magazzino  
PRELIEVO: T diventa TA (viene estratto dal magazzino)

Tale tipo di cambio utensile è selezionato ponendo **TYP='T'** nel file TYPUCU.TAB .

### c) CUA di tipo CICLICO

Esiste un magazzino a ruota o catena ed un sistema di trasporto dell'utensile desiderato da magazzino a posizione di lavoro e viceversa. Il tipo di meccanica usata non può eseguire due o più RICERCHE consecutive nel magazzino, ma deve completare un CAMBIO dopo ogni RICERCA. In altre parole, essa può eseguire solo cicli completi di RICERCA-CAMBIO, RICERCA-CAMBIO, ecc. .

Tale tipo di cambio utensile è selezionato ponendo **TYP='C'** nel file TYPUCU.TAB .

### d) CUA tipo "ACICLICO"

Sono quei cambi utensili simili al tipo CICLICO ma che possono eseguire due o più RICERCHE consecutive nel magazzino; non è necessario eseguire un CAMBIO dopo ogni RICERCA. In altre parole, possono eseguire cicli di tipo RICERCA-RICERCA-CAMBIO oppure CAMBIO-CAMBIO.

Tale tipo di cambio utensile è selezionato ponendo **TYP=' '** (spazio) nel file TYPUCU.TAB .

## 9.14.1 SEGNALI LOGICI

Per la gestione del Cambio Utensili, sono presenti diversi segnali di scambio tra CNC e PLC che sono stati associati ai sinonimi elencati al paragrafo 9.7 al quale si rimanda.

Naturalmente non sono applicabili i segnali che si riferiscono al "**CUA da Subroutine**" , non valido per i torni.

## 9.14.2 DIALOGO TCI ↔ PLC

Per l'esecuzione delle varie sequenze relative alla ricerca utensili, prelievo, deposito e cambio utensile, è previsto un dialogo tra il Tool Controller TCI e PLC. Tale dialogo è ampiamente descritto al paragrafo 9.8 al quale si rimanda.

Non si applicano i comandi relativi al carico/scarico utensili dal magazzino ed alla gestione random, in quanto funzioni non previste per il tornio.

## 9.15 VARI TIPI DI CAMBI UTENSILI AUTOMATICI {T}

In questo paragrafo si intende spiegare il comportamento del TCI nei vari tipi di CUA del tornio. Si è pensato di utilizzare delle tabelle che rendono più comprensibile la sequenza.

Il progettista del PLC deve normalmente predisporre uno *switch* (vedi esempio al paragrafo 9.11.1.1) che intercetti i comandi **TCI→PLC** ed esegua le fasi della macchina che realizzano

**CAMBIO UTENSILI**

---

---

il comando. Infatti il PLC deve comportarsi da SLAVE (non prende iniziative) e deve esclusivamente stare in attesa dei comandi che arrivano dal TCI, eseguirli e rendere il FATTCI al comando senza dover ricordare niente dell'utensile.

Occorre tener presente che il TCI ricorda sempre (anche dopo un OFF/ON CNC) l'ultimo comando al quale ha ricevuto il "fatto" e che esegue le sue scelte basandosi su questo.

Solo nel caso che il PLC preveda di poter muovere il magazzino o le torrette utensili in modo indipendente (senza richieste del TCI) occorre prendere qualche precauzione. In questo caso torna utile il segnale ITCMPOS che, messo a zero, segnala la necessità di riposizionare il magazzino stesso; il TCI provvederà a ripetere l'emissione del comando MOVCU prima di passare ad una fase successiva che richieda il magazzino in posizione.

### 9.15.1 C.Ut. a REVOLVER (TYP='R')

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTCI	FATTOM	TOOL Attivo
T20	1	CNC	Emette NEWT	X			T0
M06	1	CNC	Emette M06			X	
	2	TCI	Emette MOVCU posizione di T20 su OTCDATIH				
	3	PLC	Posiziona la torretta sul posto; ITCMPOS=1		X		
	4	TCI	Aggiorna correttori				T20
T30	1	CNC	Emette NEWT	X			
M06	1	CNC	Emette M06			X	
	2	TCI	Se ITCMPOS=1 e posto di T30= posto attuale →5				
	3	TCI	Altrimenti: emette MOVCU posto di T30 su OTCDATIH				
	4	PLC	Posiziona la torretta sul posto; ITCMPOS=1		X		
	5	TCI	Aggiorna correttori				T30
T0	1	CNC	Emette NEWT	X			
M06	1	CNC	Emette M06			X	
	2	TCI	Aggiorna correttori				T0

NOTE:

- Quando il cambio avviene tra due taglienti nella stessa posizione di torretta e ITCMPOS=1, il TCI non dà nessun comando al PLC.
- Il PLC deve mettere **ITCMPOS=0**, nel caso la torretta venga mossa senza ordini del TCI, per garantirne il riposizionamento automatico. In questo caso, infatti, il TCI riemette il comando MOVCU (con il valore della posizione del T desiderato su OTCDATIH), anche se la posizione corrisponde all'ultima comandata.
- Il PLC deve mettere **FATTCI=1** quando ha terminato di eseguire il comando di rotazione torretta ricevuto con lo strobe **NEWTCI=1**; il TCI può così considerare finita la sequenza attivata dal CNC con Txx ed M06.



### 9.15.2 C.Ut. a TORRETTA (TYP='T')

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTCI	FATTOM	TOOL Attivo
T20	1	CNC	Emette NEWT	X			T0
M06	1	CNC	Emette M06			X	
	2	TCI	Emette MOVCU posizione di T20 su ODATIH				
	3	PLC	Posiziona la torretta sul posto; ITCMPOS=1		X		
	4	TCI	Emette CUTE con TM=0				
	5	PLC	Porta l'utensile in posizione di lavoro (se necessario)		X		
	6	TCI	Aggiorna correttori				T20
T30	1	CNC	Emette NEWT	X			
M06	1	CNC	Emette M06			X	
	2	TCI	Se posto di T30= posto attuale →13				
	3	TCI	Se ITCMPOS=1 →6				
	4	TCI	Emette MOVCU posizione di T20 (T attuale) su ODATIH				
	5	PLC	Posiziona la torretta sul posto; ITCMPOS=1		X		
	6	TCI	Emette CUTE con TM=1				
	7	PLC	Reinserisce T20 se estratto		X		
	8	TCI	Aggiorna correttori				T0
	9	TCI	Emette MOVCU posizione di T30 (T nuovo) su ODATIH				
	10	PLC	Posiziona la torretta sul posto; ITCMPOS=1		X		
	11	TCI	Emette CUTE con TM=0				
	12	PLC	Porta l'utensile in posizione di lavoro (se necessario)		X		
	13	TCI	Aggiorna correttori				T30
T0	1	CNC	Emette NEWT	X			
M06	1	CNC	Emette M06			X	
	2	TCI	Se ITCMPOS=1: →5				
	3	TCI	Emette MOVCU posizione di T30 (T attuale) su ODATIH				
	4	PLC	Posiziona la torretta sul posto; ITCMPOS=1		X		
	5	TCI	Emette CUTE con TM=1				
	6	PLC	Reinserisce T30 se estratto		X		
	7	TCI	Aggiorna correttori				T0

NOTE:

- Quando il cambio avviene tra due taglienti nella stessa posizione di torretta, il TCI non dà nessun comando al PLC.
- Quando il cambio avviene tra due utensili di posto diverso, viene prima comandato un cambio per depositare in torretta l'utensile attivo (nel suo posto); viene poi ricercato e montato l'utensile desiderato.
- Con il comando CUTE, il PLC esegue l'estrazione o il reinserimento dell'utensile in torretta (se la meccanica lo prevede) in base allo stato del segnale TM (se=1 significa Utensile Montato in posizione di lavoro e quindi occorre fare il deposito in torretta).
- Il PLC deve mettere **ITCMPOS=0**, nel caso la torretta venga mossa senza ordini del TCI, per garantirne il riposizionamento automatico prima del cambio. In questo caso, infatti, il TCI riemette un comando MOVCU con il valore della posizione del T attivo su OTCDATIH, prima del comando CUTE di deposito.
- Il PLC deve mettere **FATTCI=1** quando ha terminato di eseguire il comando ricevuto con lo strobe **NEWTCI=1**; il TCI può così passare alla fase successiva o considerare finita tutta la sequenza attivata dal CNC con Txx ed M06.

9.15.3 C.Ut. a CATENA ACICLICO (TYP=' ') - SENZA Ut. Intermedio (UTI=0)

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTCI	FATTOM	TOOL Cercato	TOOL Attivo
T20	1	CNC	Emette NEWT					T0
	2	PLC		X				
	3	TCI	Emette MOVCU posizione di T20 su ODATIH					
	4	PLC	Posiziona la catena sul posto; ITCMPOS=1		X		T20	
M06	1	CNC	Emette M06			X		
	2	TCI	Emette CUTE con TM=0					
	3	PLC	Porta l'utensile in posizione di lavoro		X			
	4	TCI	Aggiorna correttori					T20
T30	1	CNC	Emette NEWT					
	2	PLC		X				
	3	TCI	Se posto di T30= posto attuale →12 di M06					
	4	TCI	Emette MOVCU posizione di T20 (T attuale)					
	5	PLC	Posiziona la catena sul posto; ITCMPOS=1		X		T20	
M06	1	CNC	Emette M06			X		
	2	TCI	Se ITCMPOS=1 →5					
	3	TCI	Emette MOVCU posizione di T20 (T attuale)					
	4	PLC	Posiziona la catena sul posto; ITCMPOS=1		X		T20	
	5	TCI	Emette CUTE con TM=1					
	6	PLC	Reinserisce T20 in catena		X			
	7	TCI	Aggiorna correttori					T0
	8	TCI	Emette MOVCU posizione di T30 (T nuovo)					
	9	PLC	Posiziona la catena sul posto; ITCMPOS=1		X		T30	
	10	TCI	Emette CUTE con TM=0					
	11	PLC	Porta l'utensile in posizione di lavoro		X			
	12	TCI	Aggiorna correttori					T30
T0	1	CNC	Emette NEWT					
	2	PLC		X				
M06	1	CNC	Emette M06			X		
	2	TCI	Se ITCMPOS=1: →5					
	3	TCI	Emette MOVCU posizione di T30 (T attuale)					
	4	PLC	Posiziona la catena sul posto; ITCMPOS=1		X		T30	
	5	TCI	Emette CUTE con TM=1					
	6	PLC	Reinserisce T30 in catena		X			
	7	TCI	Aggiorna correttori					T0





## NOTE:

- Come si può notare dalle varie sequenze della tabella, il PLC non deve preoccuparsi d'altro che eseguire i comandi del TCI.
- Quando il cambio avviene tra due taglienti dello stesso portautensili, il TCI non dà nessun comando al PLC ma aggiorna subito i correttori.
- Quando il cambio avviene tra due utensili di posto diverso, viene prima comandato un cambio per depositare in catena l'utensile attivo (nel suo posto); viene poi ricercato e montato l'utensile desiderato.
- Con il comando CUTE, il PLC esegue l'estrazione o il reinserimento dell'utensile in magazzino in base allo stato del segnale TM (se=1 significa Utensile Montato in posizione di lavoro e quindi occorre fare il deposito).
- Il PLC deve mettere **ITCMPOS=0**, nel caso la catena venga mossa senza ordini del TCI, per garantirne il riposizionamento automatico prima del cambio. In questo caso, infatti, il TCI riemette un comando MOVCU con il valore della posizione del T attivo su OTCDATIH, prima del comando CUTE di deposito (con TM=1). Il TCI emette poi un nuovo comando MOVCU con il valore della posizione del T desiderato su OTCDATIH, prima del definitivo comando CUTE di prelievo (con TM=0).
- Il PLC deve mettere **FATTCI=1** quando ha terminato di eseguire il comando ricevuto con lo strobe **NEWTCI=1**; il TCI può così passare alla fase successiva o considerare finita tutta la sequenza attivata dal CNC con Txx ed M06.

9.15.4 C.Ut. a CATENA ACICLICO (TYP=' ') - CON Ut. Intermedio (UTI=1 o 2)

Richiesta	Sequenza	ATTORE	AZIONE	FATTOT	FATTCI	FATTOM	TOOL Cercato	TOOL Intermedio	TOOL Attivo
T20	1	CNC	Emette NEWT					vuoto	T0
	2	PLC		X					
	3	TCI	Emette MOVCU posizione di T20 su ODATIH						
	4	PLC	Posiziona la catena sul posto; ITCMPOS=1		X		T20		
	5	TCI	UTI=1: Emette PREUTE posizione T20 UTI=2: sequenza finita.						
	6	PLC	Muove T20 da catena in posto intermedio		X			T20	
M06	1	CNC	Emette M06			X			
	2	TCI	Emette CUTE con TM=0						
	3	PLC	Porta l'utensile in posizione di lavoro		X			vuoto	
	4	TCI	Aggiorna correttori						T20
T30	1	CNC	Emette NEWT						
	2	PLC		X					
	3	TCI	Se posto di T30= posto attuale: fine sequenza						
	4	TCI	Emette MOVCU posizione di T30 (T nuovo)						
	5	PLC	Posiziona la catena sul posto; ITCMPOS=1		X		T30		
	6	TCI	UTI=1: Emette PREUTE posizione T30 UTI=2: sequenza finita.						
	7	PLC	Muove T30 da catena in posto intermedio		X			T30	
M06	1	CNC	Emette M06			X			
	2	TCI	Se posto di T30= posto attuale: aggiorna correttori; fine sequenza						
	3	TCI	UTI=1: →5 UTI=2: Emette PREUTE posizione T30.						
	4	PLC	Muove T30 da catena in posto intermedio		X			T30	
	5	TCI	Emette CUTE						
	6	PLC	Cambia l'utensile (monta T30; toglie T20)		X			T20	
	7	TCI	Aggiorna correttori						T30
	8	TCI	Emette MOVCU posizione di T20 (T vecchio)						
	9	PLC	Posiziona la catena sul posto; ITCMPOS=1		X		T20		
	10	TCI	Emette DEPUTE posizione T20						
	11	PLC	Deposita T20 in catena		X			vuoto	
T0	1	CNC	Emette NEWT						
	2	PLC		X					
M06	1	CNC	Emette M06			X			
	2	TCI	Emette MOVCU posizione di T30 (T attuale)						
	3	PLC	Posiziona la catena sul posto; ITCMPOS=1		X		T30		
	4	TCI	Emette CUTE						
	5	PLC	Cambia l'utensile (toglie T30)		X			T30	
	6	TCI	Aggiorna correttori						T0
	7	TCI	Emette DEPUTE posizione T30						
	8	PLC	Reinserisce T30 in catena		X			vuoto	



NOTE:

- Quando il cambio avviene tra due taglienti dello stesso portautensili, il TCI non dà nessun comando al PLC ma aggiorna subito i correttori.
- Quando il cambio avviene tra due utensili di posto diverso, viene prima ricercato e montato l'utensile desiderato e quindi l'utensile precedente viene depositato in catena nel suo posto dedicato. Ogni cambio utensile comporta quindi una **doppia ricerca**: la prima per cercare l'utensile nuovo ed la seconda per cercare il posto dedicato a quello che era attivo, per depositarlo.
- Quando viene programmato T0, non viene fatta nessuna ricerca. Con il successivo M06, il TCI comanda la ricerca del posto dedicato all'utensile attivo; comanda poi un cambio ed un deposito dell'utensile sul magazzino, lasciando vuoto il posto attivo e il posto intermedio.
- Per **NEWTCI** e **FATTCI** valgono le note date nelle tabelle precedenti.
- Programmando una nuova ricerca (Tyy dopo una Txx) senza un cambio (M06), il TCI comanda il DEPUTE (se UTI=1 e quindi era stato eseguito il PREUTE), la nuova ricerca MOVCU ed infine il nuovo PREUTE (sempre se UTI=1).

Sono possibili anche altri tipi di cambi utensili, ad esempio:

CICLICO (TYP='C' ) - SENZA Ut. Intermedio (UTI=0)

CICLICO (TYP='C') - CON Ut. Intermedio (UTI=1 o 2)

ma non ne vengono date le relative tabelle in quanto molto rari; è comunque possibile ricavarne la logica di funzionamento facendo riferimento a quelle date, considerando che un C.Ut CICLICO non ammette due ricerche successive e quindi il TCI aggiunge un ulteriore cambio per realizzare la sequenza RICERCA - CAMBIO - RICERCA - CAMBIO.



---

---

## 9.16 RICERCA UTENSILE NEL MAGAZZINO

Nei paragrafi precedenti si è parlato ampiamente della gestione del magazzino utensili, delle varie tipologie e dei comandi previsti per la ricerca di un posto "adatto" nelle varie fasi di carico/scarico, ricerca, prelievo, deposito, . . .

La realizzazione dei movimenti non è stata approfondita in quanto dipende dalla "meccanica" di costruzione del magazzino stesso.

Uno caso particolare è quello di una ruota o di una catena, con posti utensile equidistanti, mossa da un azionamento del tipo di quelli usati per gli assi macchina. In questo caso è comodo usare un asse PLC e la routine **movmag()** presente nella libreria PLC.

### 9.16.1 Routine "MOV MAG"

Chiamata: **movmag()**;

Scopo: realizza la movimentazione dell'asse/catena utensili gestito come asse PLC; la routine gestisce assi lineari e toni (con ricerca per la via più breve).

Parametri in Ingresso :

**ORDMAG** (unsigned short) Numero d'ordine dell'asse magazzino. In base al tipo di CNC sono accettati valori compresi tra 0 e 15.

**FEDMAG** (double) Velocità di movimento del magazzino in mm/min (in caso di assi lineari) o in gradi/min (in caso di assi toni). Il CNC muove l'asse tenendo conto anche del valore scritto nell'override asse indipendente OVRFREx (default OVRFREx = 100).

**PASMAG** (double) Passo del magazzino in mm (in caso di assi lineari) o in gradi (in caso di assi toni).

Segnali in Uscita:

**ITCMPOS = 1** : Magazzino in posizione.

**FATRCMU = 1** : Ricerca eseguita. Con questo segnale il PLC può dare Fatto del Nuovo comando da TCI (**FATTCI**)

Note:

La routine usa la funzione `AxMovAbsWork(ORDMAG, quota_mag, FEDMAG)` calcolando la quota in base a **PASMAG** ed al posto desiderato (presente su **OTCDATIH** con il comando MOVCU del TCI).

L'asse deve essere assegnato in gestione al PLC (caratterizzato dunque dal relativo bit **OAXPLC#=1**) e quindi deve avere il SET fatto ed essere attiva la richiesta **IAXPLC#**. Per le particolarità di funzionamento di un asse PLC si rimanda al relativo **CAPITOLO 4**.

La routine usa il Sequenziatore **SEQ30** che non è più quindi libero per altri usi.

La routine segnala con il messaggio **MECS09** (Messaggio PLC 19409) eventuali errori di chiamata.

Il segnale **ITCMPOS** è descritto al paragrafo SEGNALI LOGICI di questo capitolo.

Il segnale **FATRCMU** può essere ritardato rispetto all'effettivo completamento del movimento dell'asse. Per far questo occorre caricare, con la funzione **lotime(...)**, il ritardo desiderato nel temporizzatore 94 appositamente previsto:

```
lotime(94,xx);
```

dove: 94 = numero del temporizzatore

xx = tempo di ritardo espresso in secondi (ad es. = 0.3 sec.)



---

**INDICE ANALITICO**

# 294 = giri/min. di cambio gamma .....	5-15	AKAGG .....	9-24
# 295 = delta quota oscillazione .....	5-16	AKNCU .....	9-23
# 296 = giri/min. di cambio gamma .....	5-16	ALIMDRYR .....	1-5
# 297 = delta quota oscillazione .....	5-16	ALIMFEED .....	1-5
#252 .....	5-16	ALLARITA.MU .....	6-4
#278 .....	5-16	allarme 31403 .....	9-17
#292 = cambio gamma desiderato .....	5-16	Allarme CNC .....	6-21
#293 = cambio gamma desiderato .....	5-15	ALLARMI .....	6-2
<ACC: . . . > .....	3-33	ALRnn .....	6-3
<AGC: . . . > .....	3-19	ANALOG .....	7-11
<AMP: . . . > .....	3-8	ANTISLITTAMENTO .....	3-59
<BLC:C> .....	5-21	AOVR .....	2-37
<BLC:sigla_asse> .....	2-4; 2-13	AOVSS .....	2-34; 2-37
<CGA: . . . > .....	3-18	APPARATO FRESA/TORNIO .....	1-22
<DMP: . . . > .....	3-8	APVOL .....	2-29
<FED: > .....	4-2	APVOL3 .....	2-29
<FRE: . . . > .....	4-1	APVOL4 .....	2-29
<GAN:OFF; . . . > .....	3-18	AREE U: Accesso .....	6-27
<GAN:ON; . . . > .....	3-17	ASKQMENU .....	7-7
<PND: . . . > .....	3-7	ASS_IPOCOR .....	3-14
<SAX: . . . > .....	3-12	Asse agganciato .....	2-6
<SBL:sigla_asse> .....	2-13	Asse agganciato in contouring .....	2-6
<TAC: . . . > .....	3-33	Asse agganciato in velocità .....	2-6
<TRU<STP>: . . . > .....	3-54	asse bloccabile meccanicamente .....	2-4
AALRnn .....	6-3	asse bloccato meccanicamente .....	2-4
AATTnn .....	6-3	Asse CNC .....	2-2
AB_SHIFT .....	7-3	asse di Proprietà del PLC .....	4-2
ABIKRCAN .....	2-29	ASSE DUMMY .....	3-12
ABIL .....	2-11	Asse entrato in soglia .....	2-6
ABILIKR .....	2-29	Asse fermo .....	2-6
ABLAVMF .....	1-19; 2-7	asse Gestito dal PLC .....	4-2
ABMND .....	5-10	asse in emergenza .....	2-3
ABMND5 .....	5-10	Asse in emergenza .....	2-3
ACG .....	5-10	Asse in moto .....	2-6
ACGS .....	5-10	Asse non arrivato .....	2-6
Aciclico .....	9-52	Asse OFF .....	2-5
Aciclico catena .....	9-56	ASSE OFF .....	2-21
ACKFIN .....	9-23	ASSE PENDOLANTE .....	3-7
ACKIKR .....	7-11	ASSE STEP .....	3-12
ACTALAT .....	6-3	Asse-mandrino senza trasduttore .....	5-19
ACTQMENU .....	7-7	ASSI .....	2-1
addint( ) .....	6-4	Assi Assoluti .....	2-18
addlong( ) .....	6-4	Assi CNC .....	3-42
addstr( ) .....	6-4	Assi CNC fermi .....	2-7
AGGANCIO ASSI .....	2-8	Assi CNC in movimento di lavoro .....	2-7
aggax( ) .....	2-10	assi digitali .....	2-40
aggmnd( ) .....	1-20	ASSI GANTRY .....	3-17
aggmnd( ) .....	5-9; 5-19	ASSI in HOLD .....	1-3
AggMndCnc( ) .....	5-26	Assi Indipendenti .....	3-42
AggMndFrv( ) .....	5-26	ASSI INDIPENDENTI .....	4-1
aggvol( ) .....	2-29	ASSI INTERPOLANTI (CNC) .....	2-2
AIOVF .....	2-33	Assi Sercos .....	2-18
AIOVR .....	2-33	ASSI VIRTUALI .....	3-57
AIOVS .....	2-33; 5-5	Assi: Funzioni Speciali .....	3-1



Assi: Modifica Accelerazione.....	3-33; 3-49	Cambio o Ricerca in Corso.....	9-16
assirtrc ( ).....	3-30	Cambio Utensile in Corso.....	9-16
ATR=Y.....	2-18	Cambio Utensile Manuale.....	9-15
Attenzione su asse.....	2-3	CAMBIO UTENSILI.....	9-1
ATTENZIONI.....	6-2	Cambio utensili nel Tornio.....	9-49
ATTESA MANDRINO FERMO.....	5-29	CAN-Bus.....	2-30
ATTnn.....	6-3	carico e scarico utensili dal magazzino.....	9-31
autoapprendimento.....	7-14	Carico utensile.....	9-32
AUTOAPPRENDIMENTO.....	6-29	CARICO/SCARICO UTENSILE.....	9-28
AUTOMATICO.....	1-2; 2-9	CARUCS.....	9-28
av11_drf().....	2-25; 2-26	CERC.....	1-19
AX_TP_MASK.....	3-54	CERDEA.....	9-29
AxAcc ( ).....	3-34	CERPOS.....	9-29
AXDUMMY.....	3-14	CERPOV.....	9-29
AxFeed ( ).....	4-14	CERTPC.....	9-29
AxJogNVelRap ( ).....	4-13	CGAMMA.....	1-20; 5-16
AxJogNVelWork ( ).....	4-13	Cicli di misura.....	3-54
AxJogPVelRap ( ).....	4-13	Ciclico.....	9-52
AxJogPVelWork ( ).....	4-13	CICLICO.....	9-17
AxJogSpRap ( ).....	4-13	CK1SEC.....	1-5; 6-1
AxJogSpWork ( ).....	4-13	KCN.....	7-3
AxMov ( ).....	4-10	CLASSI DI F.A.....	1-9
AxMovAbsRap ( ).....	4-8	Classificazione CUA Tornio.....	9-51
AxMovAbsWork.....	9-60	CLOCK 1 SECONDO.....	1-5
AxMovAbsWork ( ).....	4-8	CNC in ALLARME.....	1-4
AxMovInc ( ).....	4-11	CNC in HOLD.....	1-4; 5-4
AxMovIncRap ( ).....	4-8	cnc_all ( ).....	6-21
AxMovIncWork ( ).....	4-8	CNOK.....	7-3
AxMovRound ( ).....	4-9	CNREL.....	1-24
AxMovRoundDec ( ).....	4-9	CNTCOK.....	9-16
AxMovRoundInc ( ).....	4-9	CODICE.....	7-3
AxMovRoundNoZero ( ).....	4-9	CODICR.....	2-30
AxPlcAn ( ).....	4-15	CODIKR.....	2-30
AxPlcModeSetOff ( ).....	4-15	CODITAS.....	2-30
AxPlcModeSetOn ( ).....	4-15	CODTAS.....	7-2
AxsFnzAxmOff ( ).....	3-46	COMANDI MANUALI DA PLC.....	6-28
AxsFnzAxmOn ( ).....	3-45	comando da PLC a TCI.....	9-20
AxSynch ( ).....	3-46	comando da TCI a PLC.....	9-19
azionamenti digitali.....	2-40	COMAXS.....	2-7; 2-12
BANDA.....	1-19	COMAZ.....	6-1
BLCAX.....	2-4; 2-11; 4-3	COMOVR.....	2-37
BLCMND.....	5-9	COMOVSS.....	2-34; 2-37
BLCMNS.....	5-9	COMPENSAZIONE DINAMICA USURA UTENSILE.....	3-62
Blocca Cambio Utensile.....	9-16	COMPENSAZIONE TERMICA.....	3-1; 3-68
Blocca T.....	9-16	compensazione termica assi da PLC.....	3-4
BLOCCAGGIO ASSI.....	2-13	COMPILAZIONE CONDIZIONALE.....	1-24
BLOCCO.....	1-3	Condizioni iniziali.....	1-2; 1-6
BLOCCO ASSI.....	1-4	Console Remota.....	7-12
BLOCCO CNC.....	1-4; 5-4	Contouring.....	2-2
Blocco Movimentazione Magazzino.....	9-15	CONVERTITORI A/D E D/A.....	6-9
BLOCK RETRACE.....	3-29; 3-51	Convertitori A/D: Bufferizzazione.....	6-12
BOOT.....	6-6	convertitori Digitali/Analogici (D/A).....	2-1
bufferizzazione convertitori A/D.....	6-13	COPIA.....	3-9
C.Ut. a catena aciclico.....	9-58	copia().....	3-11
C.Ut. a CATENA ACICLICO.....	9-38	COR.TAR.....	3-64
C.Ut. a CATENA CICLICO.....	9-46	coremecs().....	7-13
cambio gamma mandrino.....	5-14		



## CAMBIO UTENSILI

---

correttori .....	9-1; 9-4	ECSLINK.....	2-40
Correzione tabellare errore assi.....	3-64	EDD .....	2-40
COSTUSER.....	2-4; 2-43; 3-11; 4-6	ENCABSL .....	2-17
CR_ACTIV.....	2-29; 2-34	ENCABST .....	2-17
CR_CODCO.....	7-14	ESECUZIONE .....	1-3
CR_FATTO.....	7-14	eseguibile esterno.....	6-29
CR_JOYON.....	7-15	Eventi del CNC .....	6-5
CR_PLANE.....	7-15	EVENTI ESTERNI .....	3-53
CR_RTCP.....	7-15	ExcludeAxTravelLimit ( ).....	3-52
CR_START.....	7-14	F.A. D.....	9-15
CR_STORE.....	7-15	F.A. ESTERNE .....	1-8
CR_UHAL.....	7-15	F.A. SUBROUTINE.....	1-9; 1-16
CR_VAX.....	7-15	F.A. T .....	9-15
CR_VIEW0.....	7-15	fasertrc ( ).....	3-30
CR_VVOL1.....	2-24; 7-14	FAST.....	VIII; 1-2; 1-5
CUA.....	9-1	FASUB .....	1-16
CUA aciclico.....	9-14; 9-52	FATFA2.....	1-12
CUA ciclico.....	9-14; 9-52	FATFA3.....	1-12
CUA da PLC.....	9-2; 9-49	FATRMCU .....	9-60
CUA da Subroutine .....	9-2; 9-21	FATSAT .....	1-12
CUA.SUB.....	9-21	FATSMN .....	1-13; 5-5
CUAT.....	9-22	FATTCI .....	9-19; 9-27
CUM.SUB .....	9-21	FATTOD.....	1-12; 9-15
CUMAG.TAB .....	9-1; 9-49	FATTOM.....	1-11
CUMAN .....	9-27	FATTOS.....	1-12; 5-5
CUMANS .....	9-22	FATTOT .....	1-12; 9-15
CUSTOM.....	7-6	FEDMAG.....	9-60
CUTE.....	9-27	Feed Assi.....	3-36
CUTES .....	9-22	Feed di un Asse PLC.....	4-14
cuttrrc ( ) .....	3-31	FEED HOLD .....	2-11
Dati da PLC a TCI .....	9-20	FeedAllAxs ( ) .....	3-58
Dati da TCI a PLC .....	9-19; 9-21	Feed-hold assi CNC .....	2-3
DEF001 .....	2-29	Filettatura.....	5-24
DELTA_AXDUMMY .....	3-15	FILTRO .....	2-34; 2-37
DEPUCS.....	9-27; 9-28	FILTROF.....	2-37
DEPUTE .....	9-27	FILTROR.....	2-37
DERN_nn .....	1-21	FILTROS.....	2-37
DERP_nn.....	1-21	FILTROSS .....	2-34
DF_SYN.INC.....	9-25	FINECORSA.....	2-21
DGABIL .....	2-41	FINESTRE VIDEO PERSONALIZZATE.....	8-1
DGDROK.....	2-41	FL_IPOCOR .....	3-15
DGMICR.....	2-41	FLGMEN .....	7-7
DGMON.....	2-41	FLGMENW .....	7-7
DGPOW.....	2-41	FMDRYRUN .....	1-5
DGSETP.....	2-41	formato attuale del CNC .....	8-6
DGVERS .....	2-41	FUNZIONI AUSILIARIE .....	1-1; 1-8
dialogo tra TCI e PLC.....	9-27	G184 .....	1-19
Digitalizzazione.....	3-9	G201 .....	1-19
dimensione dell'utensile .....	9-4	G309 .....	9-49
DRESET .....	2-42; 2-43	G345 .....	5-28
DRF .....	2-23	G380 .....	3-65
Drive ON.....	2-2	G630 .....	1-22
DRIVETOP .....	2-43	G633 .....	5-2
DRY RUN .....	1-5	G79x .....	9-4; 9-51
due Tastiere.....	7-12	G800-G900 .....	4-10
e_u_wrw( ).....	7-7	G84 .....	1-18
ECSCLOG.TXT .....	6-5	G84ENT .....	1-18

---



G85 .....	1-18	IFMDRF.....	2-24; 2-32
G86 .....	1-19	IGM1 .....	5-6
G87 .....	1-19	IGMn .....	5-7
GAMMA .....	5-14	IHLD .....	2-8; 4-4
gamma mandrino.....	5-6	IJOGN .....	2-7; 4-5; 4-6
GANTRY .....	3-17	IJOGP .....	2-7; 4-4; 4-6
Gantry Multislave .....	3-17	IJSPL.....	2-7
Gantry: assi assoluti .....	3-25	IJSPR .....	2-7
Gantry: Digitalizzazione .....	3-22	IJVLL.....	2-8
Gantry: Homing assi Sercos incrementali .	3-25	IJVLR .....	2-7
Gantry: Presa di zero.....	3-22	IKR .....	7-10
Gantry: riaggancio degli assi .....	3-28	ILIPx .....	9-25
gestovr().....	2-37	ILS .....	2-4; 2-19
GestOvr3() .....	2-34	ILSAZ .....	6-1
gestovrc() .....	2-39	IMNDATT .....	1-13; 5-5; 5-7
gestwplc() .....	8-1; 8-2	IMPOS.....	9-16
get_CorrFileCode() .....	3-65	INAL .....	2-5
get_CorrFileNumber ().....	3-64	Indramat .....	3-25; 3-54
get_qr_pintl ().....	3-5	INIMAG .....	9-28; 9-29
get_T_Axis_Override().....	3-62	init_T_Axis() .....	3-61
GetExecutionTime ().....	6-15	iniz_drf() .....	2-32
GetPPSel () .....	6-15	iniz_drf() .....	2-25
GetSerialNumber().....	6-39	INJS .....	2-8; 2-13
HLDFRE .....	2-8; 4-4	INPOUT.....	7-10
hldfre () .....	4-7	INTC.....	9-20
hldfre() .....	3-12	IOPLC_nread_tt().....	9-5
HOLD .....	1-4	IOPLCBlockPPSel () .....	6-15
I/O su tastiera .....	7-9	IOPLCGet_nread_tt() .....	9-6
IAKDAT .....	9-19	IOPLCGetRecord() .....	6-38
IANA.....	4-3	IOPLCReadCNCLanguage () .....	6-22
IAPPATT .....	1-13; 1-23	IOPLCReadFile().....	6-37
IAXOFF .....	2-5	IOPLCRET . 6-2; 6-9; 6-16; 6-22; 6-33; 6-38; 8-7; 9-6	
IAXPLC .....	2-7; 4-4; 4-5	IOPLCRunFileExe () .....	6-29
IBICU .....	9-16	IOPLCSelectGraphPP() .....	6-16
IBL.....	1-4; 2-3; 5-4	IOPLCSendSMS .....	6-8
IBLAX.....	1-4; 2-3	IOPLCShutDown () .....	6-22
IBLAXMND .....	1-4; 2-3; 5-4	IOPLCSTS . 6-1; 6-9; 6-15; 6-16; 6-22; 6-29; 6-33; 6-38; 8-7; 9-5	
IBLC .....	2-4; 4-3	IOPLCWinvisActualForm().....	8-7
IBLDRF .....	2-24	IOPLCWriteFile().....	6-32
IBSET .....	2-5	IOSTPB .....	3-53
IBSET# .....	2-17	IOVF .....	2-33
IBT .....	9-16	IOVR .....	2-33
ICORAX .....	3-2	IOVS.....	2-33; 5-5
ICRCU .....	9-16	IOVSS .....	2-33; 5-5
IDD.....	2-40	IPOCOR .....	3-14
idddrive().....	2-43	IRANN.....	9-21
IDGOFF .....	2-42	IRAP.....	2-3
IDGTAR .....	2-43	IRES.....	1-4
IDRF12 .....	2-23	IRME .....	5-6
IDRF3 .....	2-24	IRTR.....	3-67
IDRVON.....	2-2	IRUR .....	5-7
IDTC.....	9-28	is_cmd_corr () .....	1-14; 3-4
IDTC1D.....	9-20	is_cnc_work ().....	6-21
IDTCD.....	9-20	is_cnc_work() .....	2-25; 2-26
IEME .....	2-3	is_corr_ext () .....	3-3
IFEEDHLD .....	2-3		
IFHD.....	2-3		





## CAMBIO UTENSILI

---

ISAJ.....	2-7; 2-23	M11.....	2-16
ISAJ3.....	2-23	M19: Stop mandrino orientato.....	5-18
ISAM.....	5-7	M40-M45.....	1-20
IsAxPlcModeSet ( ).....	4-15	MacroMenuDev.....	7-1
ISGA.....	2-3	MAGAZZINO.....	9-29
IsMTXOn().....	6-16	Magazzino in posizione.....	9-16
ISRME.....	5-7	MANDRINO.....	5-1
ISRUR.....	5-7	Mandrino "esterno" in rotazione.....	5-6
ISST.....	3-9	MANDRINO ASSE DEL CNC.....	5-8
ISSTCPY.....	3-9	Mandrino come asse tondo.....	5-23
ISTEP.....	3-10	MANDRINO e matrici dinamiche.....	5-27
ISTEPCPY.....	3-10	Mandrino gestito dal PLC.....	5-23
IsVaxOn ( ).....	3-57	MANDRINO in HOLD.....	1-3; 5-4
ITCAKDAT.....	9-19	Mandrino in velocità.....	5-17
ITCBICU.....	9-16	Mandrino Monogamma.....	5-15
ITCBT.....	9-16	Mandrino pilotato in spazio.....	5-21
ITCCRCU.....	9-16	mandrino principale.....	5-3
ITCMPOS.....	9-16; 9-60	mandrino secondario.....	4-6
ITCOD.....	9-20; 9-28	Mandrino: abilitazione azionamento.....	5-19
ITCRANN.....	9-21; 9-34	MANDRINO: Allarmi.....	5-25
ITCSTTCI.....	9-20; 9-27; 9-28	Mandrino: bloccaggio asse.....	5-21
ITF02.....	3-10	Mandrino: gestione cambio scala.....	5-12
ITFnn.....	3-53	mandrino: gestione gamme.....	5-14
ITSTFE.....	2-23	MANUALE.....	1-2
IVFM.....	2-23	marker del trasduttore ottico.....	2-5
IVJ.....	2-8	MAXOVF.....	2-34; 2-37; 2-39
IVJ1.....	2-23	MAXOVR.....	2-34; 2-37
IVJ2.....	2-23	MAXOVS.....	2-34; 2-37
IVJ3.....	2-23	MCREVNT.....	3-55
IVOLF.....	2-23	MDI.....	1-2; 2-9; 6-28
JOG.....	2-7; 2-8	Mechatrolink.....	2-40; 2-42; 3-54; 3-55; 3-56
JOGNAZ.....	6-1	MECS07.....	6-27
JOGPAZ.....	6-1	MECS10.....	2-43
Joystick.....	7-14	MECS11.....	2-44
KCNINP.....	7-10	menu PLC attivo.....	7-7
KCNINP00.....	7-12	menu PLC: bloccaggio.....	7-7
KCNOUT.....	7-10	MESSAGGI PARAMETRIZZATI.....	6-2
KEY.....	5-24; 6-17	Messaggi tramite SMS.....	6-8
KEY#M68.....	7-11	Micro di Evento.....	3-54
KEY00.....	6-18	micro di zero.....	2-5; 2-17; 2-41
Keyboard Remoto.....	7-10	MILL.....	5-1
KT116M67.....	7-11	MINOVS.....	2-34; 2-37
LAMP.....	7-11	MISCELLANEA.....	6-1
LATHE.....	5-1	MODE_AUT.....	1-2
leds.....	7-11; 8-4	MODE_CNT.....	1-3
LETTURA.....	1-3	MODE_CPY.....	3-10
Libera Servo.....	2-4	MODE_EXE.....	1-3
LIBERA SERVO.....	2-19	MODE_FND.....	1-2
Lingua attuale del CNC.....	6-22	MODE_FST.....	1-2
Logo personalizzato.....	8-6	MODE_JOG.....	1-1
lotime ( ).....	1-21	MODE_MDI.....	1-2
M01.....	1-3	MODE_RPZ.....	1-2
M03: Rotazione oraria.....	5-18	MODE_SET.....	1-1
M05.....	5-29	modem.....	6-8
M05: Stop mandrino.....	5-18	modijog().....	2-12
M06.....	9-1	MOVCS.....	9-27; 9-28
M10.....	2-16	MOVCU.....	9-27

---



movimenti di lavoro a mandrino fermo .....	2-7	OCOK.....	1-2; 1-6
movmag().....	9-60	OCOP.....	3-10
NAXJOG .....	7-5	OCPLC.....	2-2; 2-7; 4-5; 4-6
NBLK_AXDUMMY .....	3-14	OCQIC .....	9-21
NCTS .....	7-2	OCUC.....	9-16
NEWD .....	1-12; 9-15	OCURR.....	2-43
NEWFA2.....	1-11	ODATI .....	9-19
NEWFA3.....	1-12	ODATQ .....	9-21
NEWGEN.....	1-13	ODATQ1D.....	9-21
NEWM .....	1-11	ODEA.....	9-16
NEWS .....	1-12; 5-5	ODGMON.....	2-42
NEWSAT .....	1-12; 1-23	ODRF.....	2-24
NEWSMN .....	1-13	OESCOM .....	9-20
NEWT .....	1-12; 9-15	OESE .....	1-2
NEWTCI.....	9-19; 9-27	OEXE .....	1-3
NORESET .....	1-4; 1-7; 6-1; 9-34	OFA2.....	1-11
numero di commessa .....	6-39	OFA3.....	1-12
Nuovo Tool .....	9-15	OFAS T .....	1-2
OALL.....	1-4	off_refmnd().....	5-13
OAOK .....	1-2	OffMfrM05( ).....	5-29
OAPP .....	6-30	OFIND.....	1-2
OASF .....	2-7	OG33.....	5-6; 5-24
OATCK .....	9-20	OG37.....	1-5
OAUTO.....	1-2	OHLAX.....	1-3
OAUTTC.....	9-15	OHLD .....	1-4; 5-4
OAXOFF.....	2-5	OHLMND.....	1-3; 5-4
OAXPLC .....	2-7; 4-4	OIDSLV .....	3-45
OAXRTR.....	3-67	OJNG .....	1-1
OAXSLV .....	3-45	OJOG .....	1-1
OBAG .....	2-6	OJPS.....	1-1
OBALAX .....	2-3	OKEYAS .....	6-18
OBAT .....	2-3	OKEYIM .....	6-18
OBCG .....	5-6	OKEYSN .....	6-18
OBCN .....	2-6; 4-4	OLAV.....	2-7
OBGG .....	3-20	OLIPx .....	9-25
OBGM.....	3-20	OLS .....	2-4
OBGSET.....	3-20	OM .....	1-11
OBIN .....	2-6	OM01 .....	1-3
OBLC .....	2-4; 2-11	OMAN .....	1-2
OBM0.....	2-5	OMCU .....	9-15; 9-23
OBM1.....	2-5	OMDG.....	3-10
OBMM.....	9-15	OMDGCOPY.....	3-10
OBMT.....	2-6	OMPUNTA.....	6-30
OBNA.....	2-6; 4-4	on_refmnd( ).....	5-12
OBNT.....	4-3	ONBAGM .....	2-6
OBOK .....	2-3	ONDAT.....	9-19
OBSE.....	2-5	OnMfrM05( ).....	5-29
OBSE#.....	2-17	OP_TWT .....	6-24
OBSET.....	2-5	OPARTR .....	3-67
OBVV .....	2-6	OPE_WIN.....	8-1
OCGEN.....	1-13	open_CorrFile ( ).....	3-64
OCNT.....	1-3	OPFRM .....	2-6
OCODFA2 .....	1-11	OPND.....	3-7
OCODFA3 .....	1-12	OPNT .....	9-15; 9-23
OCODM1.....	1-11	OPQ12 .....	3-7
OCODSAT .....	1-12; 1-23	OPSSL.....	2-6
OCODSMN .....	1-13; 5-2; 5-5	OPVQ1.....	3-7



---

OPZIONI SOFTWARE .....	6-24	Parametri di taratura degli assi .....	2-1
OQCOD .....	9-19	Parametri di taratura del mandrino .....	5-3
ORDMAG .....	9-60	Parametro Generico .....	1-14
ORFC .....	9-16	PARPLCnn .....	8-1
Orientamento mandrino.....	5-22	Part-Program .....	6-14
ORIGINI.....	2-8	PASEXT.....	7-3
ORIPOK.....	1-5	PASMAG.....	9-60
ORPZ.....	1-2	PIEGATUBI.....	3-59
ORTP.....	9-15; 9-23	plclusr ( ) .....	3-63
ORTR .....	3-67	plcrtrc ( ).....	3-29
OSAMAC .....	5-6	plcrtrcf ( ).....	3-30
OSET .....	1-1	plcrusr ( ).....	3-62
OSKP.....	1-3	pon .....	6-2
OSOK .....	5-6; 5-10	posti dedicati.....	9-3; 9-49
OSOKS.....	5-10	Posti dedicati .....	9-36
OSQSLV.....	3-45	POSTI DEDICATI .....	9-50
OSTM .....	5-6	posto utensile.....	9-7; 9-8
OSTM# .....	5-24	power-on.....	6-2
OSTO .....	3-10	PREUCS.....	9-27; 9-28
OSTOREO .....	6-30	PREUTE .....	9-27
OSYSMILL.....	1-5; 1-6; 1-22; 5-1	PRIIRTR.....	3-66
OSYSTURN.....	1-5; 1-6; 1-22; 5-1	PROBE ON/OFF.....	3-53
OTAWAR.....	9-16	PROBE1 .....	3-54
OTCAKTCI .....	9-20; 9-27; 9-28	PROVA PROGRAMMA .....	VIII; 1-2; 1-5
OTCAUT.....	9-15	PSET.....	1-18
OTCBMM.....	9-15; 9-34	PUCNC .....	7-2
OTCCQIC .....	9-21	PUECS.....	7-3
OTCCUAT .....	9-22	PUSER.....	7-3
OTCCUC .....	9-16	qalcmd ( ).....	6-29
OTCCUTER.....	9-17	QR_IPOCOR .....	3-15
OTCDATI.....	9-19; 9-27	qvax ( ).....	3-4
OTCDATQ .....	9-28	r_man() .....	6-39
OTCDATQD .....	9-21	RABIL.....	2-10
OTCDEA.....	9-16	RABMND .....	5-9
OTCESCOM.....	9-20; 9-28	RABMNDS.....	5-9
OTCMCU.....	9-15; 9-23	Random .....	9-3
OTCPNT.....	9-15; 9-23	Random fisso.....	9-4
OTCQCOD .....	9-19; 9-27	rd_opz ( ).....	6-24
OTCRFC.....	9-16	RDCUST .....	9-29
OTCRTP.....	9-15; 9-23	read_buf_ad ( ) .....	6-13
OTCSECU.....	9-17	read_var ( ) .....	6-25
OTCSTDAT .....	9-19	RESET .....	1-4; 1-6; 9-34
OTCWAR.....	9-16	RESET CNC .....	1-4
OTDS .....	1-12; 5-5; 9-15	Resource Builder .....	8-5
OTDSH .....	1-12	RETG84.....	7-3
OTDSL.....	1-12	RETRACE.....	3-29
OTWT .....	1-3	Retract Assi .....	VIII; 3-65
OVER .....	1-19	revolver .....	9-49
ovF_set() .....	2-35	Revolver.....	9-51; 9-54
ovF_setc().....	2-39	RICDRF .....	2-30
OVGEN.....	1-13	RICERCA.....	1-2
OVL .....	2-23	Ricerca Fisica in Corso.....	9-16
ovR_set().....	2-35	Ricerca Tool.....	9-23
OVRFRE.....	2-34; 4-5; 4-6	Ricerca Tool Prenotata.....	9-15
ovS_set() .....	2-35	RICERCA UTENSILE NEL MAGAZZINO .	9-60
OWAR .....	9-16	RICVFED .....	2-30
PARAMETRI CUSTOM.....	9-29	RICVOL.....	2-30

---



RIGABS .....	2-17	SETGAN .....	3-23
RILS .....	2-4; 2-11	SetLimFeedOff ( ) .....	3-37; 3-40
RIMPV1 .....	2-29	SetLimFeedOn ( ) .....	1-5
Riposizionamento .....	3-58	SetLimFeedOn ( ) .....	3-37
RIPOSIZIONAMENTO.....	1-2; 2-9	SetLimSpeed ( ) .....	3-40
Riposizionamento delle F.A.....	1-15	SetMovMnd( ).....	5-11
ripristina modo manuale .....	6-39	SetSpeed ( ) .....	3-41
RIPUTM .....	9-29	SetTarSpeed ( ).....	3-40
RISAMP .....	5-7; 5-9	shut-down.....	6-22
RISAMS .....	5-7; 5-9	SINCRONIZZAZIONE ASSI .....	3-44
RISST .....	3-9; 3-12	SINCRONIZZAZIONE DELLE F.A.....	1-15
RITN_nn .....	1-21	Sincronizzazioni CNC PLC .....	6-23
RITP_nn.....	1-21	Sinonimi di stato del CNC.....	1-1
RKEYAS .....	6-18	Sinonimi miscellanei .....	6-1
RKEYIM .....	6-17	Sinonimi per i temporizzatori.....	1-21
RKEYSN .....	6-17	Sinonimi per le F.A.....	1-11
ROMBO .....	1-19	Sinonimi utilizzati dal PLC per eventi esterni.3-53	
Rotational Tool Center Point .....	3-6	Sinonimi utilizzati per gli ASSI .....	2-2
RSTEP .....	3-10; 3-12	Sinonimi utilizzati per gli assi digitali .....	2-40
RSTMUON.....	7-3	Sinonimi utilizzati per gli Assi Gantry .....	3-19
RTCP .....	3-6; 7-15	Sinonimi utilizzati per gli assi indipendenti...4-3	
rw_var ( ).....	6-25	Sinonimi utilizzati per gli OVERRIDE.....	2-33
s_man().....	6-39	Sinonimi utilizzati per il mandrino.....	5-4
salva modo manuale.....	6-39	Sinonimi utilizzati per la compensazione da PLC.....	3-1; 3-69
SavAxAccTar ( ).....	3-34	Sinonimi utilizzati per la funzione COPIA ....	3-9
scalusr ( ) .....	3-63	Sinonimi utilizzati per la funzione SINCRONIZZAZIONE ASSI.....	3-45
Scarico utensile .....	9-33	Sinonimi utilizzati per la gestione FINESTRE VIDEO .....	8-1
scarusr ( ).....	3-63	Sinonimi utilizzati per la gestione IKR.....	7-10
SCAUCS.....	9-28	Sinonimi utilizzati per la gestione tastiera....	7-2
SECFA.....	1-16	Sinonimi utilizzati per l'autoapprendimento	6-30
SECONDA E TERZA F.A. ....	1-16	SKIP .....	1-3
SEGNALI DI STATO.....	1-1	SMS .....	6-8
sel_pp ( ).....	6-14	SPEED .....	5-16
selezione Part-Program.....	6-14	Speed Mandrino.....	3-40
Senso di rotazione del Mandrino "esterno".	5-7	STAGG.....	9-24
SEQ28 .....	2-13; 2-30	start_buf_ad ( ) .....	6-13
SEQ30 .....	9-60	start_corr_ext ( ).....	3-2
Sercos.....	2-42; 3-54	Stato del CNC .....	6-21
SERCOS.....	2-40; 3-56	stato utensile.....	9-8
SET .....	1-17; 2-9	STBCOD .....	7-3
SET asse .....	2-5	STBIKR .....	7-10
SET ASSI.....	2-17	STBQMENU .....	7-7
SET automatico di asse-mandrino .....	5-18	STFIN.....	9-23
set_marker ( ) .....	6-30	STNCU.....	9-23
set_text ( ).....	6-30	StoAxValMark ( ).....	6-31
SET0.....	2-17	STOP Mandrino .....	5-24
SETAL .....	2-17	stop_buf_ad ( ).....	6-13
SetAllAxs ( ) .....	3-58	stop_corr_ext ( ).....	3-3
SetAxDefFeed ( ).....	3-36	StoreAxisValue ( ) .....	6-31
SetAxLimFeed ( ).....	3-36	StoreText ( ) .....	6-31
SetAxTarFeed ( ) .....	3-36	Storico Allarmi .....	6-5
SetAxThresold ( ).....	3-50	Strobe/Ack tra TCI e PLC .....	9-24
SetAxTravellLimitOff ( ) .....	3-53	subroutine CUA.....	9-1; 9-49
SetAxTravellLimitOn ( ) .....	3-52		
SetDefFeed ( ) .....	3-36		
SetDefSpeed ( ) .....	3-40		
SetFeed ( ).....	3-37		



## CAMBIO UTENSILI

---

subroutine CUM.....	9-1; 9-49	TstStsMnd() .....	5-27
SVOL1 .....	2-29	TT.BIN.....	9-4; 9-6; 9-51
SWINTAT .....	6-27	TWIST .....	1-3
T_IPOCOR .....	3-14	TYPCU.TAB.....	9-1; 9-49
tabella utensili.....	9-4; 9-51	u_rdc( ).....	6-27
Tabella Utensili.....	9-8	u_rdf( ) .....	6-27
taglienti .....	9-1; 9-49	u_rdl( ).....	6-27
Taratura Azionamenti.....	6-1	u_rdw( ).....	6-27
TARAZ.....	6-1	u_resb( ).....	6-28
tartwt( ) .....	3-6	u_setb( ).....	6-28
TAS2700.....	7-2	u_tstb( ).....	6-28
TASBASIC.....	7-2	u_wrc( ) .....	6-28
tasfuecs( ).....	2-13; 2-31; 7-5	u_wrf( ).....	6-28
tastatore continuo come volantini Feed .....	2-28	u_wrl( ) .....	6-28
tastatore continuo come volantini Feed. ....	2-23	u_wrw( ) .....	6-28
tastcnc( ).....	1-18	Utensile ricercato .....	9-22
tastcnc( ).....	7-4	utensile rotante .....	5-18
Tasti funzione CUSTOM .....	7-6	Utensile Rotante .....	5-1
tasti operativi del CNC.....	7-4	Utensile sul Mandrino .....	9-22
tastiera .....	7-9	utensili in fine vita .....	9-16
TASTIERA.....	7-1	utensili intermedi.....	9-49
TASTIERA PORTATILE.....	7-12	utensili morti.....	9-16
TCACKAGG .....	9-24	V1202.....	9-25
TCACKFIN.....	9-23	V1912.....	1-23
TCACKNCU.....	9-23	V7005.....	1-16
TCCUMAG .....	9-17	VALLIMFD .....	1-5
TCCUMANS .....	9-22	ValueAxDefFeed ( ) .....	3-37
TCCURAS .....	9-17	ValueAxLimFeed ( ) .....	3-38
TCCUREV .....	9-17	ValueAxTarFeed ( ) .....	3-38
TCCUTES.....	9-22	ValueAxTarThreshold ( ).....	3-50
TCI.....	9-1; 9-3; 9-49; 9-50	ValueAxTarTravelLimit ( ).....	3-51
TCSTRAGG.....	9-24	ValueAxTravelLimit ( ) .....	3-52
TCSTRFIN.....	9-23	ValueDefFeed ( ) .....	3-38
TCSTRNCU.....	9-23	ValueDefModeCNC ( ).....	3-39
Teach Pendant .....	7-12	ValueDefSpeed ( ) .....	3-41
TEMPORIZZATORI.....	1-20	ValueFeed ( ) .....	3-39
TESTON .....	1-19	ValueFeedEx ( ).....	3-39
THIFA .....	1-16	ValueLimFeed ( ) .....	3-38
TIME_nn .....	1-21	ValueLimSpeed ( ).....	3-41
TIPTM.....	9-22	ValueModeCNC ( ) .....	3-39
TIPTMEM .....	9-23	ValueModeCNCEX ( ) .....	3-39
TIPTRIC.....	9-23	ValueSpeed ( ) .....	3-42
TM .....	9-22	ValueTarSpeed ( ) .....	3-42
TMEM .....	9-23	Variabili per Copia .....	3-10
Tool Controller.....	9-1; 9-49	VARIABILI: LETTURA/SCRITTURA .....	6-25
Tool Table.....	9-4; 9-51	VaxPlcOff.....	3-58
Tornio .....	9-1	VaxPlcOn ( ) .....	3-58
torretta .....	9-49	VDA.....	2-1
Torretta .....	9-14; 9-51; 9-55	VDA0.....	6-11
TORRETTA .....	9-36	VELCG .....	5-6; 5-10
TOUCH-PROBE .....	3-53	VELCGS .....	5-10
TRIC .....	9-22	Velocità mandrino .....	5-6
tst_out ( ) .....	6-23	VOL#IKR.....	2-24
tst_refmnd( ) .....	5-13	VOL1IKR.....	7-11
TstFrvCnc( ).....	5-27	Volantini .....	2-22; 2-24
TstMfrM05( ) .....	5-30	volantini gestiti in FEED.....	2-23; 2-27
TstMovMnd( ) .....	5-11	volantini gestiti in SPAZIO .....	2-23

---



---

VOLTAS.....	2-29	winplc( ).....	8-2
VVOL .....	2-24	WOOD.....	1-20
warning 31900 .....	9-17	WOPZCN .....	6-24
WDRYRUN .....	1-5	WPLCTST .....	8-1
WIN_ENT.....	8-1	WRCUST .....	9-29
WIN_ESC .....	8-1	write_var ( ) .....	6-25
WIN_OPN .....	8-1	wTestApp().....	6-31
WIN_PLC.....	8-1	wWriteWindow( ).....	8-2
WIN_VIS .....	8-1	Yaskawa.....	2-40
Window PLC .....	8-1	zero_out ( ).....	6-23