

IsoUs – Ultimate Step
Manuale di programmazione G Code

www.promax.it



PROMAX

Motion
&
Control

Le informazioni contenute nel manuale sono solo a scopo informativo e possono subire variazioni senza preavviso e non devono essere intese con alcun impegno da parte di Promax srl. Promax srl non si assume nessuna responsabilità od obblighi per errori o imprecisioni che possono essere riscontrate in questo manuale. Eccetto quanto concesso dalla licenza, nessuna parte di questa pubblicazione può essere riprodotta, memorizzata in un sistema di archiviazione o trasmessa in qualsiasi forma o con qualsiasi mezzo, elettronico, meccanico, di registrazione o altrimenti senza previa autorizzazione di Promax srl. Qualsiasi riferimento a nomi di società e loro prodotti è a scopo puramente dimostrativo e non allude ad alcuna organizzazione reale.

Rev.4.8.7 © Promax srl

1 PRAFAZIONE

Questo manuale spiega la programmazione ISO e le varie funzioni.

Il metodo di programmazione può cambiare a seconda del tipo di PC utilizzato (tastiera o TouchScreen), rimangono comunque inalterati i concetti principali.

Per la gestione dei Programmi ISO fare riferimento al manuale **INTERFACCIA OPERATORE**.

2 REGOLE DI ISOUS

2.1 BLOCCO

Un blocco è composto da una o più funzioni ISO che terminano con il carattere Cr/Lf (Return)

La sequenza di più blocchi compone un programma. Il programma viene poi gestito dall' interfaccia operatore. **I caratteri che compongono il BLOCCO devono essere necessariamente MAIUSCOLI.**

L' editor di Isous effettua un conversione in automatico del carattere in maiuscolo.

G1X100Y100Z10

2.2 NUMERO DI LINEA O NUMERO DI BLOCCO

Il numero di linea o numero di blocco definisce la posizione del blocco all' interno del PROGRAMMA.

Tale posizione risulta utile per alcune funzionalità (ripartenza da blocco, retrace ecc.)

Isous ignora i marcatori di linea Nxx prima del blocco, pertanto il numero di linea reale è quello visualizzato sulla finestra di editor ISO

| ISO CODE | |
|----------|----------------|
| 1 | N10 G1X100Y100 |
| 2 | N30 G4F2 |

I caratteri **N10** e **N30** vengono ignorati da Isous, pertanto la linea reale è quella indicata sulla barra sinistra. In questo caso 1 e 2

2.3 ESECUZIONE DEL PROGRAMMA

L' esecuzione viene effettuata da interfaccia operatore Isous. Il programma viene eseguito e può essere messo in PAUSA o terminato tramite STOP.

2.4 NOMI DEGLI ASSI

ISOUS non fa distinzione particolare con il nome degli ASSI. Non utilizza una convenzione particolare che definisce assi rotativi e lineari. Questi vengono definiti in configurazione. Isous gestisce al massimo 9 ASSI con la seguente nomenclatura:

X,Y,Z,A,B,C,U,V,W

2.5 ORIGINE MACCHINA

Tutti gli assi fanno riferimento ad un origine macchina che viene effettuata all' accensione di questa. L' origine macchina segue una procedura di HOMING che può differenziarsi dalla tipologia di Drives utilizzati.

2.6 ORIGINE PEZZO O ZERO PEZZO

ISOUS gestisce 256 Origini Pezzo diverse. Ogni origine Pezzo definisce un punto di manualmente tramite interfaccia Isous, sia da istruzioni PartProgram. Le origini possono essere SOSPESE, RIPRESE e DISABILITATE

2.7 OFFSET PEZZO

Come per le origini PEZZO Isous gestisce 256 Offset Pezzo. Un Offset Pezzo è in pratica una nuova origine che può essere gestita solamente da PartProgram. Gli offset possono essere SOSPESI, RIPRESI e DISABILITATI.

2.8 TESTE

ISOUS Gestisce 256 Teste di lavorazione diverse per la singola macchina.

Ogni testa viene riferita allo zero macchina e quindi presettata (non occorre utilizzare ORIGINI PEZZO e OFFSET PEZZO per la testa). La testa utilizzata viene richiamata da Partprogram con l' istruzione Tn. Questa automaticamente inserisce gli offset della testa dallo ZERO MACCHINA.

2.9 FUNZIONI MODALI

Si definiscono funzioni MODALI quelle la cui attivazione rimane per tutto il programma in esecuzione e anche per i successivi. G0,G1,G2,G3 ec. Sono funzioni modali e pertanto non è necessario ripeterle fino a che un'altra funzione esclude la precedente. G1 esclude G0,G2 e G3, G2 esclude G0,G1 e G3 ecc.

```
G1X100Y100
X300Y120Z10
X450
```

In questo esempio è stata utilizzata la funzione G1 Modale. Nel blocco 2 e 3 non è stato ripetuta G1 in quanto rimane attiva.

2.10 CODICI RICONOSCIUTI

Quasi tutti i codici e le istruzioni riconosciute vengono evidenziate con un colore. È probabile che in alcuni casi questo non si manifesti. Ciò non significa che l'istruzione non venga riconosciuta.

L'editor di IsoUs quando non riconosce un'istruzione la sottolinea di ROSSO e attiva la finestra di visualizzazione errori.

2.11 IMPOSTAZIONE DEI VALORI NUMERICI

IsoUs gestisce due tipi di valori numerici;

INTERI

DECIMALI

I valori interi vengono utilizzati per quelle grandezze definite (Numero di Loop ecc.) Se al posto di un valore intero viene messo uno decimale IsoUs genera un errore avvertendo l'operatore.

I valori decimali vengono utilizzati per le grandezze REALI (quote assi, raggio ecc.)

IsoUs utilizza come separatore decimale il . (PUNTO) e non la , (VIRGOLA). Viene in questo caso generato un errore di token non riconosciuto.

2.12 COMMENTI AL PROGRAMMA

IsoUs non utilizza la convenzione classica dei CNC per il commento al programma

Il commento viene iniziato con i caratteri “//” e termina con il fine linea

L'editore di IsoUs evidenzia di colore VERDE i commenti.

I commenti possono stare anche a destra di un blocco.

Il commento non viene eseguito nel PartProgram tutto il suo contenuto viene scartato

```
// QUESTO è UN COMMENTO
G1X100Y100 // ANCHE QUESTO è UN COMMENTO
```

2.13 INIZIO DI UN PROGRAMMA

Un PartProgram IsoUs inizia in modo normale o con istruzioni valide o con commenti

Generalmente all'inizio vengono inserite le funzioni M preparatorie della macchina.

2.14 FINE DEL PROGRAMMA

La fine di un PartProgram termina con l'ultima istruzione eseguibile.

Generalmente sono M preparatorie della macchina.

3 ISTRUZIONI ISO US

3.1 ISTRUZIONI G RICONOSCIUTE

| Codici G | Descrizione | TASK1 | TASK2 |
|----------------------------|--|-------|-------|
| G0 | Traslazione in RAPIDO degli ASSI | NO | NO |
| G0.1 | Traslazione in RAPIDO ASSE SINGOLO Con accelerazione programmata | NO | NO |
| G0.2 | Traslazione in RAPIDO ASSI per Asse tangenziale | NO | NO |
| G1 | Interpolazione lineare degli assi alla F programmata | NO | NO |
| G1.1 | Sospende G1-G2-G3 impostando G0 | NO | NO |
| G1.2 | Riprende funzione salvata da G1.1 | NO | NO |
| G2 | Interpolazione circolare ORARIA | NO | NO |
| G3 | Interpolazione circolare ANTIORARIA | NO | NO |
| G4 | Pausa temporizzata | SI | SI |
| G4.1 | Addizionale tempi su Calc Time | NO | NO |
| G10 | Abilita OVERRIDE ASSI esterno su potenziometro | SI | SI |
| G11 | Disabilita OVERRIDE ASSI esterno su potenziometro | SI | SI |
| G17 | Piano di interpolazione su X-Y | NO | NO |
| G18 | Piano di interpolazione su Z-X | NO | NO |
| G18.1 | Piano di interpolazione su Z-X ma non nel pannello Preview | NO | NO |
| G19 | Piano di interpolazione su Y-Z | NO | NO |
| G19.1 | Piano di interpolazione su Y-Z ma non nel pannello Preview | NO | NO |
| G20 | Programmazione Quote in Pollici | NO | NO |
| G21 | Programmazione Quote in Millimetri | NO | NO |
| G22 | Scambia gli assi del piano di interpolazione | NO | NO |
| G23 | Ripristina gli assi del piano di interpolazione | NO | NO |
| G24 | Abilita Specularità orizzontale | NO | NO |
| G25 | Disabilita Specularità orizzontale G24 | NO | NO |
| G26 | Scambia una coppia di assi a piacere | NO | NO |
| G27 | Sospende G26 | NO | NO |
| G28 (1028) | Riprende G26 | NO | NO |
| G30 | Attiva inserimento automatico RACCORDI su spigoli | NO | NO |
| G31 | Sospende G30 | NO | NO |
| G32 | Riprende G30 | NO | NO |
| G33 | Attiva inserimento automatico SMUSSI su spigoli | NO | NO |
| G34 | Sospende G33 | NO | NO |
| G35 | Riprende G33 | NO | NO |
| G36 | Gestione assi rotativi | NO | NO |
| G40 | Annulla compensazione OFFSET UTENSILE | NO | NO |
| G41 | Abilita compensazione a SINISTRA OFFSET UTENSILE | NO | NO |
| G42 | Abilita compensazione a DESTRA OFFSET UTENSILE | NO | NO |

| | | | |
|---|---|----|----|
| G43 | Attiva correzione lunghezza utensile | NO | NO |
| G44 | Disattiva correzione lunghezza utensile G44 | NO | NO |
| G44.1 G44.2 | Sospende riprende correzione Lunghezza Utensile | NO | NO |
| G45 | Abilita correzione zero utensile da Testa T selezionata | NO | NO |
| G46 | Disabilita correzione zero utensile da Testa T selezionata G45 | NO | NO |
| G47 | Modo impegno/disimpegno utensile per combinazione G41 G42 | NO | NO |
| G48 | Definisce asse di profondità | NO | NO |
| G49 | Gestione MILD MODE – Smorzamento Spigoli | NO | NO |
| G50 | Rotazione del piano di lavoro | NO | NO |
| G51 | Sospende rotazione del piano di lavoro G50 | NO | NO |
| G52 | Riprende rotazione del piano di lavoro G50 | NO | NO |
| G53 | Analoga a G98 (per compatibilità su ISO standard) | NO | NO |
| G54 G54.n | Zero pezzo da memoria ASSE X (possono seguire i parametri XYZ ecc) | NO | NO |
| G55 | Zero pezzo da memoria ASSE Y | NO | NO |
| G56 | Zero pezzo da memoria ASSE Z | NO | NO |
| G57 | Zero pezzo da memoria ASSE XY | NO | NO |
| G58 | Zero pezzo da memoria ASSE YZ | NO | NO |
| G59 | Zero pezzo da memoria ASSE XZ | NO | NO |
| G60 | Abilita interpolazione VELOCE (senza fermata sui tratti) | NO | NO |
| G61 | Abilita interpolazione con fermata sul TRATTO | NO | NO |
| G62 | Attende ASSI FERMI | NO | NO |
| G63 | Abilita PX_MOVETO (interpolazione 3D fuori dal piano) senza fermata in G1 | NO | NO |
| G64 | Abilita interpolazione sul piano PX_LINETO (interpolazione 2D nel piano) in G1 | NO | NO |
| G65 | Abilita interpolazione 3D Px_moveto con fermata decisa dal CN da parametri | NO | NO |
| G66 | AFC – Adaptive Feed Control | NO | NO |
| G66 X-100 | NEW AFC – Adaptive Feed Control | NO | NO |
| G67 | Usa la combinazione Px_lineto (interpolazione 2D nel piano) e Px_moveto (interpolazione 3D fuori dal piano) in G1 con fermata se Px_Moveto NON DEVE ESSERE ABILITATO G63 | NO | NO |
| G68 | Usa sempre Px_lineto (interpolazione 2D nel piano) in G1 ASSI TRASPORTATI NON DEVE ESSERE ABILITATO G63 | NO | NO |
| G69 | LHK – Profondità buffer Look Ahead su virtual CN | NO | NO |
| G70 | Setta il piano di interpolazione su una coppia di assi a piacere | NO | NO |
| G71 G71.1 G71.2 | Ricerca home asse Abilita Asse Disabilita Asse | NO | NO |
| G72 | Filtro N.U.R.B.S (Non Uniform Rational Bspline) | NO | NO |
| G73 | Filtro Noise | NO | NO |

| | | | |
|---|---|----|----|
| G74 | Filtro RLS (Remove Len Segment) | NO | NO |
| G75 | Abilita interpolazione 2D sul piano e 3D fuori dal PIANO con fermata SGL3D | NO | NO |
| G80 (G1080) | Pausa forzata da codice ISO | SI | SI |
| G81 (G1081) | Gestione LIMITI secondari ASSI | SI | SI |
| G82 (G1082) | Zero Assi nel punto attuale con OFFSET sensore di acquisizione (origine pezzo) | NO | NO |
| G83 (G1083) | Forza aggiornamento contatori CPU1 | NO | NO |
| G84 (G1084) | Preset contatori CPU | NO | NO |
| G85 | Offset Assi nel punto attuale con OFFSET sensore di acquisizione (offset pezzo) | NO | NO |
| G86 | Preset hardware asse sul modulo 360 gradi | NO | NO |
| G87 | Sospensione offset testa selezionato con Hn | NO | NO |
| G88 | Ripresa offset testa selezionato con Hn | NO | NO |
| G89 | Preset contatore asse in modo HardWare | NO | NO |
| G90 | Movimentazione con quote ASSOLUTE | NO | NO |
| G91 G91.1 G91.2 | Movimentazione con quote RELATIVE Forza movimentazione Assoluta salvando lo stato di G91 Reimposta lo stato salvato con G91.1 | NO | NO |
| G92 | Zero Assi nel punto attuale (origine pezzo) | NO | NO |
| G93 | Offset Assi a quote (offset pezzo) | NO | NO |
| G94 G94.n | Zero Assi a quote (origine pezzo) Salvataggio delle origini | NO | NO |
| G95 | Offset Assi nel punto attuale (offset pezzo) | NO | NO |
| G96 | Sospensione Offset Assi G93-G95 | NO | NO |
| G97 | Ripresa Offset Assi G92-G94 | NO | NO |
| G98 | Sospensione Zero Assi G92-G94 | NO | NO |
| G99 | Ripresa Zero Assi G92-G94 | NO | NO |
| G100 | Comando sincrono per asse VIRTUALE | NO | NO |
| G101 | Comanda uno STOP ASSI nel punto attuale | SI | SI |
| G102 | Start Acquisizione quote da sensore | NO | NO |
| G102.1 | Acquisizione da sistema di visione PxVision | NO | NO |
| G102.2 | Start Acquisizione quote da sensore senza controllo ALLARME | NO | NO |
| G103 | Gestione parametri RTCP | NO | NO |
| G104 G104.1 | Abilita RTCP | NO | NO |
| G105 | Sospende RTCP | NO | NO |
| G106 | Filtro Smoothing | NO | NO |
| G107 | Gestione Macro ad interrupt | NO | NO |
| G108 | Gestione Assi Speciali | NO | NO |
| G120 | Abilita Specularità verticale | NO | NO |
| G121 | Disabilita Specularità verticale G20 | NO | NO |
| G940 | Movimentazione nel blocco escludendo OFFSET e ZERI | NO | NO |

| | | | |
|-----------------------------|------------------------|----|----|
| G1028 (G28) | Ritorno ad HOME | NO | NO |
| G1050 | Disabilita Scaling | NO | NO |
| G1051 | Abilita Scaling | NO | NO |
| G1080 (G80) | Disabilita Cicli Fissi | NO | NO |
| G1081 (G81) | Foratura | NO | NO |
| G1082 (G82) | Foratura | NO | NO |
| G1083 (G83) | Foratura | NO | NO |
| G1084 (G84) | Maschiatura | NO | NO |

3.2 ISTRUZIONI ISO RICONOSCIUTE

| Codice | Descrizione | TASK1 | TASK2 |
|--------------------|----------------------------------|-------|-------|
| X | Quota asse X | NO | NO |
| Y | Quota asse Y | NO | NO |
| Z | Quota asse Z | NO | NO |
| A | Quota asse A | NO | NO |
| B | Quota asse B | NO | NO |
| C | Quota asse C | NO | NO |
| U | Quota asse U | NO | NO |
| V | Quota asse V | NO | NO |
| W | Quota asse W | NO | NO |
| QX | Quota Canale 0 | NO | NO |
| QY | Quota Canale 1 | NO | NO |
| QZ | Quota Canale 2 | NO | NO |
| QA | Quota Canale 3 | NO | NO |
| QB | Quota Canale 4 | NO | NO |
| QC | Quota Canale 5 | NO | NO |
| QU | Quota Canale 6 | NO | NO |
| QV | Quota Canale 7 | NO | NO |
| QW | Quota Canale 8 | NO | NO |
| DX | Quota relativa asse X | NO | NO |
| DY | Quota relativa asse Y | NO | NO |
| DZ | Quota relativa asse Z | NO | NO |
| DA | Quota relativa asse A | NO | NO |
| DB | Quota relativa asse B | NO | NO |
| DC | Quota relativa asse C | NO | NO |
| DU | Quota relativa asse U | NO | NO |
| DV | Quota relativa asse V | NO | NO |
| DW | Quota relativa asse W | NO | NO |
| OX | Imposta numero origine su asse X | NO | NO |

| | | | |
|--------------------|---|----|----|
| <u>OY</u> | Imposta numero origine su asse Y | NO | NO |
| <u>OZ</u> | Imposta numero origine su asse Z | NO | NO |
| <u>OA</u> | Imposta numero origine su asse A | NO | NO |
| <u>OB</u> | Imposta numero origine su asse B | NO | NO |
| <u>OC</u> | Imposta numero origine su asse C | NO | NO |
| <u>OU</u> | Imposta numero origine su asse U | NO | NO |
| <u>OV</u> | Imposta numero origine su asse V | NO | NO |
| <u>OW</u> | Imposta numero origine su asse W | NO | NO |
| <u>DOX</u> | Imposta offset origine su asse X | NO | NO |
| <u>DOY</u> | Imposta offset origine su asse Y | NO | NO |
| <u>DOZ</u> | Imposta offset origine su asse Z | NO | NO |
| <u>DOA</u> | Imposta offset origine su asse A | NO | NO |
| <u>DOB</u> | Imposta offset origine su asse B | NO | NO |
| <u>DOC</u> | Imposta offset origine su asse C | NO | NO |
| <u>DOU</u> | Imposta offset origine su asse U | NO | NO |
| <u>DOV</u> | Imposta offset origine su asse V | NO | NO |
| <u>DOW</u> | Imposta offset origine su asse W | NO | NO |
| <u>F</u> | FEED assi | NO | NO |
| <u>S</u> | Speed mandrino | SI | SI |
| <u>N</u> | Numeratore di linea (NON NECESSARIO) | NO | NO |
| <u>D</u> | Correttore utensile | NO | NO |
| <u>M</u> | Funzione M | SI | SI |
| <u>HM</u> | Funzione HM | SI | SI |
| <u>H</u> | Seleziona testa porta utensile | NO | NO |
| <u>I</u> | Selezione utensile di lavoro | NO | NO |
| <u>I</u> | Coordinata X del CENTRO per G2 G3 | NO | NO |
| <u>J</u> | Coordinata Y del CENTRO per G2 G3 | NO | NO |
| <u>K</u> | Coordinata Z del CENTRO per G2 G3 (opzionale) | NO | NO |
| <u>R</u> | Raggio dell' arco per G2 G3 | NO | NO |
| <u>USER_ZERO</u> | Indice vettore ZERO PEZZO (origine pezzo) | NO | NO |
| <u>USER_OFFSET</u> | Indice vettore OFFSET PEZZO | NO | NO |
| <u>STOP_MODE</u> | Definisce il modo di STOP assi da pulsante | NO | NO |
| <u>PAUSE_MODE</u> | Definisce come deve agire la pausa sul CNC | NO | NO |
| <u>FILTER_MODE</u> | Definisce il modo dei filtri | NO | NO |

3.3 ISTRUZIONI PER IL CONTROLLO DEL FLUSSO DI PROGRAMMA

| Codice | Descrizione | TASK1 | TASK2 |
|--------------------|---|-------|-------|
| <u>IF</u> | Ciclo IF condizionale | SI | SI |
| <u>ELSE</u> | Parte del costrutto IF | SI | SI |
| <u>END IF</u> | Chiusura ciclo IF | SI | SI |
| <u>SELECT-CASE</u> | Ciclo Select – Case | SI | SI |
| <u>LOOP</u> | Ciclo LOOP iterativo | SI | SI |
| <u>END LOOP</u> | Chiusura ciclo LOOP | SI | SI |
| <u>FOR</u> | Ciclo Iterativo For | SI | SI |
| <u>NEXT</u> | Fine Ciclo For | SI | SI |
| <u>BREAK</u> | Interrompe un ciclo FOR o WHILE | SI | SI |
| <u>CONTINUE</u> | Continua un ciclo FOR o WHILE | SI | SI |
| <u>WHILE</u> | Ciclo iterativo WHILE | SI | SI |
| <u>END WHILE</u> | Fine Ciclo WHILE | SI | SI |
| <u>GOTO</u> | Salto a LABEL o numero di linea | SI | SI |
| <u>GOSUB</u> | Richiamo di una Soubrutine | SI | SI |
| <u>RETURN</u> | Ritorna da una Soubrutine | SI | SI |
| <u>@</u> | Definizione di una LABEL | SI | SI |
| <u>//</u> | Commento al programma | SI | SI |
| <u>END PROGRAM</u> | Termine del programma | SI | SI |
| <u>WAIT INPUT</u> | Attende lo stato di un input digitale con Time Out | SI | SI |
| <u>TWAIT INPUT</u> | Attende lo stato di un input digitale con Time Out e Time Stato | SI | SI |
| <u>ERROR</u> | Forza un uscita da programma con errore | SI | SI |

3.4 ISTRUZIONI PER IL MULTI TASK

| Codice | Descrizione | TASK1 | TASK2 |
|----------------------|-------------------------------------|-------|-------|
| <u>TASK.RUN</u> | Avvia l' esecuzione di un TASK | SI | SI |
| <u>TASK.STOP</u> | Stop esecuzione TASK | SI | SI |
| <u>TASK.PAUSE</u> | Pause esecuzione TASK | SI | SI |
| <u>TASK.READVAR</u> | Legge una variabile di un TASK | SI | SI |
| <u>TASK.WRITEVAR</u> | Scrive una variabile di un TASK | SI | SI |
| <u>TASK.STATUS</u> | Legge lo stato di un TASK | SI | SI |
| <u>TASK.LOADCMD</u> | Carica un CMD in un TASK | SI | SI |
| <u>TASK.PRIORITY</u> | Setta la priorità di un TASK | SI | SI |
| <u>USTASK</u> | Apri una sezione di Gcode per TASK1 | NO | NO |
| <u>ENDUSTASK</u> | Fine sezione Gcode per TASK1 | NO | NO |

3.5 ISTRUZIONI PER UTILIZZO GENERICO

| Codice | Descrizione | TASK1 | TASK2 |
|-------------------------------------|---|-------|-------|
| <u>SDO_DL</u> | Sdo download (can open) | SI | SI |
| <u>SDO_UL</u> | Sdo upload (can open) | SI | SI |
| <u>GET</u> | Legge un parametro delle funzioni HM | SI | SI |
| <u>READ_PARMAC</u> | Legge un parametro macchina | SI | SI |
| <u>WRITE_PARMAC</u> | Scrive un parametro macchina | SI | SI |
| <u>LOAD_VAR</u> | Carica un file specificato di variabili | SI | SI |
| <u>GET_VAR</u> | Legge una variabile dal file caricato | SI | SI |
| <u>WRITE_VAR</u> | Scrive una variabile nella lista di memoria | SI | SI |
| <u>SAVE_VAR</u> | Salva il file su hard disk | SI | SI |
| <u>DIM_VAR</u> | Dimensiona una lista di memoria | SI | SI |
| <u>CLEAR_VAR</u> | Rimuove tutti i valori dalla lista attuale | SI | SI |
| <u>REMOVE_VAR</u> | Rimuove un valore dalla lista attuale | SI | SI |
| <u>FILE_EXISTS</u> | Controlla se esiste il file indicato | SI | SI |
| <u>OPT</u> | Setta opzioni di compilazione in Run Time | SI | SI |
| <u>IMPORT</u> | Importazione di file esterni in vari formati | NO | NO |
| <u>END_IMPORT</u> | Termina un programma chiamato con IMPORT | NO | NO |
| <u>PA(n,par)</u> | Set quota ASSOLUTA posizionatori e controllo | SI | SI |
| <u>PD(n,par)</u> | Set quota RELATIVA posizionatori | SI | SI |
| <u>PF(n)</u> | Feed posizionatore | SI | SI |
| <u>PS(n)</u> | Stop posizionatore | SI | SI |
| <u>_PM(n,par)</u> | Lettura stato posizionatore | SI | SI |
| <u>RESUME_T</u> | Riprende l' esecuzione dall' ultima istruzione Tn utilizzata | NO | NO |
| <u>DIM</u> | Dimensiona un ARRAY in memoria | SI | SI |
| <u>DEBUG_INFO</u> | Scrive informazioni su file LOG | SI | SI |
| <u>SET_TABPAR</u> | Setta una Tabella parametri | SI | SI |
| <u>SAVE_T</u> | Salva la tabella utensile nel file di configurazione | SI | SI |
| <u>IF_RUN</u> | Controllo tipo di RUN Normale (\$[X7]=0) | SI | SI |
| <u>IF_NOTRUN</u> | Controllo tipo di RUN SIMULATO (\$[X7]<>0) | SI | SI |
| <u>USEFORM</u> | Avvia un FORM compilato con UsFormManager | SI | SI |

3.6 ISTRUZIONI PER IL CONTROLLO REMOTO

| Codice | Descrizione | TASK1 | TASK2 |
|---|--|-------|-------|
| <u>REMOTE.LOAD</u> | Load Part Program su CN remoto | SI | SI |
| <u>REMOTE.RUN</u> | Run Part Program su CN remoto | SI | SI |
| <u>REMOTE.STOP</u> | Stop Part Program su CN remoto | SI | SI |
| <u>REMOTE.PAUSE</u> | Pausa Part Program su CN remoto | SI | SI |
| <u>REMOTE.STATUS</u> | Lettura Status su CN remoto | SI | SI |
| <u>REMOTE.MOVE</u> | Lettura stato movimento assi su CN remoto | SI | SI |
| <u>REMOTE.INFO</u> | Lettura informazioni su CN remoto | SI | SI |
| <u>REMOTE.AXIS</u> | Lettura quote su CN remoto | SI | SI |
| <u>REMOTE.GROUP</u> | Lettura Gruppo quote su CN remoto | SI | SI |
| <u>REMOTE.READISOVAR</u> <u>REMOTE.READVARNAME</u> | Lettura variabile ISO su CN remoto | SI | SI |
| <u>REMOTE.WRITEISOVAR</u> <u>REMOTE.WRITEVARNAME</u> | Scrittura variabile ISO su CN remoto | SI | SI |
| <u>REMOTE.READCNVAR</u> | Lettura variabile USER su CN remoto | SI | SI |
| <u>REMOTE.WRITECNVAR</u> | Scrittura variabile USER su CN remoto | SI | SI |
| <u>REMOTE.READINPUT</u> | Lettura stato ingresso digitale su CN remoto | SI | SI |
| <u>REMOTE.READOUT</u> | Lettura stato uscita digitale su CN remoto | SI | SI |
| <u>REMOTE.WRITEOUT</u> | Scrittura uscita digitale su CN remoto | SI | SI |

3.7 ISTRUZIONI PER IL CONTROLLO DEL MULTIPROCESSO

| Codice | Descrizione | TASK1 | TASK2 |
|--|---|-------|-------|
| <u>CNC.LOAD</u> | Load Part Program su Processo CN | SI | SI |
| <u>CNC.RUN</u> | Run Part Program su Processo CN | SI | SI |
| <u>CNC.PREVIEW</u> | Preview Part Program su Processo CN | SI | SI |
| <u>CNC.STOP</u> | Stop Part Program su Processo CN | SI | SI |
| <u>CNC.PAUSE</u> | Pausa Part Program su Processo CN | SI | SI |
| <u>CNC.STATUS</u> | Lettura Status su Processo CN | SI | SI |
| <u>CNC.STATUSBIT</u> | Lettura Status a bit su Processo CN | SI | SI |
| <u>CNC.INFO</u> | Lettura informazioni su Processo CN | SI | SI |
| <u>CNC.AXIS</u> | Lettura quote su Processo CN | SI | SI |
| <u>CNC.GROUP</u> | Lettura Gruppo quote su Processo CN | SI | SI |
| <u>CNC.READVARADDR</u> <u>CNC.READVARNAME</u> | Lettura variabile ISO su Processo CN | SI | SI |
| <u>CNC.WRITEVARADDR</u> <u>CNC.WRITEVARNAME</u> | Scrittura variabile ISO su Processo CN | SI | SI |
| <u>CNC.READPARAMAC</u> | Lettura Parametro Macchina su Processo CN | SI | SI |
| <u>CNC.WRITEPARAMAC</u> | Write Parametro Macchina su Processo CN | SI | SI |
| <u>CNC.ENABLEAXIS</u> | Abilitazione Asse su Processo CN | SI | SI |
| <u>CNC.HOMEAXIS</u> | Homing Asse su Processo CN | SI | SI |
| <u>CNC.READGENERIC</u> | Lettura variabile Generica su Processo CN | SI | SI |
| <u>CNC.WRITEGENERIC</u> | Scrittura variabile Generica su Processo CN | SI | SI |

3.8 OPERATORI LOGICI E MATEMATICI

| Codice | Descrizione | TASK1 | TASK2 |
|---------------------------------|---|-------|-------|
| + | Somma | SI | SI |
| - | Sottrazione | SI | SI |
| * | Moltiplicazione | SI | SI |
| / | Divisione | SI | SI |
| (| Parentesi aperta | SI | SI |
|) | Parentesi chiusa | SI | SI |
| [| Inizio di un espressione per contatore assi es: G1X[\$var+\$var1] | SI | SI |
|] | Fine di un espressione per contatore assi es: G1X[\$var+\$var1] | SI | SI |
| ^ | Elevamento a potenza | SI | SI |
| % | XOR | SI | SI |
| > | Maggiore | SI | SI |
| < | Minore | SI | SI |
| >= | Maggiore uguale | SI | SI |
| <= | Minore uguale | SI | SI |
| <> | Diverso | SI | SI |
| = | Uguale | SI | SI |
| | Or logico | SI | SI |
| && | And logico | SI | SI |
| | Or Bit | SI | SI |
| & | And Bit | SI | SI |
| ! | Negazione espressione | SI | SI |
| ~ | Not Bit | SI | SI |
| >> | Shift bit a sinistra | SI | SI |
| << | Shift bit a destra | SI | SI |
| \$VAR++ :xxx++ | Incremento variabile \$VAR=\$VAR+1 :1000=:1000+1 | SI | SI |
| \$VAR-- :xxx-- | Decremento variabile \$VAR=\$VAR-1 :1000=:1000-1 | SI | SI |
| \$VAR+= :xxx+= | Somma variabile \$VAR=\$VAR+exp :1000=:1000+exp | SI | SI |
| \$VAR-= :xxx-= | Sottrazione variabile \$VAR=\$VAR-exp :1000=:1000-exp | SI | SI |
| \$VAR*= :xxx*= | Moltiplicazione variabile \$VAR=\$VAR*exp :1000=:1000*exp | SI | SI |
| \$VAR/= :xxx/= | Divisione variabile \$VAR=\$VAR/exp :1000=:1000/exp | SI | SI |

3.9 MATEMATICA

| Codice | Descrizione | TASK1 | TASK2 |
|-----------------------------|---|-------|-------|
| <u>SIN</u> | Seno | SI | SI |
| <u>COS</u> | Coseno | SI | SI |
| <u>LOG</u> | Logaritmo | SI | SI |
| <u>EXP</u> | Esponenziale | SI | SI |
| <u>SQR</u> | Radice quadrata | SI | SI |
| <u>TAN</u> | Tangente | SI | SI |
| <u>ATAN</u> | ARCOTANGENTE | SI | SI |
| <u>ASIN</u> | ARCOSENO | SI | SI |
| <u>ACOS</u> | ARCOSENO | SI | SI |
| <u>INT</u> | Parte intera di un float con arrotondamento | SI | SI |
| <u>FIX</u> | Parte intera di un float senza arrotondamento | SI | SI |
| <u>ABS</u> | Valore assoluto | SI | SI |
| <u>DRG</u> | Funzioni COS,SIN,TAN,ACOS,ASIN,ATAN in gradi | SI | SI |
| <u>RAD</u> | Funzioni COS,SIN,TAN,ACOS,ASIN,ATAN in radianti | SI | SI |

3.10 VARIABILI e COSTANTI

| Codice | Descrizione | TASK1 | TASK2 |
|-------------------|---|-------|-------|
| <u>NUMERO</u> | Costante numerica es: 432.12 | SI | SI |
| <u>\$NOMEVAR</u> | Variabile generica di tipo double es: \$VAR1 | SI | SI |
| <u>\$STRU.VAR</u> | Strutture Dati | SI | SI |
| <u>:Addr</u> | Variabile per indirizzo | SI | SI |
| <u>:\$VAR</u> | Variabile per puntatore | SI | SI |
| <u>\$(Qn)</u> | Variabile Quote assi teoriche es: \$(Q0) - \$(Q100) | SI | SI |
| <u>\$(Rn)</u> | Variabile Quote assi reali es: \$(R0) - \$(R100) | SI | SI |
| <u>\$(In)</u> | Variabile ingresso digitale es: \$(I10) | SI | SI |
| <u>\$(On)</u> | Variabile uscita digitale es: \$(O10) | SI | SI |
| <u>\$(Tn)</u> | Variabile TIMER es: \$(T3) | SI | SI |
| <u>\$(Un)</u> | Variabile tabella utensile es: \$(U12) | SI | SI |
| <u>\$(Cn)</u> | Variabile contatore assi es: \$(C0) | SI | SI |
| <u>\$(Hn)</u> | Variabile testa utensile es: \$(H1) | SI | SI |
| <u>\$(Xn)</u> | Variabili parametri speciali es: \$(X1) | SI | SI |
| <u>\$(Yn)</u> | Variabile quote zero pezzo es: \$(Y0) | SI | SI |
| <u>\$(Wn)</u> | Variabile quote offset pezzo es: \$(W0) | SI | SI |
| <u>\$(Kn)</u> | Variabile user generic su CN | SI | SI |
| <u>\$(En)</u> | Limite positivo asse impostato | SI | SI |
| <u>\$(Pn)</u> | Parametri per Preview | SI | SI |
| <u>\$(Sn)</u> | Limite negativo asse impostato | SI | SI |
| <u>\$(A0)</u> | Scrittura uscita ANALOGICA per Mandrino | SI | SI |
| <u>\$(Jn)</u> | Gestione Parametri Speciali | SI | SI |

3.11 VARIABILI PREDEFINITE

| Codice | Descrizione | TASK1 | TASK2 |
|------------------|----------------------------|-------|-------|
| <u>\$ PARM 1</u> | Parametro 1 per funzione M | SI | SI |
| <u>\$ PARM 2</u> | Parametro 2 per funzione M | SI | SI |
| <u>\$ PARM 3</u> | Parametro 3 per funzione M | SI | SI |
| <u>\$ PARM 4</u> | Parametro 4 per funzione M | SI | SI |
| <u>\$ PARM 5</u> | Parametro 5 per funzione M | SI | SI |

3.12 ISTRUZIONI DI LIBRERIA USFORMS

| Codice | Descrizione | TASK1 | TASK2 |
|---|--|-------|-------|
| <u>LIB.MESSAGE</u> | Apri una Message Box | NO | NO |
| <u>LIB.SHOWFORM</u> | Apri un NsForm | NO | NO |
| <u>LIB.CLOSEFORM</u> | Chiudi un NsForm | NO | NO |
| <u>LIB.FORMPROP</u> | Imposta le proprietà del NsForm | NO | NO |
| <u>LIB.FORMTEXT</u> | Imposta il titolo (Caption) del NsForm | NO | NO |
| <u>LIB.ADDLABEL</u> | Inserisce una NsLabel nel NsForm | NO | NO |
| <u>LIB.LABELPROP</u> | Imposta le proprietà di una NsLabel | NO | NO |
| <u>LIB.LABELTEXT</u> | Scrivi un testo nella NsLabel | NO | NO |
| <u>LIB.LABELPRINT</u> | Scrivi il contenuto di una variabile di Isous nella NsLabel | NO | NO |
| <u>LIB.LABELF</u> | Imposta il formato per print valore tramite TIMER | NO | NO |
| <u>LIB.ADDBUTTON</u> | Inserisce una NsButton nel NsForm | NO | NO |
| <u>LIB.BUTTONPROP</u> | Imposta le proprietà di una NsButton | NO | NO |
| <u>LIB.BUTTONTEXT</u> | Scrivi un testo nel NsButton | NO | NO |
| <u>LIB.BUTTONPRINT</u> | Scrivi il contenuto di una variabile di Isous nel NsButton | NO | NO |
| <u>LIB.BUTTONF</u> | Imposta il formato per print valore tramite TIMER | NO | NO |
| <u>LIB.ADDINPUT</u> | Inserisce un oggetto NsInput nel Form | NO | NO |
| <u>LIB.INPUTPROP</u> | Imposta le proprietà di un oggetto NsInput | NO | NO |
| <u>LIB.INPUTSETVALUE</u> | Imposta il valore di un oggetto NsInput da una variabile Isous | NO | NO |
| <u>LIB.ADDITEXT</u> | Inserisce un ITEXT nel NsForm | NO | NO |
| <u>LIB.ITEXTPROP</u> | Imposta le proprietà di un oggetto ITEXT | NO | NO |
| <u>LIB.ITEXTSETVALUE</u> | Imposta il valore iniziale di un oggetto ITEXT | NO | NO |
| <u>LIB.ADDCHECK</u> | Inserisce un CHECK nel NsForm | NO | NO |
| <u>LIB.CHECKPROP</u> | Imposta le proprietà di un oggetto CHECK | NO | NO |
| <u>LIB.CHECKSETVALUE</u> | Imposta il valore iniziale di un oggetto CHECK | NO | NO |
| <u>LIB.CHECKTEXT</u> | Imposta Caption del CHECK | NO | NO |
| <u>LIB.ADDCOMBO</u> | Inserisce un oggetto COMBO nel Form | NO | NO |
| <u>LIB.COMBOPROP</u> | Imposta le proprietà di un oggetto COMBO | NO | NO |
| <u>LIB.COMBOSETVALUE</u> | Imposta il valore iniziale di un oggetto COMBO | NO | NO |
| <u>LIB.COMBOITEM</u> | Aggiunge un ITEM al COMBO | NO | NO |
| <u>LIB.ADDSLIDER</u> | Inserisce uno SLIDER nel NsForm | NO | NO |
| <u>LIB.SLIDERPROP</u> | Imposta le proprietà di un oggetto SLIDER | NO | NO |
| <u>LIB.SLIDERSETVALUE</u> | Imposta il valore iniziale di un oggetto SLIDER | NO | NO |
| <u>LIB.GETVAR</u> | Legge una variabile Form | NO | NO |
| <u>LIB.SETVAR</u> | Scrivi una variabile Form | NO | NO |
| <u>LIB.DEBUG</u> | Finestra per Debug Gcode | NO | NO |

3.13 ISTRUZIONI ESTESE EXD

| Codice | Descrizione | TASK1 | TASK2 |
|---|--|-------|-------|
| EXD.STL_LOAD | Carica un file STL nel Preview | SI | SI |
| EXD.READ_TOOLPAR | Legge un parametro della Tabella Utensile | SI | SI |
| EXD.READ_HEADPAR | Legge un parametro della Tabella Teste | SI | SI |
| EXD.WRITE_TOOLPAR | Scrive un parametro della Tabella Utensile | SI | SI |
| EXD.WRITE_HEADPAR | Scrive un parametro della Tabella Teste | SI | SI |
| EXD.LOAD_LAST | Carica l' ultimo utensile usato | SI | SI |
| EXD.SAVE_LAST | Salva l' utensile in uso | SI | SI |
| EXD.RESET_LAST | Reset del file ultimo utensile usato | SI | SI |
| EXD.SET_OUT | Set/Reset uscita digitale per Indirizzo - $\$(On)$ | SI | SI |
| EXD.READ_OUT | Legge uscita digitale per Indirizzo - $\$(On)$ | SI | SI |
| EXD.READ_INP | Legge ingresso digitale per Indirizzo - $\$(In)$ | SI | SI |
| EXD.WRITE_USER | Scrive una variabile sul CN per indirizzo - $\$(Kn)$ | SI | SI |
| EXD.READ_USER | Legge una variabile dal CN per indirizzo - $\$(Kn)$ | SI | SI |
| EXD.READ_DEMAND | Legge Demand Position di una asse per Indirizzo - $\$(Qn)$ | SI | SI |
| EXD.READ_REAL | Legge Real Position di una asse per Indirizzo - $\$(Rn)$ | SI | SI |
| EXD.MASK | Set Maschera Uscite Digitali | SI | SI |
| EXD.WRITE_BIT | Set/Reset bit di variabile | SI | SI |
| EXD.READ_BIT | Legge un bit di una variabile | SI | SI |
| EXD.PXV_ALL_DETECTORS | Legge tutti i detectors di PXV | SI | SI |
| EXD.PXV_SINGLE_DETECTOR | Legge un detector di PXV | SI | SI |
| EXD.PXV_READ_PROBE | Legge una PROBE di PXV | SI | SI |
| EXD.PXV_RESET_PROBE | Reset di una PROBE di PXV | SI | SI |
| EXD.PXV_SET_DETECTOR | Set del tipo di detector di PXV | SI | SI |
| EXD.PXV_SAVE_IMAGE | Salva la corrente immagine acquisita da PXV | SI | SI |
| EXD.PXV_GET_IMAGE | Cattura un Immagine da PXV | SI | SI |
| EXD.PXV_SET_JOB | Set file JOB di PXV | SI | SI |
| EXD.RUN_SCRIPT | Esegue uno SCRIPT esterno scritto in codice C# | SI | SI |
| EXD.OPEN_DLL | Apri una UsDLL .NET | SI | SI |
| EXD.CALL_DLL | Chiamata ad una UsDLL .NET | SI | SI |
| EXD.CLOSE_DLL | Chiude ad una UsDLL .NET | SI | SI |
| EXD.SYNK_DLL | Sincronizza ad una UsDLL .NET | SI | SI |
| EXD.STL_SETVIS | Visualizza o nasconde STL per Nome | SI | SI |
| EXD.STL_READVIS | Legge lo stato di visibilità di un STL per Nome | SI | SI |
| EXD.STL_SETVISIDX | Visualizza o nasconde STL per Indirizzo | SI | SI |
| EXD.STL_READVISIDX | Legge lo stato di visibilità di un STL per Indirizzo | SI | SI |

3.14 SWITCH e DIRETTIVE DI COMPILAZIONE

| Codice | Descrizione | TASK1 | TASK2 |
|------------------------------------|---|-------|-------|
| <u>IFDEF</u> | Switch di Compilazione IF | SI | SI |
| <u>ELSEDEF</u> | Switch di Compilazione ELSE | SI | SI |
| <u>ENDIFDEF</u> | Switch di Compilazione ENDIF | SI | SI |
| <u>NOAXESREADY</u> | Direttiva di compilazione Assi non Pronti | SI | SI |
| <u>ONERROR</u> | Direttiva di compilazione On Error | SI | SI |
| <u>ONSTOP</u> | Direttiva di compilazione On Stop | SI | SI |
| <u>ENDON</u> | Direttiva di compilazione End On | SI | SI |
| <u>USET</u> | Direttiva di compilazione Use T | SI | SI |
| <u>USEH</u> | Direttiva di compilazione Use H | SI | SI |

4 CONTROLLO DEL FLUSSO DEL PROGRAMMA

Isous utilizza un linguaggio esteso del codice standard ISO. Questo permette di avere istruzioni specifiche stile BASIC per il controllo del flusso del programma, aumentando la capacità di programmazione. Pertanto è possibile effettuare cicli condizionali, iterativi, salti a etichette ecc.

4.1 IF-ELSE-END_IF

Consente l'esecuzione condizionale di un gruppo di istruzioni in base al valore di un'espressione.

I cicli IF possono essere annidati senza alcun limite

Sintassi

IF condizione

[istruzioni]

ELSE

[istruzioni else]

END_IF

condizione Obbligatoria. Qualsiasi espressione numerica che può dare come risultato True o False.

Istruzioni Elenco istruzioni da eseguire se la condizione **IF** è **VERA**

istruzioni else Facoltativa. Elenco istruzioni da eseguire se la condizione **IF** è **FALSA**.

END_IF Fine del ciclo **IF ELSE**

Es:

IF \$VAR=\$VAR2*15+(\$VAR4+18)

\$VAR1=10

G1X10Y20

ELSE

\$VAR1=0

G1X0Y0

END_IF

4.2 LOOP - END_LOOP

Ciclo iterativo del programma Le istruzioni contenute all'interno del ciclo LOOP END_LOOP vengono eseguite per un numero di volte determinato da un'espressione. I cicli LOOP possono essere annidati senza alcun limite

Sintassi

LOOP variabile K

[istruzioni]

END_LOOP

variabile Obbligatoria. Qualsiasi variabile di tipo \$nome o valore numerico

K Opzionale – Se inserito in Preview viene effettuato solo 1 loop (LOOP 100 K)

Istruzioni Elenco istruzioni da eseguire durante il ciclo iterativo

END_LOOP Fine del ciclo **LOOP**

Es:

G91

\$VAR=20+(\$VAR1*5)

LOOP \$VAR

G1X1.3Y1.2

IF \$VAR1=20

GOTO EXIT_LOOP // USCITA FORZATA

END_IF

END_LOOP

@EXIT_LOOP

4.3 FOR – NEXT – BREAK - CONTINUE

Ciclo iterativo FOR-NEXT

Esegue un iterazione delle istruzioni in base alle condizioni

Sintassi

FOR InitFor;CondFor;IterFor

[istruzioni]

BREAK

CONTINUE

NEXT

| | |
|-------------------|--|
| InitFor | Obbligatoria. Inizializzazione della Variabile ciclo for di condizione |
| CondFor | Obbligatoria – Condizione del ciclo FOR – itera fine a che la condizione è soddisfatta |
| IterFor | Obbligatoria – Iterazione della Variabile del ciclo FOR – Incremento della variabile di condizione |
| Istruzioni | Elenco istruzioni da eseguire durante il ciclo iterativo |
| BREAK | Interrompe il ciclo FOR |
| CONTINUE | Salta alla prossima iterazione del ciclo FOR |
| NEXT | Fine del ciclo FOR |

Es:

G0X0Y0

\$VAR1=0

\$VAR2=0

\$VAR3=0

\$VAR4=0

FOR \$V=0;\$V<20;\$V+=2

G0X[\$VAR1]Y[\$VAR2]

G4F0.5

\$VAR1+=10

\$VAR2+=20

\$VAR+=4

IF\$VAR<20

CONTINUE

END_IF

IF\$VAR4=200

BREAK

END_IF

FOR \$V1=0;\$V1<100;\$V1++

\$VAR3+=1

NEXT

NEXT

4.4 WHILE – END_WHILE – BREAK - CONTINUE

Ciclo iterativo WHILE-END_WHILE

Itera delle istruzioni in base alla condizione

Sintassi

WHILE CondWhile

[istruzioni]

BREAK

CONTINUE

END_WHILE

CondWhile Obbligatoria – Condizione del ciclo WHILE – itera fine a che la condizione è soddisfatta

Istruzioni Elenco istruzioni da eseguire durante il ciclo iterativo

BREAK Interrompe il ciclo WHILE

CONTINUE Salta alla prossima Iterazione del ciclo WHILE

END_WHILE Fine del ciclo **WHILE**

Es:

G0X100

\$VAR1=10

\$VAR=0

\$VAR3=0

\$VAR4=5

WHILE \$VAR<\$VAR1

\$VAR++

WHILE \$VAR3<200

IF \$VAR3=204

BREAK

END_IF

\$VAR3++

END_WHILE

IF \$VAR4=10

CONTINUE

END_IF

END_WHILE

G0X200

4.5 GOTO

Salto incondizionato a etichetta (label) o numero di linea. Permette un salto del programma ad un etichetta definita o ad un numero di linea identificato con *Nxx*.

Sintassi

GOTO nomelabel Per salto a label
GOTO @numerolinea Per salto a numero di linea. Se numero di linea =-1 esegue il PartProgram dall' inizio

nomelabel Obbligatoria. Etichetta del programma definita con **@nomelabel**
@numerolinea Numero di linea identificato con **Nxx**

Es:

```
IF $VAR1=0
    GOTO CICLO_A
END_IF
IF $VAR1=1
    GOTO @100 // GOTO @-1 ESEGUE DALL' INIZIO
END_IF
// ESEGUE IL CICLO A
@ CICLO_A
G1X100Y100
.....
.....
// ESEGUE IL CICLO B
N100G1X10Y10
```

4.6 GOSUB - RETURN

Consente il salto ad un etichetta di programma con ritorno all' istruzione successiva il salto

Sintassi

GOSUB nomelabel Per salto a label
GOSUB @numerolinea Per salto a numero di linea

nomelabel Obbligatoria. Etichetta del programma definita con **@nomelabel**
@numerolinea Numero di linea identificato con **Nxx**

È obbligatorio che la label termini con un RETURN

Es:

```
IF $VAR1=0
    GOSUB CICLO_A
END_IF
IF $VAR1=1
    GOSUB @100
END_IF
// ESEGUE IL CICLO A
@ CICLO_A
G1X100Y100
RETURN
// ESEGUE IL CICLO B
N100G1X10Y10
RETURN
```

4.7 LABEL (punti di etichetta)

Le LABEL definiscono un punto di salto per le istruzioni **GOTO** e **GOSUB**. Non possono esistere LABEL con lo stesso nome in quanto creerebbero un conflitto.

Le LABEL sono esclusive per ogni PartProgram, tuttavia Isous considera PRIVATE le label dichiarate per le funzioni M o HM, pertanto queste non influiscono con le LABEL del PartProgram.

È buona norma dare dei nomi significativi alle LABEL.

Sintassi

@nomelabel

@ Obbligatoria. Identificativo di una LABEL
nomelabel Obbligatoria. Nome univoco della LABEL
 Sono ammessi tutti i caratteri escluso separatori logici e matematici

Es:

```
@ CICLO_A
G1X100Y100
.....
@ CICLO_B
G1X10Y10
```

4.8 END_PROGRAM

Termina il programma nel punto esatto dove si trova. Isous inserisce automaticamente l'istruzione END_PROGRAM alla fine del programma (Fine FILE). Tuttavia in certi casi è necessario terminare il programma in un punto diverso da quello della fine FILE.

Sintassi

END_PROGRAM

Es:

```
IF $VAR1=0
    END_PROGRAM // TERMINA IL PROGRAMMA IN MODO ANTICIPATO
END_IF
```

4.9 WAIT_INPUT

Attende lo stato di un ingresso digitale per un tempo predefinito. Se allo scadere del tempo l'ingresso digitale non si trova allo stato logico indicato è possibile generare un ALLARME interrompendo l'esecuzione del PartProgram in corso.

Sintassi

WAIT_INPUT Input Stato Timeout Allarme

Input Obbligatorio. Numero ingresso digitale da 0 a 255
Stato Obbligatorio. Stato logico al quale deve trovarsi l'ingresso digitale entro il tempo prestabilito
 0 Stato OFF – 1 Stato ON
Timeout Tempo in sec con risoluzione del decimo. Se il tempo è trascorso è possibile generare un allarme interrompendo il PartProgram
TimeOut=0 attesa infinita
Allarme Codice di allarme generato (fare riferimento al file di configurazione)
 0 – Nessun allarme

Es:

```
WAIT_INPUT 2 1 5.3 12 // attende input 2 allo stato logico 1 per 5.3 sec e
//genera allarme 12 se condizione non verificata
```

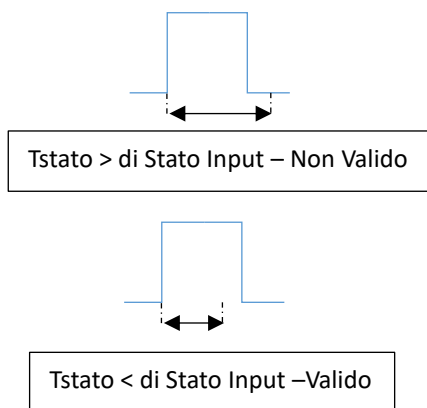

4.10 TWAIT_INPUT

Attende lo stato di un ingresso digitale per un tempo predefinito e lo stato deve trovarsi nella giusta condizione per un tempo. Questo filtra ingressi che hanno variazioni improvvise che vanno filtrate. Se allo scadere del tempo l'ingresso digitale non si trova allo stato logico indicato per il tempo indicato è possibile generare un ALLARME interrompendo l'esecuzione del PartProgram in corso.

Sintassi

TWAIT_INPUT Input Stato TStato TimeOut allarme

| | |
|----------------|---|
| Input | Obbligatorio. Numero ingresso digitale da 0 a 255 |
| Stato | Obbligatorio. Stato logico al quale deve trovarsi l'ingresso digitale entro il tempo prestabilito 0 Stato OFF – 1 Stato ON |
| Tstato | Tempo in sec con risoluzione del decimo per considerare lo Stato valido Tstato=0 Non considerato (come WAIT_INPUT) |
| Timeout | Tempo in sec con risoluzione del decimo. Se il tempo è trascorso è possibile generare un allarme interrompendo il PartProgram TimeOut=0 attesa infinita |
| Allarme | Codice di allarme generato (fare riferimento al file di configurazione) 0 – Nessun allarme |



Es:

```

TWAIT_INPUT 2 1 2 5.3 12 // attende input 2 allo stato logico 1 Tstato=2 Sec per 5.3 sec e
                          //genera allarme 12 se condizione non verificata
  
```

4.11 ERROR

Genera un uscita da PartProgram con errore. Tale istruzione serve per generare eccezioni del PartProgram al verificarsi di eventi. L'esecuzione viene interrotta generando un allarme.

Sintassi

ERROR err_code

err_code Obbligatorio. Codice di errore riferito al file di configurazione.

Es:

```

IF $VAR1=0
      ERROR 10      // TERMINA IL PARTPROGRAM CON ALLARME 10
END_IF
  
```

4.12 RESUME_T

Riprende il PartProgram dall' ultima istruzione **Tn** utilizzata.

Viene usata per gestione degli utensili Danneggiati. In pratica quando la misura utensile rileva che l' utensile è danneggiato, tramite **RESUME_T** è possibile rifare la lavorazione dall' ultimo cambio utensile effettuato.

Il parametro che segue **RESUME_T** è l' utensile alternativo a quello danneggiato. Tale parametro viene preso dalla tabella Utensili.

Sintassi

RESUME_T Tool_nr

Tool_nr Numero utensile da caricare (utensile alternativo a quello danneggiato)

4.13 SAVE_T

Salva il contenuto della tabella utensili nel file di configurazione di Isous.

Tramite Part Program è possibile modificare il contenuto della Tabella Utensili. **SAVE_T** rende le modifiche permanenti.

Sintassi

SAVE_T

T1 // SELEZIONA INDICE 1

\$(U0)=10 // SCRIVE DIAMETRO UTENSILE INDEX 0

\$(U1)=100.3 // SCRIVE LUNGHEZZA UTENSILE INDEX 1

SAVE_T // SALVA LA TABELLA IN MODO PERMANENTE

4.14 SET_TABPAR

Setta una tabella parametri ed attiva immediatamente i valori dei parametri in tabella.

Le tabelle parametri vengono create dal configuratore IsoUs e queste contengono una copia dei parametri macchina, ma con valori diversi. Questo permette una parametrizzazione dinamica da Gcode.

La TABELLA 0 è contiene i PARAMETRI DI DEFAULT.

Sintassi

SET_PARTAB Tab_nr

Tab_nr contiene il numero della tabella da attivare. La tabella 0 contiene una copia dei parametri originali

4.15 SELECT – CASE – END_SELECT

Esegue uno dei diversi gruppi di istruzioni , a seconda del valore di un'espressione

Sintassi

SELECT Test Variabile

CASE Espressione1

[istruzioni]

CASE Espressione2

[istruzioni]

.

.

CASE ELSE

[istruzioni]

END_SELECT

| | |
|------------------------------|---|
| SELECT Test Variabile | Obbligatoria deve essere una variabile \$ - Inizio ciclo |
| CASE Num | Condizione Vera – Valore=Num |
| CASE Exp | Condizione Vera – Valore=Exp |
| CASE Exp1 TO Exp2 | Condizione Vera - Valori compresi tra Exp1 1 e Exp2 |
| CASE Exp1,Exp2,Exp3.. | Condizione Vera – Valori uguali Exp1, o Exp2,o Exp3, ecc. |
| CASE IF.. | Condizione Vera – Viene presa l' espressione dopo IF |
| CASE ELSE | Facoltativa. Elenco istruzioni da eseguire se nessuna condizione CASE è vera (deve trovarsi come ultima condizione nel ciclo SELECT, prima di END_SELECT |
| END_SELECT | Fine ciclo Select |

Es:

```

SELECT $VAR
  CASE 10
    .
    .
  CASE $VAR1
    .
    .
  CASE $VAR2,15,$VAR3
    .
    .
  CASE 20 TO $VAR4
    .
    .
  CASE IF>=$VAR5
    .
    .
  CASE ELSE
    .
    .
END_SELECT

```

4.16 IF RUN

Controlla il tipo di esecuzione del Gcode. Se condizione vera **NORMALE** esecuzione.
Analogo ad **IF \$[X7]=0**

Sintassi

IF RUN

..
..
..

END_IF

4.17 IF NOTRUN

Controlla il tipo di esecuzione del Gcode. Se condizione vera esecuzione in **SIMULAZIONE**.
Analogo ad **IF \$[X7]<>0**

Sintassi

IF NOTRUN

..
..
..

END_IF

4.18 USEFORM

Apre un FORM compilato con il PlugIn **UsFormManager**

Sintassi

USEFORM "FORMNAME"

FORMNAME Nome del Form senza estensione Compilato (cartella \Data_Form)

Es:

USEFORM "FORM1"

5 ISTRUZIONI DI UTILIZZO GENERICO

5.1 SDO_DL

Istruzione CanOpen SDO Down Load (riferirsi alla documentazione CanOpen per dettagli).
Scrittura dati su nodo.

Sintassi

SDO_DL nodo index subindex len_data \$var

| | |
|-----------------|---|
| nodo | Obbligatorio. Nodo rete CanOpen a cui richiedere i dati |
| index | Obbligatorio. Indice dell' oggetto DS301 |
| subindex | Obbligatorio. Sub indice dell' oggetto DS301 |
| len | Obbligatorio. Lunghezza dati contenuti in \$Var (1 char – 2 int – 4 long) |
| \$Var | Obbligatorio. Variabile che contiene i dati da inviare al nodo. |

Es:

```
// INVIA AL NODO 2 ALL' INDICE 24596 SUBIND. 4 BYTE
// PRESI DALLA VARIABILE $VAR1
SDO_DL 2 24596 0 4 $VAR1
```

5.2 SDO_UL

Istruzione CanOpen SDO Up Load (riferirsi alla documentazione CanOpen per dettagli).
Lettura dati da nodo.

Sintassi

SDO_UL nodo index subindex len_data \$var

| | |
|-----------------|--|
| nodo | Obbligatorio. Nodo rete CanOpen a cui richiedere i dati |
| index | Obbligatorio. Indice dell' oggetto DS301 |
| subindex | Obbligatorio. Sub indice dell' oggetto DS301 |
| len | Obbligatorio. Lunghezza dati (1 char – 2 int – 4 long) |
| \$Var | Obbligatorio. Variabile di deposito dati richiesti al nodo. \$Var contiene i dati richiesti al nodo |

Es:

```
// INTERROGA IL NODO 2 ALL' INDICE 24596 SUBIND. 4 LEGGENDO 4 BYTE
// CHE VERRANNO DEPOSITATI NELLA VARIABILE $VAR1
SDO_UL 2 24596 0 4 $VAR1
```

5.3 GET

Legge i parametri per una funzione HM

ATTENZIONE I PARAMETRI SONO IN SUCCESSIONE INVERTITA RISPETTO A COME SONO STATI SCRITTI

HM 10 20 30 40

..

```
GET $PAR1 $PAR2 $PAR3 $PAR4 // LEGGE I PARAMETRI
```

quindi

```
$PAR1=40 $PAR2=30 $PAR3=20 $PAR4=10
```

5.4 READ_PARMAC

Legge un parametro macchina tramite passaggio del nome.

Sintassi

READ_PARMAC ParName \$var

ParName Obbligatorio. Nome del parametro da leggere (es: VJOG_X)

\$Var Obbligatorio. Variabile di deposito dove viene riportato il valore del parametro.

Genera un errore di runtime se il parametro letto non esiste

READ_PARMAC VJOG_X \$VAR1

5.5 WRITE_PARMAC

Scrive un parametro macchina tramite passaggio del nome.

Sintassi

WRITE_PARMAC ParName \$var

ParName Obbligatorio. Nome del parametro da leggere (es: VJOG_X)

\$Var Obbligatorio. Variabile che contiene il valore del parametro da scrivere..

Genera un errore di runtime se il parametro scritto non esiste

ATTENZIONE la scrittura dei parametri può causare errori di sistema se non viene effettuata in modo appropriato

\$VAR1=100

WRITE_PARMAC VJOG_X \$VAR1

5.6 OPT

Setta le opzioni di compilazione in RUN TIME.

Serve per cambiare da PartProgram le varie M predefinite

Sintassi

OPT Opzione valore

Opzione Opzioni disponibili:

MSTOP,MSTART,MEND,MGOBLOCK,MGORETRACE,MERROR,MPAUSE,MGOPAUSE

valore Valore ella nuova opzione M

Es:

OPT MSTOP 10 //ridefinisce la M di stop ad M10 per questo partprogram

OPT MERROR -1 //disabilita la M di errore per questo partprogram

5.7 PAUSE_MODE

Definisce il modo con cui deve agire la pausa del CNC.

I Modi sono i seguenti:

- 0** Modo normale (pausa abilitata, M di pausa Abilitata, M di ripresa abilitata)
- 1** Pausa Abilitata, M di pausa abilitata, **M di ripresa Disabilitata**
- 2** Pausa Abilitata, **M di pausa Disabilitata**, M di ripresa Abilitata
- 3** Pausa Abilitata, **M di pausa Disabilitata**, **M di ripresa Disabilitata**
- 4** **PAUSA CNC DISABILITATA**

Sintassi

PAUSE_MODE val

Val Valore o espressione del modo di pausa

Pause_Mode serve per disabilitare o limitare la funzionalità della PAUSA durante alcuni cicli di lavoro speciali: Es: CAMBIO UTENSILE ecc. Dove si potrebbe generare malfunzionamenti causati dalla pausa del CNC.

5.8 FILTER_MODE

Definisce il modo con cui devono agire filtri di IsoUS G69 o G66 e ulteriori funzioni

I Modi sono i seguenti:

- 0** **LOOK AHEAD** disabilitato su IsoUs
- 1** **LOOK AHEAD** abilitato all' ultimo valore su IsoUs
- 2** Sospende contemporaneamente **PAUSA e STOP**
- 3** Riprende contemporaneamente **PAUSA e STOP**
- 4** Sospende istruzione **SAVE_T**
- 5** Riprende istruzione **SAVE_T**

Sintassi

FILTER_MODE val

Val Valore o espressione

5.9 IMPORT

Importa un file esterno con vari formati.

Il formato predefinito è quello di ISOUS. Se è presente nella cartella di Isous il file **NsImportFile.dll**, l' importazione viene effettuata da questa, la quale può effettuare traslazione di codice

Sintassi

IMPORT "nomefile.ext"

NomeFile Contiene il percorso assoluto o relativo del file da importare.

Se **nomefile** inizia con il prefisso **\$APPPATH**, il file viene cercato all' interno delle cartelle di Isous. Il file deve essere sempre completo di estensione.

L' importazione del file viene effettuata al momento dell' istruzione **IMPORT**, di conseguenza anche la compilazione. Viene generato l' evento **StartImport** e **EndImport** quando il file termina la sua esecuzione. Il programma principale riprenderà dopo la terminazione del programma importato. Possono essere nidificati infiniti file di **IMPORT**.

Es:

IMPORT "\$APPPATH\PROJECT\IMPORT\TESTIMPORT.ISO" // importa da cartella locale project

IMPORT "C:\FILEIMPORT\TESTIMPORT.ISO" // importa da percorso assoluto

ATTENZIONE

Per terminare in modo anticipato un file chiamato con IMPORT, occorre utilizzare END_IMPORT

5.10 END_IMPORT

Termina in modo anticipato un programma chiamato dalla funzione IMPORT.
Questo permette di fare un ritorno al programma chiamante.

Sintassi

END_IMPORT

Es:

```
IF $VAR=0
    END_IMPORT
END_IF
```

5.11 STOP_MODE

Definisce il modo di STOP da pulsante

I Modi sono i seguenti:

- | | |
|----------|---|
| 0 | Modo normale |
| 1 | Lo STOP è disabilitato quando la MACRO di STOP è in ESCUZIONE Questo permette di non INTERROMPERE MAI la MACRO di STOP |
| 2 | Interrompe la MACRO DI STOP con DUE PRESSIONI consecutive del PULSANTE STOP Questo permette di INTERROMPERE la MACRO di STOP, ma deve essere premuto per due volte consecutive il pulsante di STOP |

Sintassi

STOP_MODE val

Val Valore o espressione del modo di STOP

5.12 DEBUG_INFO

Scrive informazioni nel file IsoUs_x.Log per poter essere utilizzate a livello di Debug

Sintassi

DEBUG_INFO "TextInfo" \$var

| | |
|-----------------|--|
| TextInfo | <p>Obbligatorio. Testo dell' informazione da scrivere</p> <p>TextInfo controlla alcuni caratteri speciali che definiscono l' azione svolta da DEBUG_INFO.</p> <p>@ Se inserito come PRIMO carattere di TextInfo, oltre a scrivere nel file di Log, viene anche inviato il messaggio al pannello delle notifiche</p> <p>Es: DEBUG_INFO "@TextInfo" \$var</p> <p>& Se inserito come PRIMO carattere di TextInfo, il messaggio viene inserito in un file particolare _Us_DebugInfo.txt che si trova nella cartella di IsoUs</p> <p>Es: DEBUG_INFO "&TextInfo" \$var</p> <p>\$ Se inserito come ULTIMO carattere di TextInfo, viene solamente generato un evento UsDebugInfo</p> <p>Es: DEBUG_INFO "TextInfo\$" \$var</p> |
| \$Var | <p>Obbligatorio. Variabile da scrivere successiva all' informazione. Questa permette anche di salvare alcuni valori relativi a delle variabili. Se non utilizzata inserire una variabile fittizia</p> <p>La variabile è stampata dopo il TextInfo</p> <p>UsDebug --> TextInfo : \$Var</p> |

ATTENZIONE TextInfo deve essere inserito tra VIRGOLETTE ("...")

\$VAR1=1

DEBUG_INFO "TEST DEBUG INFO" \$VAR1

6 ISTRUZIONI DI LIBRERIA GESTIONE DISPLAY REMOTE HANDWHELL

6.1 LIB.HMESSAGE

Scrive una stringa di testo nel display del Remote HandWheel Promax WHC.

Il display del WHC ha 4 RIGHE x 20 CARATTERI

Sintassi

LIB.HMESSAGE "TESTO" DURATION_TIME

TESTO → stringa di testo del messaggio da scrivere nel display

Caratteri speciali:

@RCC

Dove R=numero della riga da cui scrivere (da 0 a 3)

CC=numero della colonna su cui scrivere (da 0 a 19) indicare sempre due caratteri per la colonna)

Es: LIB.HMESSAGE "@105TESTO" 5 Scrive nella riga1 colonna 5

DURATION_TIME → Durata in Sec. Di permanenza del testo sul Display

LIB.HMESSAGE "@100TEST TEXT" 5

Scrive il testo TEST TEXT riga 1 colonna 0 per 5 secondi

7 ISTRUZIONI DI LIBRERIA USFORMS

Le istruzioni **LIB**. Servono per aprire finestre di dialogo (FORM) tramite Gcode.

Questo permette di preparare o condizionare un Part Program in base all' azione degli **OGGETTI** che possono essere inseriti nel **FORM**:

BUTTON

LABEL

CHECK BOX

INPUT

TEXT BOX

COMBO BOX

SLIDER

Tutti gli oggetti operano su Variabili Gcode e quindi è possibile cambiarne il Valore.

Tutti i messaggi di testo che possono essere rappresentati dagli Oggetti se iniziano con il prefisso "@_" possono essere presi dal dizionario di traduzioni "*UsLanguages.lang*" presente nella cartella *\Languages*.

Per semplificare la composizione delle finestre è possibile usa il PlugIn **UsFormManager** per ottenere una programmazione visuale del **FORM**.

7.1 LIB.MESSAGE

Apri una MessageBox (finestra di avviso) con varie modalità.

Il flusso del programma viene sospeso fino all' esecuzione dell' azione richiesta (pressione bottone della MessageBox). Utilizzata per dare avvisi o imporre scelte.

Sintassi

LIB.MESSAGE "TESTO" "CAPTION" BUTTON ICON

TESTO

Testo del messaggio \N a capo

CAPTION

Titolo del messaggio

BUTTON (valore o variabile)

0 OK
1 SI - NO

ICON (valore o variabile)

0 ERROR
1 WARNING
2 INFORMATION
3 QUESTION
4 NONE

Es:

```
LIB.MESSAGE "CONTINUE\NPART PROGRAM EXECUTION???" "WARNING!!!" 1 1
```

```
$BTN=${X13} // LEGGE IL PULSANTE PREMUTO
```

```
IF $BTN=2
```

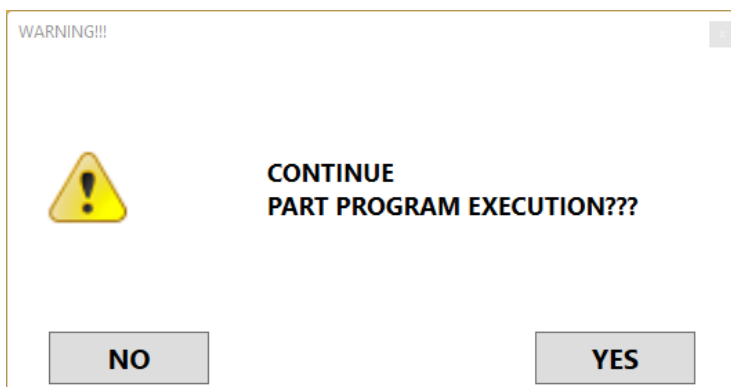
```
    END_PROGRAM // TERMINA IL PROGRAMMA SE PREMUTO NO
```

```
END_IF
```

La funzione LIB.MESSAGE riporta il codice pulsante premuto nella variabile speciale **\$(X13)**.

Questa contiene:

- 0 → Pulsante **OK** premuto
- 1 → Pulsante **SI** premuto
- 2 → Pulsante **NO** premuto



7.2 LIB.SHOWFORM

Apre un Form tipo NsForm. Le proprietà devono essere impostate dall' istruzione **LIB.FORMPROP**.

Il form viene comunque chiuso al termine del PartProgram . **È possibile visualizzare solo un NsForm.**

Sintassi

LIB.SHOWFORM

7.3 LIB.CLOSEFORM

Chiude un Form tipo NsForm precedentemente aperto.

Sintassi

LIB.CLOSEFORM

7.4 LIB.FORMPROP

Imposta le proprietà di un NsForm. Le proprietà possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM)

Sintassi

LIB.FORMPROP "PROPNAME" PROPVALUE (valore numerico o variabile)

PROPNAME (stringa di caratteri tra virgolette) :

VISIBLE

PropValue 0 (o < di ZERO) NsForm Invisibile

PropValue 1 (o > di ZERO) NsForm Visibile

WIDTH

PropValue Valori > di zero imposta la larghezza del NSForm

HEIGHT

PropValue Valori > di zero imposta l' altezza del NSForm

LEFT

PropValue Valori > di zero imposta la posizione nello schermo riferita al lato sinistro del NsForm

TOP

PropValue Valori > di zero imposta la posizione nello schermo riferita al lato alto del NsForm

BACKCLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore di sfondo (vedi tabella riferimento colori)

STARTPOSITION (va impostata prima dello SHOWFORM)

PropValue 0 (o < di ZERO) Viene presa la posizione di **LEFT** e **TOP**

PropValue 1 (o > di ZERO) NsForm è centrato nello schermo

WINDOWSTATE

PropValue 0 (o < di ZERO) La dimensione è impostata da **WIDTH** e **HEIGHT**

PropValue 1 (o > di ZERO) La dimensione è quella massima consentita dallo schermo.

USESTYLE

PropValue 0 Usa BackColor e ForeColor per form e tutti gli OGGETTI

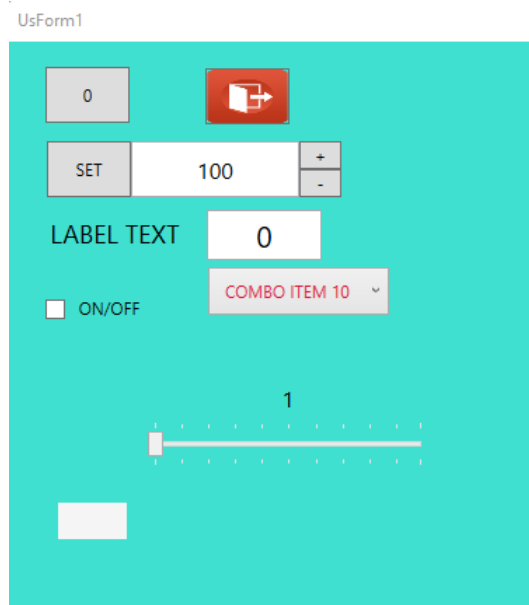
PropValue 1 Viene usato lo stile di IsoUs impostato per form e tutti gli OGGETTI.

Es:

```

LIB.FORMPROP "WIDTH" 400
LIB.FORMPROP "HEIGHT" 450
LIB.FORMPROP "LEFT" 748
LIB.FORMPROP "TOP" 293
LIB.FORMPROP "STARTPOSITION" 1
LIB.FORMPROP "WINDOWSTATE" 0
LIB.FORMPROP "BACKCOLOR" 134
LIB.FORMTEXT "UsForm1"
LIB.ADDLABEL "LbI0"
LIB.LABELPROP "LbI0" "WIDTH" 111
LIB.LABELPROP "LbI0" "HEIGHT" 34
LIB.LABELPROP "LbI0" "LEFT" 22
LIB.LABELPROP "LbI0" "TOP" 120
LIB.LABELPROP "LbI0" "DIMFONT" 18
LIB.LABELPROP "LbI0" "ALIGN" 1
LIB.LABELPROP "LbI0" "VISIBLE" 1
LIB.LABELPROP "LbI0" "ENABLED" 1
LIB.LABELPROP "LbI0" "COLORON" 113
LIB.LABELPROP "LbI0" "COLOROFF" 138
LIB.LABELTEXT "LbI0" "LABEL TEXT"
LIB.ADDITEXT "Txt0"
LIB.ITEXTPROP "Txt0" "WIDTH" 82
LIB.ITEXTPROP "Txt0" "HEIGHT" 36
LIB.ITEXTPROP "Txt0" "LEFT" 143
LIB.ITEXTPROP "Txt0" "TOP" 120
LIB.ITEXTPROP "Txt0" "DIMFONT" 21
LIB.ITEXTPROP "Txt0" "ALIGN" 1
LIB.ITEXTPROP "Txt0" "VISIBLE" 1
LIB.ITEXTPROP "Txt0" "ENABLED" 1
LIB.ITEXTPROP "Txt0" "AUTOSET" 1
LIB.ITEXTPROP "Txt0" "NDECIMAL" 3
LIB.ITEXTPROP "Txt0" "MIN" -2000000000
LIB.ITEXTPROP "Txt0" "MAX" 2000000000
LIB.ITEXTPROP "Txt0" "DEST" ${K15}
LIB.ITEXTSETVALUE "Txt0" ${K15}
LIB.ADDCHECK "Check0"
LIB.CHECKPROP "Check0" "WIDTH" 73
LIB.CHECKPROP "Check0" "HEIGHT" 46
LIB.CHECKPROP "Check0" "LEFT" 27
LIB.CHECKPROP "Check0" "TOP" 168
LIB.CHECKPROP "Check0" "DIMFONT" 12
LIB.CHECKPROP "Check0" "ALIGN" 1
LIB.CHECKPROP "Check0" "VISIBLE" 1
LIB.CHECKPROP "Check0" "AUTOSET" 1
LIB.CHECKPROP "Check0" "ENABLED" 1
LIB.CHECKSETVALUE "Check0" ${O10}
LIB.CHECKPROP "Check0" "DEST" ${O10}
LIB.CHECKTEXT "Check0" "ON/OFF"
LIB.ADDCOMBO "Combo0"
LIB.COMBOPROP "Combo0" "WIDTH" 129
LIB.COMBOPROP "Combo0" "HEIGHT" 34
LIB.COMBOPROP "Combo0" "LEFT" 144
LIB.COMBOPROP "Combo0" "TOP" 161
LIB.COMBOPROP "Combo0" "DIMFONT" 12
    
```

LIB.FORMPROP "USESTYLE" 0



LIB.FORMPROP "USESTYLE" 1



```

LIB.COMBOPROP "Combo0" "ALIGN" 1
LIB.COMBOPROP "Combo0" "VISIBLE" 1
LIB.COMBOPROP "Combo0" "ENABLED" 1
LIB.COMBOPROP "Combo0" "AUTOSET" 1
LIB.COMBOPROP "Combo0" "FOREGROUND" 19
LIB.COMBOPROP "Combo0" "DEST" $VAR1
LIB.COMBOSETVALUE "Combo0" 2
LIB.COMBOITEM "Combo0" "COMBO ITEM 10" 1
LIB.COMBOITEM "Combo0" "COMBO ITEM 20" 2
LIB.COMBOITEM "Combo0" "COMBO ITEM 30" 3
LIB.ADDSLIDER "Slider0"
LIB.SLIDERPROP "Slider0" "WIDTH" 200
LIB.SLIDERPROP "Slider0" "HEIGHT" 35
LIB.SLIDERPROP "Slider0" "LEFT" 101
LIB.SLIDERPROP "Slider0" "TOP" 237
LIB.SLIDERPROP "Slider0" "DIMFONT" 16
LIB.SLIDERPROP "Slider0" "VISIBLE" 1
LIB.SLIDERPROP "Slider0" "VISLABEL" 1
LIB.SLIDERPROP "Slider0" "ENABLED" 1
LIB.SLIDERPROP "Slider0" "AUTOSET" 1
LIB.SLIDERPROP "Slider0" "NDECIMAL" 3
LIB.SLIDERPROP "Slider0" "TICKP" 3
LIB.SLIDERPROP "Slider0" "TICKF" 10
LIB.SLIDERPROP "Slider0" "MIN" 1
LIB.SLIDERPROP "Slider0" "MAX" 100
LIB.SLIDERPROP "Slider0" "LARGE" 10
LIB.SLIDERPROP "Slider0" "SMALL" 1
LIB.SLIDERPROP "Slider0" "DEST" $PIPP01
LIB.SLIDERSETVALUE "Slider0" 0
LIB.ADDBUTTON "Btn0"
LIB.BUTTONPROP "Btn0" "WIDTH" 60
LIB.BUTTONPROP "Btn0" "HEIGHT" 40
LIB.BUTTONPROP "Btn0" "LEFT" 28
LIB.BUTTONPROP "Btn0" "TOP" 18
LIB.BUTTONPROP "Btn0" "DIMFONT" 12
LIB.BUTTONPROP "Btn0" "ALIGN" 1
LIB.BUTTONPROP "Btn0" "CLOSEFORM" 0
LIB.BUTTONPROP "Btn0" "VISIBLE" 1
LIB.BUTTONPROP "Btn0" "ENABLED" 1
LIB.BUTTONPROP "Btn0" "INPUTUPDATE" 0
LIB.BUTTONPROP "Btn0" "DEST" $VAR1
LIB.BUTTONPRINT "Btn0" "I" @VART
LIB.BUTTONPROP "Btn0" "COLORON" 113
LIB.BUTTONPROP "Btn0" "COLOROFF" 138
LIB.ADDINPUT "Input0"
LIB.INPUTPROP "Input0" "WIDTH" 120
LIB.INPUTPROP "Input0" "HEIGHT" 40
LIB.INPUTPROP "Input0" "LEFT" 29
LIB.INPUTPROP "Input0" "TOP" 71
LIB.INPUTPROP "Input0" "DIMFONT" 16
LIB.INPUTPROP "Input0" "ALIGN" 1
LIB.INPUTPROP "Input0" "VISIBLE" 1
LIB.INPUTPROP "Input0" "ENABLED" 1
LIB.INPUTPROP "Input0" "AUTOSET" 1
LIB.INPUTPROP "Input0" "NDECIMAL" 3
LIB.INPUTPROP "Input0" "MIN" -2000000000
LIB.INPUTPROP "Input0" "MAX" 2000000000
LIB.INPUTPROP "Input0" "INCREMENT" 1
LIB.INPUTPROP "Input0" "DEST" $PIPP0
LIB.INPUTSETVALUE "Input0" 100
LIB.ADDLABEL "Lbl1"
LIB.LABELPROP "Lbl1" "WIDTH" 49
LIB.LABELPROP "Lbl1" "HEIGHT" 26
LIB.LABELPROP "Lbl1" "LEFT" 37
LIB.LABELPROP "Lbl1" "TOP" 329
LIB.LABELPROP "Lbl1" "DIMFONT" 18
LIB.LABELPROP "Lbl1" "ALIGN" 1
LIB.LABELPROP "Lbl1" "VISIBLE" 1
LIB.LABELPROP "Lbl1" "ENABLED" 1
LIB.LABELF "Lbl1" "B" ${O10}
LIB.LABELPROP "Lbl1" "TIMER" 200
LIB.LABELPROP "Lbl1" "COLORON" 113
LIB.LABELPROP "Lbl1" "COLOROFF" 138
LIB.ADDBUTTON "Btn1"
LIB.BUTTONPROP "Btn1" "WIDTH" 60
LIB.BUTTONPROP "Btn1" "HEIGHT" 40
LIB.BUTTONPROP "Btn1" "LEFT" 142
LIB.BUTTONPROP "Btn1" "TOP" 19
LIB.BUTTONPROP "Btn1" "DIMFONT" 12
LIB.BUTTONPROP "Btn1" "ALIGN" 1
LIB.BUTTONPROP "Btn1" "CLOSEFORM" 1
LIB.BUTTONPROP "Btn1" "VISIBLE" 1
LIB.BUTTONPROP "Btn1" "ENABLED" 1
LIB.BUTTONPROP "Btn1" "INPUTUPDATE" 0
LIB.BUTTONPROP "Btn1" "COLORON" 113
LIB.BUTTONPROP "Btn1" "COLOROFF" 138
LIB.BUTTONTEXT "Btn1" "(@)EXIT"
LIB.SHOWFORM

```

7.5 LIB.FORMTEXT

Imposta Caption di un NsForm. Le proprietà possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM)

Sintassi

LIB.FORMTEXT "TEXT"

TEXT (stringa di caratteri tra virgolette) :

Testo (titolo) della Caption.

Es:

LIB.FORMTEXT "CAPTION"

7.6 LIB.ADDLABEL

Inserisce una NsLabel in un NsForm. Le proprietà della NsLabel possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM).

Possono essere inserite un numero a piacere di NsLabel

Sintassi

LIB.ADDLABEL "NOME"

NOME (stringa di caratteri tra virgolette) :

Nome univoco senza spazi e caratteri speciali della NsLabel

Es:

LIB.ADDLABEL "LBL"

7.7 LIB.LABELPROP

Imposta le proprietà di una NsLabel. Le proprietà possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM). Ma prima di impostare la proprietà occorre che la NsLabel sia stata inserita con l' istruzione **LIB.ADDLABEL**.

Sintassi

LIB.LABELPROP "LBLNAME" "PROPNAME" PROPVALUE (valore numerico o variabile)

LBLNAME Nome della NsLabel

PROPNAME (stringa di caratteri tra virgolette) :

VISIBLE

PropValue 0 (o < di ZERO) NsLabel Invisibile

PropValue 1 (o > di ZERO) NsLabel Visibile (default)

AUTOSIZE

PropValue 0 (o < di ZERO) dimensione presa da WHIDTH e HEIGHT

PropValue 1 (o > di ZERO) dimensione automatica in base alla lunghezza del testo (default)

ENABLED

PropValue 0 (o < di ZERO) NsLabel abilitata (default)

PropValue 1 (o > di ZERO) NsLabel disabilitata

WIDTH

PropValue Valori > di zero imposta la larghezza del NSLabel (se autosize = 0)

HEIGHT

PropValue Valori > di zero imposta l' altezza del NSLabel (se autosize = 0)

LEFT

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato sinistro del NSLabel

TOP

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato alto del NSLabel

DIMFONT

PropValue Valori compresi tra 6 e 100. Imposta la dimensione del font

BACKCLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore di sfondo (vedi tabella riferimento colori)

FORECOLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore del testo (vedi tabella riferimento colori)

COLORON

PropValue Valori compresi tra 0 e 140. Imposta il colore ON per formato ON/OFF (LABELF – LABELPRINT)

COLOROFF

PropValue Valori compresi tra 0 e 140. Imposta il colore OFF per formato ON/OFF (LABELF – LABELPRINT)

STYLE (default 0)

PropValue: FontWeight

0 Normale **1** Medium **2** Bold **3** Ultra Bold

ALIGN (default 0)

PropValue:

0 Middle Left **1** Middle Center **2** Middle Right

3 Top Left **4** Top Center **5** Top Right

6 Bottom Left **7** Bottom Center **8** Bottom Right

BORDER (default 0)

PropValue:

0 None **1** Fixed Single **2** Fixed 3D

TIMER (default 0)

PropValue Valori maggiori di zero (millisecondi) abilitano la stampa continua della variabile tramite TIMER.
Deve essere impostato **LABELF**

7.8 LIB.LABELTEXT

Scrivere il testo nella NSLabel.

Sintassi

LIB.LABELTEXT "LBLNAME" "TEXT"

LBLNAME Nome della NSLabel

TEXT (stringa di caratteri tra virgolette)

Es:

LIB.LABELTEXT "LBL" "MY LABEL"

7.9 LIB.LABELPRINT

Scrivere il valore di una Variabile Isous nel campo testo della NsLabel

Tipi di VAR: \$,;,\$[I], \$[O], \$[Q], \$[R], \$[P], \$[E], \$[S], \$[U], \$[X], \$[Y], \$[W], \$[K]

Sintassi

LIB.LABELPRINT“LBLNAME” “FORMAT” VAR


LBLNAME Nome della NsLabel


FORMAT Formato di stampa:

| | |
|---------|---|
| “B” | ON/OFF visualizza COLORON-COLOROFF se non usati STILI , oppure immagine standard IsoUs del LED ON/OFF |
| “I” | Viene visualizzata solo la parte intera del valore |
| “Fndec” | Viene visualizzato un numero di decimali indicati in Ndec |

B (USESTYLE=1)  (off)  (on)

B (USESTYLE=0)  (COLOROFF)  (COLORON)

I  33

F3  52.676

7.10 LIB.LABELF

Setta il formato della variabile per stampa tramite TIMER

Tipi di VAR: \$,;,\$[I], \$[O], \$[Q], \$[R], \$[P], \$[E], \$[S], \$[U], \$[X], \$[Y], \$[W], \$[K]

Sintassi

LIB.LABELF “LBLNAME” “FORMAT” VAR

LBLNAME Nome della NsLabel

FORMAT Formato di stampa:

| | |
|---------|---|
| “B” | ON/OFF visualizza COLORON-COLOROFF se non usati STILI , oppure immagine standard IsoUs del LED ON/OFF |
| “I” | Viene visualizzata solo la parte intera del valore |
| “Fndec” | Viene visualizzato un numero di decimali indicati in Ndec |

7.11 LIB.ADDBUTTON

Inserisce un NsButton in un NsForm. Le proprietà del NsButton possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM)

Possono essere inseriti un numero a piacere di NsButton

Sintassi

LIB.ADDBUTTON “NOME”

NOME (stringa di caratteri tra virgolette) :

Nome univoco senza spazi e caratteri speciali della NsButton

Es:

LIB.ADDBUTTON "BTN"

7.12 LIB.BUTTONPROP

Imposta le proprietà di una NsButton. Le proprietà possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM). Ma prima di impostare la proprietà occorre che NsButton sia stato inserito con l'

istruzione **LIB.ADDBUTTON**.

Sintassi

LIB.BUTTONPROP "BTNNAME" "PROPNAME" PROPVALUE (valore numerico o variabile)

BTNNAME Nome del NsButton

PROPNAME (stringa di caratteri tra virgolette) :

VISIBLE

PropValue 0 (o < di ZERO) NsButton Invisibile

PropValue 1 (o > di ZERO) NsButton Visibile (default)

ENABLED

PropValue 0 (o < di ZERO) NsButton abilitato (default)

PropValue 1 (o > di ZERO) NsButton disabilitato

WIDTH

PropValue Valori > di zero imposta la larghezza del NsButton

HEIGHT

PropValue Valori > di zero imposta l' altezza del NsButton

LEFT

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato sinistro del NsButton

TOP

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato alto del NsButton

DIMFONT

PropValue Valori compresi tra 6 e 100. Imposta la dimensione del font

BACKCLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore di sfondo (vedi tabella riferimento colori)

FORECOLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore del testo (vedi tabella riferimento colori)

COLORON

PropValue Valori compresi tra 0 e 140. Imposta il colore ON per formato ON/OFF (BUTTONF – BUTTONPRINT)

COLOROFF

PropValue Valori compresi tra 0 e 140. Imposta il colore OFF per formato ON/OFF (BUTTONF – BUTTONPRINT)

STYLE (default 0)

PropValue: FontWeight

0 Normale

1 Medium

2 Bold

3 Ultra Bold

ALIGN (default 1)

PropValue:

0 Middle Left

1 Middle Center

2 Middle Right

3 Top Left

4 Top Center

5 Top Right

6 Bottom Left

7 Bottom Center **8** Bottom Right

DEST

Variabile destinazione scrittura. Indica la variabile destinazione dove viene scritto il valore di Source

Tipi di VAR: \$,;,\$[O],\$[P],\$[K], \$[A]

SOURCE

Variabile sorgente. Valore da scrivere in DEST

Alla pressione del tasto il valore della variabile sorgente (o il valore fisso) viene scritto in DEST.

Tipi di VAR: \$,;,\$[I], \$[O], \$[Q], \$[R], \$[P], \$[E], \$[S], \$[U], \$[X], \$[Y], \$[W], \$[K]

CLOSEFORM (default 0)

PropValue 0 (o < di ZERO) Alla pressione il form non viene chiuso (default)

PropValue 1 (o > di ZERO) Alla pressione chiude il FORM

TIMER (default 0)

PropValue Valori maggiori di zero (millisecondi) abilitano la stampa continua della variabile tramite TIMER.

Deve essere impostato **BUTTONF**

INPUTUPDATE (default 0)

PropValue =1 al click tutti gli oggetti di input presenti (ITEXT,INPUT,ICHECK) aggiornano la variabile destinazione

7.13 LIB.BUTTONTEXT

Scrive il testo nel NsButton.

Sintassi

LIB.BUTTONTEXT "BTNNAME" "TEXT"

BTNNAME Nome della NsButton

TEXT (stringa di caratteri tra virgolette)

Se il nome del testo viene preceduto dal prefisso (@)text, viene visualizzata l'immagine con nome **text.png** salvata nella cartella **\DataForm\Images**

LIB.BUTTONTEXT "BTNNAME" "EXIT"



LIB.BUTTONTEXT "BTNNAME" "(@)EXIT"



(immagine **Exit.png** salvata in **\DataForm\Images**)

7.14 LIB.BUTTONPRINT

Scrive il valore di una Variabile Isous nel campo testo della NsButton

Tipi di VAR: **NUM,\$,,\$[I], \$[O], \$[Q], \$[R], \$[P], \$[E], \$[S], \$[U], \$[X], \$[Y], \$[W], \$[K]**

Sintassi

LIB.BUTTONPRINT "BTNNAME" "FORMAT" VAR

BTNNAME Nome della NsButton

FORMAT Formato di stampa:

FORMAT Print Format:

- "B" ON/OFF visualizza **COLORON-COLOROFF** se non usati **STILI**, oppure immagine standard **isoUs** del **LED ON/OFF**
- "I" Viene visualizzata solo la parte intera del valore
- "Fndec" Viene visualizzato un numero di decimali indicati in Ndec

| | | | | |
|----------------|--|------------|--|-----------|
| B (USESTYLE=1) | | (off) | | (on) |
| B (USESTYLE=0) | | (COLOROFF) | | (COLORON) |
| I | | | | |
| F3 | | | | |

7.15 LIB.BUTTONF

Setta il formato della variabile per stampa tramite TIMER

Tipi di VAR: \$,;,\$[I], \$[O], \$[Q], \$[R], \$[P], \$[E], \$[S], \$[U], \$[X], \$[Y], \$[W], \$[K]

Sintassi

LIB.BUTTONF "BTNNAME" "FORMAT" VAR

BTNNAME Nome della NsButton

FORMAT Formato di stampa:

FORMAT Print Format:

| | |
|---------|---|
| "B" | ON/OFF visualizza COLORON-COLOROFF se non usati STILI , oppure immagine standard IsoUs del LED ON/OFF |
| "I" | Viene visualizzata solo la parte intera del valore |
| "Fndec" | Viene visualizzato un numero di decimali indicati in Ndec |

7.16 LIB.ADDINPUT

Inserisce un oggetto NsInput in un NsForm. Le proprietà del NsInput possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM)

Possono essere inseriti un numero a piacere di NsInput.

L' oggetto NsInput serve esclusivamente per inserimento di valori numerici nelle variabili di Isous

Sintassi

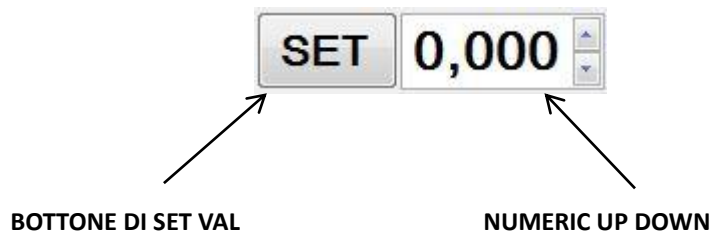
LIB.ADDINPUT"NAME"

NAME (stringa di caratteri tra virgolette) :

Nome univoco senza spazi e caratteri speciali della NsInput

Es:

LIB.ADDINPUT "INP"



Pulsante **SET** Scrive il valore nella variabile "DEST"

Set **AUTOSET=1** Il valore viene scritto in modo automatico al cambiamento del valore

7.17 LIB.INPUTPROP

Imposta le proprietà di un NsInput. Le proprietà possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM). Ma prima di impostare la proprietà occorre che NsInput sia stato inserito.

Sintassi

LIB.INPUTPROP "INPNAME" "PROPNAME" PROPVALUE (valore numerico o variabile)

INPNAME Nome del NsInput

PROPNAME (stringa di caratteri tra virgolette) :

VISIBILE

PropValue 0 (o < di ZERO) NsInput Invisibile

PropValue 1 (o > di ZERO) NsInput Visibile (default)

ENABLED

PropValue 0 (o < di ZERO) NsInput abilitato (default)

PropValue 1 (o > di ZERO) NsInput disabilitato

WIDTH

PropValue Valori > di zero imposta la larghezza del NsInput

LEFT

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato sinistro del NsInput

TOP

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato alto del NsInput

DIMFONT

PropValue Valori compresi tra 6 e 100. Imposta la dimensione del font

BACKCLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore di sfondo (vedi tabella riferimento colori)

FORECOLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore del testo (vedi tabella riferimento colori)

ALIGN (default 0)

PropValue:

0 Left **1** Center **2** Right

DEST

Variabile destinazione scrittura. Indica la variabile destinazione dove viene scritto il valore

Tipi di VAR: \$,;,\$[O],\$[P],\$[K], \$[A]

.

MIN

PropValue Valore Minimo inseribile (double)

MAX

PropValue Valore Massimo inseribile (double)

NDECIMAL

PropValue Valori > di zero *Numero di decimali visualizzati*

INCREMENT

PropValue Valore di incremento da pulsante **up down** (double)

AUTOSET

PropValue 0 (o < ZERO) Per scrivere il valore nella variabile DEST occorre premere il pulsante SET

PropValue 1 (o > ZERO) Il valore viene automaticamente scritto nella variabile dest al variare dello stesso.

7.18 LIB.INPUTSETVALUE

Imposta il valore iniziale di un oggetto NsInput

Sintassi**LIB.INPUTSETVALUE "INPNAME" VAR**

INPNAME Nome della NsInput

VAR Variabile da cui prendere il valore

Tipi di VAR: NUM,\$,;,\$[I], \$[O], \$[Q], \$[R], \$[P], \$[E], \$[S], \$[U], \$[X], \$[Y], \$[W], \$[K]

Es:

\$VAR=103.2569

LIB.INPUTSETVALUE "INP" \$VAR //IMPOSTA IL VALORE A 103.2569

7.19 LIB.ADDITEXT

Inserisce un oggetto ITEXT in un NsForm. Le proprietà di ITEXT possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM)

L' oggetto ITEXT serve esclusivamente per inserimento di valori numerici nelle variabili di Isous

Sintassi

LIB.ADDITEXT "NOME"

NOME (stringa di caratteri tra virgolette) :

Nome univoco senza spazi e caratteri speciali

Es:

LIB.ADDITEXT "TEXT1"

7.20 LIB.ITEXTPROP

Imposta le proprietà di un ITEXT. Le proprietà possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM). Ma prima di impostare la proprietà occorre che ITEXT sia stato inserito.

Sintassi

LIB.ITEXTPROP "TXTNAME" "PROPNAME" PROPVALUE (valore numerico o variabile)

TXTNAME Nome ITEXT

PROPNAME (stringa di caratteri tra virgolette) :

VISIBLE

PropValue 0 (o < di ZERO) ITEXT Invisibile

PropValue 1 (o > di ZERO) ITEXT Visibile (default)

ENABLED

PropValue 0 (o < di ZERO) ITEXT abilitato (default)

PropValue 1 (o > di ZERO) ITEXT disabilitato

WIDTH

PropValue Valori > di zero imposta la larghezza ITEXT

HEIGHT

PropValue Valori > di zero imposta l' altezza ITEXT

LEFT

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato sinistro di ITEXT

TOP

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato alto di ITEXT

DIMFONT

PropValue Valori compresi tra 6 e 100. Imposta la dimensione del font

BACKCLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore di sfondo (vedi tabella riferimento colori)

FORECOLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore del testo (vedi tabella riferimento colori)

ALIGN (default 0)

PropValue:

0 Left **1** Center **2** Right

DEST

Variabile destinazione scrittura. Indica la variabile destinazione dove viene scritto il valore

Tipi di VAR: \$,;, \$[O], \$[P], \$[K], \$[A]

MIN

PropValue Valore Minimo inseribile (double)

MAX

PropValue Valore Massimo inseribile (double)

NDECIMAL

PropValue Valori > di zero *Numero di decimali visualizzati*

AUTOSET

PropValue 0 (o < ZERO) Per scrivere il valore nella variabile DEST un aggiornamento tramite bottone UPDATEINPUT

PropValue 1 (o > ZERO) Il valore viene automaticamente scritto nella variabile dest quando premuto CR

7.21 LIB.ITEXTSETVALUE

Imposta il valore iniziale di un oggetto ITEXT

Sintassi

LIB.ITEXTSETVALUE "TXTNAME" VAR

TXTNAME Nome ITEXT

VAR Variabile da cui prendere il valore

Tipi di VAR: **NUM,\$,,\$[I], \$[O], \$[Q], \$[R], \$[P], \$[E], \$[S], \$[U], \$[X], \$[Y], \$[W], \$[K]**

Es:

\$VAR=103.2569

LIB.ITEXTSETVALUE "TXT" \$VAR //IMPOSTA IL VALORE A 103.2569

7.22 LIB.ADDCHECK

Inserisce un oggetto CHECK in un NsForm. Le proprietà di CHECK possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM)

L' oggetto CHECK serve esclusivamente per inserimento di valori ON/OFF se valore 0 CHECK UNCHECKED se valore >0 CHECK CHECKED

Sintassi

LIB.ADDCHECK "NOME"

NOME (stringa di caratteri tra virgolette) :

Nome univoco senza spazi e caratteri speciali

Es:

LIB.ADDCHECK "CHECK1"

7.23 LIB.CHECKPROP

Imposta le proprietà di un CHECK. Le proprietà possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM). Ma prima di impostare la proprietà occorre che CHECK sia stato inserito.

Sintassi

LIB.CHECKPROP "CHECKNAME" "PROPNAME" PROPVALUE (valore numerico o variabile)

CHECKNAME Nome CHECK

PROPNAME (stringa di caratteri tra virgolette) :

VISIBLE

PropValue 0 (o < di ZERO) CHECK Invisibile

PropValue 1 (o > di ZERO) CHECK Visibile (default)

ENABLED

PropValue 0 (o < di ZERO) CHECK abilitato (default)

PropValue 1 (o > di ZERO) CHECK disabilitato

WIDTH

PropValue Valori > di zero imposta la larghezza del CHECK

HEIGHT

PropValue Valori > di zero imposta altezza del CHECK

LEFT

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato sinistro di CHECK

TOP

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato alto di CHECK

DIMFONT

PropValue Valori compresi tra 6 e 100. Imposta la dimensione del font

BACKCLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore di sfondo (vedi tabella riferimento colori)

FORECOLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore del testo (vedi tabella riferimento colori)

ALIGN (default 0)

PropValue:

0 Left **1** Center **2** Right

DEST

Variabile destinazione scrittura. Indica la variabile destinazione dove viene scritto il valore

Tipi di VAR: \$,;,\$[O],\$[P],\$[K], \$[A]

AUTOSET

PropValue 0 (o < ZERO) Per scrivere il valore nella variabile DEST un aggiornamento tramite bottone UPDATEINPUT

PropValue 1 (o > ZERO) Il valore viene automaticamente scritto nella variabile dest al CLICK

7.24 LIB.CHECKSETVALUE

Imposta il valore iniziale di un oggetto CHECK

Sintassi

LIB.CHECKSETVALUE "CHECKNAME" VAR

CHECKNAME Nome ITEXT

VAR Variabile da cui prendere il valore

Tipi di VAR: NUM,\$,;,\$[I], \$[O], \$[Q], \$[R], \$[P], \$[E], \$[S], \$[U], \$[X], \$[Y], \$[W], \$[K]

Es:

\$VAR=1

LIB.CHECKSETVALUE "CHECK" \$VAR //IMPOSTA IL VALORE A 1 CHECKED

7.25 LIB.CHECKTEXT

Caption Check

Sintassi

LIB.CHECKTEXT "CHECKNAME" "TEXT"

CHECKNAME Nome CHECK

TEXT (stringa di caratteri tra virgolette)

7.26 LIB.ADDCOMBO

Inserisce un oggetto COMBO in un NsForm. Le proprietà di CHECK possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM)

L' oggetto COMBO serve esclusivamente per inserimento di valori predefiniti contenuti negli ITEMS del COMBO

Sintassi

LIB.ADDCOMBO "NOME"

NOME (stringa di caratteri tra virgolette) :

Nome univoco senza spazi e caratteri speciali

Es:

LIB.ADDCOMBO "COMBO1"

7.27 LIB.COMBOPROP

Imposta le proprietà di un COMBO. Le proprietà possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM). Ma prima di impostare la proprietà occorre che COMBO sia stato inserito.

Sintassi

LIB.COMBOPROP "COMBONAME" "PROPNAME" PROPVALUE (valore numerico o variabile)

COMBONAME Nome COMBO

PROPNAME (stringa di caratteri tra virgolette) :

VISIBLE

PropValue 0 (o < di ZERO) COMBO Invisibile

PropValue 1 (o > di ZERO) COMBO Visibile (default)

ENABLED

PropValue 0 (o < di ZERO) COMBO abilitato (default)

PropValue 1 (o > di ZERO) COMBO disabilitato

WIDTH

PropValue Valori > di zero imposta la larghezza del COMBO

HEIGHT

PropValue Valori > di zero imposta altezza del COMBO

LEFT

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato sinistro di COMBO

TOP

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato alto di COMBO

DIMFONT

PropValue Valori compresi tra 6 e 100. Imposta la dimensione del font

FORECOLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore del testo (vedi tabella riferimento colori)

ALIGN (default 0)

PropValue:

0 Left **1** Center **2** Right

DEST

Variabile destinazione scrittura. Indica la variabile destinazione dove viene scritto il valore dell' ITEM

Tipi di VAR: \$,;, \$[O], \$[P], \$[K], \$[A]

AUTOSET

PropValue 0 (o < ZERO) Per scrivere il valore nella variabile DEST un aggiornamento tramite bottone UPDATEINPUT

PropValue 1 (o > ZERO) Il valore viene automaticamente scritto nella variabile dest al set ITEM

7.28 LIB.COMBOSETVALUE

Inizializza il COMBO all' ITEM con relativo valore

Sintassi

LIB.COMBOSETVALUE "COMBONAME" VAR

COMBONAME Nome COMBO

VAR Variabile da cui prendere il valore

Tipi di VAR: **NUM,\$,;**

Es:

\$VAR=2

LIB.COMBOSETVALUE "COMBO" \$VAR //IMPOSTA IL VALORE

7.29 LIB.COMBOITEM

Aggiunge un ITEM al COMBO

Sintassi

LIB.COMBOITEM "COMBONAME" "ITEMNAME" VAR

COMBONAME Nome COMBO

ITEMNAME Testo ITEM

VAR Variabile da cui prendere il valore dell' ITEM

Tipi di VAR: **NUM,\$,;**

Es:

LIB.COMBOITEM "COMBO" "ITEM 1" 1 //ITEM 1 IMPOSTA VALORE 1

LIB.COMBOITEM "COMBO" "ITEM 2" 2 //ITEM 2 IMPOSTA VALORE 2

LIB.COMBOITEM "COMBO" "ITEM 3" 3 //ITEM 3 IMPOSTA VALORE 3

7.30 LIB.ADDSLIDER

Inserisce un oggetto SLIDER in un NsForm. Le proprietà di SLIDER possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM)

L' oggetto SLIDER serve esclusivamente per inserimento di valori di Variabili tramite cursore

Sintassi

LIB.ADDSLIDER "NOME"

NOME (stringa di caratteri tra virgolette) :

Nome univoco senza spazi e caratteri speciali

Es:

LIB.ADDSLIDER"SLIDER1"

7.31 LIB.SLIDERPROP

Imposta le proprietà di uno SLIDER. Le proprietà possono essere impostate prima che l' NsForm sia visualizzato (prima dell' istruzione LIB.SHOWFORM). Ma prima di impostare la proprietà occorre che SLIDER sia stato inserito.

Sintassi

LIB.SLIDERPROP "SLIDERNAME" "PROPNAME" PROPVALUE (valore numerico o variabile)

SLIDERNAME Nome SLIDER

PROPNAME (stringa di caratteri tra virgolette) :

VISIBLE

PropValue 0 (o < di ZERO) SLIDER Invisibile

PropValue 1 (o > di ZERO) SLIDER Visibile (default)

VISLABEL

PropValue 0 (o < di ZERO) LABEL SLIDER Invisibile

PropValue 1 (o > di ZERO) LABEL SLIDER Visibile (default)

ENABLED

PropValue 0 (o < di ZERO) SLIDER abilitato (default)

PropValue 1 (o > di ZERO) SLIDER disabilitato

WIDTH

PropValue Valori > di zero imposta la larghezza dello SLIDER

HEIGHT

PropValue Valori > di zero imposta l' altezza dello SLIDER

LEFT

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato sinistro di SLIDER

TOP

PropValue Valori > di zero imposta la posizione nel NsForm riferita al lato alto di SLIDER

DIMFONT

PropValue Valori compresi tra 6 e 100. Imposta la dimensione del font

BACKCLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore di sfondo (vedi tabella riferimento colori)

FORECOLOR

PropValue Valori compresi tra 0 e 140. Imposta il colore del testo (vedi tabella riferimento colori)

LARGE

PropValue definisce la quantità di variazione quando la casella di scorrimento subisce una variazione notevole

SMALL

PropValue definisce la quantità di variazione quando la casella di scorrimento subisce una variazione leggera

TICKP

PropValue definisce la posizione dei segni di graduazione:

0 None **1** TopLeft **2** BottomRight **3** Both

TICKF

PropValue definisce la frequenza dei segni di graduazione:

DEST

Variabile destinazione scrittura. Indica la variabile destinazione dove viene scritto il valore

Tipi di VAR: \$,;,\$[O],\$[P],\$[K], \$[A]

MIN

PropValue Valore Minimo (double)

MAX

PropValue Valore Massimo (double)

NDECIMAL

PropValue Valori > di zero *Numero di decimali visualizzati*

AUTOSET

PropValue 0 (o < ZERO) Per scrivere il valore nella variabile DEST un aggiornamento tramite bottone UPDATEINPUT

PropValue 1 (o > ZERO) Il valore viene automaticamente scritto alla variazione della barra di scorrimento

7.32 LIB.SLIDERSETVALUE

Imposta il valore iniziale di un oggetto SLIDER

Sintassi**LIB.SLIDERSETVALUE "SLIDERNAME" VAR**

SLIDERKNAME Nome SLIDER

VAR Variabile da cui prendere il valore

Tipi di VAR: **NUM,\$,;,\$[I], \$[O], \$[Q], \$[R], \$[P], \$[E], \$[S], \$[U], \$[X], \$[Y], \$[W], \$[K]**

Es:

\$VAR=100

LIB.SLIDERSETVALUE "SLIDER" \$VAR //IMPOSTA IL VALORE A 100

7.33 LIB.GETVAR

Permette di leggere una variabile FORM (tipo double) in una variabile IsoUs.

Le variabili FORM sono proprietarie del FORM in uso e sono presenti un massimo di 1000 variabili.

La Variabile 999 indica se il FORM è Aperto

999=1 Form Aperto

999=0 Form Chiuso

Sintassi**LIB.GETVAR SOURCEINDEX DEST**

SOURCEINDEX Indice della variabile FORM (da 0 a 999) da leggere, può essere un numero o una variabile IsoUs

DEST Variabile IsoUs di destinazione

Es:

\$VARFORM=100

LIB.GETVAR \$VARFORM \$VAR //LEGGE LA VARIABILE 100 IN \$VAR

LIB.GETVAR 10 \$VAR //LEGGE LA VARIABILE 10 IN \$VAR

7.34 LIB.SETVAR

Permette di scrivere una variabile FORM (tipo double) da una variabile IsoUs.

Le variabili FORM sono proprietarie del FORM in uso e sono presenti un massimo di 1000 variabili

Sintassi**LIB.SETVAR DESTINDEX SOURCE**

DESTINDEX Indice della variabile FORM (da 0 a 999) da scrivere, può essere un numero o una variabile IsoUs

SOURCE Variabile IsoUs sorgente

Es:

\$VARFORM=100

\$SOURCE=10.123

LIB.SETVARVAR \$VARFORM \$ SOURCE //SCRIVE LA VARIABILE 100 AL VALORE 10.123

LIB.SETVAR 10 \$ SOURCE //LEGGE LA VARIABILE 10 AL VALORE 10.123

7.35 LIB.DEBUG

Permette di scrivere del Testo e Valori di variabili direttamente da Gcode in una finestra.
Utilizzato per Debug del Gcode

Sintassi

LIB.DEBUG "TESTO" VAR

TESTO Testo da scrivere

VAR Variabile da scrivere o valore

Es:

\$VAR=10

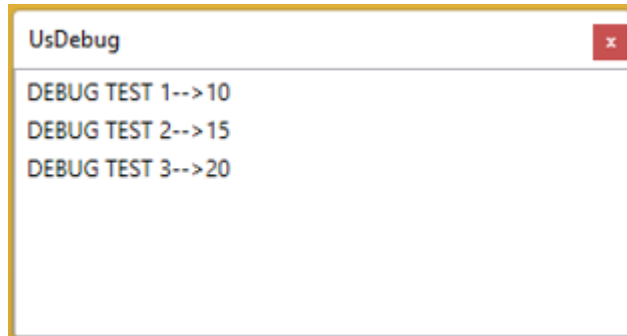
LIB.DEBUG "DEBUG TEST 1" \$VAR

\$VAR=15

LIB.DEBUG "DEBUG TEST 2" \$VAR

\$VAR=20









































LIB.DEBUG "DEBUG TEST 3" \$VAR


























```
UsDebug
DEBUG TEST 1-->10
DEBUG TEST 2-->15
DEBUG TEST 3-->20
```

7.36 TABELLA RIFERIMENTO COLORI

Di seguito viene rappresentata la tabella di riferimento colori da impostare nelle varie proprietà.

| | | | | | | | | |
|----|----------------|---|----|----------------------|---|-----|-------------------|---|
| 0 | ALICEBLUE |  | 35 | DARKSLATEGRAY |  | 70 | LIGHTSALMON |  |
| 1 | ANTIQUEWHITE |  | 36 | DARKTURQUOISE |  | 71 | LIGHTSEAGREEN |  |
| 2 | AQUA |  | 37 | DARKVIOLET |  | 72 | LIGHTSKYBLUE |  |
| 3 | AQUAMARINE |  | 38 | DEEPPINK |  | 73 | LIGHTSLATEGRAY |  |
| 4 | AZURE |  | 39 | DEEPSKYBLUE |  | 74 | LIGHTSTEELBLUE |  |
| 5 | BEIGE |  | 40 | DIMGRAY |  | 75 | LIGHTYELLOW |  |
| 6 | BISQUE |  | 41 | DODGERBLUE |  | 76 | LIME |  |
| 7 | BLACK |  | 42 | FIREBRICK |  | 77 | LIMEGREEN |  |
| 8 | BLANCHEDALMOND |  | 43 | FLORALWHITE |  | 78 | LINEN |  |
| 9 | BLUE |  | 44 | FORESTGREEN |  | 79 | MAGENTA |  |
| 10 | BLUEVIOLET |  | 45 | FUCHSIA |  | 80 | MAROON |  |
| 11 | BROWN |  | 46 | GAINSBORO |  | 81 | MEDIUMAQUAMARINE |  |
| 12 | BURLYWOOD |  | 47 | GHOSTWHITE |  | 82 | MEDIUMBLUE |  |
| 13 | CADETBLUE |  | 48 | GOLD |  | 83 | MEDIUMORCHID |  |
| 14 | CHARTREUSE |  | 49 | GOLDENROD |  | 84 | MEDIUMPURPLE |  |
| 15 | CHOCOLATE |  | 50 | GRAY |  | 85 | MEDIUMSEAGREEN |  |
| 16 | CORAL |  | 51 | GREEN |  | 86 | MEDIUMSLATEBLUE |  |
| 17 | CORNFLOWERBLUE |  | 52 | GREENYELLOW |  | 87 | MEDIUMSPRINGGREEN |  |
| 18 | CORNSILK |  | 53 | HONEYDEW |  | 88 | MEDIUMTURQUOISE |  |
| 19 | CRIMSON |  | 54 | HOTPINK |  | 89 | MEDIUMVIOLETRED |  |
| 20 | CYAN |  | 55 | INDIANRED |  | 90 | MIDNIGHTBLUE |  |
| 21 | DARKBLUE |  | 56 | INDIGO |  | 91 | MINTCREAM |  |
| 22 | DARKCYAN |  | 57 | IVORY |  | 92 | MISTYROSE |  |
| 23 | DARKGOLDENROD |  | 58 | KHAKI |  | 93 | MOCCASIN |  |
| 24 | DARKGRAY |  | 59 | LAVENDER |  | 94 | NAVAJOWHITE |  |
| 25 | DARKGREEN |  | 60 | LAVENDERBLUSH |  | 95 | NAVY |  |
| 26 | DARKKHAKI |  | 61 | LAWNGREEN |  | 96 | OLDLACE |  |
| 27 | DARKMAGENTA |  | 62 | LEMONCHIFFON |  | 97 | OLIVE |  |
| 28 | DARKOLIVEGREEN |  | 63 | LIGHTBLUE |  | 98 | OLIVEDRAB |  |
| 29 | DARKORANGE |  | 64 | LIGHTCORAL |  | 99 | ORANGE |  |
| 30 | DARKORCHID |  | 65 | LIGHTCYAN |  | 100 | ORANGERED |  |
| 31 | DARKRED |  | 66 | LIGHTGOLDENRODYELLOW |  | 101 | ORCHID |  |
| 32 | DARKSALMON |  | 67 | LIGHTGRAY |  | 102 | PALEGOLDENROD |  |
| 33 | DARKSEAGREEN |  | 68 | LIGHTGREEN |  | 103 | PALEGREEN |  |
| 34 | DARKSLATEBLUE |  | 69 | LIGHTPINK |  | 104 | PALETURQUOISE |  |

| | | | | | |
|-----|---------------|---|-----|-------------|---|
| 105 | PALEVIOLETRED |  | 140 | YELLOWGREEN |  |
| 106 | PAPAYAWHIP |  | | | |
| 107 | PEACHPUFF |  | | | |
| 108 | PERU |  | | | |
| 109 | PINK |  | | | |
| 110 | PLUM |  | | | |
| 111 | POWDERBLUE |  | | | |
| 112 | PURPLE |  | | | |
| 113 | RED |  | | | |
| 114 | ROSYBROWN |  | | | |
| 115 | ROYALBLUE |  | | | |
| 116 | SADDLEBROWN |  | | | |
| 117 | SALMON |  | | | |
| 118 | SANDYBROWN |  | | | |
| 119 | SEAGREEN |  | | | |
| 120 | SEASHELL |  | | | |
| 121 | SIENNA |  | | | |
| 122 | SILVER |  | | | |
| 123 | SKYBLUE |  | | | |
| 124 | SLATEBLUE |  | | | |
| 125 | SLATEGRAY |  | | | |
| 126 | SNOW |  | | | |
| 127 | SPRINGGREEN |  | | | |
| 128 | STEELBLUE |  | | | |
| 129 | TAN |  | | | |
| 130 | TEAL |  | | | |
| 131 | THISTLE |  | | | |
| 132 | TOMATO |  | | | |
| 133 | TRANSPARENT |  | | | |
| 134 | TURQUOISE |  | | | |
| 135 | VIOLET |  | | | |
| 136 | WHEAT |  | | | |
| 137 | WHITE |  | | | |
| 138 | WHITESMOKE |  | | | |
| 139 | YELLOW |  | | | |

8 Istruzioni Estese

8.1 EXD.STL_LOAD

Permette di caricare un File STL (Modello 3D) nel Preview che viene preso come riferimento per la simulazione, rendendo più reale. Valido solo se utilizzata la simulazione **RealMachine**.

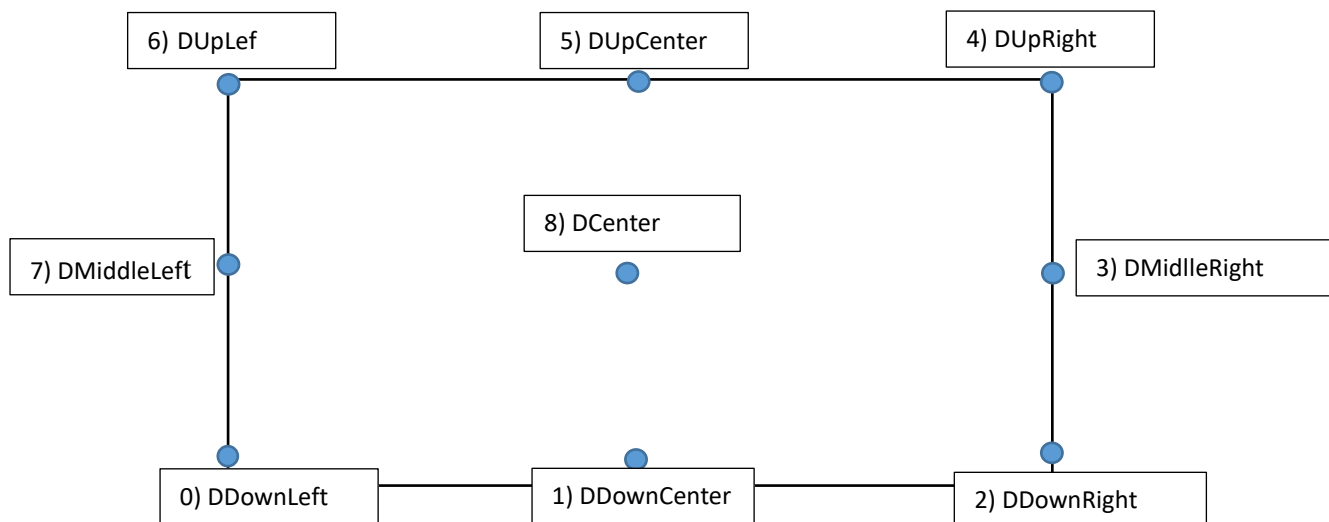
Sintassi

EXD.STL_LOAD "STLNAME" Xval Yval Zval Aval

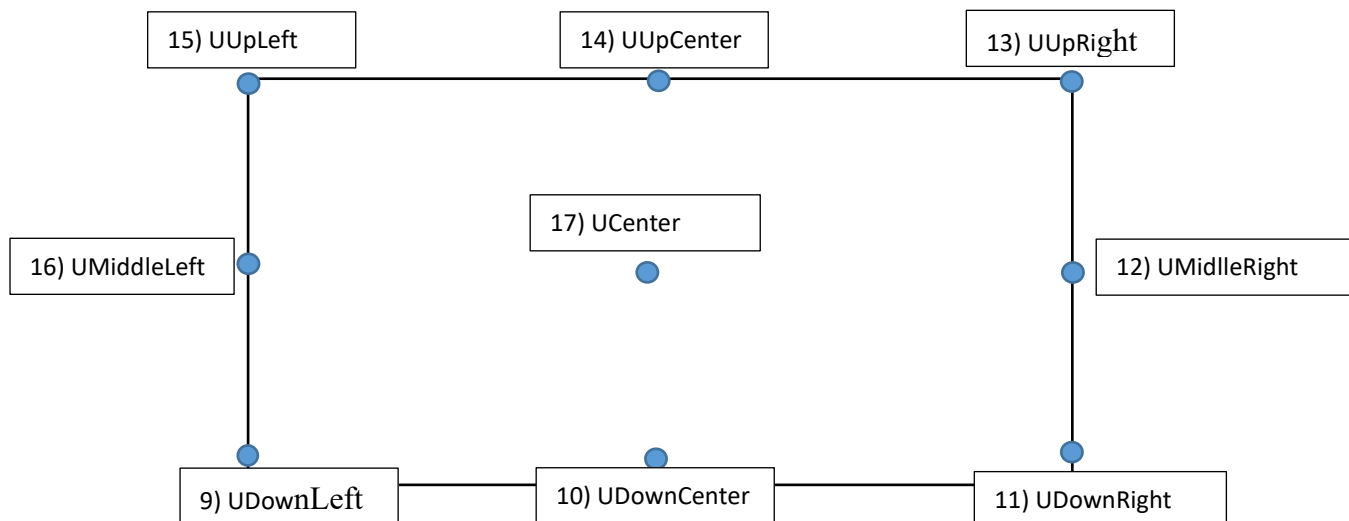
| | |
|----------------|--|
| STLNAME | Nome del file STL (senza estensione) il file deve trovarsi nella cartella \UsMachines\DynamicStl |
| X | Opzionale Centro X file STL (se omissso viene preso centro in MachineSettings) |
| Y | Opzionale Centro Y file STL (se omissso viene preso centro in MachineSettings) |
| Z | Opzionale Centro Z file STL (se omissso viene preso centro in MachineSettings) |
| A | Opzionale Allineamento STL: (se omissso viene preso in MachineSettings) |

VALORI PER PARAMETRO A

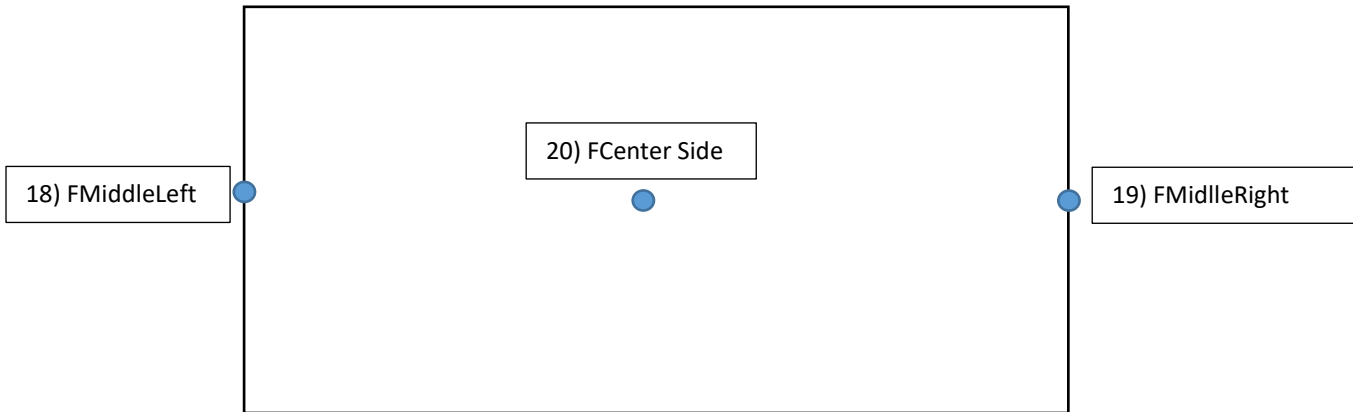
SIDE DOWN



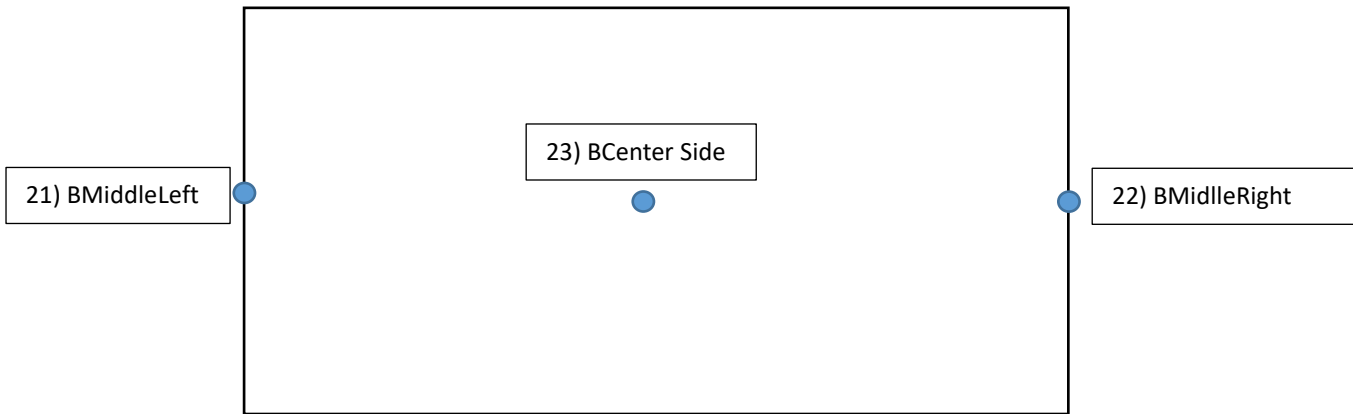
SIDE UP



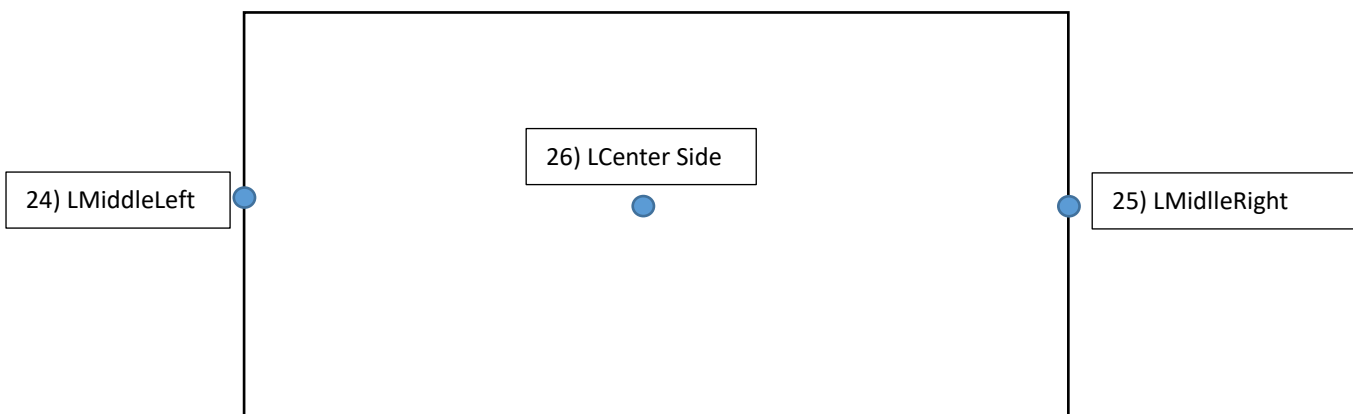
SIDE FRONT



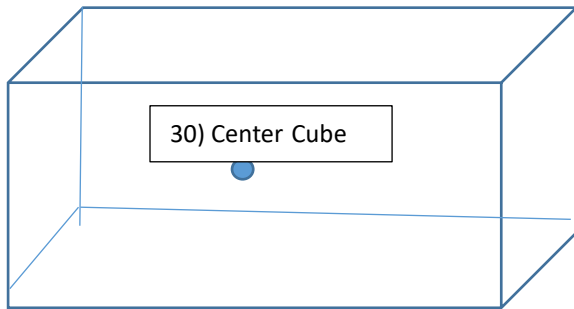
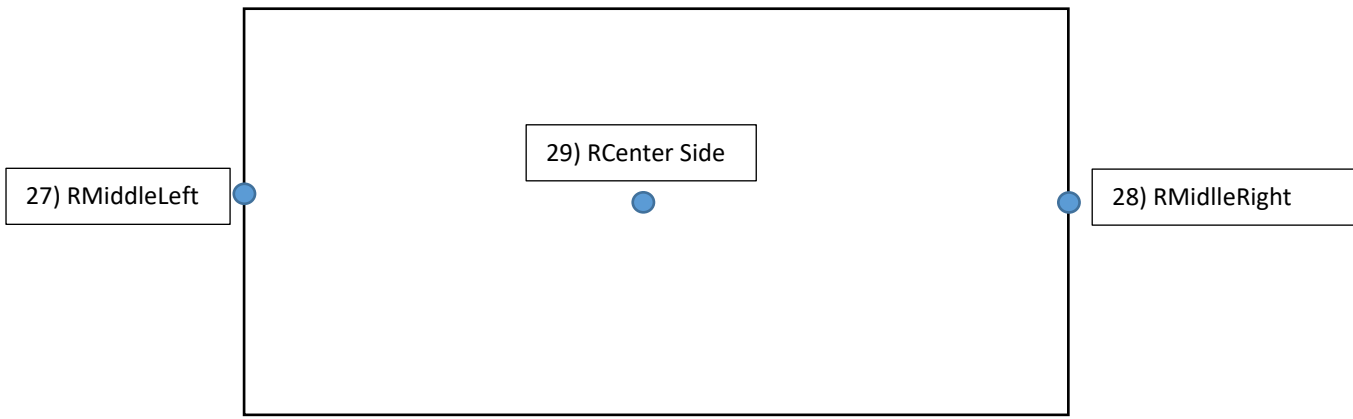
SIDE BACK



SIDE LEFT



SIDE RIGHT

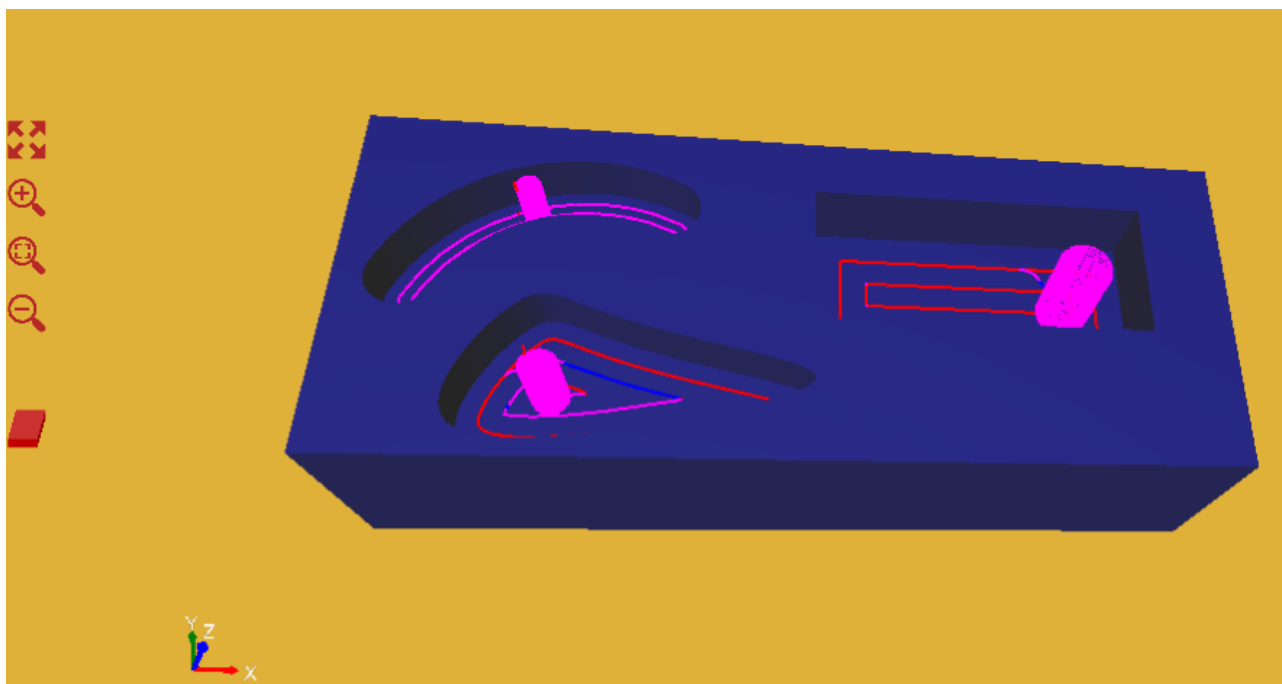


31) Default definito in XML

32) Model Origin – Origine presa da STL

Es:

EXD.STL_LOAD "TEST"



8.2 EXD.READ_TOOLPAR

Permette di leggere un parametro della tabella utensile. Analogo ad \$(Un] ma permette di indirizzare qualsiasi indice della tabella e non solo quella selezionata con Tn.

Sintassi

EXD.READ_TOOLPAR TabIndex ParIndex \$DEST

\$TabIndex Indice della tabella utensile desiderata (valore numerico o variabile)
\$ParIndex Indice del parametro (valore numerico o variabile)
\$DEST Variabile destinazione

Es:

\$TABINDEX=2

\$PARINDEX=1

EXD.READ_TOOLPAR 2 1 \$Dest //LEGGE IL PARAMETRO 1 DIAMETRO UTENSILE DALLA TABELLA 2

EXD.READ_TOOLPAR \$TABINDEX \$PARINDEX \$Dest

8.3 EXD.READ_HEADPAR

Permette di leggere un parametro della tabella teste. Analogo ad \$(Hn] ma permette di indirizzare qualsiasi indice della tabella e non solo quella selezionata con Hn.

Sintassi

EXD.READ_HEADPAR TabIndex ParIndex \$DEST

\$TabIndex Indice della tabella teste desiderata (valore numerico o variabile)
\$ParIndex Indice del parametro (valore numerico o variabile)
\$DEST Variabile destinazione

Es:

\$TABINDEX=2

\$PARINDEX=1

EXD.READ_HEADPAR 2 1 \$Dest //LEGGE IL PARAMETRO 1 DALLA TABELLA 2

EXD.READ_HEADPAR \$TABINDEX \$PARINDEX \$Dest

8.4 EXD.WRITE_TOOLPAR

Permette di scrivere un parametro della tabella utensile. Analogo ad \$(Un] ma permette di indirizzare qualsiasi indice della tabella e non solo quella selezionata con Tn.

Sintassi

EXD.WRITE_TOOLPAR TabIndex ParIndex \$SOURCE

\$TabIndex Indice della tabella utensile desiderata (valore numerico o variabile)
\$ParIndex Indice del parametro (valore numerico o variabile)
\$SOURCE Variabile sorgente del valore da scrivere

Es:

\$TABINDEX=2

\$PARINDEX=1

\$SOURCE =10

EXD.WRITE_TOOLPAR 2 1 \$SOURCE //SCRIVE IL PARAMETRO 1 DIAMETRO UTENSILE DELLA TABELLA 2 AL VALORE 10

EXD.WRITE_TOOLPAR \$TABINDEX \$PARINDEX \$SOURCE

8.5 EXD.WRITE_HEADPAR

Permette di scrivere un parametro della tabella teste. Analogo ad \$[Hn] ma permette di indirizzare qualsiasi indice della tabella e non solo quella selezionata con Hn.

Sintassi

EXD.WRITE_HEADPAR TabIndex ParIndex \$SOURCE

\$TabIndex Indice della tabella teste desiderata (valore numerico o variabile)

\$ParIndex Indice del parametro (valore numerico o variabile)

\$SOURCE Variabile sorgente del valore da scrivere

Es:

\$TABINDEX=2

\$PARINDEX=1

\$SOURCE =10

EXD.WRITE_HEADPAR 2 1 \$SOURCE //SCRIVE IL PARAMETRO 1 DELLA TABELLA 2 AL VALORE 10

EXD.WRITE_HEADPAR \$TABINDEX \$PARINDEX \$SOURCE

8.6 EXD.LOAD_LAST

Carica in una variabile l' indice dell' Ultima Tabella utensile selezionata con Tn salvata nel file ("UTENSILE.INF")

Sintassi

EXD.LOAD_LAST \$DEST

\$DEST Variabile destinazione

Es:

EXD.LOAD_LAST \$DEST

8.7 EXD.SAVE_LAST

Salva nel file ("UTENSILE.INF") la corrente Tabella utensile selezionata con Tn

Sintassi

EXD.SAVE_LAST

Es:

EXD.SAVE_LAST

8.8 EXD.RESET_LAST

Azzerà il file ("UTENSILE.INF") Nessun utensile usato

Sintassi

EXD.RESET_LAST

Es:

EXD.RESET_LAST

8.9 EXD.SET_OUT

Set/Reset di un uscita digitale tramite Indice. Come **\$(On)** ma vengono utilizzate delle variabili per indice uscita e stato

Sintassi

EXD.SET_OUT \$OutIndex \$State

Es:

\$INDEX=10

\$STATE=1

EXD.SET_OUT \$INDEX \$STATE //SET USCITA NUMERO 10

8.10 EXD.READ_OUT

Read di un uscita digitale tramite Indice. Come **\$(On)** ma vengono utilizzate delle variabili per indice uscita

Sintassi

EXD.READ_OUT \$OutIndex \$Dest

Es:

\$INDEX=10

EXD.READ_OUT \$INDEX \$DEST //LEGGE USCITA NUMERO 10 IN \$DEST

8.11 EXD.READ_INP

Legge un ingresso digitale tramite Indice. Come **\$(In)** ma vengono utilizzate delle variabili per indice ingresso

Sintassi

EXD.READ_INP \$InpIndex \$Dest

Es:

\$INDEX=10

EXD.READ_INP \$INDEX \$DEST //LEGGE INGRESSO NUMERO 10 IN \$DEST

8.12 EXD.WRITE_USER

Scrive una Variabile User Generic sul CN tramite Indice. Come **\$(Kn)** ma vengono utilizzate delle variabili per indice variabile e valore

Sintassi

EXD.WRITE_USER \$Index \$Value

Es:

\$INDEX=10

\$VALUE=100

EXD.WRITE_USER \$INDEX \$VALUE //SCRIVE USER 10 AL VALORE 100

8.13 EXD.READ_USER

Legge una Variabile User Generic dal CN tramite Indice. Come $\$(Kn)$ ma viene utilizzata una variabile per indice

Sintassi

EXD.READ_USER $\$Index$ $\$Dest$

Es:

$\$INDEX=10$

EXD.REA_USER $\$INDEX$ $\$DETS$ //LEGGE USER 10 IN DEST

8.14 EXD.READ_DEMAND

Legge la demand position di un asse. Come $\$(Qn)$ ma vengono utilizzate delle variabili per indice asse

Come per $\$(Qn)$ se valore <100 cioè da 0 a 8 viene presa la posizione comprensiva di OFFSET (Zero pezzo ecc.)

Se valore >= 100 cioè da 100 a 108 viene presa la posizione senza OFFSET

Sintassi

EXD.READ_DEMAND $\$Axis$ $\$Dest$

Es:

$\$Axis=1$

EXD.READ_DEMAND $\$AXIS$ $\$DEST$ //LEGGE LA DEMAND POS ASSE 1 CON OFFSET IN $\$DEST$

$\$Axis=101$

EXD.READ_DEMAND $\$AXIS$ $\$DEST$ //LEGGE LA DEMAND POS ASSE 1 SENZA OFFSET IN $\$DEST$

8.15 EXD.READ_REAL

Legge la real position di un asse. Come $\$(Rn)$ ma vengono utilizzate delle variabili per indice asse

Come per $\$(Rn)$ se valore <100 cioè da 0 a 8 viene presa la posizione comprensiva di OFFSET (Zero pezzo ecc.)

Se valore >= 100 cioè da 100 a 108 viene presa la posizione senza OFFSET

Sintassi

EXD.READ_REAL $\$Axis$ $\$Dest$

Es:

$\$Axis=1$

EXD.READ_REAL $\$AXIS$ $\$DEST$ //LEGGE LA REAL POS ASSE 1 CON OFFSET IN $\$DEST$

$\$Axis=101$

EXD.READ_REAL $\$AXIS$ $\$DEST$ //LEGGE LA REAL POS ASSE 1 SENZA OFFSET IN $\$DEST$

8.16 EXD.MASK

Setta una maschera per le Uscite Digitali in modo da condizionarne lo stato.

Sintassi

EXD.MASK Type OutIndex

| | |
|-----------------|---|
| Type | Variabile o valore numerico che definisce il tipo di maschera |
| Type=0 | Forza uscita a OFF |
| Type=1 | Forza Uscita ad ON |
| Type=2 | Inverte stato Uscita |
| Type=3 | Reset della maschera |
| OutIndex | Indice dell' uscita digitale |

Es:

```
$TYPE=0
```

```
$OUTINDEX=10
```

```
EXD.MASK $TYPE $OUTINDEX //FORZA USCITA 10 A OFF
```

```
EXD.MASK 3 10 //RESET MASK OUT 10
```

8.17 EXD.WRITE_BIT

Set/Reset bit di una variabile

Sintassi

EXD.WRITE_BIT \$BIT \$VAR \$VAL

| | |
|------------|--|
| BIT | Variabile o Valore che contiene il numero di bit (da 0 a 31) |
| VAR | Variabile destinazione |
| VAL | Variabile o Valore che contiene lo stato del Bit (0 Reset 1 Set) |

Es:

```
$BIT=5
```

```
$VAL=1
```

```
EXD.WRITE_BIT $BIT $VARDEST $VAL //SET BIT 5 DI VARDEST
```

```
EXD.WRITE_BIT 5 $VARDEST 0 //RESET BIT 5 DI VARDEST
```

8.18 EXD.READ_BIT

Legge lo stato del bit di una variabile

Sintassi

EXD.READ_BIT \$BIT \$VAR \$DEST

| | |
|------------|--|
| BIT | Variabile o Valore che contiene il numero di bit (da 0 a 31) |
| VAR | Variabile Sorgente |
| VAL | Variabile destinazione che contiene lo stato del bit letto (0 o 1) |

Es:

```
$BIT=5
```

```
EXD.READ_BIT $BIT $VARSOURCE $VARDEST //LEGGE BIT 5 DI VARSOURCE
```

```
EXD.READ_BIT 5 $VARSOURCE $VARDEST // LEGGE BIT 5 DI VARSOURCE
```

8.19 EXD.PXV_ALL_DETECTORS

Legge lo stato di tutti i detectors settati del sistema PXV

Es. per leggere utensili in rastrelliera

Sintassi

EXD.PXV_ALL_DETECTORS \$SHOWMODE \$TOOLS \$ISOK

SHOWMODE Determina se deve essere mostrata un immagine nella simulazione di PXV Variabile o valore
Usata per Debug

SHOWMODE=0 Nessuna Immagine mostrata

SHOWMODE=1 Viene Mostrata l' immagine **FILTRATA**

SHOWMODE=2 Viene Mostrata l' immagine **ORIGINALE**

TOOLS Ritorna lo stato dei detectors mappati a bit (max 32 detectors da 0-31)

Bit=0 Detector non attivo

Bit=1 Detector attivo

ISOK Se =-1 **ERRORE LETTURA DETECTORS** Altrimenti contiene il numero dei detectors letti

Es:

```
EXD.PXV_ALL_DETECTORS 0 $TOOLS $ISOK //LEGGE I DETECTORS
IF $ISOK<>-1 //LETTURA OK
    EXD.READ_BIT 0 $TOOLS $VARDEST //LEGGE DETECTOR 0
    IF $VARDEST=1 //ATTVO
        ....
    ELSE //NON ATTVO
        ....
    END_IF
END_IF
```

8.20 EXD.PXV_SINGLE_DETECTOR

Legge lo stato di un detector settato nel sistema PXV

Sintassi

EXD.PXV_SINGLE_DETECTOR \$NRDET \$SHOWMODE \$VAR

NRDET Numero del detector da leggere (da 0-31) Variabile o Valore

SHOWMODE Vedi EXD.PXV_ALL_DETECTORS

\$VAR Valore letto

-1 ERRORE

0 NON ATTVO

1 ATTIVO

Es:

```
EXD.PXV_SINGLE_DETECTOR 0 0 $RET //LEGGE IL DETECTOR 0
IF $RET<>-1 //LETTURA OK
    IF $RET=1 //ATTVO
        ....
    ELSE //NON ATTVO
        ....
    END_IF
END_IF
```

8.21 EXD.PXV_READ_PROBE

Legge le coordinate di una Probe settato nel sistema PXV
Ritorna X,Y dove si trova la probe

Sintassi

EXD.PXV_READ_PROBE \$PROBE \$XVAL \$YVAL \$ISOK

PROBE Numero della Probe da leggere Variabile o Valore
XVAL Se ISOK=1 Posizione X della Probe
YVAL Se ISOK=1 Posizione Y della Probe
ISOK Se =-1 ERRORE LETTURA PROBE

Es:

```
EXD.PXV_READ_PROBE 0 $VX $VY $ISOK //LEGGE LA PROBE 0
IF $ISOK=1 //LETTURA OK
    GO X[$VX] Y[$VY] //SPOSTA ALLA POSIZIONE DELLA PROBE
END_IF
```

8.22 EXD.PXV_RESET_PROBE

Disattiva una Probe settato nel sistema PXV
Questo genera un ERRORE se verrà letta la Probe

Sintassi

EXD.PXV_RESET_PROBE \$PROBE

PROBE Numero della Probe da disattivare Variabile o Valore

Es:

```
EXD.PXV_RESET_PROBE 0 //RESET PROBE 0
```

8.23 EXD.PXV_SET_DETECTOR

Definisce il tipo di Detector (usati nel system PXV) da usare

Sintassi

EXD.PXV_SET_DETECTOR \$MODE

MODE Tipo di Detector Variabile o Valore

| | |
|---------------|---------------------|
| MODE=0 | Contrast Detector |
| MODE=1 | Brightness Detector |
| MODE=2 | Area Detector |
| MODE=3 | Color Detector |

Es:

```
EXD.PXV_SET_DETECTOR 0 //SETTA IL MODO CONTRAST DETECTOR
```

8.24 EXD.PXV_SAVE_IMAGE

Salva su file la corrente immagine acquisita da PXV

Può essere utile per CATTURARE stati della macchina.

Usata in combinazione con **EXD.PXV_GET_IMAGE**

Viene salvata un immagine FILTRATA e una ORIGINALE nei seguenti nomi:

| | |
|-------------------------------|--|
| _PxvImgFiltered.jpg | Immagine FILTRATA |
| _PxvImgNoFiltered.jpg | Immagine ORIGINALE |
| _PxvImgNoFiltered1.jpg | Immagine ORIGINALE IN ALTERNATIVA A _PxvImgNoFiltered.jpg |

Sintassi

EXD.PXV_SAVE_IMAGE

8.25 EXD.PXV_GET_IMAGE

Cattura un immagine da sistema PXV.

L'immagine può essere salvata su file da **EXD.PXV_SAVE_IMAGE**

Sintassi

EXD.PXV_GET_IMAGE \$ISOK

| | | |
|-------------|----------|--------------------------------|
| ISOK | 0 | ERRORE CATTURA IMMAGINE |
| | 1 | OK |

Es:

```
EXD.PXV_GET_IMAG $ISOK      //CATTURA UN IMMAGINE
IF $ISOK=1 //CATTURA OK
    EXD.PXV_SAVE_IMAGE      //SALVA L' IMMAGINE CATTURATA
END_IF
```

8.26 EXD.PXV_SET_JOB

Setta un JOB File di configurazione di PXV

Il JOB deve essere creato da PxVisionBrowser

Sintassi

EXD.PXV_SET_JOB «JOBNAME» \$ISOK

| | | |
|----------------|---|-----------------------|
| JOBNAME | Nome del file JOB (Tra Virgolette) | |
| ISOK | 0 | ERRORE SET JOB |
| | 1 | OK |

Es:

```
EXD.PXV_SET_JOB "MYJOB" $ISOK
```

8.27 EXD.RUN_SCRIPT

Questa funzione permette di eseguire uno **SCRIPT** scritto in codice **C#**. Lo **SCRIPT** viene compilato in **RUN.-TIME** e pertanto non necessita nessun ambiente di sviluppo esterno per la scrittura e compilazione.

Lo **SCRIPT** esterno ovviamente ha l'accesso a tutte le risorse del processo di **ISO US** in corso, pertanto può condividere **VARIABILI** sia in lettura che scrittura.

Ovviamente questo aumenta la **POTENZIALITA'** del Gcode in quanto si accede ad un **LINGUAGGIO DI PROGRAMMAZIONE** molto potente e strutturato. Tutti i parametri della funzione **EXD.RUN_SCRIPT** sono separati da **VIRGOLA “,”** e hanno un identificativo iniziale (*,&,@,X) che determina il tipo di parametro.

Lo **SCRIPT** deve trovarsi nella **CARTELLA DI ISO US->Us_Script** e deve avere estensione **.USS**.

Possono essere **SCRITTI** ed **ESEGUITI** un numero a piacere di **SCRIPT**

Lo SCRIPT termina quando viene effettuato il RETURN dalla funzione MAIN.

IDENTIFICATIVI :

- *)** Identifica il nome dello Script (senza estensione) **OBBLIGATORIO**
Un solo Parametro di questo tipo può essere scritto
- &)** Identifica un parametro passato alla funzione **Main** dello script **NON OBBLIGATORIO**
Possono essere scritti da 0 a N parametri di questo tipo
Possono essere:
Variabili \$
Variabili :
Valori DIRETTI
- @)** Identifica un valore di ritorno dello script **NON OBBLIGATORIO**
Possono essere scritti da 0 a N parametri di questo tipo
Possono essere:
Variabili \$
- X)** Identifica modalità di esecuzione **NON OBBLIGATORIO**
Un solo Parametro di questo tipo può essere scritto
Possono essere:
VALORI DIRETTI 0 o 1
X0, NON viene eseguito in DEMO (Preview) DEFAULT
X1, viene eseguito in DEMO (Preview)

Sintassi

EXD.RUN_SCRIPT *SCRIPT_NAME, &\$VAL,&:ADDR,&100,@\$RET,@\$RET1

SCRIPT_NAME Nome dello SCRIPT SENZA ESTENSIONE

& Parametri di INPUT

@ Parametri di OUPUT

Es:

\$VAR=127.14

:5000=37.18

EXD.RUN_SCRIPT *TESTSCRIPT,&\$VAR,&:5000,&500.14,@\$RET,@\$RET1,@\$RET2

//\$VAR PRIMO VALORE DI INPUT (127.14)

//:5000 SECONDO VALORE DI INPUT (37.18)

//500.14 TERZO VALORE DI INPUT

//\$RET PRIMO VALORE RESTITUITO

//\$RET1 SECONDO VALORE RESTITUITO

//\$RET2 TERZO VALORE RESTITUITO

REGOLE DI SCRITTURA CODICE SCRIPT

Il codice **SCRIPT** deve seguire delle **REGOLE** in modo che possa essere **COMPILATO** ed **ESEGUITO** regolarmente. **ISO US** crea un **ESEMPIO** base nella cartella **Us_Script-> UsExample.uss** in modo di avere una linea guida semplice da poter modificare.

REGOLE BASE:**namespace**Deve sempre essere **UsFunctions****Classe**Deve sempre essere **UsFunctions** e di tipo **PUBBLICA****Funzione di ingresso**

Deve sempre essere di tipo:

public static bool Main(UsWork.IsoUs MyIsoUs,double[] Par,out double[] ValOut)**dove:**

MyIsoUs (di tipo **UsWork.IsoUs**) è l'istanza del **FRAMEWORK** del **PROCESSO** in corso tramite la quale abbiamo accesso a tutte le risorse di **ISOUS**

fare riferimento

https://www.promax.it/file_download/ITA/IsoUs%20Framework.pdf

Par Array di **double** che contiene i valori di **INPUT** (se presenti) nell'esatta successione di inserimento nella funzione **EXD.RUN_SCRIPT**

Se non è **PRESENTE** nessun parametri di **INPUT**, **Par=null**

ValOut Array di **double** che contiene i valori di **OUTPUT** (se presenti)

I valori di output devono essere scritti nell'esatta successione di come verranno **LETTI** dalla funzione **EXD.RUN_SCRIPT**.

Es: **EXD.RUN_SCRIPT *XXX,&\$x,@\$VAL1,@\$VAL2,@\$\$VAL3**

\$VAL1=ValOut[0]

\$VAL2=ValOut[1]

\$VAL3=ValOut[2]

Se nessun valore di **OUTPUT** è previsto impostare a **ValOut=null**

Il metodo **Main** torna un valore di tipo **BOOL**:

True **SCRIPT ESEGUITO REGOLARMENTE**

False **ERRORE ESECUZIONE**

ERRORE ESECUZIONE SCRIPT

Il codice script può generare **ERRORI** di **COMPILAZIONE** o **ECCEZIONI DI ESECUZIONE**.

Da un punto di vista di **Gcode** viene **GENERATO UN ALLARME GENERIC** "ERRORE ESECUZIONE SCRIPT".

Per capire meglio l'origine dell'**ERRORE** viene scritto un file (nella cartella di **IsoUS**) "**_UsScript.Err**" questo contiene informazioni del compilatore relative all'errore generato.

ESEMPIO CODICE SCRIPT

```

@ASSEMBLY System.dll,Microsoft.CSharp.dll,System.Core.dll,System.Data.dll,ComSynk.dll,Compiler.dll,UsWork.dll
//@ASSEMBLY references - add new references if necessary
using System;
using System.Windows;
namespace UsFunctions
{
    public class UsFunctions
    {
        //Function Main 3 par input return 3 ValOut
        public static bool Main(UsWork.IsoUs MyIsoUs,double[] Par,out double[] ValOut)
        {
            ValOut=null;
            try
            {
                ValOut=new double[3]; // 3 ValOut
                ValOut[0]=Par[0]*2; // Par input 0*2
                ValOut[1]=Par[1]*2; // Par input 1*2
                ValOut[2]=Par[2]*2; // Par input 2*2
                for(int n=0;n<300;n++) //Write IsoUs Var From 1000 to 1300
                    MyIsoUs.MyMaster.GetLink.MyCpu.VarCpu[0].VarIsoNs[1000+n]=n*10;
                TestCall(); //Test function call
                return true; // return OK
            }
            Catch { return false; // Error Script}
        }
        //Test Function Call
        static void TestCall()
        {
            ...
            ...
        }
    }
}

```

Il codice sopra viene richiamato con:

\$VAR=127.14

:5000=37.18

EXD.RUN_SCRIPT *TESTSCRIPT,&\$VAR,&:5000,&500.14,@\$RET,@\$RET1,@\$RET2

3 Parametri di input

Par[0]=127.14 -> \$VAR

Par[1]=37.18 -> :5000

Par[2]=500.14

3 Valori di Output

ValOut[0]=Par[0]*2 -> 254.28

ValOut[1]=Par[1]*2 -> 74.36

ValOut[2]=Par[2]*2 -> 1000.28

Inoltre scrive le variabili IsoUs dall' indirizzo 1000 fino a 1300 con valori da 0 a 300

LIMITI CODICE SCRIPT

Teoricamente non esistono limiti nell' codice dello SCRIPT.

Occorre porre attenzione al tempo di esecuzione del codice SCRIPT che non sia superiore a qualche secondo (questo dipende dall' applicazione e IsoUs in corso), in quanto lo SCRIPT BLOCCA l' esecuzione di IsoUs.

8.28 EXD.OPEN_DLL

Questa funzione permette di eseguire una **DLL .NET**. La **DLL** deve essere **SCRITTA** utilizzando l'ambiente di sviluppo Visual Studio nel codice **C#** o **VbNet**.

In modo analogo allo **SCRIPT**, la **DLL** ha l'accesso a tutte le risorse del processo di **ISO US** in corso, pertanto può condividere **VARIABILI** sia in lettura che scrittura. La **DLL** può permettere, al contrari dello **SCRIPT**, di aprire dei **FORM** che possono essere usati nel Gcode in esecuzione.

Tutti i parametri della funzione **EXD.OPEN_DLL** sono separati da **VIRGOLA “,”** e hanno un identificativo iniziale (*****,**&**,**@**,**X**) che determina il tipo di parametro.

La **DLL** deve trovarsi nella **CARTELLA DI ISO US->Us_Dll** e deve avere estensione **.DLL**.

Possono essere **SCRITTI** ed **ESEGUITI** un numero a piacere di **DLL**

Lo DLL termina quando viene invocato il metodo EXD.CLOSE_DLL.

Viene invocato il metodo `public bool Load(UsWork.IsoUs IsoUs, double[] ParInput, out double[] ValOut)`

IDENTIFICATIVI :

- *)** Identifica il nome della **DLL** (senza estensione) **OBBLIGATORIO**
Un solo Parametro di questo tipo può essere scritto
- &)** Identifica un parametro passato alla funzione **Load** della **DLL NON OBBLIGATORIO**
Possono essere scritti da 0 a N parametri di questo tipo
Possono essere:
Variabili \$
Variabili :
Valori DIRETTI
- @)** Identifica un valore di ritorno della Funzione **Load NON OBBLIGATORIO**
Possono essere scritti da 0 a N parametri di questo tipo
Possono essere:
Variabili \$
- X)** Identifica modalità di esecuzione **NON OBBLIGATORIO**
Un solo Parametro di questo tipo può essere scritto
Possono essere:
VALORI DIRETTI 0 o 1
X0, NON viene eseguito in DEMO (Preview) DEFAULT
X1, viene eseguito in DEMO (Preview)

La DLL DEVE IMPORTARE UN INTERFACCIA UsWork.IsoUs.IUsDll.

Questa impone la presenza dei seguenti Metodi e Proprietà:

`public bool Load(UsWork.IsoUs IsoUs, double[] ParInput, out double[] ValOut)`

Deve sempre essere chiamata per **PRIMA ED UNA SOLA VOLTA**

IsoUs Processo attuale di IsoUs
ParInput Array di double di parametri di **INPUT** se presenti, altrimenti **null**
ValOut Array di double di parametri di **OUTPUT** se presenti, altrimenti **null**
Ritorna **True** Load **DLL OK**
 False Load **DLL ERROR**

`public bool CallFunc(double[] ParInput, out double[] ValOut)`

Deve sempre essere chiamata a **PIACERE** durante il **CICLO** del Gcode

La **DLL** deve essere aperta da **Load**

ParInput Array di double di parametri di **INPUT** se presenti, altrimenti **null**
ValOut Array di double di parametri di **OUTPUT** se presenti, altrimenti **null**
Ritorna **True** Call **DLL OK**
 False Call **DLL ERROR**

`public void Close()`

Esegue la chiusura della **DLL**.

È compito del programmatore durante **Close** di rilasciare tutte le risorse della **DLL**

Il metodo Close viene chiamato in **MODO AUTOMATICO** alla fine del Gcode o durante un **ALLARME** Gcode

```
public int Synk { get; set; }
```

Serve per **SINCRONIZZARE** la **DLL** con il Gcode senza bloccare l' applicazione

Il **SYNK** deve essere impostato **INIZIALMENTE** (durante il Load) ad un valore di -1.

Questo significa che **NESSUN SINCRONISMO** è effettuato

Valori **DIVERSI** da -1 **ABILITANO IL SINCRONISMO E SBLOCCANO IL GCODE PER LA PROSSIMA LINEA.**

Generalmente è usato durante l' attesa di **INPUT PARAMETRI** da un **FORM**.

Il valore di **RITORNO** (quando diverso da -1) può essere **LETTO** da Gcode

Sintassi

```
EXD.OPEN_DLL *DLL_NAME, &$VAL,&:ADDR,&100,@$RET,@$RET1
```

DLL_NAME Nome della DLL SENZA ESTENSIONE

& Parametri di INPUT

@ Parametri di OUPUT

Es:

```
//$VAR PRIMO VALORE DI INPUT (127.14)
//:5000 SECONDO VALORE DI INPUT (37.18)
//500.14 TERZO VALORE DI INPUT
//$RET PRIMO VALORE RESTITUITO
//$RET1 SECONDO VALORE RESTITUITO
//$RET2 TERZO VALORE RESTITUITO
$VAR=127.14
:5000=37.18
EXD.OPEN_DLL *TESTUSDLL,&$VAR,&:5000,&500.14,@$RET,@$RET1,@$RET2 //OPEN
EXD.SYNK_DLL *TESTUSDLL,@$SYNCKRET //WAIT SYNK
IF $SYNCKRET=1 // IF SYNK = 1 END
  END_PROGRAM
END_IF
$_LOOP=:100
$COUNT=0
LOOP $_LOOP
  F[:101]
  G1X250Y250Z-50A150B50
  G62
  F[:101]
  X0Y0Z0A0B0
  G62
  $COUNT++
  EXD.CALL_DLL *TESTUSDLL,&$COUNT // CALL FOR UPDATE FORM VALUES
END_LOOP
EXD.SYNK_DLL *TESTUSDLL,@$SYNCKRET // SYNK WAIT END
EXD.CLOSE_DLL *TESTUSDLL // CLOSE
```

8.29 EXD.CALL_DLL

Chiamata ad una DLL precedentemente aperta.

Generalmente viene usato per aggiornare parametri alla DLL.

Viene invocato il metodo `public bool CallFunc(double[] ParInput, out double[] ValOut)`

Sintassi

EXD.CALL_DLL *DLL_NAME, &\$VAL,&:ADDR,&100,@\$RET,@\$RET1

DLL_NAME Nome della DLL SENZA ESTENSIONE

& Parametri di INPUT

@ Parametri di OUPUT

Es:

`EXD.CALL_DLL *TESTUSDLL,&$COUNT // CALL FOR UPDATE FORM VALUES`

8.30 EXD.CLOSE_DLL

Chiude un DLL aperta.

Viene invocato il metodo `public void Close()`

Sintassi

EXD.CLOSE_DLL *DLL_NAM

DLL_NAME Nome della DLL SENZA ESTENSIONE

Es:

`EXD.CLOSE_DLL *TESTUSDLL`

8.31 EXD.SYNK_DLL

Sincronizza il Gcode con una DLL aperta

Permette di attendere alcune operazioni, tipo input dati, dalla DLL prima di continuare il Gcode

Viene chiamata la proprietà `public int Synk { get; set; }`

Per ABILITARE il synk la DLL deve scrivere la Proprietà Synk !=-1

Sintassi

EXD.SYNK_DLL *DLL_NAM,@SYNKRET

DLL_NAME Nome della DLL SENZA ESTENSIONE

SYNKRET Valore di ritorno del SYNK (il valore è sempre diverso da -1, poiché il valore -1 è utilizzato per SYNK non effettuato) **NON OBBLIGATORIO**

Es:

`EXD.SYNK_DLL *TESTUSDLL,@$SYNKRET // SYNK WAIT IN $SYNKRET IL VALORE DELLA PROPRIETA' SYNK`

8.32 EXD.STL_SETVIS

Permette di visualizzare o nascondere un componente STL nella simulazione **REALMACHINE**.
Il componente viene definito per **NOME** come nel file *MachineSettings.xml*

Sintassi

EXD.STL_SETVIS "STLNAME" Xvis

| | | |
|----------------|--------------------------|----------|
| STLNAME | Nome del componente STL | |
| Xvis | Stato di visualizzazione | |
| | X0 | Nascosto |
| | X1 | Visibile |

Es:

\$VIS=0

EXD.STL_SETVIS "STLPART1" \$VIS // NASCONDE IL COMPONENTE STLPART1

8.33 EXD.STL_READVIS

Permette di leggere lo stato di visibilità di un componente STL nella simulazione **REALMACHINE**.
Il componente viene definito per **NOME** come nel file *MachineSettings.xml*

Sintassi

EXD.STL_READVIS "STLNAME" \$STATE

| | | |
|----------------|----------------------------------|----------|
| STLNAME | Nome del componente STL | |
| \$STATE | Variabile di ritorno dello stato | |
| | \$STATE =0 | Nascosto |
| | \$STATE =1 | Visibile |

Es:

EXD.STL_READVIS "STLPART1" \$STATE // LEGGE LO STATO DEL COMPONENTE STLPART1

8.34 EXD.STL_SETVISIDX

Permette di visualizzare o nascondere un componente STL nella simulazione **REALMACHINE**.
Il componente viene definito per **INDIRIZZO** come inserito nel file *MachineSettings.xml*

Sintassi

EXD.STL_SETVISIDX ADDR VIS

| | | |
|-------------|---|----------|
| ADDR | Indirizzo del componente del componente STL può essere una variabile o un valore numerico | |
| VIS | Stato di visualizzazione può essere una variabile o un valore numerico | |
| | 0 | Nascosto |
| | 1 | Visibile |

Es:

\$VIS=0

\$ADDR=5

EXD.STL_SETVISIDX \$ADDR \$VIS // NASCONDE IL COMPONENTE CON INDIRIZZO 5

EXD.STL_SETVISIDX 7 0 // NASCONDE IL COMPONENTE CON INDIRIZZO 7

8.35 EXD.STL_READVISIDX

Permette di leggere lo stato di visibilità di un componente STL nella simulazione **REALMACHINE**.
Il componente viene definito per **INDIRIZZO** come inserito nel file **MachineSettings.xml**

Sintassi

EXD.STL_READVISIDX ADDR \$STATE

ADDR Indirizzo del componente del componente STL può essere una variabile o un valore numerico

\$STATE Variabile di ritorno dello stato

\$STATE =0 Nascosto

\$STATE =1 Visibile

Es:

```
EXD.STL_READVISIDX 5 $STATE // LEGGE LO STATO DEL COMPONENTE INDIRIZZO 5
```

REGOLE DI SCRITTURA US DLL

Le DLL scritte per IsoUs devono seguire delle **REGOLE** in modo che possano essere richiamate dal Gcode.
ISO US crea un **ESEMPIO** base nella cartella **Us_Dll-> ExampleLink.txt** questo contiene un LINK da dove è possibile scaricare un Progetto Visual Studio scritto in C#

REGOLE BASE:

namespace Deve sempre essere **UsDll**
Classe Deve sempre essere **UsDll** e derivare l' INTERFACCIA **UsWork.IsoUs.IUsDll**
public class UsDll:UsWork.IsoUs.IUsDll

I metodi e le Proprietà necessarie sono descritte in [EXD.OPEN DLL](#)

ERRORE ESECUZIONE DLL

Da un punto di vista di Gcode viene **GENERATO UN ALLARME GENERICO "ERRORE ESCUZIONE DLL"**.
Per capire meglio l' origine dell' **ERRORE** viene scritto un file (nella cartella di **IsoUS**) "**_UsDll.Err**" questo contiene informazioni del compilatore relative all' errore generato.
Oppure testare tramite il **DEBUG** di Visual studio.

ESEMPIO USDLL IN VISUAL STUDIO C#

Progetto completo https://www.promax.it/file_download/ZIP/TestUsDll.zip

Classe UsDll

```
using System.Windows;
using TestUsDll;
```

```
namespace UsDll
{
    public class UsDll:UsWork.IsoUs.IUsDll
    {
        //Gcode IsoUs for Test
        /*
            $VAR=127.14
            :5000=37.18
            EXD.OPEN_DLL *TESTUSDLL,&$VAR,&:5000,&500.14,@$RET,@$RET1,@$RET2
            EXD.SYNK_DLL *TESTUSDLL,@$SYNCKRET
            IF $SYNCKRET=1
                END_PROGRAM
            END_IF
            $_LOOP=:100
            $COUNT=0
            LOOP $_LOOP
                F[:101]
                G1X250Y250Z-50A150B50
                G62
                F[:101]
                X0Y0Z0A0B0
                G62
                $COUNT++
                EXD.CALL_DLL *TESTUSDLL,&$COUNT
            END_LOOP
            EXD.SYNK_DLL *TESTUSDLL,@$SYNCKRET
            EXD.CLOSE_DLL *TESTUSDLL
        */
        internal UsWork.IsoUs _MyIsoUs; // instance of UsWork
        TestW _TestW; //Test Window
        internal int Loop; //Loop Number
        #region Events
        /// <summary>
        /// Values changed
        /// Update form labels
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        void _MyIsoUs_AxesDemandValueChanged(object sender, UsWork.AxesValueArg e)
        {
            _TestW.LblValX.Content = e.Format[0];
            _TestW.LblValY.Content = e.Format[1];
            _TestW.LblValZ.Content = e.Format[2];
            _TestW.LblValA.Content = e.Format[3];
            _TestW.LblValB.Content = e.Format[4];
        }
        #endregion Events
    }
}
```

```

#region IUsDll
    /// <summary>
    /// IsoUs Instance of current Framework running
    /// Load Dll
    /// ParInput Input parameters
    /// ValOut Output values
    /// </summary>
    /// <param name="IsoUs"></param>
    /// <param name="ParInput"></param>
    /// <param name="ValOut"></param>
    /// <returns></returns>
    public bool Load(UsWork.IsoUs IsoUs, double[] ParInput, out double[] ValOut)
    {
        Synk = -1;
        ValOut = new double[ParInput.Length]; // ValOut the same length of ParInput
        for (int n = 0; n < ParInput.Length; n++)
            ValOut[n] = ParInput[n] * 2; // Copy values * 2
        _MyIsoUs = IsoUs; // save IsoUs instance
        _SetEvents(); // Set IsoUs events
        _TestW = new TestW(this); // test Window
        _TestW.Owner = (Window)_MyIsoUs.Tag; // Owner
        _TestW.Show(); // Show
        return true; // return OK
    }
    /// <summary>
    /// Call Function
    /// ParInput Input parameters
    /// ValOut Output values
    /// </summary>
    /// <param name="ParInput"></param>
    /// <param name="ValOut"></param>
    /// <returns></returns>
    public bool CallFunc(double[] ParInput, out double[] ValOut)
    {
        ValOut = null; // no ValOut
        int _Val = (int)ParInput[0]; // Get ParInput[0]
        if (_Val == Loop) // if cycle reached
        {
            _TestW.LblLoop.Content = "END PCS: " + _Val.ToString(); // update label END PCS
            _TestW.BtnExit.IsEnabled = true; // enable Exit button
        }
        else
            _TestW.LblLoop.Content = _Val.ToString(); //Update PCS Number
        return true;
    }
    /// <summary>
    /// Close Dll
    /// </summary>
    public void Close()
    {
        _TestW.Close(); // Close Window
    }

```

```
/// <summary>
/// Synk Dll
/// return SYNK value if not -1
/// </summary>
public int Synk { get; set; }

#endregion IUsDll
/// <summary>
/// Stop the current Gcode execution
/// called by form button
/// </summary>
internal void Stop()
{
    _MyIsoUs.UsGcodeRun.StopGcode(); // Stop Gcode
}
/// <summary>
/// Set events
/// </summary>
void _SetEvents()
{
    _MyIsoUs.AxesDemandValueChanged += _MyIsoUs_AxesDemandValueChanged; // axes values changed
    _MyIsoUs.MyMaster.ResetEvent(); // reset events for update labels
}
}
```


TestW.xaml

```

<Window x:Name="TestW1" x:Class="TestUsDll.TestW"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:TestUsDll"
  mc:Ignorable="d"
  Title="TestW" Height="400" Width="400" WindowStyle="None">
<Grid x:Name="MyGrid" >
  <Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
  </Grid.RowDefinitions>
  <Grid Grid.Row="0">
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="0.7*" />
      <ColumnDefinition Width="0.8*" />
      <ColumnDefinition Width="0.7*" />
      <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <Label x:Name="LblNrLoop" Tag="2" Grid.Column="0" Content="NrLoop" HorizontalAlignment="Center"
  Height="40" FontSize="18" VerticalAlignment="Center" />
    <TextBox x:Name="TxtLoop" Tag="0" Grid.Column="1" HorizontalAlignment="Center" Height="30" Text="50"
  FontSize="20" VerticalAlignment="Center" Width="90" TextAlignment="Center" PreviewTextInput="TxtLoop_Preview-
  TextInput" />
    <Label x:Name="LblCurrentLoop" Grid.Column="2" Tag="2" Content="Current" HorizontalAlignment="Center"
  Height="40" FontSize="18" VerticalAlignment="Center" />
    <Label x:Name="LblLoop" Tag="2" Grid.Column="3" FontSize="18" Content="0" HorizontalAlignment="Left"
  Height="40" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" VerticalAlignment="Center"
  />
  </Grid>
  <Grid Grid.Row="1">
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="*" />
      <ColumnDefinition Width="*" />
      <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <Label x:Name="LblFeed" Tag="2" Grid.Column="0" Content="Feed" HorizontalAlignment="Center"
  Height="40" FontSize="18" VerticalAlignment="Center" Width="50" />
    <TextBox x:Name="TxtF" Tag="0" Grid.Column="1" HorizontalAlignment="Center" Height="30" FontSize="20"
  Text="1" VerticalAlignment="Center" Width="100" TextAlignment="Center" PreviewTextInput="TxtF_PreviewTextIn-
  put" />
    <Button x:Name="BtnSetF" Grid.Column="2" Tag="1" Content="Set Feed" ToolTip="Set Work Feed" Horizontal-
  Alignment="Center" FontSize="16" Height="30" VerticalAlignment="Center" Width="80" Click="BtnSetF_Click" />
  </Grid>
  <Grid Grid.Row="2">
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>

```

```

    <ColumnDefinition Width="5*"/>
  </Grid.ColumnDefinitions>
  <Border x:Name="BLblX" Grid.Column="0" Tag="3" Height="40" Width="40" VerticalAlignment="Center"
HorizontalAlignment="Center" >
    <Label x:Name="LblX" Tag="2" Content="X" Height="40" FontSize="24" HorizontalContentAlignment="Cen-
ter" VerticalAlignment="Center" Width="40"/>
  </Border>
  <Border x:Name="BLblValX" Grid.Column="1" Tag="4" Height="40" Width="200" VerticalAlignment="Center"
HorizontalAlignment="Center">
    <Label x:Name="LblValX" Tag="2" Content="..." Height="40" FontSize="24" HorizontalContentAlign-
ment="Center" VerticalAlignment="Center" />
  </Border>
</Grid>
<Grid Grid.Row="3">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*"/>
    <ColumnDefinition Width="5*"/>
  </Grid.ColumnDefinitions>
  <Border x:Name="BLblY" Grid.Column="0" Tag="3" Height="40" Width="40" VerticalAlignment="Center"
HorizontalAlignment="Center" >
    <Label x:Name="LblY" Tag="2" Content="Y" Height="40" FontSize="24" HorizontalContentAlignment="Cen-
ter" VerticalAlignment="Center" Width="40"/>
  </Border>
  <Border x:Name="BLblValY" Grid.Column="1" Tag="4" Height="40" Width="200" VerticalAlignment="Center"
HorizontalAlignment="Center" >
    <Label x:Name="LblValY" Tag="2" Content="..." Height="40" FontSize="24" HorizontalContentAlign-
ment="Center" VerticalAlignment="Center" />
  </Border>
</Grid>

<Grid Grid.Row="4">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*"/>
    <ColumnDefinition Width="5*"/>
  </Grid.ColumnDefinitions>
  <Border x:Name="BLblZ" Tag="3" Grid.Column="0" Height="40" Width="40" VerticalAlignment="Center" Hor-
izontalAlignment="Center" >
    <Label x:Name="LblZ" Tag="2" Content="Z" Height="40" FontSize="24" HorizontalContentAlignment="Cen-
ter" VerticalAlignment="Center" Width="40"/>
  </Border>
  <Border x:Name="BLblValZ" Grid.Column="1" Tag="4" Height="40" Width="200" VerticalAlignment="Center"
HorizontalAlignment="Center" >
    <Label x:Name="LblValZ" Tag="2" Content="..." Height="40" FontSize="24" HorizontalContentAlign-
ment="Center" VerticalAlignment="Center" />
  </Border>
</Grid>
<Grid Grid.Row="5">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*"/>
    <ColumnDefinition Width="5*"/>
  </Grid.ColumnDefinitions>
  <Border x:Name="BLblA" Grid.Column="0" Tag="3" Height="40" Width="40" VerticalAlignment="Center" Hor-
izontalAlignment="Center">
    <Label x:Name="LblA" Tag="2" Content="A" Height="40" FontSize="24" HorizontalContentAlignment="Cen-
ter" VerticalAlignment="Center" Width="40"/>
  </Border>

```

```

    <Border x:Name="BLblValA" Grid.Column="1" Tag="4" Height="40" Width="200" VerticalAlignment="Center"
HorizontalAlignment="Center" Grid.Row="1">
    <Label x:Name="LblValA" Tag="2" Content="..." Height="40" FontSize="24" HorizontalContentAlign-
ment="Center" VerticalAlignment="Center" />
    </Border>
</Grid>
<Grid Grid.Row="6">
    <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="5*" />
    </Grid.ColumnDefinitions>
    <Border x:Name="BLblB" Grid.Column="0" Tag="3" Height="40" Width="40" VerticalAlignment="Center"
HorizontalAlignment="Center" >
    <Label x:Name="LblB" Tag="2" Content="B" Height="40" FontSize="24" HorizontalContentAlignment="Cen-
ter" VerticalAlignment="Center" Width="40" />
    </Border>
    <Border x:Name="BLblValB" Grid.Column="1" Tag="4" Height="40" Width="200" VerticalAlignment="Center"
HorizontalAlignment="Center" >
    <Label x:Name="LblValB" Tag="2" Content="..." Height="40" FontSize="24" HorizontalContentAlign-
ment="Center" VerticalAlignment="Center" />
    </Border>
</Grid>
<Grid Grid.Row="7">
    <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <Button x:Name="BtnStop" Grid.Column="0" Tag="1" IsEnabled="False" Content="STOP" ToolTip="Stop Gcode
and Exit" HorizontalAlignment="Center" Height="30" VerticalAlignment="Center" Width="60" Click="BtnStop_Click" />
    <Button x:Name="BtnRun" Grid.Column="1" Tag="1" Content="Run" ToolTip="Run Gcode" HorizontalAlign-
ment="Center" Height="30" VerticalAlignment="Center" Width="60" Click="BtnRun_Click" />
    <Button x:Name="BtnExit" Grid.Column="2" Tag="1" ToolTip="Exit" Content="Exit" HorizontalAlignment="Cen-
ter" Height="30" VerticalAlignment="Center" Width="60" Click="BtnExit_Click" />
    </Grid>
</Grid>
</Window>

```

TestW.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace TestUsDll
{
    public partial class TestW : Window
    {
        UsDll.UsDll _MyUsDll;
        //Style components
        /*
        MainGridStyle      -> Grid
        PluginButton       -> Button
        PluginButtonSelect -> Button
        BtnVisType         -> Button
        ButtonCommandsStyle -> Button
        TextBoxStyle       -> TextBox
        LabelStyle         -> Label
        LabelSelector      -> Label
        LabelSelector1     -> Label
        LabelAbsStyle      -> Label
        LabelMonitorStyle  -> Label
        LabelCommandsStyle -> Label
        LabelNameAxesStyle -> Label
        LabelFlashStyle    -> Label
        LabelHomeStyle     -> Label
        LabelInc           -> Label
        UsPreviewCheckBox  -> CheckBox
        CheckBoxStyle      -> CheckBox
        UsContextMenuStyle -> ContextMenu
        UsGridViewHeaderStyle -> DataGridColumnHeader
        UsDataGridCellStyle -> DataGridCell
        UsProgressBar      -> ProgressBar
        TabItemStyle       -> TabItem
        BorderSelector     -> Border
        BorderSelector1    -> Border
        BorderStyle        -> Border
        BorderAbsStyle     -> Border
        BorderMonitorStyle -> Border
        BorderNameAxesStyle -> Border
        BorderInc          -> Border
        SliderThumbStyle   -> Thumb
        NotifyListViewFont -> ListViewItem

```

```

NotifyListViewStyle -> ListView
*/
public TestW(UsDII.UsDII MyUsDII)
{
    _MyUsDII = MyUsDII;
    InitializeComponent();
    _SetStyle(); // Set Style from IsoUs
}
#region Events
/// <summary>
/// Move Window
/// </summary>
/// <param name="e"></param>
protected override void OnMouseLeftButtonDown(MouseButtonEventArgs e)
{
    base.OnMouseLeftButtonDown(e);
    DragMove();
}
/// <summary>
/// Run Gcode SYNK OK
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void BtnRun_Click(object sender, RoutedEventArgs e)
{
    _MyUsDII.Loop = int.Parse(TxtLoop.Text); // Loop Value
    _MyUsDII._MyIsoUs.MyMaster.GetLink.MyCpu.VarCpu[0].VarIsoNs[100] = _MyUsDII.Loop; // Write IsoUs Var
Loop
    _MyUsDII._MyIsoUs.MyMaster.GetLink.MyCpu.VarCpu[0].VarIsoNs[101] = double.Parse(TxtF.Text,
    _MyUsDII._MyIsoUs.NewProvider); // Write FEED Value
    _MyUsDII.Synk = 0; // Synk Ok
    BtnStop.IsEnabled = true;
    BtnExit.IsEnabled = false;
    BtnRun.IsEnabled = false;

}
/// <summary>
/// Exit
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void BtnExit_Click(object sender, RoutedEventArgs e)
{
    _MyUsDII.Synk = 1; // SYNK EXIT
}
/// <summary>
/// Stop Gcode
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void BtnStop_Click(object sender, RoutedEventArgs e)
{
    _MyUsDII.Stop();
}

```

```

/// <summary>
/// Set Feed
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void BtnSetF_Click(object sender, RoutedEventArgs e)
{
    _MyUsDll._MyIsoUs.MyMaster.GetLink.MyCpu.VarCpu[0].VarIsoNs[101] = double.Parse(TxtF.Text,
    _MyUsDll._MyIsoUs.NewProvider); // Set Gcode Feed
}
/// <summary>
/// Feed Preview Text Input
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void TxtF_PreviewTextInput(object sender, TextCompositionEventArgs e)
{
    TextBox _Txt = (TextBox)sender;
    _ChecksNumeric(e, _Txt.Text, true, false);
}
/// <summary>
/// Loop Preview Text Input
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void TxtLoop_PreviewTextInput(object sender, TextCompositionEventArgs e)
{
    TextBox _Txt = (TextBox)sender;
    _ChecksNumeric(e, _Txt.Text, false, false);
}
#endregion Events
/// <summary>
/// Check if string contains only numerical value
/// </summary>
/// <param name="e">TextCompositionEventArgs PreviewTextInput Argument</param>
/// <param name="_Text">String Text for check</param>
/// <param name="_DecimalPoint">True accept decimal point</param>
/// <param name="_NegativeValue">True accept negative value</param>
void _ChecksNumeric(TextCompositionEventArgs e, string _Text, bool _DecimalPoint, bool _NegativeValue)
{
    if (!_DecimalPoint)
        _ChecksNumeric(e, _Text, 0, _NegativeValue);
    else
        _ChecksNumeric(e, _Text, 20, _NegativeValue);
}
/// <summary>
/// Check if string contains only numerical value
/// </summary>
/// <param name="e">TextCompositionEventArgs PreviewTextInput Argument</param>
/// <param name="_Text">String Text for check</param>
/// <param name="_NrDecimal">Int32 Number of Decimal Digit</param>
/// <param name="_NegativeValue">True accept negative value</param>
void _ChecksNumeric(TextCompositionEventArgs e, string _Text, Int32 _NrDecimal, bool _NegativeValue)
{
    int result;
    if (e.Text == "-")

```

```

{
    if (!_NegativeValue)
        e.Handled = true;
    if (_Text == "")
        return;
    if (_Text.Contains('-'))
    {
        e.Handled = true;
        return;
    }
    TextBox _Txt = (TextBox)e.OriginalSource;
    if (_Txt.CaretIndex == 0)
        return;
    e.Handled = true;
    return;
}
if (e.Text == ".")
{
    if (_NrDecimal <= 0)
        e.Handled = true;
    if (_Text.Contains("."))
        e.Handled = true;
    return;
}
if (!(int.TryParse(e.Text, out result)))
    e.Handled = true;
// Check Decimal after dot
Int32 AfterDot;
Int32 PosDot;
PosDot = _Text.IndexOf('.');
if (PosDot == -1)
    return;
// get char after dot
AfterDot = _Text.Length - PosDot - 1;
TextBox _MText = (TextBox)e.Source;

if (AfterDot >= _NrDecimal && _MText.CaretIndex > PosDot)
    e.Handled = true;
}
/// <summary>
/// Set SType
/// </summary>
void _SetStyle()
{
    _SetStyle(MyGrid);
}

```

```

/// <summary>
/// Set Styles
/// Set Components Styles like to IsoUs
/// </summary>
void _SetStyle(Grid _Grid)
{
    for(int n=0;n<_Grid.Children.Count;n++)
    {
        int _Tag = -1;
        object _Obj = _Grid.Children[n];
        if(_Obj is Grid)
        {
            Grid _Grd = (Grid)_Obj;
            _Grd.Style = _MyUsDll._MyIsoUs.UsStylesProvider.GetStyle("MainGridStyle");
            _SetStyle(_Grd);
        }
        Label _Lbl = null;
        Border _Brd = null;
        Button _Btn = null;
        TextBox _Txt = null;
        if (_Obj is Label)
        {
            _Lbl = (Label)_Obj;
            _Tag = int.Parse(_Lbl.Tag.ToString());
        }
        if (_Obj is Button)
        {
            _Btn = (Button)_Obj;
            _Tag = int.Parse(_Btn.Tag.ToString());
        }
        if (_Obj is TextBox)
        {
            _Txt = (TextBox)_Obj;
            _Tag = int.Parse(_Txt.Tag.ToString());
        }
        if (_Obj is Border)
        {
            _Brd = (Border)_Obj;
            object _Obj1 = _Brd.Child;
            _Lbl = (Label)_Obj1;
            _Tag = int.Parse(_Brd.Tag.ToString());
        }
        switch (_Tag)
        {
            case 0:
                _Txt.Style = _MyUsDll._MyIsoUs.UsStylesProvider.GetStyle("TextBoxStyle");
                break;
            case 1:
                _Btn.Style = BtnExit.Style = _MyUsDll._MyIsoUs.UsStylesProvider.GetStyle("PlugInButton");
                break;
            case 2:
                _Lbl.Style = _MyUsDll._MyIsoUs.UsStylesProvider.GetStyle("LabelNameAxesStyle");
                break;
            case 3:
                _Brd.Style = _MyUsDll._MyIsoUs.UsStylesProvider.GetStyle("BorderNameAxesStyle");
                _Lbl.Style = _MyUsDll._MyIsoUs.UsStylesProvider.GetStyle("LabelNameAxesStyle");
        }
    }
}

```



```
break;  
case 4:  
_Brd.Style = _MyUsDll._MyIsoUs.UsStylesProvider.GetStyle("BorderStyle");  
_Lbl.Style = _MyUsDll._MyIsoUs.UsStylesProvider.GetStyle("LabelAbsStyle");  
break;  
}  
}  
}  
}
```

The screenshot displays a CNC control interface with a golden-brown background. At the top, it shows 'NrLoop' with a value of 50 and 'Current' with a value of 4. Below this, there is a 'Feed' field set to 10 and a 'Set Feed' button. The main area contains five rows of coordinate readouts: X (69.540), Y (69.540), Z (-13.908), A (41.724), and B (13.908). At the bottom, there are three buttons: 'STOP', 'Run', and 'Exit'.

| Coordinate | Value |
|------------|---------|
| X | 69.540 |
| Y | 69.540 |
| Z | -13.908 |
| A | 41.724 |
| B | 13.908 |

9 SWITCH e DIRETTIVE DI COMPILAZIONE

Gli switch di compilazione, permettono di dare delle direttive al compilatore tramite il costrutto:

IFDEF

ELSEDEF

ENDIFDEF

Questo permette di compilare solamente parti interessate che rientrano nella logica dello switch selezionato. Attualmente esiste un solo switch disponibile a livello di sistema:

AXES

Che contiene il numero di assi configurati per l' applicazione:

Es: per configurazione 3 ASSI

```
IFDEF AXES=3 // AXES NUMBER=3
  G1 X100 Y100 Z200
ENDIFDEF
IFDEF AXES=4 //AXES NUMBER=4
  G1 X100 Y100 Z200 A200
ENDIFDEF
```

Nell' esempio sopra riportato, viene solamente compilato il codice che rientra nella logica dello switch, cioè 3 Assi, quindi il codice **G1X100Y100Z200A200** viene scartato (non compilato). In una condizione normale senza l' utilizzo dei costrutti **IFDEF**, la riga **G1X100Y100Z200A200** avrebbe generato un errore di ASSE NON CONFIGURATO (A)

È anche possibile aggiungere degli switch di compilazione personalizzati, vedere il capitolo

3.5.2 del manuale [utility di sistema](#) – **Aggiungere un Parametro Custom**

Le direttive del compilatore, sono comandi che interessano il Gcode in uso

9.1 IFDEF

Istruzione **IF** sullo switch selezionato:

```
IFDEF AXES=3
```

Se lo condizione è vera, li codice contenuto tra **IFEDF** e **ELSEDEF** o **ENDIFDEF**, verrà compilato, altrimenti viene scartato e compilato l' eventuale codice inserito in **ELSEDEF**

In un costrutto **IFDEF** possono essere utilizzati i seguenti operatori:

```
=      Uguale
>      Maggiore
<      Minore
>=     Maggiore Uguale
<=     Minore Uguale
<>    Diverso
```

9.2 ELSEDEF

Istruzione **ELSE** sullo switch selezionato:

```
IFDEF AXES=3
  G1 X100
ELSEDEF
  G1 Y30
ENDIFDEF
```

Se la condizione non è vera, cioè **AXES<>3** li codice contenuto tra **ELSEDEF** e **ENDIFDEF**, verrà compilato

9.3 ENDIFDEF

Fine del costrutto **IFDEF**

9.4 NOAXESREADY

Questa direttiva di compilazione (generalmente inserita all' inizio del PartProgram) permette di eseguire il Gcode anche se gli assi non sono pronti (Homing e Enable effettuati).

Questo fa sì poter usare le istruzioni **G71** per gestione degli assi

NOAXESREADY

```
G71.1 X // ENABLE X
G71.1 Y // ENABLE Y
G71 X // HOMING X
G71 Y // HOMING Y
```

9.5 ONERROR

Questa direttiva di compilazione permette di programmare una parte di codice che viene eseguita al verificarsi di un Errore (tipo macro di Errore)

Successivamente viene comunque eseguita la **Macro di Errore** se programmata

ONERROR

```
$(O1)=0 // DISABLE OUT
$(O2)=0 // DISABLE OUT
ENDON
```

9.6 ONSTOP

Questa direttiva di compilazione permette di programmare una parte di codice che viene eseguita quando il CN va in STOP (tipo macro di Stop)

Successivamente viene comunque eseguita la **Macro di STOP** se programmata

ONSTOP

```
$(O1)=0 // DISABLE OUT
$(O2)=0 // DISABLE OUT
ENDON
```

9.7 ENDON

Fine direttiva **ONERROR** e **ONSTOP**

9.8 USET

Questa direttiva di compilazione permette di lanciare automaticamente un **MACRO** quando viene eseguita l' istruzione **Tn**

```
USET 5 // VIENE SEMPRE ESEGUITA LA MACRO M5 DOPO L' ISTRUZIONE Tn
```

9.9 USEH

Questa direttiva di compilazione permette di lanciare automaticamente un **MACRO** quando viene eseguita l' istruzione **Hn**

```
USEH 6 // VIENE SEMPRE ESEGUITA LA MACRO M6 DOPO L' ISTRUZIONE Hn
```

10 ISTRUZIONI PER IL CONTROLLO MULTIPROCESSO

L'istruzione per il controllo del MultiProcesso, servono per gestire più CNC che sono eseguiti nella stessa unità PC. Non necessita di nessuna configurazione particolare, in quanto questa è automaticamente rilevata dal file IsoUs.cfg.

10.1 CNC.LOAD

Carica un programma ISO su un processo.

Sintassi

CNC.LOAD CN TypeRun InEditor "PATH"

CN Numero di Processo da 1 a 8

TypeRun 0 Solo Load
1 Load e Run
2 Load e Preview
3 Load, Preview e Run

InEditor 0 Non Inviato a UsEditor
1 Inviato a UsEditor

PATH (tra virgolette) Se **PATH** inizia con il prefisso **\$APPPATH**, il file viene cercato all'interno delle cartelle di IsoUs. Il file deve essere sempre completo di estensione.

Es:

CNC.LOAD 1 1 1 "\$APPPATH\PROJECT\IMPORT\TEST.ISO" **//percorso relativo**

CNC.LOAD 1 1 1 "C:\FILE\TEST.ISO" **//percorso assoluto**

10.2 CNC.RUN

Run Gcode su un processo.

Sintassi

CNC.RUN CN

CN Numero di Processo da 1 a 8
Se **CN=-1** Run contemporaneo su tutti i processi disponibili

Es:

CNC.RUN 1 **//RUN SU PROCESSO 1**

CNC.RUN -1 **//RUN SU TUTTI I PROCESSI**

10.3 CNC.PREVIEW

Preview Gcode su un processo.

Sintassi

CNC.PREVIEW CN

CN Numero di Processo da 1 a 8
Se **CN=-1** Preview contemporaneo su tutti i processi disponibili

Es:

CNC.PREVIEW 1 **//PREVIEW SU PROCESSO 1**

CNC.PREVIEW -1 **//PREVIEW SU TUTTI I PROCESSI**

10.4 CNC.STOP

Stop Gcode su un processo.

Sintassi

CNC.STOP CN

CN Numero di Processo da 1 a 8
Se **CN=-1** Stop contemporaneo su tutti i processi disponibili

Es:

CNC.STOP 1 //STOP SU PROCESSO 1

CNC. STOP -1 //STOP SU TUTTI I PROCESSI

10.5 CNC.PAUSE

Pausa Gcode su un processo.

Sintassi

CNC.PAUSE CN

CN Numero di Processo da 1 a 8
Se **CN=-1** Pause contemporaneo su tutti i processi disponibili

Es:

CNC.PAUSE 1 //PAUSE SU PROCESSO 1

CNC. PAUSE -1 //PAUSE SU TUTTI I PROCESSI

10.6 CNC.STATUS

Lettura Status su un processo.

Sintassi

CNC.STATUS CN \$VAR

CN Numero di Processo da 1 a 8

\$VAR Variabile di ritorno lettura mappata a bit

| | | | | | |
|---------------|-----------|---------------|--|---------------|-------------------|
| Bit 0 | Run | Bit 1 | Errore | Bit 2 | Assi in Movimento |
| Bit 3 | Pausa | Bit 4 | Home X Ok | Bit 5 | Home Y OK |
| Bit 6 | Home Z Ok | Bit 7 | Home A Ok | Bit 8 | Home B Ok |
| Bit 9 | Home C Ok | Bit 10 | Home U Ok | Bit 11 | Home V Ok |
| Bit 12 | Home W Ok | Bit 13 | Enable X | Bit 14 | Enable Y |
| Bit 15 | Enable Z | Bit 16 | Enable A | Bit 17 | Enable B |
| Bit 18 | Enable C | Bit 19 | Enable U | Bit 20 | Enable V |
| Bit 21 | Enable W | Bit 22 | Pronto al Run (Home e enable Ok su tutti gli assi) | | |

Es:

CNC.STATUS 1 \$VAR

\$VAR1=\$VAR & 1 //TEST RUN

IF \$VAR1 = 1

...// RUN

10.7 CNC.STATUSBIT

Letture Status a bit su un processo.

Sintassi

CNC.STATUS CN NrBit \$VAR

| | |
|--------------|--------------------------------------|
| CN | Numero di Processo da 1 a 8 |
| NrBit | Numero Bit (vedi CNC.STATUS) |
| \$VAR | Variabile di ritorno lettura (0 o 1) |

Es:

```
CNC.STATUSBIT 1 1 $VAR
```

```
IF $VAR = 1
```

```
...// RUN
```

10.8 CNC.INFO

Letture Informazioni su un processo.

Sintassi

CNC.INFO CN InfoType \$VAR

| | |
|-----------------|--|
| CN | Numero di Processo da 1 a 8 |
| InfoType | Tipo informazione |
| | <ul style="list-style-type: none"> 0 Lettura Linea Teorica in esecuzione 1 Lettura Linea Reale in esecuzione 2 Lettura risoluzione quote Assi 3 Lettura risoluzione Feed |
| \$VAR | Variabile di ritorno lettura |

10.9 CNC.AXIS

Letture Informazioni Asse su un processo.

Sintassi

CNC.AXIS CN AxisIndex AxisType \$VAR

| | |
|------------------|---|
| CN | Numero di Processo da 1 a 8 |
| AxisIndex | Indice Asse da 0 a 8 |
| AxisType | Tipo informazione |
| | <ul style="list-style-type: none"> 0 Lettura Quota Teorica Assoluta 1 Lettura Quota Reale Assoluta 2 Lettura Quota Teorica Assoluta (Syncro modo preciso) 3 Lettura Quota Reale Assoluta (Syncro modo preciso) 4 Lettura Valore Totale Offset (Origini, Offset Hn G43 ecc.) |
| \$VAR | Variabile di ritorno lettura |

10.10 CNC.GROUP

Lettura Gruppo Informazioni Asse su un processo.

Sintassi

CNC.GROUP CN NrAxis AxisType \$VAR

| | |
|-----------------|--|
| CN | Numero di Processo da 1 a 8 |
| NrAxis | Numero Assi Da Leggere |
| AxisType | Tipo informazione |
| | 0 Lettura Quota Teorica Assoluta |
| | 1 Lettura Quota Reale Assoluta |
| | 2 Lettura Valore Totale Offset (Origini, Offset Hn G43 ecc.) |
| \$VAR | Variabile di ritorno lettura di Partenza o Array |

Es:

```
// ALLOCARE 3 VARIABILI PER 3 ASSI
```

```
$VARX=0
```

```
$VARY=0
```

```
$VARZ=0
```

```
CNC.GROUP 1 3 0 $VARX
```

```
// $VARX=QUOTA ASSE X
```

```
// $VARY=QUOTA ASSE Y
```

```
// $VARZ=QUOTA ASSE Z
```

Es:

```
// DICHIARARE ARRAY
```

```
DIM $ARR 3
```

```
CNC.GROUP 1 3 0 $ARR
```

```
// $ARR[0]=QUOTA ASSE X
```

```
// $ARR[1]=QUOTA ASSE Y
```

```
// $ARR[2]=QUOTA ASSE Z
```

10.11 CNC.READVARADDR

Lettura Gcode Variabile per indirizzo su un processo.

Sintassi

CNC.READVARADDR CN AddrVar \$VAR

| | |
|----------------|----------------------------------|
| CN | Numero di Processo da 1 a 8 |
| AddrVar | Indirizzo Variabile da 0 a 32767 |
| \$VAR | Variabile di ritorno lettura |

10.12 CNC.READVARNAME

Lettura Gcode Variabile per Nome su un processo.

Sintassi

CNC.READVARNAME CN \$VAR "NAMEVAR"

| | |
|----------------|--|
| CN | Numero di Processo da 1 a 8 |
| \$VAR | Variabile di ritorno lettura |
| NAMEVAR | Nome Variabile in Maiuscolo senza \$(tra virgolette) |

10.13 CNC.WRITEVARADDR

Scrittura Gcode Variabile per indirizzo su un processo.

Sintassi

CNC.WRITEVARADDR CN AddrVar \$VAR

| | |
|----------------|--|
| CN | Numero di Processo da 1 a 8 Se CN=-1 Scrittura contemporanea su tutti i processi disponibili |
| AddrVar | Indirizzo Variabile da 0 a 32767 |
| \$VAR | Variabile Sorgente |

10.14 CNC.WRITEVARNAME

Scrittura Gcode Variabile per Nome su un processo.

Sintassi

CNC.WRITEVARNAME CN \$VAR "NAMEVAR"

| | |
|----------------|--|
| CN | Numero di Processo da 1 a 8 Se CN=-1 Scrittura contemporanea su tutti i processi disponibili |
| \$VAR | Variabile Sorgente |
| NAMEVAR | Nome Variabile destinazione in Maiuscolo senza \$ (tra virgolette) |

10.15 CNC.READPARAMAC

Lettura Parametro Macchina di un processo.

Sintassi

CNC.READPARAMAC CN \$VAR "PARNAME"

| | |
|----------------|--|
| CN | Numero di Processo da 1 a 8 |
| \$VAR | Variabile di ritorno lettura |
| PARNAME | Nome Parametro Macchina (tra virgolette) |

10.16 CNC.WRITEPARAMAC

Scrittura Parametro Macchina di un processo.

Sintassi

CNC.WRITEPARAMAC CN \$VAR "PARNAME"

| | |
|----------------|--|
| CN | Numero di Processo da 1 a 8 |
| \$VAR | Variabile Sorgente |
| PARNAME | Nome Parametro Macchina (tra virgolette) |

10.17 CNC.ENABLEAXIS

Abilitazione/Disabilitazione Asse di un processo. (Attende sequenza terminata)

Sintassi

CNC.ENABLEAXIS CN AXIS STATE

| | |
|--------------|---|
| CN | Numero di Processo da 1 a 8 |
| AXIS | Indice Asse |
| STATE | Stato Asse 0 Disabilitato 1 Abilitato |

10.18 CNC.HOMEAXIS

Homing Asse di un processo. (Attende sequenza terminata) CNC.STOP per interrompere

Sintassi**CNC.HOMEAXIS CN AXIS**

CN Numero di Processo da 1 a 8
AXIS Indice Asse

10.19 CNC.READGENERIC

Lettura variabile tipo Generico (vedere sotto Type Generic)

Sintassi**CNC.READGENERIC CN TYPE \$PAR \$VAR**

CN Numero di Processo da 1 a 8
TYPE Tipo di Generic
\$PAR Parametro Addizionale (se gestito altrimenti mettere \$PAR=0)
\$VAR Variabile Lettura

10.20 CNC.WRITEGENERIC

Scrittura variabile tipo Generico (vedere sotto Type Generic)

Sintassi**CNC.WRITEGENERIC CN TYPE \$PAR \$VAR**

CN Numero di Processo da 1 a 8
TYPE Tipo di Generic
\$PAR Parametro Addizionale (se gestito altrimenti mettere \$PAR=0)
\$VAR Variabile Scrittura

Type Generic

| TYPE | READ | WRITE | Descrizione | \$PAR |
|------|------|-------|---|------------------------------|
| 0 | V | | Lettura quote teoriche (\$[Qn]) | Indice Asse di lettura |
| 1 | V | | Lettura quote reali (\$[Rn]) | Indice Asse di lettura |
| 2 | V | | Lettura Ingresso Digitale (\$[In]) | Indice Ingresso (da 0 a 255) |
| 3 | V | V | Lettura/Scrittura Uscita Digitale (\$[On]) | Indice Uscita (da 0 a 255) |
| 4 | V | V | Lettura/Scrittura Parametro Tabella Utensile (\$[Un]) | Indice Parametro |
| 5 | V | | Lettura Parametro Tabella Testa (\$[Hn]) | Indice Parametro |
| 6 | V | | Lettura SPEED (\$[X0]) | Non usato |
| 7 | V | | Lettura Abilitazione Origine Pezzo (\$[X3]) | Non usato |
| 8 | V | | Lettura Abilitazione Offset Pezzo (\$[X4]) | Non usato |
| 9 | V | | Lettura FEED (\$[X8]) | Non usato |
| 10 | V | | Lettura valore Origine (\$[Yn]) | Indice Asse |
| 11 | V | | Lettura valore Offset (\$[Wn]) | Indice Asse |
| 12 | V | | Lettura H impostata | Non usato |
| 13 | V | | Lettura T impostato | Non usato |
| 14 | V | V | Lettura/Scrittura User Generic (\$[Kn]) | Indice User Generic |
| 15 | | V | Scrittura Uscita analogica per SPINDLE | Non Usato |

11 ISTRUZIONI PER IL CONTROLLO REMOTO

L'istruzione per il controllo remoto, servono per gestire più CNC in rete LAN con scambio dati fra di essi. Questi devono essere precedentemente configurati come da Capitolo “CONFIGURAZIONE CN REMOTI”
Tale istruzioni possono generare i seguenti **RunTimeError**:

- E1082** CNC remoto non connesso
- E1083** Operazione fallita su CNC remoto
- E1084** Il part program del CNC remoto contiene ERRORI di compilazione
- E1085** L'operazione di RUN su CNC remoto è fallita
- E1086** Non è stato trovato il file da caricare sul CNC remoto (Remote.load)
- E1087** Assi non pronti sul CNC remoto
- E1088** Programma già in RUN su CNC remoto
- E1089** Numero assi non presenti sul CNC remoto
- E1090** Variabile ISO non disponibile su CNC remoto

11.1 REMOTE.LOAD

Carica un programma ISO su un CN REMOTO

Sintassi

REMOTE.LOAD CN RUN “PATH”

- CN** Numero di CN configurato dal file *clientNs.cfg*
- RUN** **0** Il file viene caricato senza esecuzione
1 Il file viene automaticamente eseguito dopo il load
- PATH** Contiene il percorso assoluto o relativo del file da importare.
Se **PATH** inizia con il prefisso **\$APPPATH**, il file viene cercato all'interno delle cartelle di Isous. Il file deve essere sempre completo di estensione.

Es:

```
REMOTE.LOAD 0 1 "$APPPATH\PROJECT\IMPORT\TEST.ISO" //percorso relativo
REMOTE.LOAD 0 1 "C:\FILE\TEST.ISO" //percorso assoluto
```

11.2 REMOTE.RUN

Manda in esecuzione un programma su CN REMOTO (il programma deve essere stato precedentemente caricato)

Sintassi

REMOTE.RUN CN

- CN** Numero di CN configurato dal file *clientns.cfg*

Es:

```
REMOTE.RUN 0 //run su CN 0
```

11.3 REMOTE.STOP

Manda in STOP un programma su CN REMOTO

Sintassi

REMOTE.STOP CN

- CN** Numero di CN configurato dal file *clientns.cfg*

Es:

```
REMOTE.STOP 0 //stop su CN 0
```

11.4 REMOTE.PAUSE

Manda in PAUSA un programma su CN REMOTO

Sintassi

REMOTE.PAUSE CN

CN Numero di CN configurato dal file *clients.cfg*

Es:

```
REMOTE.PAUSE 0 //pausa su CN 0
```

11.5 REMOTE.STATUS

Legge lo stato del CN REMOTO

Sintassi

REMOTE.STATUS CN \$VAR

CN Numero di CN configurato dal file *clients.cfg*

\$VAR Variabile di deposito dello stato del CN remoto

Ritorna in \$VAR:

bit 0 → CN IN ERRORE

bit 1 → CN IN RUN

bit 2 → CN IN MOVIMENTO

bit 3 → CN IN PAUSA

Es:

```
REMOTE.LOAD 0 0 "..."  
REMOTE.LOAD 1 0 "..."  
REMOTE.RUN 0  
REMOTE.RUN 1  
@LBL_1  
REMOTE.STATUS 0 $VAR //READ STATUS CN 0 IN $VAR  
REMOTE.STATUS 1 $VAR1 // READ STATUS CN 1 IN $VAR1  
$V1=$VAR&2 //TEST BIT RUN  
$V2=$VAR1&2  
IF $V1=2 || $V2=2 //WAIT IF RUN  
  GOTO LBL_1  
END_IF
```

11.6 REMOTE.MOVE

Legge lo stato MOVIMENTO del CN REMOTO

Sintassi

REMOTE.MOVE CN \$VAR

CN Numero di CN configurato dal file *clients.cfg*

\$VAR Variabile di deposito dello stato movimento assi del CN remoto

Ritorna in \$VAR:

0 → CN ASSI FERMI E BUFFER VUOTO

1 → CN ASSI IN MOVIMENTO

Es:

```
REMOTE.MOVE 0 $VAR //legge lo stato movimento in $VAR
```

11.7 REMOTE.INFO

Legge informazioni dal CN REMOTO

Sintassi

REMOTE.INFO CN TIPO \$VAR

| | |
|--------------|---|
| CN | Numero di CN configurato dal file <i>clientns.cfg</i> |
| TIPO | Tipo informazione richiesta: 0 → Richiesta numero linea teorica in lavorazione 1 → Richiesta numero linea reale in lavorazione 2 → Richiesta actual FEED 3 → Richiesta % Override 4 → Richiesta Tabella utensile selezionata (T) 5 → Richiesta Testa utensile selezionata (H) 6 → Richiesta Risoluzione quote assi impostate (1000,10000 ecc.) 7 → Richiesta Risoluzione FEED impostata (1000 ecc.) |
| \$VAR | Variabile di deposito delle formazioni |

Es:

REMOTE.INFO 0 0 \$VAR //legge il numero di linea in lavorazione in \$VAR

11.8 REMOTE.AXIS

Legge informazioni quote assi dal CN REMOTO

Sintassi

REMOTE.AXIS CN AXIS TIPO \$VAR

| | |
|--------------|--|
| CN | Numero di CN configurato dal file <i>clientns.cfg</i> |
| AXIS | Numero asse 0=X - 1=Y - 2=Z - 3=A - 4=B - 5=C - 6=U -7=V - 8=W |
| TIPO | Tipo informazione quote richiesta: 0 → Richiesta Quota teorica assoluta allo zero macchina 1 → Richiesta Quota teorica relativa allo zero pezzo 2 → Richiesta Quota reale assoluta allo zero macchina 3 → Richiesta Quota reale relativa allo zero pezzo 4 → Richiesta Quota teorica assoluta allo zero macchina (in syncro con CN) 5 → Richiesta Quota reale assoluta allo zero macchina (in syncro con CN) 6 → Richiesta valore ZERO pezzo impostato 7 → Richiesta valore OFFSET pezzo impostato 8 → Richiesta NUMERO ZERO pezzo impostato 9 → Richiesta NUMERO OFFSET pezzo impostato |
| \$VAR | Variabile di deposito delle formazioni |

Es:

REMOTE.AXIS 0 1 0 \$VAR //legge quota asse Y

11.9 REMOTE.GROUP

Legge gruppo informazioni quote assi dal CN REMOTO

Sintassi

REMOTE.AXIS CN NAXIS TIPO \$VAR

| | |
|--------------|--|
| CN | Numero di CN configurato dal file <i>clientns.cfg</i> |
| NAXIS | Numero di assi contemporanei da leggere |
| TIPO | Tipo informazione quote richiesta: 0 → Richiesta Quota teorica assoluta allo zero macchina 1 → Richiesta Quota teorica relativa allo zero pezzo 2 → Richiesta Quota reale assoluta allo zero macchina 3 → Richiesta Quota reale relativa allo zero pezzo 4 → Richiesta Quota teorica assoluta allo zero macchina (in syncro con CN) 5 → Richiesta Quota reale assoluta allo zero macchina (in syncro con CN) 6 → Richiesta valore ZERO pezzo impostato 7 → Richiesta valore OFFSET pezzo impostato 8 → Richiesta NUMERO ZERO pezzo impostato 9 → Richiesta NUMERO OFFSET pezzo impostato |
| \$VAR | Variabile iniziale di deposito delle formazioni Le altre variabili vengono prese in successione alla variabile iniziale |

Es:

`$VARX=0 //Inizializza tre variabili`

`$VARY=0`

`$VARZ=0`

`REMOTE.GROUP 0 3 0 $VARX //legge quota asse X,Y,Z in VARX,VARY,VARZ`

11.10 REMOTE.READISOVAR

Legge una variabile ISO per INDIRIZZO dal CN REMOTO

Sintassi

REMOTE.READISOVAR CN ADDR \$VAR

| | |
|--------------|---|
| CN | Numero di CN configurato dal file <i>clientns.cfg</i> |
| ADDR | Indirizzo variabile REMOTA ISO da leggere da 10 a 32768 (da 0 a 9 sono riservate) |
| \$VAR | Variabile di deposito della lettura |

Es:

`REMOTE.READISOVAR 0 10 $VAR //legge la prima variabile`

11.11 REMOTE.READVARNAME

Legge una variabile ISO per NOME dal CN REMOTO

Sintassi

REMOTE.READVARNAME CN \$VAR "RemoteVar"

| | |
|------------------|---|
| CN | Numero di CN configurato dal file <i>clientns.cfg</i> |
| \$VAR | Variabile di deposito della lettura |
| RemoteVar | Nome della variabile remota senza \$ iniziale |

Es:

`REMOTE.READVARNAME 0 $VAR "VAR1" //legge la variabile $VAR1 su CN remoto`

11.12 REMOTE.WRITEISOVAR

Scriva una variabile ISO per INDIRIZZO su CN REMOTO

Sintassi

REMOTE.WRITEISOVAR CN ADDR \$VAR

| | |
|--------------|--|
| CN | Numero di CN configurato dal file <i>clients.cfg</i> |
| ADDR | Indirizzo variabile REMOTA ISO da scrivere da 10 a 32768 (da 0 a 9 sono riservate) |
| \$VAR | Variabile contenente il valore da scrivere |

Es:

\$VAR=100.1

REMOTE.WRITEISOVAR 0 10 \$VAR //scrive la prima variabile

11.13 REMOTE.WRITENAMEVAR

Scriva una variabile ISO per NOME su CN REMOTO

Sintassi

REMOTE.WRITENAMEVAR CN \$VAR "RemoteVar"

| | |
|------------------|--|
| CN | Numero di CN configurato dal file <i>clients.cfg</i> |
| \$VAR | Variabile contenente il valore da scrivere |
| RemoteVar | Nome della variabile remota senza \$ iniziale |

Es:

\$VAR=100.1

REMOTE.WRITENAMEVAR 0 \$VAR "VAR1" //scrive la variabile remota \$VAR1

11.14 REMOTE.READCNVAR

Legge una variabile USER dal CN REMOTO

Sintassi

REMOTE.READCNVAR CN ADDR \$VAR

| | |
|--------------|--|
| CN | Numero di CN configurato dal file <i>clients.cfg</i> |
| ADDR | Indirizzo variabile REMOTA USER da leggere da 0 a 9 |
| \$VAR | Variabile di deposito della lettura |

Es:

REMOTE.READCNVAR 0 1 \$VAR //legge la user 1

11.15 REMOTE.WRITECNVAR

Scriva una variabile USER sul CN REMOTO

Sintassi

REMOTE.WRITECNVAR CN ADDR \$VAR

| | |
|--------------|--|
| CN | Numero di CN configurato dal file <i>clients.cfg</i> |
| ADDR | Indirizzo variabile REMOTA USER da scrivere da 0 a 9 |
| \$VAR | Variabile contenente il valore da scrivere |

Es:

\$VAR=10

REMOTE.WRITECNVAR 0 1 \$VAR //scrive il valore 10 nella USER 1

11.16 REMOTE.READINPUT

Legge un ingresso digitale dal CN REMOTO

Sintassi

REMOTE.READINPUT CN NINP \$VAR

| | |
|--------------|---|
| CN | Numero di CN configurato dal file <i>clientns.cfg</i> |
| NINP | Numero ingresso digitale da leggere da 0 a 255 (max configurati su CN remoto) |
| \$VAR | Variabile di deposito della lettura |
| | 0 → Ingresso OFF |
| | 1 → Ingresso ON |

Es:

REMOTE.READINPUT 0 1 \$VAR //legge l' ingresso digitale numero 2

11.17 REMOTE.READOUT

Legge un uscita digitale dal CN REMOTO

Sintassi

REMOTE.READOUT CN NOUT \$VAR

| | |
|--------------|---|
| CN | Numero di CN configurato dal file <i>clientns.cfg</i> |
| NOUT | Numero uscita digitale da leggere da 0 a 255 (max configurati su CN remoto) |
| \$VAR | Variabile di deposito della lettura |
| | 0 → Uscita OFF |
| | 1 → Uscita ON |

Es:

REMOTE.READIOUT 0 1 \$VAR //legge l' uscita digitale numero 2

11.18 REMOTE.WRITEOUT

Scrive un uscita digitale sul CN REMOTO

Sintassi

REMOTE.WRITEOUT CN NOUT \$VAR

| | |
|--------------|--|
| CN | Numero di CN configurato dal file <i>clientns.cfg</i> |
| NOUT | Numero uscita digitale da scrivere da 0 a 255 (max configurati su CN remoto) |
| \$VAR | Variabile contenente il valore di scrittura |
| | 0 → Uscita OFF |
| | 1 → Uscita ON |

Es:

\$VAR=1

REMOTE.WRITEOUT 0 1 \$VAR //Setta l' uscita digitale numero 2

12 Configurazione CNC Remoti

Prima di utilizzare le funzioni **REMOTE.xxxx**, occorre configurare i vari CNC in RETE LAN.

I CNC possono essere anche processi sullo stesso PC, oppure trovarsi su PC diversi ma comunque uniti da una stessa LAN. Quest' ultima deve necessariamente avere un indirizzo IP STATICO (vedi configurazione IP statico sul sistema operativo).

Una rete di CNC remoti è composta dalle seguenti parti:

NSCLIENT

NSSERVER

NSCLIENT

Questo rappresenta la configurazione di uno o più CNC che aprono la connessione verso un CNC NS SERVER. Per configurare i CNC client è sufficiente che sia presente un file di nome "ClienteNs.cfg" nella cartella di installazione di Isous.

ClientNs.cfg

[USER_CN 0]

CN=IpAddr,port → CN 0 del processo 0

CN=IpAddr,port → CN 1 del processo 0

.
.

[USER_CN 1]

CN=IpAddr,port → CN 0 del processo 1 (se presente)

CN=IpAddr,port → CN 1 del processo 1 (se presente)

Dove:

IpAddr → Indirizzo IP del server per il CN 0,1,2 ecc.

port → Numero della porta di connessione del server per il CN 0,1,2 ecc.

NSSERVER

Rappresenta un CNC server in rete LAN. Questo per essere configurato come server, deve avere installato il PLUGIN "ServerNs.dll" (in Plugin\Bin) con la seguente installazione nel file UsPlugin.cfg:

```
<Set Name="UsServer" Image="" Title="@_US SERVER" Path="UsPlugin\Bin\UsServer.dll"
NameSpace="UsServer.GestUsServer" Autorun="False" Button="False" Enable="True" />
```

Inoltre deve essere presente il file "ServerNs.cfg" nella cartella di installazione di Isous server che configura la porta di ascolto del processo server.

ServerNs.cfg

[USER_CN 0]

Port → Porta di ascolto per il CN 0

[USER_CN 1]

Port → Porta di ascolto per il CN 1 (se presente)

.
.

Nota

Attenzione nella configurazione delle Porte. Occorre scegliere valori non occupati da altri processi del sistema operativo (in genere valori alti es. 36000)

Esempio di configurazione CNC REMOTI

Configurazione su LAN locale, Indirizzo IP del server **10.0.0.34** Port **36000**

Il PC che contiene il processo server deve quindi avere un IP statico **10.0.0.34**

Il PC che contiene il processo client, può avere lo stesso indirizzo IP se si tratta dello stesso PC, oppure deve avere un indirizzo congruo alla rete in uso es: 10.0.0.50

ClientNs.cfg

[USER_CN 0]

CN=10.0.0.34,36000

ServerNs.cfg

[USER_CN 0]

36000

13 ISTRUZIONI MATEMATICHE

Isous mette a disposizione le principali istruzioni matematiche che sicuramente riescono a risolvere le equazioni più complesse.

13.1 SIN

Istruzione seno di un angolo in gradi

Sin ritorna un valore di tipo DOUBLE.

Sintassi

SIN (espressione)

espressione Obbligatorio. Espressione che identifica l' angolo in radianti

Es:

\$VAR=SIN(\$VAR1*3)

13.2 COS

Istruzione coseno di un angolo in gradi

Cos ritorna un valore di tipo DOUBLE .

Sintassi

COS (espressione)

espressione Obbligatorio. Espressione che identifica l' angolo in radianti

Es:

\$VAR=COS(\$VAR1*3)

13.3 LOG

Istruzione Logaritmo – Restituisce la base del Logaritmo naturale

Log ritorna un valore di tipo DOUBLE .

Sintassi

LOG (espressione)

espressione Obbligatorio. Espressione

Es:

\$VAR=LOG(\$VAR1*3)

13.4 EXP

Istruzione Esponenziale (calcola l' esponente dell' espressione)

Exp ritorna un valore di tipo DOUBLE.

Sintassi

EXP (espressione)

espressione Obbligatorio. Espressione

Es:

\$VAR=EXP(10.15)

13.5 INT

Intero di un valore arrotondato

INT ritorna la parte intera di una valore DOUBLE arrotondato per eccesso

Sintassi

INT (espressione)

espressione Obbligatorio. Espressione numerica

Es:

\$VAR=35.14

\$VAR1=INT(\$VAR)

13.6 FIX

Intero di un valore troncato

FIX ritorna la parte intera di una valore DOUBLE troncato

Sintassi

FIX (espressione)

espressione Obbligatorio. Espressione numerica

Es:

\$VAR=35.14

\$VAR1=INT(\$VAR)

13.7 ABS

Valore assoluto

ABS ritorna il valore assoluto di un DOUBLE

Sintassi

ABS (espressione)

espressione Obbligatorio. Espressione numerica

Es:

\$VAR=-35.14

\$VAR1=ABS(\$VAR)

13.8 DRG

Abilita il passaggio angoli espressi in **GRADI** alle funzioni COS,SIN,TAN,ACOS,ASIN,ATAN

DI default ad ogni esecuzione PART PROGRAM il passaggio degli angolo è espresso in radianti

Sintassi

DRG

13.9 RAD

Abilita il passaggio angoli espressi in **RADIANTI** alle funzioni COS,SIN,TAN,ACOS,ASIN,ATAN

DI default ad ogni esecuzione PART PROGRAM il passaggio degli angolo è espresso in radianti

Sintassi

RAD

13.10 SQR

Istruzione Radice quadrata

Sqr ritorna un valore di tipo DOUBLE.

Sintassi

SQR (espressione)

espressione Obbligatorio. Espressione

Es:

\$VAR=SQR(\$VAR1)

13.11 TAN

Istruzione Tangente di un angolo in gradi

Tan ritorna un valore di tipo DOUBLE.

Sintassi

TAN (espressione)

espressione Obbligatorio. Espressione che identifica l' angolo in radianti

Es:

\$VAR=TAN(\$VAR1*3)

13.12 ATAN

Istruzione ArcoTangente di un numero

La funzione **Atan** utilizza come argomento numero il rapporto tra due lati di un triangolo rettangolo e restituisce l'angolo corrispondente espresso in radianti.

Il risultato viene espresso in radianti nell'intervallo da $-\text{PiGreco}/2$ a $\text{PiGreco}/2$.

Sintassi

ATAN (espressione)

espressione Obbligatorio. Espressione che identifica il numero

Es:

$\$VAR=ATAN(\$VAR1*3)$

13.13 ASIN

Istruzione ArcoSeno

Sintassi

ASIN (espressione)

espressione Obbligatorio. Espressione che identifica il numero

Es:

$\$VAR=ASIN(\$VAR1*3)$

13.14 ACOS

Istruzione ArcoCoSeno

Sintassi

ACOS (espressione)

espressione Obbligatorio. Espressione che identifica il numero

Es:

$\$VAR=ACOS(\$VAR1*3)$

14 UTILIZZO DEGLI OPERATORI LOGICI

In questo capitolo vengono spiegati l' utilizzo degli operatori tralasciando gli operatori matematici (+-*/).

14.1 PARENTESI ()

Separano il livelli di un espressione dando precedenza a ciò che è contenuto all' interno delle parentesi.

14.2 ESPRESSIONI IN ISO - PARENTESI []

Le parentesi QUADRE, vengono utilizzate da Isous in modo particolare. In pratica sono valide solo per alcuni tipi di variabili e per le istruzioni ISO sottoelencate. Praticamente nelle istruzioni ISO identificano al loro interno un ESPRESSIONE o una VARIABILE. Questo nasce dal fatto che originalmente il codice ISO accetta solo numeri per i CAMPI, onde rimanere rigidi a questo standard ma comunque avere la possibilità di un linguaggio più flessibile e potente, Isous utilizza le parentesi [] per identificare al suo interno un espressione o una variabile.

ISTRUZIONI ISO CHE POSSONO UTILIZZARE LE PARENTESI []

| | |
|------------------|--|
| Contatori Assi | X,Y,Z,A,B,C,U,V,W – DX,DY,DZ,DA,DB,DC,DU,DV,DW |
| Feed Assi | F |
| Speed | S |
| Correttore | D |
| Centro Arco | I,J |
| Raggio arco | R |
| Tabella utensile | T |
| Testa utensile | H |
| Indice Origini | USER_ZERO |
| Indice Offset | USER_OFFSET |

Es:

```
G1X1000 // CONVEZIONE NORMALE ISO GCODE
G1DX1000 // QUOTA INCREMENTALE (LO STESSO CON G91)
G1X[$VAR] // UTILIZZO DI UNA VARIABILE COME SPOSTAMENTO ASSI
G1X[$VAR+$VAR1*SQR(18)] // UTILIZZO DI UN ESPRESSIONE
```

14.3 CONTROLLO VARIABILI

Per il controllo sulle variabili vengono utilizzati i normali operatori = > < <> >= <= !

Questi vengono inseriti nei cicli **IF ELSE END_IF** e permettono di effettuare cicli condizionati al valore di una o più variabili. Gli operatori di uguaglianza possono essere combinati agli altri operatori logici OR (|) AND (&&). Ovviamente possono essere controllati tutti i livelli di espressione utilizzando le ().

Es:

```
IF $VAR=0 && $VAR1=5
    G1X100
ELSE
    G1X0
END_IF
```

14.4 CONTROLLO DEI BIT

Isous mette a disposizione anche operatori che permettono di controllare lo stato dei bit di una variabile. Questi possono essere utilizzati per cicli **IF ELSE END_IF** o per **settare/resettare** i singoli bit della variabile è anche possibile effettuare shift dei BIT sia a destra che a sinistra e la negazione dello stato dei singoli bit. Tutte le operazioni a Bit vengono effettuate su variabili di TIPO INTERO (pertanto viene esclusa la parte decimale).

Es:

```

IF $VAR & 4           // CONTROLLO BIT 3
      G1X100
ELSE
      G1X0
END_IF

$VAR=$VAR | 1 // SETTA BIT 1
$VAR=$VAR & 1 // RESETTA TUTTI I BIT ESCLUSO BIT 1

$VAR=1
$VAR=$VAR<<3 // SHIFT A SINISTRA DI 3 POSIZIONI

$VAR=7
$VAR=$VAR&~1

```

15 TIPI DI VARIABILI UTILIZZATE DA ISOUS

Isous utilizza le VARIABILI per aumentare le capacità di programmazione. Le variabili utilizzate da Isous dipendono dal tipo e pertanto possono essere INTERE o DOUBLE (float doppia precisione). Le variabili possono essere combinate nelle varie somme, sottrazione ecc. come tutti il linguaggi di programmazione.

15.1 COSTANTI NUMERICHE

Valori numerici sotto forma di costante separate da PUNTO DECIMALE.

Es:

```

G1X100.13
$VAR=25.143

```

15.2 VARIABILI DI UTILIZZO GENERICO

Le variabili di utilizzo generico sono tutte di tipo **DOUBLE** e servono per memorizzare dei valori che possono essere poi cambiati o utilizzati dal PartProgram. Quando la variabile viene utilizzata per istruzioni ISO (X,Y,Z ecc) deve essere contenuta tra **parentesi quadre [...]**.

| | |
|----------------|--------------------------------|
| Valore minimo | - 1.79769313486232 E308 |
| Valore massimo | 1.79769313486232 E308 |

Il numero massimo di variabili utilizzabili è 2048 vengono riconosciute con un NOME ALFANUMERICO ed il prefisso obbligatorio \$.

Es:

```

$VAR=1
$VAR=$VAR1*5
$VAR=SQR($VAR1)
G1X[$VAR]

```

15.3 VARIABILI PER INDIRIZZO

La gestione delle variabili oltre che per nome, \$VAR, può anche essere effettuata per **INDIRIZZO**, cioè con un valore da 0 a 32767. Questo in alcuni casi risulta più comodo.

:1000=10 Assegna il valore 10 alla variabile con INDIRIZZO 1000

ATTENZIONE

Le variabili per indirizzo sono comunque condivise dalle variabili \$, cioè anche queste hanno alla fine un indirizzo associato dal compilatore, pertanto onde evitare sovrapposizioni quando si usano le variabili per INDIRIZZO è consigliabile valore alti tipo 10000 o maggiori

15.4 VARIABILI PER PUNTATORE

Simili alle variabili per INDIRIZZO, ma quest' ultimo è passato tramite una variabile \$.

\$VAR=1000

:\$VAR=10 Assegna il valore 10 alla variabile con INDIRIZZO 1000

\$VAR++

:\$VAR=20 Assegna il valore 20 alla variabile con INDIRIZZO 1001

Seguono le stesse regole delle variabili per indirizzo, con eccezione che queste possono generare un errore di RUN TIME se il valore dell' indirizzo supera il limite max delle variabili.

15.5 VARIABILI ARRAY - DIM

Le variabili ARRAY sono utilizzate come normali variabili di utilizzo **GENERICO**, ma possono essere **INDICIZZATE**.

Prima di essere utilizzate è necessario dichiarare la dimensione dell' ARRAY con **DIM** e la variabile sulla quale è appoggiato:

Es:

```
DIM $ARR 10
```

Nell' esempio sopra è stato dichiarato un ARRAY di nome ARR di 10 elementi.

Gli ELEMENTI sono indicizzati partendo da 0, pertanto il valore in questo caso è da 0 a 9

UTILIZZO DELL' ARRAY

Le variabili ARRAY contengono valori del tipo delle **VARIABILI DI UTILIZZO GENERICO** (double), l' unica differenza è che per accedere ad esse è necessario inserire il valore dell' indice racchiuso tra “[.]”.

Es:

```
DIM $ARR 10
$INDEX=0
$VAR=2
LOOP 10
    $ARR[$INDEX]=$VAR*$INDEX
    $INDEX=$INDEX+1
END_LOOP
$INDEX=0
LOOP 10
    G1 X[$ARR[$INDEX]]
    $INDEX=$INDEX+1
END_LOOP
```

15.6 VARIABILI QUOTE ASSI TEORICHE/REALI

Queste variabili contengono le quote teoriche degli assi e pertanto sono di sola lettura.

Il numero massimo di queste variabili è 9. Possono essere utilizzate nel PartProgram per leggere la posizione degli assi

in un determinato momento.

Variabili quote ASSI teoriche - ReadOnly

\$(Qn) Con **n** compreso tra **0 e 8** che identifica l' asse. La variabile ritorna il valore della quota **TEORICA RELATIVA** cioè inclusa dei vari **OFFSET** dovuti agli **ZERO PEZZO** ecc.

Se "**n**" è un valore compreso tra **100 e 108** che identifica l' asse (togliendo 100). La variabile ritorna il valore della quota **TEORICA ASSOLUTA** allo **ZERO MACCHINA**.

Variabili quote ASSI reali – ReadOnly

\$(Rn) Con **n** compreso tra **0 e 8** che identifica l' asse. La variabile ritorna il valore della quota **REALE RELATIVA** cioè inclusa dei vari **OFFSET** dovuti agli **ZERO PEZZO** ecc.

Se "**n**" è un valore compreso tra **100 e 108** che identifica l' asse (togliendo 100). La variabile ritorna il valore della quota **REALE ASSOLUTA** allo **ZERO MACCHINA**.

| Quote Relative | Quote Assolute |
|-----------------------|-----------------------|
| n=0 ASSE X | n=100 ASSE X |
| n=1 ASSE Y | n=101 ASSE Y |
| n=2 ASSE Z ecc | n=102 ASSE Z ecc |

Es:

```
IF $(Q0)>100.3 // CONTROLLA QUOTA TEORICA ASSE X
    G101
END_IF
$VAR=$(R1) // LEGGE QUOTA REALE ASSE Y
```

15.7 VARIABILI INGRESSI/USCITE DIGITALI

Queste servono per leggere gli ingressi digitali del CN o settare/resettare le uscite digitali. Per quanto riguarda quest' ultime, occorre sempre tenere presente che un eventuale ciclo PLC sul CN potrebbe avere la precedenza su queste. La possibilità di scrivere uscite digitali e leggere ingressi, permette di utilizzare nel PartProgram una logica PLC in modo immediato. Questo tipo di variabili considera solamente valori 0 e 1. **Queste variabili quando inserite nel PartProgram permettono l' accesso alle I/O in modo diretto, ma non attendono nessun sincronismo con gli ASSI. Quindi in alcuni casi occorre utilizzare l' istruzione G62 (attesa fine movimento assi) per avere un perfetto sincronismo con l' operazione di I/O svolta**

Variabili ingressi digitali – - ReadOnly

\$(In) Dove "**n**" è un valore compreso tra **0 e 255** che identifica l' ingresso
Valore=0 ingresso a OFF **Valore=1** ingresso a ON

Variabili uscite digitali – Read/Write

\$(On) Dove **n** è un valore compreso tra 0 e 255 che identifica l' uscita
Valore=0 uscita a OFF **Valore=1** uscita a ON

Es:

```
IF $(I0)=1 // SE L' INGRESSO 1 e' ATTIVO
    $(O5)=1 // ALZA L' USCITA 6
ELSE
    $(O5)=0 // ALTRIMENTI AZZERA
END_IF
```


15.8 VARIABILI TIMER

Queste servono per settare i TIMER di sistema. I timer vengono utilizzati per generare dei tempi utilizzati poi nel PartProgram per scopi generici. Il numero massimo di TIMER disponibili è 10. Il TIMER viene settato con un tempo che ha una risoluzione del millisecondo . Una volta che il TIMER è stato settato, questo automaticamente decrementa il proprio valore fin a che non raggiunge 0.

Variabili timer – Read/Write

\$(Tn) Dove “n” è un valore compreso tra 0 e 9 che identifica il timer da attivare

Es:

```
$(T0)=1500           // TIMER 1 A 1500 MILLISECONDI (1.5 SECONDI)
@CICLO
IF $(I0)=1
    GOTO EXIT        // ESCE SE INGRESSO 1 SETTATO
END_IF
IF $(T0)=0           // TEST TIMER
    ERROR 10         // GENERA ERRORE INTERROMPENDO IL PARTPROGRAM
END_IF
GOTO CICLO           // CICLA
@EXIT
```

15.9 VARIABILI TABELLA UTENSILE

Queste servono leggere i PARAMETRI della TABELLA utensile selezionata con **Tn**. Di seguito vengono descritti i parametri della TABELLA UTENSILE. Sono disponibili fino ad un massimo di 20 parametri per TABELLA UTENSILE.

| Address | Descrizione |
|-----------|---------------------------------|
| 0 | Diametro utensile |
| 1 | Lunghezza utensile |
| 2 | Velocità rotazione Max utensile |
| 3 | User Define 1 |
| 4 | User Define 2 |
| 5 | User Define 3 |
| 6 | User Define 4 |
| 7 | User Define 5 |
| 8 | User Define 6 |
| 9 | User Define 7 |
| 10 | User Define 8 |
| 11 | User Define 9 |
| 12 | User Define 10 |
| 13 | User Define 11 |
| 14 | User Define 12 |
| 15 | User Define 13 |
| 16 | User Define 14 |
| 17 | User Define 15 |
| 18 | User Define 16 |
| 19 | User Define 17 |

Variabili Testa Utensile – Read/Write

\$[Un] Dove “n” è un valore compreso tra 0 e 19 che identifica l' indirizzo del parametro

Es:

```
$VAR=${U0}           // LEGGE DIAMETRO UTENSILE TABELLA SELEZIONATA
```

15.10 VARIABILI TESTA UTENSILE

Queste servono leggere/scrivere i PARAMETRI della TESTA utensile selezionata con **Hn**. Di seguito vengono descritti i parametri della TESTA UTENSILE. Sono disponibili fino ad un massimo di 20 parametri per TESTA UTENSILE.

| Address | Descrizione |
|---------|--------------------------------------|
| 0 | Offset testa da 0 macchina su asse X |
| 1 | Offset testa da 0 macchina su asse Y |
| 2 | Offset testa da 0 macchina su asse Z |
| 3 | Offset testa da 0 macchina su asse A |
| 4 | Offset testa da 0 macchina su asse B |
| 5 | Offset testa da 0 macchina su asse C |
| 6 | Offset testa da 0 macchina su asse U |
| 7 | Offset testa da 0 macchina su asse V |
| 8 | Offset testa da 0 macchina su asse W |
| 9 | Offset sensore da 0 testa su asse X |
| 10 | Offset sensore da 0 testa su asse Y |
| 11 | Offset sensore da 0 testa su asse Z |
| 12 | Offset sensore da 0 testa su asse A |
| 13 | Offset sensore da 0 testa su asse B |
| 14 | Offset sensore da 0 testa su asse C |
| 15 | Offset sensore da 0 testa su asse U |
| 16 | Offset sensore da 0 testa su asse V |
| 17 | Offset sensore da 0 testa su asse W |
| 18 | User Define 1 |
| 19 | User Define 2 |

Variabili Testa Utensile – ReadOnly

\$(Hn) Dove “n” è un valore compreso tra 0 e 19 che identifica l' indirizzo del parametro

Es:

`$VAR=$(H0) // LEGGE OFFSET X`

15.11 VARIABILI CONTATORI ASSI

Servono per leggere i contatori assi settati dall' interpolatore. Questi contengono le quote assi dove si trovano attualmente oppure dove si dovrebbero trovarsi al seguito a delle movimentazioni assi. In genere queste variabili anticipano la posizione dell' asse prima che questo sia posizionato. Questo risulta utile per la ripartenza da blocco.

Variabili Contatore Assi – ReadOnly

\$(Cn) Dove “n” è un valore compreso tra 0 e 8 che identifica l' asse

15.12 VARIABILI LIMITI ASSI

Servono per leggere i limiti assi impostati nei parametri macchina

Variabili Limiti Assi Positivi – ReadOnly

\$(Nn) Dove “n” è un valore compreso tra 0 e 8 che identifica l' asse

Variabili Limiti Assi Negativi – ReadOnly

\$(Sn) Dove “n” è un valore compreso tra 0 e 8 che identifica l' asse

15.13 STRUTTURE DATI

IsoUs può gestire variabili presente in strutture dati. Le strutture servono per raccogliere variabili di tipo \$ appartenenti ad un insieme. Questo semplifica la programmazione in quanto rende più semplice l' utilizzo delle variabili.

Le strutture dati sono sempre **GLOBALI** e pertanto sono visibili anche nelle funzioni **M**. Le strutture vengono create dal PlugIn **UsStructManager** (vedi documentazione sullo store).

Una struttura viene definita da una **CATEGORIA** (nome della struttura) e da una o più **VARIABILI** (dati della struttura).

Es. di struttura definita:

| NOME | TOOLPAR |
|------------------|----------------|
| VARIABILI | PAR1 |
| | PAR2 |
| | PAR3 |

Pertanto a livello di **GCODE** viene usata la seguente sintassi:

\$TOOLPAR.PAR1=10

\$TOOLPAR.PAR2=10+\$TOOLPAR.PAR1

Le strutture sono sempre disponibili una volta definite con **UsStructManager** e vengono gestite come una normale variabile

15.14 PARAMETRI PER PREVIEW

Questo tipo di variabile, legge o imposta i parametri per il Preview (Simulazione Gcode) in modo di adattarlo al tipo di Gcode.

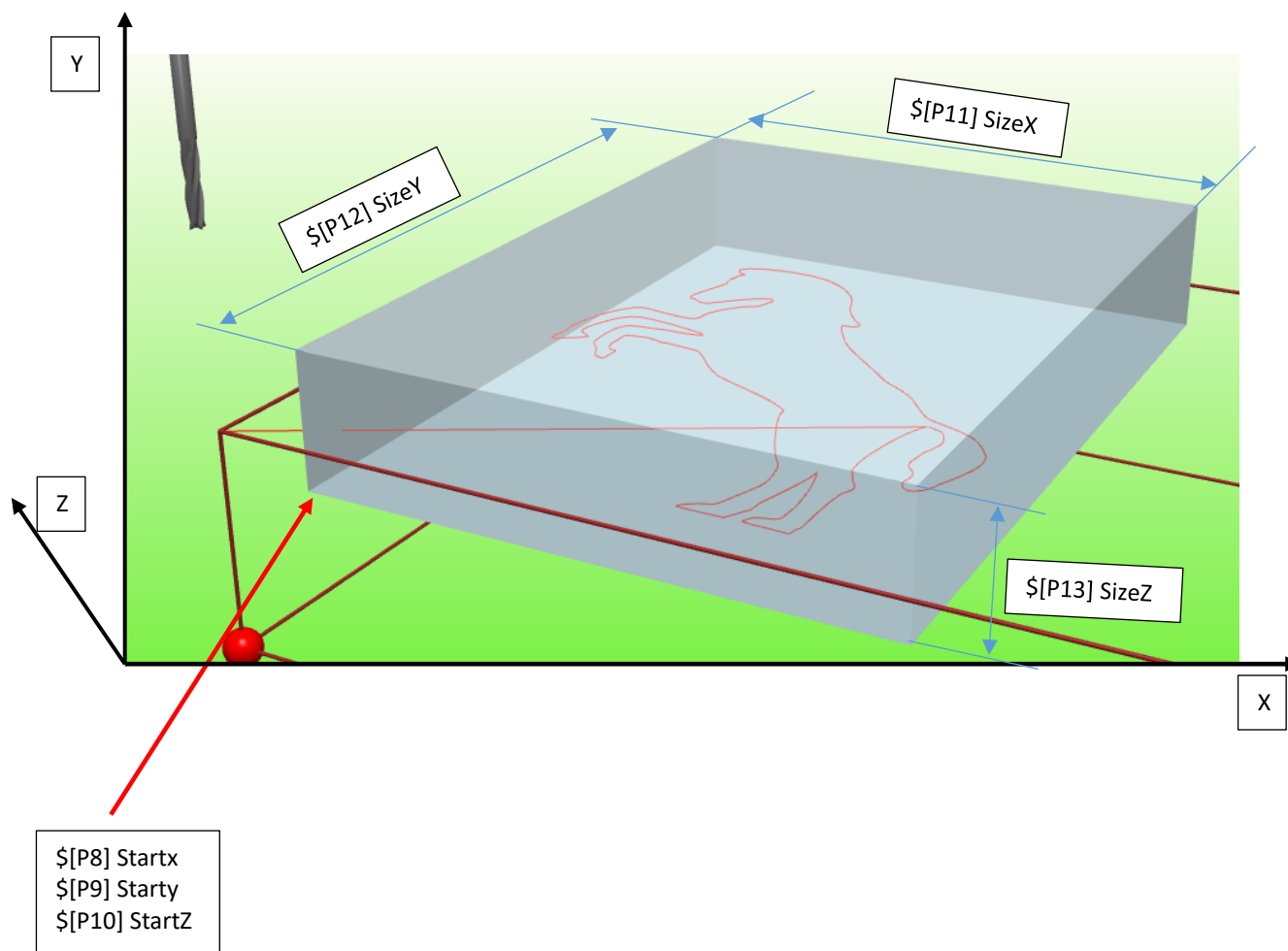
\$(Pn) Dove “n” è un valore compreso tra 0 e 16 che identifica il tipo di Parametro

| Pn | Descrizione | Valori |
|----|---|---|
| 0 | Abilita/Disabilita Asse rotativo | 0 Disabilita 1 Abilita |
| 1 | Setta direzione asse Z per asse rotativo | 0 Positiva verso il basso 1 Negativa verso il basso |
| 2 | Tipo di simulazione | 0 Line 1 Mesh 2 Tornio 3 Close Mesh 4 LayerT |
| 3 | Abilita/Disabilita Offset teste | 0 Disabilitato 1 Abilitato |
| 4 | Seleziona tipo di cursore | 0 Pointer 1 RTCP 10-49 Tipo Tools dove 10 Tool0-11 Tool1 ecc. 50-99 Tipo Blade dove 50 blade0 -51 Blade1 ecc |
| 5 | Indice Asse Blade (Lama) | Da 0 a numero assi-1 |
| 6 | Offset Blade (Lama) | Da 0 a 360 |
| 7 | Diametro Utensile | Valore positivo |
| 8 | Box Start X | Vedi figura sotto |
| 9 | Box Start Y | Vedi figura sotto |
| 10 | Box Start Z | Vedi figura sotto |
| 11 | Box Size X | Vedi figura sotto |
| 12 | Box Size Y | Vedi figura sotto |
| 13 | Box Size Z | Vedi figura sotto |
| 14 | Show box | Valore >0 disegna il box con i parametri da P8 a P13. Questo parametro deve essere messo per ultimo una volta inseriti tutti i parametri P8..P13 |
| 15 | LASER ON/OFF per RMS (Real machine Simulation) | Valore>0 Abilita la simulazione Laser Acceso per visualizzazione punto di fuoco in simulazione macchina |
| 16 | TOOL ON/OFF per RMS (Real machine Simulation) | Valore>0 Abilita la simulazione Mandrino Acceso per visualizzazione punto di lavorazione in simulazione macchina |
| 17 | TEST COLLISION ON/OFF per RMS (Real machine Simulation) | Valore>0 Abilita il run con test collisione assi |
| 18 | TOOL LENGHT per RMS (Real machine Simulation) | Valore che aggiorna la lunghezza utensile in preview es: dopo G102 |
| 19 | SHOW MODEL 3D BOX per RMS (Real machine Simulation) | Valore>0 rende visibile il modello del pezzo 3D |
| 20 | Profondità MODEL 3D BOX per RMS (Real machine Simulation) | Profondità del modello del pezzo 3D |

| | | |
|--------------|---|--|
| 21 | Larghezza MODEL 3D BOX per RMS (Real machine Simulation) | Larghezza del modello del pezzo 3D |
| 22 | Altezza MODEL 3D BOX per RMS (Real machine Simulation) | Altezza del modello del pezzo 3D |
| 23 | <i>Simula Acquisizione Sul Pezzo come da G102 per RMS (Real machine Simulation). Se SHOW MODEL 3D BOX acquisisce sul Box, altrimenti sul Piano Macchina</i> | <i>Valore = 1 Acquisisce Valore = 0 Reset</i> |
| 24 | <i>OFFSET sul \${P23}</i> | <i>Aggiunge un offset all' acquisizione del pezzo con \${P23}</i> |
| 25-36 | <i>RESERVED</i> | <i>Lettura MinX, MaX, MinY, MaxY, MinZ, MaxZ MaX1, MinY1, MaxY1, MinZ1, MaxZ1</i> |
| 37 | <i>Preview Laser Raster</i> | <i>Valore =1 Abilita Preview Laser Raster Mode In \${p7} Diametro Spot Laser Valore = 0 Disabilita</i> |

Disegno del Box ingombro pezzo

Tramite i parametri P8..P14 è possibile disegnare un Box che evidenzia le dimensioni di un pezzo in lavorazione. Questo permette di rendere più realistica la simulazione, in quanto questa verrà visualizzata negli ingombri del pezzo.

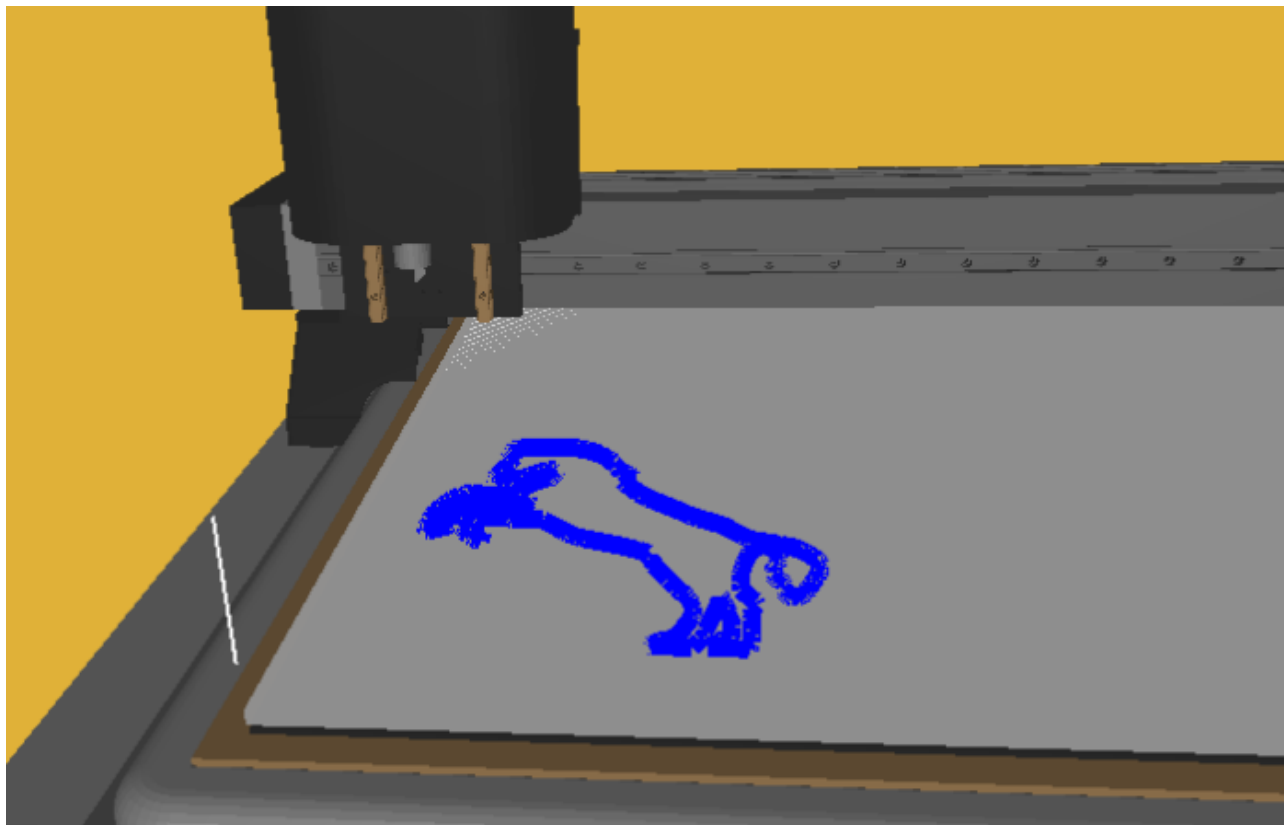
**Esempio:**

```

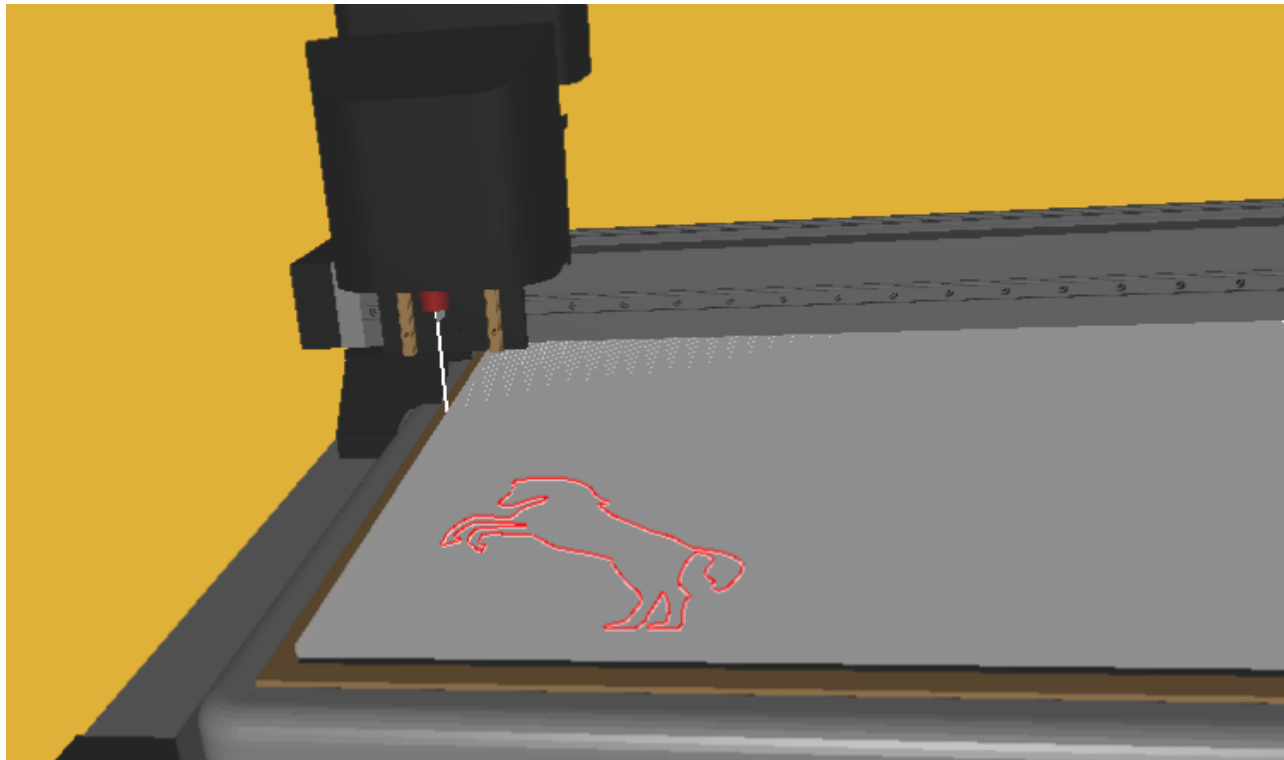
 $\$[P8]=50$  // START X
 $\$[P9]=150$  // START Y
 $\$[P10]=0$  // START Z
 $\$[P11]=300$  // SIZE X
 $\$[P12]=200$  // SIZE Y
 $\$[P13]=70$  // SIZE Z
 $\$[P14]=1$  // SHOWS BOX

```

Disegno con Tool On e lavorazione Fuori pezzo



Disegno con Tool On e lavorazione Interno pezzo



Acquisizione origine Pezzo con \${P23}

Es:

H2

 $\${P23}=1$

G1X0Y0

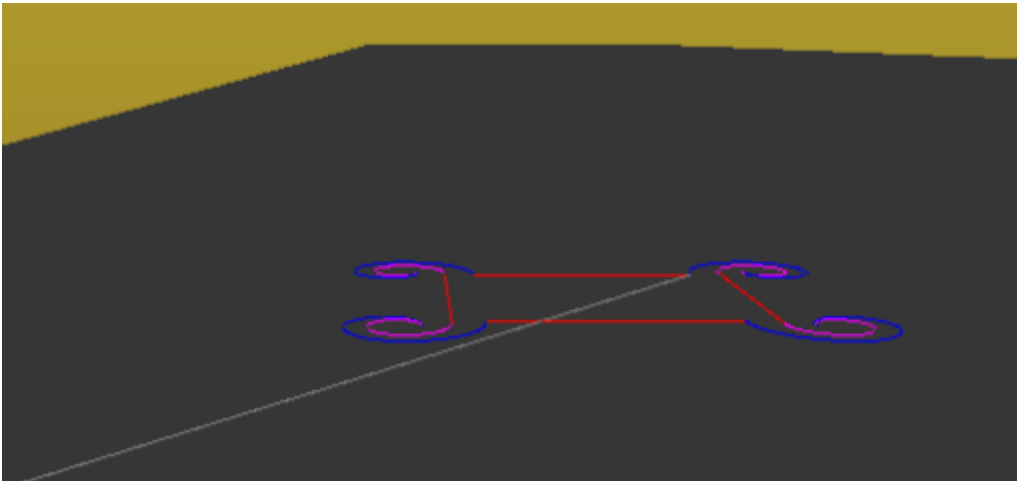
G60

G0X202.018Y271.361

G2X203.211Y280.401I236.814J271.368

.

.



15.15 VARIABILI GLOBAL

Il prefisso GLOBAL viene applicato solamente alle variabili di utilizzo GENERICO (\$nome). Tale prefisso ha ragione di esistere solamente nelle funzioni M o HM. Questo poichè per tali funzioni Isous considera private sia le LABEL sia le VARIABILI, di conseguenza se si rende necessario condividere una variabile con il PartProgram principale o con un'altra funzione M/HM deve essere preceduta dalla parola chiave GLOBAL. Questo fa in modo che la variabile dichiarata sia visibile a tutti i moduli con lo stesso nome.

Es:

```
//ESEMPIO DI REALIZZAZIONE MACRO DI PAUSA
//DICHARA GLOBALE $SAVEX E $SAVEY
GLOBAL $SAVEX
GLOBAL $SAVEY
G62                // ATTENDE ASSI FERMI
$SAVEX=${Q0}    // SALVA LA QUOTA ASSE X
$SAVEY=${Q1}    // SALVA LA QUOTA ASSE Y
GOXOYO
// ESEMPIO DI REALIZZAZIONE MACRO RIPRESA DA PAUSA
//DICHARA GLOBALE $SAVEX E $SAVEY
GLOBAL $SAVEX
GLOBAL $SAVEY
GOX[$SAVEX] Y[$SAVEY] // RIPORTA GLI ASSI AL PUNTO SALVATO DALLA MACRO DI PAUSA
G62
```

15.16 VARIABILI PREDEFINITE PER FUNZIONI M

Queste variabili servono per settare i parametri alle funzioni M definite su CN. Questo poichè le funzioni M possono essere parametrizzate in modo da condizionare il funzionamento in base al valore di uno o più parametri.

I parametri devono essere scritti prima di chiamare la funzione M.

Il numero di parametri passati alla funzione M è deciso dalla configurazione di Isous, comunque ha un valore massimo di 10 parametri (default 5 parametri)

ATTENZIONE:

Il CN vede i parametri solamente come valori interi .

Es:

```
// INVOCA UNA M SU CN SCRIVENDO PRIMA DUE PARAMETRI
$_PARAM_1=100
$_PARAM_2=300
M15
```

15.17 VARIABILI PARAMETRI SPECIALI

Servono per leggere alcuni parametri impostati su Isous. In seguito sono riassunti i vari parametri a cui è possibile accedere tramite queste variabili.

- \$(X0)** Legge la SPEED attuale impostata con la funzione SPEED - *ReadOnly*
- \$(X1)** Legge l' indice dello ZERO PEZZO attuale - *ReadOnly*
- \$(X2)** Legge l' indice dell' OFFSET PEZZO attuale - *ReadOnly*
- \$(X3)** Legge abilitazione ZERO PEZZO – torna 1 se abilitato 0 se disabilitato - *ReadOnly*
- \$(X4)** Legge abilitazione OFFSET PEZZO – torna 1 se abilitato 0 se disabilitato – *ReadOnly*
- \$(X5)** Legge il numero di testa impostato con H – *ReadOnly*
- \$(X6)** Legge il numero di utensile impostato con T – *ReadOnly*
- \$(X7)** Legge il tipo di RUN in corso – *ReadOnly*
 - 0 RUN NORMALE
 - 1 RUN IN SIMULAZIONE
 - 2 RUN PER RIPRESA DA BLOCCO
 - 3 RUN PER RETRACE
 - 4 RUN PER CALCOLO TEMPI ESECUZIONE GCODE
- X8** Legge l' ultima FEED impostata – *ReadOnly*
- X9** Legge programmazione quote – *ReadOnly*
 - 0 G90 absolute
 - 1 G91 relative
- X10** Legge piano di lavoro impostato – *ReadOnly*
 - 0 XY G17
 - 1 XZ G18
 - 2 YZ G19
 - 1 Altri
- X11** Legge correttore lunghezza utensile impostato – *ReadOnly*
 - 0 Nessun correttore impostato
 - 1 Valore (+/-) del correttore impostato
- X12** Legge Numero asse correttore lunghezza utensile impostato – *ReadOnly*
 - 0 Asse X
 - 1 Asse Y
 - 2 Asse Z
 - 3 Asse A
 - 4 Asse B
 - 5 Asse C
 - 6 Asse U
 - 7 Asse V
 - 8 Asse W
- X13** Legge il tasto premuto da LIB MESSAGE– *ReadOnly*
- X14** Legge il tipo di GoBlock invocato– *ReadOnly*
 - 0 GoBlock da linea o marker
 - 1 GoBlock da M6Tn
- X15** Legge l' indice dell' asse selezionato per il JOG (Es: 0=X – 1=Y ecc.) – *ReadOnly*
- X16** Legge lo stato dell' asse in VELOCITA'
 - 0 Asse Fermo
 - 1 Asse in Accelerazione
 - 2 Asse In Decelerazione
 - 3 Asse in Velocità raggiunta

- X17** *Legge se Run da CMD – ReadOnly*
0 RUN Normale (no CMD)
1 RUN da CMD
2 RUN SCRIPT da CMD
- X18** *Parametro P1 CMD – ReadOnly*
X19 *Parametro P2 CMD – ReadOnly*
X20 *Parametro P3 CMD – ReadOnly*
X21 *Parametro P4 CMD – ReadOnly*
X22 *Parametro P5 CMD – ReadOnly*
X23 *Parametro P6 CMD – ReadOnly*
X24 *Parametro P7 CMD – ReadOnly*
X25 *Parametro P8 CMD – ReadOnly*
X26 *Parametro P9 CMD – ReadOnly*
X27 *Parametro P10 CMD – ReadOnly*
X28 *Parametro X cicli Fissi – ReadOnly*
X29 *Parametro Y cicli Fissi – ReadOnly*
X30 *Parametro Z cicli Fissi – ReadOnly*
X31 *Parametro R cicli Fissi – ReadOnly*
X32 *Parametro P cicli Fissi – ReadOnly*
X33 *Parametro Q cicli Fissi – ReadOnly*
X34 *Parametro K cicli Fissi – ReadOnly*
X35 *Parametro F cicli Fissi – ReadOnly*
X36 *Parametro A cicli Fissi – ReadOnly*
X37 *Parametro B cicli Fissi – ReadOnly*
X38 *Reserved*
- X39** *Legge Attivazione G43 – ReadOnly*
0 Disattivata
1 Attivata
- X40-X48** *Legge Stato Home Axis (X,Y,Z,A,B,C,U,V,W) – ReadOnly*
0 Home non Effettuato
1 Home Effettuato
- X49-X57** *Legge Stato Enable Axis (X,Y,Z,A,B,C,U,V,W) – ReadOnly*
0 Enable non Effettuato
1 Enable Effettuato
- \$(X58)** *Legge sospensione ZERO PEZZO – torna 1 se sospeso 0 se non sospeso - ReadOnly*
- \$(X59)** *Legge sospensione OFFSET PEZZO – torna 1 se sospeso 0 se non sospeso – ReadOnly*
- \$(X60)** *Legge lo stato STOP_MODE impostato - ReadOnly*
- \$(X61)** *Legge lo stato PAUSE_MODE impostato – ReadOnly*
- \$(X62)** *Legge lo stato FILTER_MODE impostato - ReadOnly*
- \$(X63)** *Legge se abilitata specularità orizzontale – ReadOnly*
0 Disabilitata
1 Abilitata
- \$(X64)** *Legge se abilitata specularità verticale – ReadOnly*
0 Disabilitata
1 Abilitata
- \$(X65)** *Legge codice G ripresa da Retrace – ReadOnly*
0 G0
1 G1
2 G2
3 G3

15.18 VARIABILI ZERO E OFFSET PEZZO

Queste variabili contengono le quote dello ZERO e OFFSET PEZZO IMPOSTATE dell' ' indice abilitato.

Il numero massimo di queste variabili è 9. Possono essere utilizzate nel PartProgram per leggere la posizione degli zeri e offset attuali.

Variabili zero pezzo - ReadOnly

\$(Yn) Dove n è un valore compreso tra 0 e 8 che identifica l' asse. La variabile ritorna il valore impostato per lo zero pezzo dell' asse indicato dell' indice attuale in uso.

Variabili offset pezzo - ReadOnly

\$(Wn) Dove n è un valore compreso tra 0 e 8 che identifica l' asse. La variabile ritorna il valore impostato per l' offset pezzo dell' asse indicato dell' indice attuale in uso.

n=0 ASSE X -n=1 ASSE Y- n=2 ASSE Z ecc.

15.19 VARIABILI USER GENERIC

Queste variabili vengono utilizzate per scambio di dati generici sul CN. Il numero massimo disponibile è **30**. Possono essere lette e scritte e contengono dei valori interi a 32 bit.

Generalmente vengono utilizzate per scambiare dei valori con l' applicazione sul CN.

Variabili User - Read/Write

\$(Kn) Dove n è un valore compreso tra **0 e 29** che identifica la variabile interessata

15.20 VARIABILI MARKER

Il marker sono delle normali variabili inserite nel PartProgram. Isous permette di riprendere il PartPrgram quando queste variabili hanno raggiunto un certo valore. Nella programmazione EVOLUTA, con l' utilizzo di CICLI LOOP e delle VARIABILI, la ripresa dal numero di LINEA non è sufficiente, in quanto le quote assi possono essere passate da valore delle variabili che sono detreminate da un ciclo LOOP. Utilizzando i MARKER, è possibile riprendere il PartProgram dal valore di uno di questi e non dal numero di linea, quindi è possibile discriminare la ripresa all' interno di cicli LOOP.I MARKER vengono definiti dall' istruzione ISO **MARKER \$NOMEVAR DESCRIZIONE**.

I Marker sono recuperabili dal PlugIn **RIPRESA DA BLOCCO**

Es:

Nell' esempio seguente viene definita una variabile Marker di nome **\$INC**.

Questa rappresenta nel LOOP il pezzo in esecuzione

MARKER \$INC NUMERO PEZZI

\$VAR=0

\$INC=0

F5

G1X0Y0

LOOP 10

\$INC=\$INC+1

G1X200

\$VAR=\$VAR+50

G0X0Y[\$VAR]

END_LOOP

E' possibile quindi attivare la ripresa del PartProgram quando la variabile **\$INC** (il MARKER) assume un certo valore.

15.21 VARIABILE USCITA ANALOGICA PER MANDRINO

Serve per scrivere un valore in VOLT sul l' uscita Analogica generalmente utilizzata per controllo **SPEED MANDRINO**.

Variabile Analog Output - Read/Write

\$(A0)=val Dove **val** dipende dalla risoluzione del canale impostato

15.21.1 Canali Impostabili

Sono impostabili fino a **16** diversi canali. Questi vengono impostati dai **PARAMETRI MACCHINA**, tabella **SPINDLE** dal parametro **SPEED_ANALOG_CH**.

I canali da 0-15 fanno riferimento all' uscita analogica che può trovarsi sulle schede **NGIO-NGPP** per **NGWARP** oppure **NGMSX** per **NGMEVO**. In pratica il **canale 0** è la prima uscita analogica della **PRIMA** scheda **NGIO-NGPP-NGMSX**, il **canale 1** è la seconda uscita delle suddette schede, il **canale 2** è la prima uscita analogica della **SECONDA** scheda **NGIO-NGPP-NGMSX** ecc.

Il Canale **NGMEVO PWM** è l' uscita analogica presente solo su la scheda **NGMEVO** (opzionale)

I canali **SPEED_X,Y,Z,A,B,C,U,V,W** fanno riferimento al controllo asse in velocità per la funzione **G108.4** e **G108.5**.

In pratica indicano quale asse in velocità viene agganciato dalla funzione **G108.4**

Es: se indicato **SPEED_X**, la funzione **G108.4** si riferisce all' asse X per l' aggancio in velocità e la funzione **S** imposta la velocità su questo asse

15.21.2 Risoluzione dei Canali

I canali hanno la seguente risoluzione:

Canali da 0 a 15 **12 bit** **Valori da 0 a 2047**

Canale NGMEVO PWM **8 bit** **Valori da 0 a 255**

(attenzione il canale PWM può saturare, cioè raggiungere il valore di 10V prima di 255)

La risoluzione del canale va impostata nei **PARAMETRI MACCHINA**, tabella **SPINDLE** dal parametro **ANALOG_BIT_RES**. Pertanto impostare i seguenti valori per questo parametro:

Canali da 0 a 15 **2048**

NGMEVO PWM **Max 256 ma generalmente un valore da 210 a 230**

(Impostare per prima 256 come valore, controllare scrivendo nella variabile **\$(A0)** un valore compreso tra 210 e 255 e misurare la tensione in uscita, quando è **10V**, scrivere questo valore nel parametro **ANALOG_BIT_RES**)

Esempio M3 accensione mandrino oraria

```

READ_PARMAC "SPEEDMAXSPINDLE" $MAX_RPM // READ MAX RPM AT 10V
READ_PARMAC "ANALOG_BIT_RES" $BIT_RES // READ BIT RESOLUTION
$CURRENT_SPEED=$[X0] // READ THE CURRENT SPEED SET BY S GCODE FUNCTION
$ANALOG_OUT=$CURRENT_SPEED*$BIT_RES/$MAX_RPM // CALC
$(A0)=$ANALOG_OUT // WRITE
$(O1)=1 // SET CW DIRECTION
$(O2)=1 // START

```

15.22 GESTIONE MACRO e Parametri Gcode

La variabile J (Lettura/Scrittura) serve per gestire le varie Macro Attivazione/Disattivazione

| \$(Jn) | Valore | Descrizione |
|---------|--------|---|
| \$(J0) | 1 | Sospende: Mstop,Merror,Mpause,MEndProgram,Mtime,MinterruptDI, MinterruptDO, M50003,M50004,M60001,M60002,M60003,M60004 |
| | 0 | Riprende: Mstop,Merror,Mpause,MEndProgram,Mtime,MinterruptDI, MinterruptDO, M50003,M50004,M60001,M60002,M60003,M60004 |
| \$(J1) | 1 | Sospende: Mstop |
| | 0 | Riprende: Mstop |
| \$(J2) | 1 | Sospende: Merror |
| | 0 | Riprende: Merror |
| \$(J3) | 1 | Sospende: Mpause |
| | 0 | Riprende: Mpause |
| \$(J4) | 1 | Sospende: MEndProgram |
| | 0 | Riprende: MEndProgram |
| \$(J5) | 1 | Sospende: Mtime |
| | 0 | Riprende: Mtime |
| \$(J6) | 1 | Sospende: MinterruptDI |
| | 0 | Riprende: MinterruptDI |
| \$(J7) | 1 | Sospende: MinterruptDO |
| | 0 | Riprende: MinterruptDO |
| \$(J8) | 1 | Sospende: M50003 |
| | 0 | Riprende: M50003 |
| \$(J9) | 1 | Sospende: M50004 |
| | 0 | Riprende: M50004 |
| \$(J10) | 1 | Sospende: M60001 |
| | 0 | Riprende: M60001 |
| \$(J11) | 1 | Sospende: M60002 |
| | 0 | Riprende: M60002 |
| \$(J12) | 1 | Sospende: M60003 |
| | 0 | Riprende: M60003 |
| \$(J13) | 1 | Sospende: M60004 |
| | 0 | Riprende: M60004 |
| \$(J14) | 1 | Sospende: STOP – Non Attivo solo Emergenza |
| | 0 | Riprende: STOP |
| \$(J15) | 1 | Sospende: PAUSA – Non Attiva sia da CN che Da PC |
| | 0 | Riprende: PAUSA |
| \$(J16) | 1 | Attiva aggiornamento contatori in automatico da funzione Hn o G43 |
| | 0 | Disattiva aggiornamento contatori in automatico da funzione Hn o G43 |
| \$(J17) | | Gestione numero campionamenti file di debug RESERVED |
| \$(J18) | | Gestione numero elementi file di debug RESERVED |
| \$(J19) | | Gestione numero Assi file di debug RESERVED |
| \$(J20) | 0 | Centri ASSOLUTI per G2 G3 parametro ARC_REL |
| | 1 | Centri RELATIVI per G2 G3 parametro ARC_REL |
| \$(J21) | 0-13 | Indica la profondità in blocchi del buffer per G66 X-100 (default 0 viene presa quella del CN) Valori compresi tra 0-13 1-16 2-32 3-64 4-128 5-256 6-512 7-1024 8-2048 9-4096 10-8192 11-16384 12-32768 13-65536 |
| \$(J22) | 0-1 | Valore =1 Abilita conversione S su G100 per gestione potenza Laser SVal->G100 XVal Valore=0 Disabilita (al RUN sempre disabilitata) |

16 GESTIONE DEI POSIZIONATORI

Isous mette a disposizione la completa gestione di uno o più assi posizionatori. Questi possono essere di diversa tipologia e comunque la loro gestione è affidata all' applicazione VTB del CNC in uso

16.1 PA-PD(n,par)espr - Set posizione assoluta e controllo posizionatore

L' istruzione PA, permette di muovere il posizionatore ad una quota assoluta oppure, di effettuare alcune operazioni di controllo su esso.

PD muove il posizionatore di una quota RELATIVA

| | |
|-------------|---|
| n | Numero di posizionatore da 0 a valore max posizionatori gestiti dall' applicazione VTB |
| par | Parametro di controllo opzionale |
| null | Start posizionatore con attesa posizione raggiunta a target impostato in espr |
| 0 | Start posizionatore senza attesa posizione raggiunta a target impostato in espr |
| 1 | Start posizionatore con attesa posizione raggiunta a target impostato in espr |
| 2 | Start HOMING (l' espressione deve essere messa ma non utilizzata) |
| 3 | Se espressione >0 abilita l' asse posizionatore Se l' espressione <=0 Disabilita l' asse posizionatore |

Es:

```
PA(0)1000 // Muove a quota 1000 il posizionatore 0
PA(0,0)1000 // Muove a quota 1000 il posizionatore 0 senza attesa fine mov
PA(0)[$VAR] // Muove a quota $VAR il posizionatore 0
PA(0,2)1 // START HOMING
PA(0,3)1 // ENABLE POSIZIONATORE 0
```

16.2 PF(n)espr - Set FEED posizionatore

Imposta la velocità del posizionatore. Espr valore della FEED

n Numero di posizionatore da 0 a valore max posizionatori gestiti dall' applicazione VTB

Es:

```
PF(0)100 // posizionatore 0 FEED 100
PF(0)[$VAR] // posizionatore 0 FEED $VAR
```

16.3 PS(n) - STOP Movimento posizionatore

Stop movimento

n Numero di posizionatore da 0 a valore max posizionatori gestiti dall' applicazione VTB

Es:

```
PS(0) // Stop posizionatore 0
```

16.4 _PM(n,par) - Lettura stato posizionatore

Legge lo stato del posizionatore

n Numero di posizionatore da 0 a valore max posizionatori gestiti dall' applicazione VTB

par Parametro

| | |
|----------|---|
| 0 | Legge stato del movimento- 0 asse fermo - 1 asse in movimento |
| 1 | Legge la quota TEORICA |
| 2 | Legge la quota REALE |
| 3 | Torna 1 asse abilitato – 0 asse disabilitato |
| 4 | Torna 1 HOMING fatto– 0 HOMING non fatto |
| 5 | Torna 1 asse in ALLARME – 0 OK |

Es:

```
$VAR=_PM(0,par)
```


17 IMPOSTAZIONE DELLE LAVORAZIONI

Vengono definite in questo capitolo tutte le funzioni relative alla movimentazione degli assi e alle loro condizioni di lavoro.

17.1 DEFINIZIONE DELLE QUOTE ASSI

Definiscono la posizione degli assi per il tipo di interpolazione utilizzata (G0-G1-G2-G3).

Il codice dell' asse può essere seguito da un valore numerico, o da un' espressione che identifica la quota.

Codici ASSI

| | |
|----------|---|
| X | Asse X |
| Y | Asse Y |
| Z | Asse Z |
| A | Asse A |
| B | Asse B |
| C | Asse C |
| U | Asse U |
| V | Asse V |
| W | Asse V |
| I | Centro dell' arco su asse X |
| J | Centro dell' arco su asse Y |
| K | Centro dell' arco su asse Z (valido solo per G18 -G19) |
| R | Raggio dell' arco |

Sintassi

CODICE valore o espressione

Es:

| | |
|------------------------------------|----------------------------|
| X100Y20 | // QUOTE DIRETTE |
| DX100Y20 | // INCREMENTALI E ASSOLUTE |
| X[\$VAR]Y[\$VAR1] | // VARIABILI |
| X[COS(\$VAR)+\$VAR1] | // ESPRESSIONI |

17.1.1 Definizione quote Assi per indirizzo Canale

| | | | |
|-----------|-----------------|-----------|-----------------|
| QX | Canale 0 | QY | Canale 1 |
| QZ | Canale 2 | QA | Canale 3 |
| QB | Canale 4 | QC | Canale 5 |
| QU | Canale 6 | QV | Canale 7 |
| QW | Canale 8 | | |

La definizione delle quote assi per indirizzo canale, permette di fare riferimento **SEMPRE** ad un indirizzo del canale, non considerando il NOME DEGLI ASSI

Es se gli assi sono nominati nel seguente modo:

| | |
|---|----------|
| X | Canale 0 |
| Y | Canale 1 |
| Z | Canale 2 |
| B | Canale 3 |
| C | Canale 4 |
| W | Canale 5 |

QX,QY ecc. fanno sempre riferimento al canale e non al nome dell' asse

Quindi

G1 X100 Y200 B300

Equivale a

G1 QX100 QY200 QA300

17.1.2 G90 - PROGRAMMAZIONE CON QUOTE ASSOLUTE**Sintassi****G90****Tipo funzione***MODALE (default)***Revoca***G91***Descrizione**

Definisce lo spostamento degli assi riferito allo **ZERO IN USO** (ZERO MACCHINA, ZERO PEZZO, OFFSET PEZZO). Se non è stato utilizzata nessuna origine pezzo o offset pezzo le quote con G90 sono riferite all'0 macchina.

17.1.3 G91 - PROGRAMMAZIONE CON QUOTE INCREMENTALI**Sintassi****G91 G91.1 G91.2****Tipo funzione***MODALE***Revoca***G90***Descrizione**

Definisce lo spostamento degli assi riferito alla posizione attuale di questi

G91.1 Forza la movimentazione con valori assoluti G90 salvando lo stato attuale

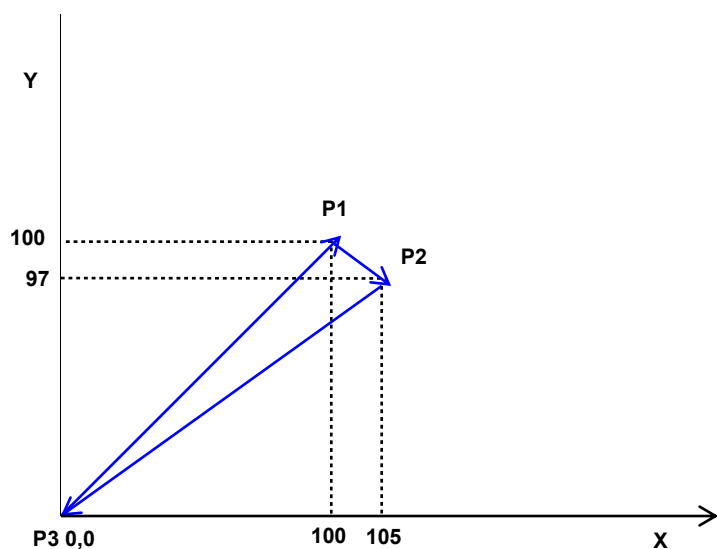
G91.2 Reimposta lo stato salvato con G91.1 (G90 o G91)

17.1.4 Definizione delle coordinate incrementali assolute**Es:**

```

G90           // QUOTE ASSOLUTE
G1X100Y100   // P1 ASSI A 100,100
G91           // QUOTE INCREMENTALI
X5Y-3        // P2 ASSI A 105,97
G90           // QUOTE ASSOLUTE
X0Y0         // P3 A 0,0

```



17.2 ORIGINI PEZZO

Isous può utilizzare fino a **256 diverse origini pezzo** selezionabile tramite la funzione **G94 G92** e l'istruzione **USER_ZERO** (in alternativa OX,Oy, ecc.). Un origine pezzo o zero pezzo definisce un punto di riferimento quote rispetto all'origine macchina.

17.2.1 Origine Pezzo da INDICE

Queste origine possono essere settate in modo immediato prendendo il valore contenuto nell'indice del file "ZERI.VAL". A differenza di USER_ZERO, questo possono avere indici diversi per ogni ASSE.

Sintassi

OXIndex OYIndex

Tipo funzione

MODALE

Revoca

G92-G82

Parametri

Index – Indice dell'origine per l'asse contenuto nel file "ZERI.VAL"

Descrizione

Definisce la nuova origine degli assi in base al valore contenuto nell'indice impostato

Es: **OX0 OY1 OZ3** // Imposta origine X su indice 0, Y su indice 1, Z su indice 3

17.2.2 G94 - Definizione di un origine pezzo a quote definite da parametro

Sintassi

G94 Xvalore Yvalore (DX,DY ecc. valori relativi)

Tipo funzione

MODALE

Revoca

G92-G82

Parametri

X,Y,Z,A,B,C,U,V,W (QX,QY,QZ,QA,QB,QC,QU,QV,QW) seguito dal valore di origine

Descrizione

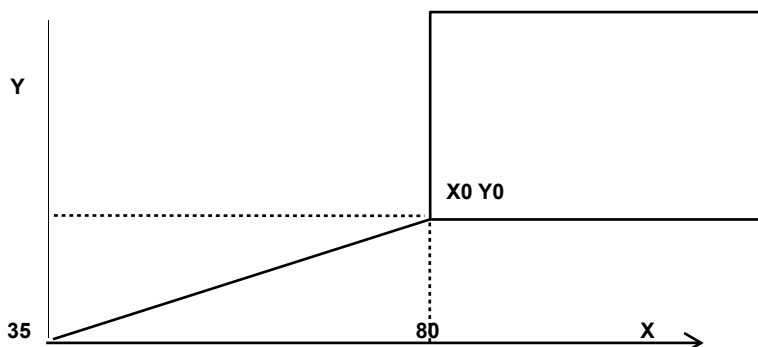
Definisce la nuova origine degli assi. Gli assi che non vengono passati come parametri non cambiano di origine.

Il nuovo punto di ZERO per il singolo asse sarà definito dal valore passato come parametro.

Il valore passato come parametro è riferito allo **ZERO MACCHINA** ed è sempre in valore assoluto quindi non influenzato da **G90 G91**. L'origine pezzo è automaticamente abilitata dall'istruzione G94.

Es:

G94 X80Y35 // DEFINISCE LA NUOVA ORIGINE PEZZO AD 80,35



17.2.3 G94.n – Salva le attuali origini nel file**Sintassi****G94.n ecc. X Y Z**

Tipo funzione

IMMEDIATA**Parametri****X,Y,Z,A,B,C,U,V,W** *Origini degli assi da salvare***.n** *Indice dello origini da salvare*

Descrizione

G94.n salva le attuali origini dell' indice passato in G94.n degli assi selezionati

Es:

G94.1 XY // SALVA LE ORIGINI DELL' INDICE 1 PER GLI ASSI X,Y**17.2.4 G54-G55-G56-G57-G58-G59 Definizione di un origine PEZZO da FILE in memoria****Sintassi****G54 (funzione composta per gestione degli assi superiori a Z)****G54 XYZ o G54.n XYZ - G54 QX QY QZ o G54.n QX QY QZ**

Tipo funzione

MODALE

Revoca

G92-G82**Parametri****X,Y,Z,A,B,C,U,V,W (QX,QY,QZ,QA,QB,QC,QU,QV,QW)****.n** *Indice dello zero pezzo da utilizzare. Se omessa usa l' indice indicato da USER_ZERO*

Descrizione

Definisce la nuova origine degli assi. Gli assi che non vengono passati come parametri non cambiano di origine. Il nuovo punto di ZERO per il singolo asse sarà **PRESO DAL FILE DEGLI ZERI GENERATO DAL PLUGIN ZEROPEZZO**.

L' origine pezzo è automaticamente abilitata dall' istruzioni.

Se a **G54** non viene passato nessun parametro viene preso per default l' **ASSE X**.

G54 può utilizzare l' indice indicato dopo il punto

G54.n dove **n** è un valore da 0 a 255 che indica l' indice dello ZERO PEZZO da utilizzare.Il parametro non **OBBLIGATORIO** **.n** non sporca la variabile **USER_ZERO**, pertanto questa rimane al suo ultimo valore.In modo analogo: (queste usano sempre l' indice indicato da **USER_ZERO**)**G55 ZERO ASSE Y G56 ZERO ASSE Z G57 ZERO ASSI XY G58 ZERO ASSI YZ****G59 ZERO ASSI XZ**

Rispetto alla funzione G94, queste prendono gli ZERO PEZZO da un file salvato in memoria del PC. Pertanto gli ZERO PEZZO possono essere ripresi anche dopo lo spegnimento della macchina.

Es:

USER_ZERO 0 // Setta l'indice 0 (opzione di default)**G54 XYZ** // Zero pezzo su X Y Z**USER_ZERO 1** // Setta l'indice 1**G55** // Zero pezzo Y su indice 1**G54.2 XYZ** // Zero pezzo su X Y Z su indice 2 Non e' necessario **USER_INDEX**

17.2.5 G92 - Definizione di un origine pezzo nel punto attuale degli assi**Sintassi****G92 XYZ ecc.****Tipo funzione****MODALE****Revoca****G94-G82****Parametri****X,Y,Z,A,B,C,U,V,W (QX,QY,QZ,QA,QB,QC,QU,QV,QW) asse da azzerare****Descrizione**

Definisce la nuova origine degli assi nel punto attuale dove si trovano.

Gli assi indicati vengono azzerati nel punto attuale.

Se a G92 non viene passato nessun parametro vengono presi automaticamente tutti gli ASSI**17.2.6 G82 - Definizione di un origine pezzo nel punto attuale degli assi da sensore****(Spostata su G1082 se USE_G80_CYCLES=TRUE)****Sintassi****G82 XYZ G82 QX QY QZ ecc.****ecc.****Tipo funzione****MODALE****Revoca****G94-G92****Parametri****X,Y,Z,A,B,C,U,V,W (QX,QY,QZ,QA,QB,QC,QU,QV,QW) asse da azzerare****Descrizione**

Definisce la nuova origine degli assi nel punto attuale dove si trovano considerando l' acquisizione da sensore (G102).

Gli assi indicati vengono azzerati nel punto attuale togliendo l' offset del sensore dallo zero macchina.

Se a G82 non viene passato nessun parametro vengono presi automaticamente tutti gli ASSI**Questa funzione è da utilizzare quando si effettuano acquisizioni origini tramite G102.****17.2.7 G98 – G53 - Sospensione origine pezzo****Sintassi****G98****G53****Tipo funzione****MODALE****Revoca****G82 G92 G94 -G54-G55-G56-G57-G58-G59**

Descrizione

Sospende tutti gli zero pezzi riportando l' origine allo ZERO MACCHINA (se non abilitati gli OFFSET ASSI G93 G95)

17.2.8 G99 - Ripresa origine pezzo**Sintassi****G99****Tipo funzione***MODALE***Revoca***G98***Descrizione**

Riprende le origini pezzo programmate con G92 G94 disabilitando G98.

17.2.9 G940 – Movimentazione assi escludendo ORIGINI e OFFSET solo nel blocco attuale**Sintassi****G940 G1Xval Yval****Tipo funzione***Autocancellante***Descrizione**

Le movimentazioni assi che seguono nel BLOCCO G940 vengono effettuate riferite allo ZERO MACCHINA.

17.2.10 USER_ZERO - Indice del vettore ORIGINE ASSI**Sintassi****USER_ZERO** espressione

Tipo funzione

MODALE**Parametri**

espressione o valore che indica l' indice dello zero assi da utilizzare

Descrizione

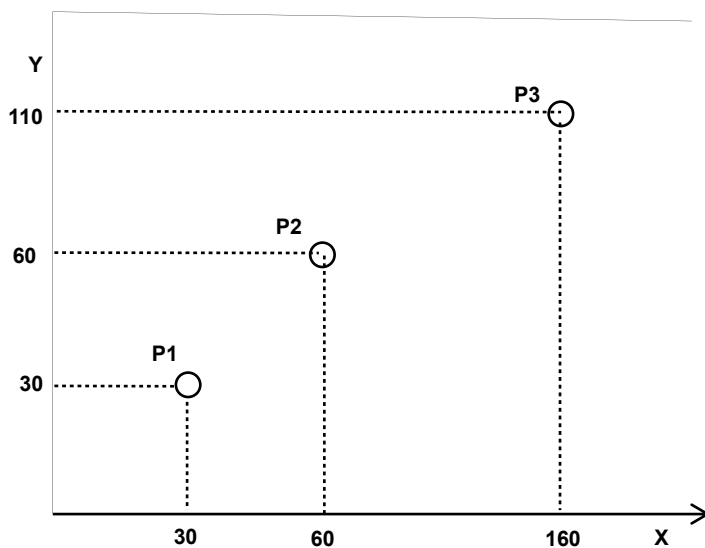
Viene abilitato l' origine pezzo contenuta nella posizione del vettore indicizzato da USER_ZERO.

Es:

```

USER_ZERO 0
G94 X20Y20           // NELLA POSIZIONE 0
USER_ZERO 1
G94 X50Y50           //NELLA POSIZIONE 1
USER_ZERO 2
G94 X150Y100 //NELLA POSIZIONE 2
USER_ZERO 0           // ABILITA INDICE 0
G1X10Y10             // P1
USER_ZERO 1           // ABILITA INDICE 1
G1X10Y10             // P2
USER_ZERO 2           // ABILITA INDICE 2
G1X10Y10             // P3

```



17.3 OFFSET PEZZO

Isous può utilizzare fino a **256 diverse offset pezzo** selezionabile tramite la funzione **G93 G95** e l'istruzione **USER_OFFSET**. Un offset pezzo è un ulteriore origine pezzo addizionale rispetto a quelle utilizzate con G92 G94 . Gli offset pezzo definiscono un punto di riferimento quote rispetto all' origina macchina. Per gli esempi vedere ORIGINE PEZZO.

17.3.1 G93 - Definizione di un offset pezzo a quote definite da parametro

Sintassi

G93 Xvalore **Y**valore (DX,DY ecc. valori relativi)

G93 P(n)valore **DP(n)**valore per posizionatori

Oppure **DOX**valore **DOY**valore ecc.

Tipo funzione

MODALE

Revoca

G85 G95

Parametri

X,Y,Z,A,B,C,U,V,W, (QX,QY,QZ,QA,QB,QC,QU,QV,QW) P (nr posiz) seguito dal valore di offset

Descrizione

Definisce il nuovo offset degli assi. Gli assi che non vengono passati come parametri non cambiano di offset.

Il nuovo punto di OFFSET per il singolo asse sarà definito dal valore passato come parametro.

Il valore passato come parametro è riferito allo **ZERO MACCHINA** ed è sempre in valore assoluto quindi non influenzato da **G90 G91**.

Il nuovo offset è automaticamente abilitato dall' istruzione G93.

17.3.2 G95 - Definizione di un offset pezzo nel punto attuale degli assi

Sintassi

G95 XYZ ecc.

Tipo funzione

MODALE

Revoca

G85 G93

Parametri

X,Y,Z,A,B,C,U,V,W (QX,QY,QZ,QA,QB,QC,QU,QV,QW) asse da assegnare offset

Descrizione

Definisce il nuovo offset degli assi nel punto attuale dove si trovano.

Gli assi indicati prendono come offset il punto attuale.

Se a G95 non viene passato nessun parametro vengono presi automaticamente tutti gli ASSI

17.3.3 G85 - Definizione di un offset pezzo nel punto attuale degli assi da sensore**Sintassi****G85 XYZ ecc. G85 QX QY QZ ecc.****Tipo funzione****MODALE****Revoca****G93-G95****Parametri****X,Y,Z,A,B,C,U,V,W (QX,QY,QZ,QA,QB,QC,QU,QV,QW) asse da azzerare****Descrizione**

Definisce il nuovo offset degli assi nel punto attuale dove si trovano considerando l' acquisizione da sensore (G102).
 Gli assi indicati vengono azzerati nel punto attuale togliendo l' offset del sensore dallo zero macchina.

Se a G85 non viene passato nessun parametro vengono presi automaticamente tutti gli ASSI

Questa funzione è da utilizzare quando si effettuano acquisizioni origini tramite G102.

17.3.4 G86 – Preset asse in modo hardware sul modulo 360 gradi**Sintassi****G86 XYZ ecc. G86 QX QY QZ ecc.****Tipo funzione****IMMEDIATA****Parametri****X,Y,Z,A,B,C,U,V,W (QX,QY,QZ,QA,QB,QC,QU,QV,QW) asse da presetare in modo hardware sul modulo 360 gradi****Descrizione****Istruzione per gestione di assi rotativi**

G86 esegue un preset asse in modo Hardware sul modulo 360 gradi.

In pratica l' asse viene presetato ai gradi relativi alla posizione assoluta.

Es:

Se la quota asse ha un valore di 1.450 il preset viene effettuato a 10 gradi

17.3.5 G96 - Sospensione offset pezzo**Sintassi****G96****Tipo funzione****MODALE****Revoca****G85 G93 G95****Descrizione**

Sospende tutti gli offset pezzi riportando l' origine allo ZERO MACCHINA (se non abilitati LE ORIGINI PEZZO G92 G94)

17.3.6 G97 - Ripresa offset pezzo

Sintassi

G97

Tipo funzione

MODALE

Revoca

G96

Descrizione

Riprende gli offset pezzo programmati con G93 G95 disabilitando G96.

17.3.7 USER_OFFSET - Indice del vettore OFFSET ASSI

Sintassi

USER_OFFSET espressione

Tipo funzione

MODALE

Parametri

espressione o valore che indica l' indice dello offset assi da utilizzare

Descrizione

Viene abilitato l' offset pezzo contenuta nella posizione del vettore indicizzato da USER_OFFSET.

Es:

```

USER_OFFSET 0
G94 X20Y20           // NELLA POSIZIONE 0
USER_OFFSET 1
G94 X50Y50           //NELLA POSIZIONE 1
USER_OFFSET 2
G94 X150Y100 //NELLA POSIZIONE 2
USER_OFFSET 0 // ABILITA INDICE 0
G1X10Y10             // P1
USER_OFFSET 1 // ABILITA INDICE 1
G1X10Y10             // P2
USER_OFFSET 2 // ABILITA INDICE 2
G1X10Y10             // P3

```

17.4 Selezione della testa di lavoro – Funzione H

Isous può utilizzare fino a **256 diverse teste porta utensili**.

Ognuna di queste viene riferita ai punti di Zero Macchina reali tramite la TABELLA TESTE.

Normalmente nessuna testa viene abilitata, tramite la funzione Hn è possibile selezionare una delle 256 teste disponibili (Prima testa valore 0). Automaticamente verranno abilitati tutti gli offset relativi alla testa selezionata.

17.4.1 Hn – Seleziona testa utensile (mandrino)

Sintassi

H espressione

Tipo funzione

MODALE

Parametri

espressione o valore che indica la testa da utilizzare (prima testa valore 0)

Descrizione

Vengono abilitati gli offset relativi alla testa selezionata.

Es:

H0 // ABILITA PRIMA TESTA

17.4.2 G87 – Sospensione offset testa

G87

Tipo funzione

MODALE

Revoca

G88

Descrizione

Vengono sospesi gli offset testa.

Es:

G87 // SOSPENSIONE OFFSET TESTA

17.4.3 G88 – Ripresa offset testa

G88

Tipo funzione

MODALE

Revoca

G87

Descrizione

Vengono ripresi gli offset testa.

Es:

G88 // RIPRENDE OFFSET TESTA

17.5 GESTIONE ASSI ROTATIVI

Isous può gestire assi rotativi in diverse modalità. Queste vengono definite dalla funzione G36.

17.5.1 G36 - Definizione di un asse rotativo

Sintassi

G36 Xmodo NOMEASSE

Tipo funzione

MODALE

Parametri

Modo: Definisce il tipo di asse rotativo:

0 → Disabilitato

1 → Strada più corta

2 → Sempre positivo

3 → Sempre negativo

4 → Strada più corta ma disabilitato durante G2/G3

NOMEASSE nome dell' asse rotativo (es A,B ecc.)

Descrizione

La funzione G36 gestisce gli assi rotativi. Questi ovviamente dipendono dalla tipologia della macchina.

Strada più corta

Il CN sceglie il percorso più corto per portarsi nel punto indicato.

Il modo 4 è uguale al modo 1 ma nei tratti G1/G2 l' asse rotativo non viene considerato

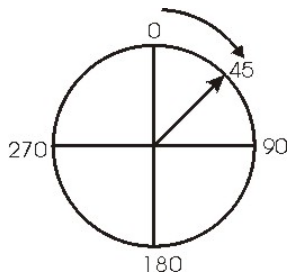
Es:

G36 X1 A

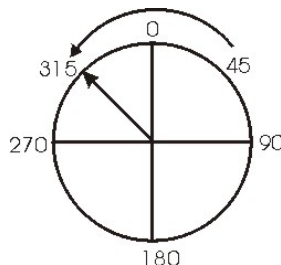
G1 A45

G1 A315

primo movimento G1 A45



secondo movimento G1 A315



Sempre Positivo/Negativo

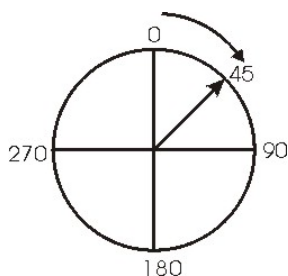
L' asse ruota sempre nel senso indicato non considerando la strada più corta

G36 X2 A

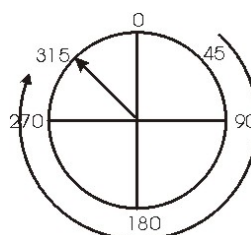
G1 A45

G1 A315

primo movimento G1 A45



secondo movimento G1 A315



17.6 PRESET HARDWARE CONTATORI ASSI

In alcuni casi è necessario presetare i contatori encoder degli assi in modo Hardware. Questo permette di evitare Overflow per assi rotativi in un unico senso.

Le funzioni di preset assi in modo HardWare vanno utilizzate con molta attenzione, in quanto se l' asse contiene un errore di spazio. Questo non viene considerato e quindi è possibile perdere degli impulsi encoder durante la fase di preset (solo assi soggetti ad errore encoder).

17.6.1 G89 – Preset contatore asse in modo HardWare

Sintassi

G89 Xvalore Yvalore (DX,DY ecc. valori relativi)

Tipo funzione

DIRETTA

Revoca

Tutte le funzioni di offset assi e zero pezzo

Parametri

X,Y,Z,A,B,C,U,V,W (QX,QY,QZ,QA,QB,QC,QU,QV,QW) seguito dal valore di offset

Descrizione

Definisce il nuovo valore del contatore asse nella posizione attuale dove si trova.

17.7 PRESET CONTATORI ASSI DOPO PAUSA e Prima delle macro speciali

In alcuni casi è necessario presetare i contatori degli assi dopo una ripresa da pausa oppure da retrace. Questo poichè durante la pausa gli assi si sono mossi utilizzando funzioni particolari (tipo shift ecc.).

Non presetando correttamente i contatori, possono verificarsi movimenti anomali durante la fase di interpolazione assi. Generalmente i contatori assi sono sempre aggiornati e quindi non risulta necessario un aggiornamento, tuttavia ci sono dei casi come quelli sopra elencati che rendono necessario un aggiornamento. Utilizzare sempre un aggiornamento assi alla fine delle M di ripresa da PAUSA i da RETRACE è comunque una cosa consigliabile.

17.7.1 G84 – Preset contatori CPU

(Spostata su G1084 se USE_G80_CYCLES=TRUE)

Sintassi

G84 X Y Z G84 QX QY QZ....

Tipo funzione

DIRETTA

Parametri

X,Y,Z,A,B,C,U,V,W (QX,QY,QZ,QA,QB,QC,QU,QV,QW) asse da presetare

Se non indicato nessun asse vengono presetati tutti

Descrizione

La funzione G84 copia i valori delle quote assi dal processo in corso (tipicamente ripresa da PAUSA o da RETRACE) al processo di LAVORO

17.7.2 G83 – Preset contatori CPU 1

(Spostata su G1083 se USE_G80_CYCLES=TRUE)

Sintassi

G83

Tipo funzione

DIRETTA

Descrizione

La funzione G83 serve esclusivamente per forzare l' aggiornamento dei contatori della CPU 1, cioè quella che lavora

sulle MACRO di PAUSA,STOP,ERRORE.

E' consigliabile iniziare il codice della MACRO con l' istruzione G83

17.8 SELEZIONE DEL PIANO DI LAVORO

Il piano di lavoro o piano di contornatura definisce:

- Il piano dell' interpolazione circolare
- il piano del calcolo della fermata sul tratto o meno (interpolazione veloce G60)
- il piano dove viene calcola la compensazione offset utensile

17.8.1 G17 - Piano di lavoro su X-Y

Sintassi

G17

Tipo funzione

MODALE

Revoca

G18 G19 G70

17.8.2 G18 - Piano di lavoro su X-Z

Sintassi

G18

Tipo funzione

MODALE

Revoca

G17 G19 G70

17.8.3 G18.1 - Piano di lavoro su X-Z Ma non in Preview

Sintassi

G18.1

Tipo funzione

MODALE

Revoca

G17 G19 G70

Viene settato il piano di lavoro su X-Z ma rimane inalterato su X-Y nel PREVIEW

17.8.4 G19 - Piano di lavoro su Y-Z

Sintassi

G19

Tipo funzione

MODALE

Revoca

G17 G18 G70

17.8.5 G19.1 - Piano di lavoro su Y-Z Ma non in Preview**Sintassi****G19.1**

Tipo funzione

MODALE

Revoca

*G17 G18 G70*Viene settato il piano di lavoro su Y-Z ma rimane inalterato su X-Y nel PREVIEW**17.8.6 G70 - Piano di lavoro su coppia di assi a piacere****Sintassi****G70 ASSE1 ASSE2**

Tipo funzione

MODALE

Revoca

G17 G18 G19

Parametri

Nome della coppia di assi su cui si intende selezionare il piano di lavoro

Descrizione

Seleziona il piano di lavoro sulla coppia di assi a piacere

G70XA // seleziona il piano sugli assi X e A**G70QXQA** // seleziona il piano su Canale 0 Canale 3

17.9 RICERCA HOME ASSE DA PART PROGRAM

Isous permette di utilizzare una funzione G (G71) che esegue l' home dell' asse indicato. Questo può tornare utile quando utilizziamo ASSI mandrini che hanno doppia funzione:

ASSE IN VELOCITA'
ASSE CONTROLLATO

Spesso quando l' asse passa da velocità ad asse controllato è necessario effettuare una ricerca di homing

17.9.1 G71 – Home asse

Sintassi

G71 ASSE

Tipo funzione

IMMEDIATA

Parametri

Nome dell' asse da azzerare

Descrizione

Effettua la ricerca di home sull' asse indicato

Il tipo di ricerca è definito dal parametro macchine RZERO_MODE.

Il PartProgram sospende l' esecuzione fino a che la ricerca non è stata terminata.

E' comunque attivo lo STOP PROGRAM. Può essere generato un errore (1005 Time Out Home Asse) se l' asse non completa la ricerca di ZERO nel tempo indicato nei parametri di configurazione TIME_OUT_HOME

Es:

G71 X

17.9.2 G71.1 – Abilita Asse

Sintassi

G71.1 ASSE

Tipo funzione

IMMEDIATA

Parametri

Nome dell' asse da abilitare

Descrizione

Abilita l' asse Indicato. Operazione necessaria prima di G71. Può essere generato un errore (1006 Time Out Enable Asse)

Es:

G71.1 X

17.9.3 G71.2 – Disabilita Asse

Sintassi

G71.2 ASSE

Tipo funzione

IMMEDIATA

Parametri

Nome dell' asse da disabilitare

Descrizione

Disabilita l' asse Indicato.

17.10 FUNZIONI DI MOVIMENTAZIONE ASSI

Vengono spiegate tutte le funzioni di movimentazione assi adottate da Isous.

17.10.1 *G0 - Movimentazione con Velocità rapida*

Sintassi

G0

Tipo funzione

MODALE

Revoca

G1 G2 G3

Descrizione

G0 definisce il tipo di movimentazione lineare con Velocità massima degli assi.

La Velocità massima del singolo asse viene definita nei parametri macchina.

Isous utilizza uno speciale calcolo della Velocità degli assi riducendo al minimo lo STRESS di questi durante la movimentazione in RAPIDO. Al contrario di altri CN che in G0 spostano gli in modo **NON INTERPOLATO**, Isous muove sempre in **INTERPOLAZIONE**, ma calcolando il vettore ottimale di questa. In questo modo si ottiene sempre un **MOVIMENTO FLUIDO** dove tutti gli assi partono allo stesso istante e arrivano contemporaneamente.

Viene inoltre utilizzata un parametro di **ACCELERAZIONE** diverso da quello di lavoro. è consigliabile utilizzare G0 solo per spostamenti ad utensile ALTO.

Es:

G0X0Y0Z0 // SPOSTA NEL PUNTO 0,0,0

17.10.2 *G0.1 - Movimentazione con Velocità rapida Asse Singolo con Accelerazione privata*

Sintassi

G0.1 NomeAsse val

Tipo funzione

Immediata

Revoca

G1 G2 G3 (impone G0)

Descrizione

G0.1 definisce il tipo di movimentazione lineare con Velocità massima di un singolo asse con accelerazione presa dal parametro **ACC_G0.1_NomeAsse**. Al contrario di G0, questa funzione muove un asse alla volta ma non utilizza come accelerazione il parametro **ACC_G0**. Di fatto è possibile programmare accelerazioni diverse per ogni singolo asse.

Le altre caratteristiche risultano uguale a G0. Se viene programmato più di un asse, viene generato un errore.

Es:

G0.1 X0 // SPOSTA NEL PUNTO 0

17.10.3 **G0.2 - Movimentazione con Velocità rapida – Per asse Tangenziale****Sintassi****G0.2**

Tipo funzione

MODALE

Revoca

G1 G2 G3**Descrizione**

G0.2 definisce il tipo di movimentazione lineare con Velocità massima degli assi simile a G0

La differenza è che la velocità viene calcolata solo per la coppia di assi del piano di lavoro, mentre i restanti assi vengono

TRASPORTATI.

Pertanto eventuali assi **TANGENZIALI** non influiscono sulla velocità degli **ASSI DEL PIANO DI LAVORO**

Es:

G0.2 X100Y100A150 // SPOSTA NEL PUNTO 100,100,150 dove A è un asse tangenziale

17.10.4 G1 - Interpolazione lineare ad F programmata**Sintassi****G1**

Tipo funzione

MODALE

Revoca

G0 G2 G3**Descrizione**

G1 definisce il modo di interpolazione lineare a Velocità programmata. Se i movimenti sono effettuati sul piano di lavoro, ed è abilitata la funzione di **INTERPOLAZIONE SENZA FERMATA SUL TRATTO G60**, Isous calcola automaticamente la fermata sugli spigoli. G1 viene eseguito alla F programmata e alla percentuale di questa selezionata dal potenziometro di OVERRIDE ASSI (se non disabilitato da G11). A seconda del CN, si possono ottenere prestazioni diverse sulla quantità di tratti G1 elaborati nell'unità di tempo. I Cn della classe NG35 elaborano circa **350 Tratti al Sec**, mentre i CN della classe NGM13 circa **120**. Questo definisce la Velocità massima di lavoro quando si devono eseguire MICRORETTE. Tutti i CN hanno un buffer di movimentazione che Isous cerca di mantenere sempre PIENO (quando si lavora in modalità G60). Se questo buffer si svuota, il CN ferma gli assi e quindi si ottiene il fastidioso **"EFFETTO SINGHIOZZO"** sulla movimentazione del percorso.

Es:

G1X300Y200Z20**17.10.5 G1.1 – Sospensione G1-G2-G3 e Impostazione G0****Sintassi****G1.1**

Tipo funzione

Immediata

Revoca

G1 - G2 - G3**Descrizione**

G1.1 Imposta per i movimenti successivi forzando **G0**, salvando la funzione **G** attualmente impostata

Es:

G1.1X300Y200Z20**17.10.6 G1.2 – Riprende G1-G2-G3****Sintassi****G1.2**

Tipo funzione

Immediata

Revoca

G0**Descrizione**

G1.2 Riprende impostazione **G1-G2-G3** sospesa con **G1.1**.

In pratica viene impostata la funzione salvata da **G1.1**

Es:

G1.2X300Y200Z20

17.10.7 G2/G3 - Interpolazione circolare ad F programmata**Sintassi****G2 (oraria)****G3 (antioraria)**

Tipo funzione

MODALE

Revoca

G0 G1**Descrizione**

G2 G3 definisce il modo di interpolazione circolare a Velocità programmata. Se i movimenti sono effettuati sul piano di lavoro, ed è abilitata la funzione di **INTERPOLAZIONE SENZA FERMATA SUL TRATTO G60**, Isous calcola automaticamente la fermata sugli spigoli. G2 G3 viene eseguito alla F programmata e alla percentuale di questa selezionata dal potenziometro di OVERRIDE ASSI (se non disabilitato da G11).

G2 o G3 possono prendere indifferentemente come parametri il centro I,J o il raggio R.

Non è comunque possibile effettuare un'intera circonferenza con il solo parametro del RAGGIO (è necessario utilizzare il centro I,J).

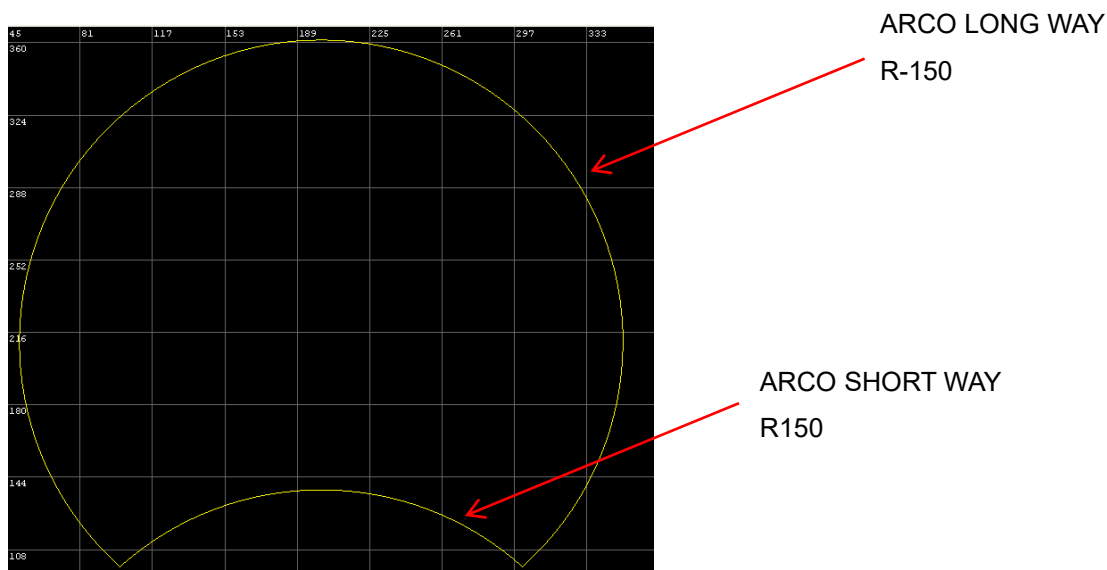
ARCO SHORT WAY E LONG WAY

Utilizzando il raggio come parametro, è possibile definire due tipi di arco che passano negli stessi punti.

Se il valore del RAGGIO è positivo viene generato un ARCO SHORT WAY (con la corda più corta).

Se il valore del RAGGIO è negativo viene generato un ARCO LONG WAY (con la corda più lunga).

Questi sono gli archi generati dall' esempio



Es:

G0X100Y100**G2X300Y100R-150** // ARCO LONG WAY**G0X100Y100****G2X300Y100R150** // ARCO SHORT WAY

17.10.8 G30 - Inserimento automatico raccordi su spigoli**Sintassi****G30 Rraggio Aangolo**

Tipo funzione

MODALE

Revoca

G33

Parametri

R valore del raggio di raccordo (raggio=0 annulla la funzione)**A** Soglia in gradi dell' angolo di inserimento

Descrizione

G30 inserisce automaticamente dei raccordi circolare su spigoli. Il raggio di raccordo è definito dal parametro R, mentre il parametro A definisce la soglia in gradi dello spigolo. Sotto a questa soglia la funzione G30 non interviene lasciando il tratto inalterato. Questa funzione risulta utile per addolcire i movimenti in percorsi SPIGOLOSI.

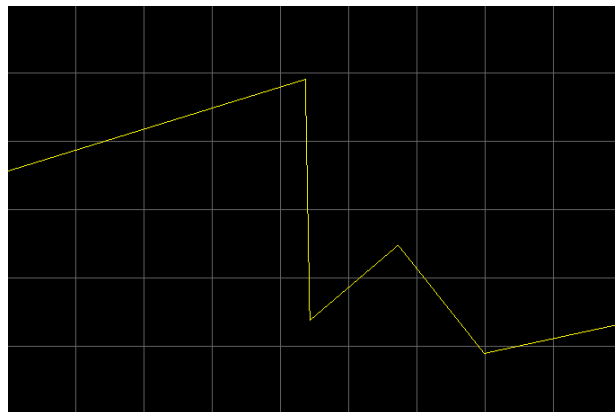
Il raccordo viene inserito solamente se questo può effettivamente essere contenuto all' interno dei due tratti.

Per disabilitare definitivamente la funzione è sufficiente inserire un RAGGIO=0.

Es:

// PERCORSO SENZA ATTIVAZIONE DI G30

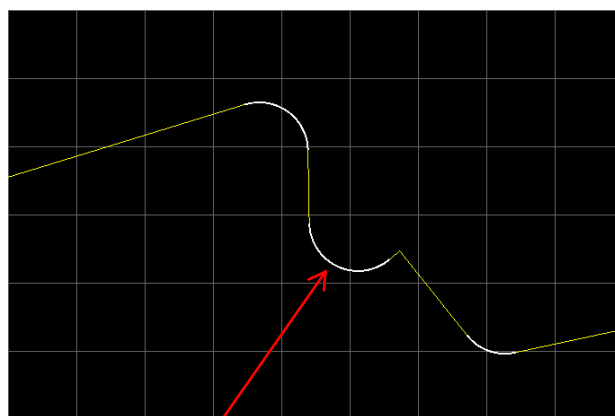
```
G0 X75.45 Y157.46
G1 X107.84 Y167.43
G1 X108.3 Y142.73
G1 X117.36 Y150.44
G1 X126.2 Y139.34
G1 X140.92 Y142.51
```



Es:

// PERCORSO CON ATTIVAZIONE DI G30

```
G0 X75.45 Y157.46
G30 R5 A20 // RAGGIO 5 ANG. 20 GRADI
G1 X107.84 Y167.43
G1 X108.3 Y142.73
G1 X117.36 Y150.44
G1 X126.2 Y139.34
G1 X140.92 Y142.51
```



R=5 mm

17.10.9 ***G31 - Sospensione Inserimento automatico raccordi su spigoli******Sintassi*****G31****Tipo funzione*****MODALE******Revoca******G30*****Descrizione**

Sospende G30. Questa può essere ripresa con G32.

17.10.10 ***G32 - Ripresa Inserimento automatico raccordi su spigoli******Sintassi*****G32****Tipo funzione*****MODALE******Revoca******G31*****Descrizione**

Riprende G30 sospesa da G31.

17.10.11 G33 - Inserimento automatico smussi su spigoli**Sintassi****G33 Rlen Aangolo****Tipo funzione****MODALE****Revoca****G30****Parametri****R** *lunghezza dello smusso (len=0 annulla la funzione)***A** *Soglia in gradi dell' angolo di inserimento***Descrizione**

G33 inserisce automaticamente degli smussi lineari su spigoli. La lunghezza dello smusso è definita dal parametro R, mentre il parametro A definisce la soglia in gradi dello spigolo. Sotto a questa soglia la funzione G33 non interviene lasciando il tratto inalterato.

Questa funzione risulta utile per addolcire i movimenti in percorsi SPIGOLOSI.

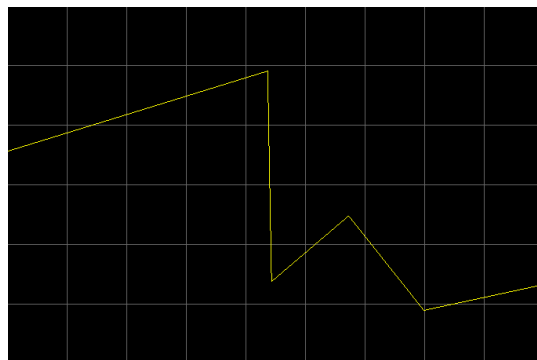
Lo smusso viene inserito solamente se questo può effettivamente essere contenuto all' interno dei due tratti.

Per disabilitare definitivamente la funzione è sufficiente inserire un LEN=0.

Es:

// PERCORSO SENZA ATTIVAZIONE DI G30

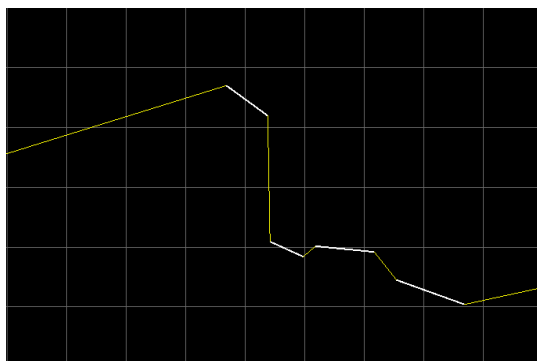
```
G0 X75.45 Y157.46
G1 X107.84 Y167.43
G1 X108.3 Y142.73
G1 X117.36 Y150.44
G1 X126.2 Y139.34
G1 X140.92 Y142.51
```



Es:

// PERCORSO CON ATTIVAZIONE DI G33

```
G0 X75.45 Y157.46
G33 R5 A20 // LEN 5MM ANG. 20 GRADI
G1 X107.84 Y167.43
G1 X108.3 Y142.73
G1 X117.36 Y150.44
G1 X126.2 Y139.34
G1 X140.92 Y142.51
```



Len=5 mm

17.10.12 ***G34 - Sospensione Inserimento automatico smussi su spigoli******Sintassi*****G34****Tipo funzione*****MODALE******Revoca******G33*****Descrizione**

Sospende G33. Questa può essere ripresa con G35.

17.10.13 ***G35- Ripresa Inserimento automatico smussi su spigoli******Sintassi*****G35****Tipo funzione*****MODALE******Revoca******G34*****Descrizione**

Riprende G33 sospesa da G34.

17.10.14 G102 – Start acquisizione quote da sensore**Sintassi****G102****G102.2****Tipo funzione***Immediata***Revoca****Descrizione**

Avvia lo start acquisizione assi da sensore.

Gli Assi vengono mossi alle posizioni indicato bei contatori X,Y,Z ecc. La quota viene presa in valore Assoluto o relativo a seconda del parametro G90 G91. Vengono considerati i vari **ZERO PEZZO** e **OFFSET ASSI** selezionati. Possono essere mossi uno i più assi contemporaneamente. La velocità viene presa dal parametro macchina **ACQ_VEL**.

Il modo di acquisizione viene indicato dal parametro macchina **ACQ_MODE**.

L' esecuzione può essere interrotta o dall' intercettazione del **SENSORE** sul pezzo, o al raggiungimento della quota di **TARGET**.

Nel caso dell' intercettazione del **SENSORE** sul pezzo il PartProgram continua l' esecuzione dal **BLOCCO SUCCESSIVO G102**.

Nel caso del raggiungimento della quota di **TARGET** il PartProgram viene interrotto con **ERRORE ACQ SENSORE**.

G102 Attende che gli assi siano fermi prima di procedere all' acquisizione.

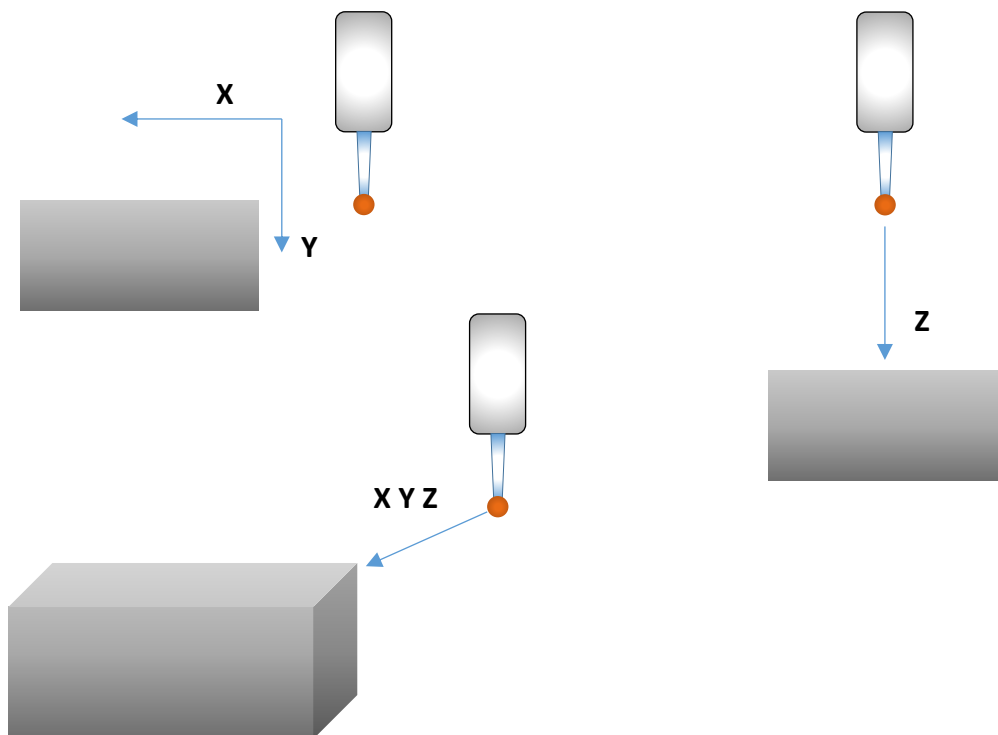
G102.2 se raggiunto il **TARGET** non **GENERA NESSUN ALLARME**

Es:**G90****G102 X100**

// ACQUISISCE SPOSTANDO X SOLAMENTE A QUOTA 100

G92 X

// ZERO PEZZO NEL PUNTO ACQUISITO



17.10.15 G102.1 – Acquisizione da sistema PxVision**Sintassi****G102.1 XYZABCUVWFS (parametri)**

Tipo funzione

Immediata

Revoca

Descrizione

IsoUs può acquisire dati dal sistema di visione **PxVision**. Per attivare la funzione **G102.1** deve essere installato il PlugIn: **UsPxVisionAcq**

Tramite questa funzione è possibile si possono effettuare le seguenti operazioni:

- Ricerca Origini Pezzo**
- Misurazione Lunghezza e Diametro utensile**
- Misurazioni Oggetti**
- Presenza Oggetti**

Ovviamente oltre al PlugIn è necessario che sia anche installato il sistema **PxVision**

La funzione **G102.1** utilizza il parametro **X** per definire la funzione di acquisizione (riferirsi al manuale di PxVision per il significato delle varie funzioni di acquisizione)

PARAMETRO X *Definizione tipo di acquisizione (obbligatorio)***Marker Detection** Ricerca un marker e ne riporta il centro

- X=0** Circular Marker Detection
- X=1** Square Marker Detection
- X=2** Type1 Marker Detection
- X=3** Hole Marker Detection

Acquisizione da Sensore Utilizza un sensore virtuale e ne riporta i dati di acquisizione

- X=100** Singolo Punto
- X=101** Origini Pezzo
- X=102** Misurazione Utensile

Detector Determina la presenza di oggetti

- X=200** Zona di Grigio
- X=201** Zona di Contrasto
- X=202** Zona di Luminosità
- X=203** Colori
- X=204** Aree

Misure Permette la misurazione di oggetti generici

- X=300** Misura di una sezione
- X=301** Calibro
- X=302** Ricerca Linea tramite punti
- X=303** Ricerca Cerchio tramite punti
- X=304** Misura di una GAP in una sezione

PARAMETRO Y *Definisce come sono i valori di ritorno dei dati riferiti alle quote assi (non obbligatorio default 0)*

- Y=0** I valori sono riportati in **RELATIVO** cioè riferiti alle origini, offset ecc. settati da IsoUs
- Y=1** I valori sono riportati in **ASSOLUTI** cioè riferiti allo zero macchina

I dati restituiti dalla funzione **G102.1** sono riportati nelle variabili per indirizzo “:”.
L’ indirizzo di partenza viene configurato dal **PlugIn UsPxVisionAcq** ed è nel seguente modo:

START_ADDRESS **DATA_INPUT** Dati di input (al momento riservati)
START_ADDRESS+100 **DATA_OUTPUT** Dati restituiti dalla funzione

Es: se viene configurato dal **PlugIn UsPxVisionAcq** l’ indirizzo di partenza **10000**, **DATA_OUTPUT** si trova a **10100**.
Si consiglia di utilizzare INDIRIZZI da 10000 a 15000

G102.1 X0-1-2-3 MARKER DETCTION

G102.1 X0 e G102.1 X2 (Circular, Type1)

Parametri Aggiuntivi - Nessuno

DATA_OUTPUT 1 Acq Ok 0 Acq Error
DATA_OUTPUT +1 Centro X Marker (mm) Riferito al centro della camera
DATA_OUTPUT +2 Centro Y Marker (mm) Riferito al centro della camera
DATA_OUTPUT +3 **Raggio** Marker (mm)
DATA_OUTPUT +4 **Angolo** inclinazione (gradi)

G102.1 X1 (Square)

Parametri Aggiuntivi - Nessuno

DATA_OUTPUT 1 Acq Ok 0 Acq Error
DATA_OUTPUT +1 Centro X Marker (mm) Riferito al centro della camera
DATA_OUTPUT +2 Centro Y Marker (mm) Riferito al centro della camera
DATA_OUTPUT +3 **Larghezza** Marker (mm)
DATA_OUTPUT +4 **Altezza** Marker (mm)
DATA_OUTPUT +5 **Angolo** inclinazione (gradi)

G102.1 X3 (Hole)

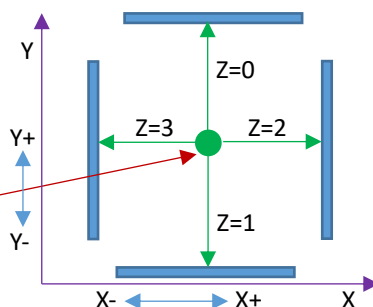
Parametri Aggiuntivi - Nessuno

DATA_OUTPUT 1 Acq Ok 0 Acq Error
DATA_OUTPUT +1 Centro X Marker (mm) Riferito al centro della camera
DATA_OUTPUT +2 Centro Y Marker (mm) Riferito al centro della camera
DATA_OUTPUT +3 **Raggio** Marker (mm)

G102.1 X100-101-102 Acquisizione da sensore

G102.1 X100 Bval Zval (Singolo punto)

Start Probe
Posizione Asse X,Y
+/- Offset parametro B



Parametri Aggiuntivi

B Opzionale Indica Offset in mm +/- nella direzione di acquisizione impostata con Z del Probe (default 10mm)

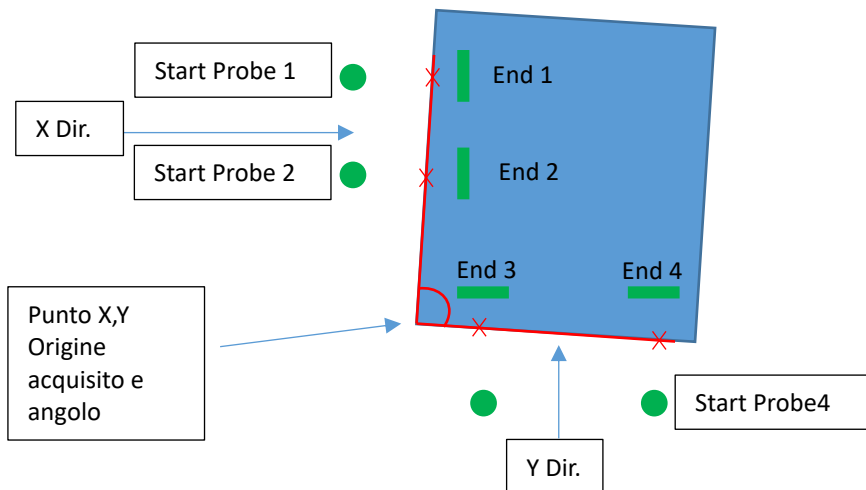
Z Obbligatorio indica la direzione di acquisizione del probe

- Z=0** **Y POSITIVA**
- Z=1** **Y NEGATIVA**
- Z=2** **X POSITIVA**
- Z=3** **X NEGATIVA**

DATA_OUTPUT 1 Acq Ok 0 Acq Error
DATA_OUTPUT +1 X Probe (mm) **Riferito allo zero macchina**
DATA_OUTPUT +2 Y Probe (mm) **Riferito allo zero macchina**

G102.1 X101 Vval (Origini Pezzo)

Per questa funzione occorre configurare uno o più **JOB** da **PxVisionBrowser** con **4 ProbeSinglePoints** che definiscono **2 punti di Start e End in direzione X** e **2 Punti di Start e End in direzione Y**



Parametri Aggiuntivi

V Obbligatorio indica l' indice del JOB (Inseriti nel Plugin UsPxVisionAcq) da usare

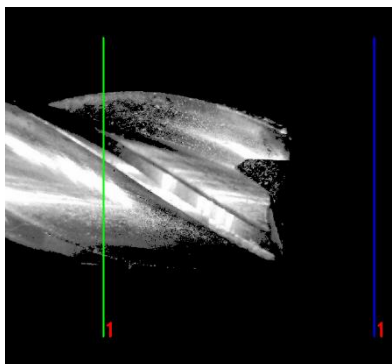
| | | |
|-----------------------|------------------|--------------------|
| DATA_OUTPUT | 1 Acq Ok | 0 Acq Error |
| DATA_OUTPUT +1 | X Origine | |
| DATA_OUTPUT +2 | Y Origine | |
| DATA_OUTPUT +3 | Angolo | |

G102.1 X102 Vval (Misura Utensile)

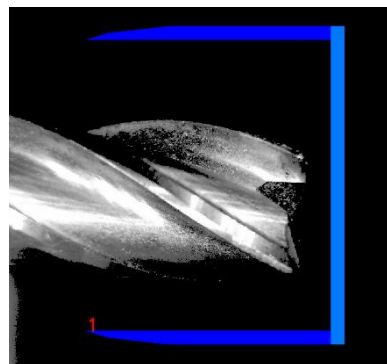
Rilevazione lunghezza e diametro utensile

Per l' utilizzo di queste funzioni è necessario configurare uno o più **JOB** dall' applicazione **PxVisionBrowser** che contiene i parametri per **ProbeLine** misurazione Lunghezza utensile e **Calliper** misurazione Diametro
 Configurare la ProbeLine e il Calliper in modo che possano contenere le dimensioni maggiori dell' utensile

ProbeLine



Calliper



Parametri Aggiuntivi

V Obbligatorio indica l' indice del JOB (Inseriti nel Plugin UsPxVisionAcq) da usare

| | | |
|-----------------------|--------------------------------|--------------------|
| DATA_OUTPUT | 1 Acq Ok | 0 Acq Error |
| DATA_OUTPUT +1 | Lunghezza Utensile (mm) | |
| DATA_OUTPUT +2 | Diametro Utensile (mm) | |

G102.1 X200-201-202-203-204 Detector

Per l' utilizzo di queste funzioni è necessario configurare i Detector dall' applicazione **PxVisionBrowser**

G102.1 X200 G102.1 X201 G102.1 X202 X203 X204 (Gray,Contrast,Brightness,Color,Area)**Parametri Aggiuntivi - Nessuno**

| | | |
|-----------------------|-----------------|---------------------------------|
| DATA_OUTPUT | 1 Acq Ok | 0 Acq Error (detector 1) |
| DATA_OUTPUT +1 | 1 Acq Ok | 0 Acq Error (detector 2) |
| DATA_OUTPUT +2 | 1 Acq Ok | 0 Acq Error (detector 3) |
| .. | | |

Ecc.

G102.1 X300-301-302-303-304 Misure

Per l' utilizzo di queste funzioni è necessario configurare le Misure dall' applicazione **PxVisionBrowser**

G102.1 X300 G102.1 X301 (Sezione,Calibro)**Parametri Aggiuntivi - Nessuno**

| | | |
|-----------------------|-------------------------------|-------------------------------|
| DATA_OUTPUT | 1 Acq Ok | 0 Acq Error (Config 1) |
| DATA_OUTPUT +1 | Misura (mm) (Config 1) | |
| DATA_OUTPUT +2 | 1 Acq Ok | 0 Acq Error (Config 2) |
| DATA_OUTPUT +3 | Misura (mm) (Config 2) | |

..

Ecc.

G102.1 X302 (Ricerca Linea)**Parametri Aggiuntivi - Nessuno**

| | | |
|-----------------------|--|-------------------------------|
| DATA_OUTPUT | 1 Acq Ok | 0 Acq Error (Config 1) |
| DATA_OUTPUT +1 | Lunghezza Linea (mm) (Config 1) | |
| DATA_OUTPUT +2 | Angolo Linea (gradi) (Config 1) | |
| DATA_OUTPUT +3 | 1 Acq Ok | 0 Acq Error (Config 2) |
| DATA_OUTPUT +4 | Lunghezza Linea (mm) (Config 2) | |
| DATA_OUTPUT +5 | Angolo Linea (gradi) (Config 2) | |

..

Ecc.

G102.1 X303 (Ricerca Cerchio)**Parametri Aggiuntivi - Nessuno**

| | | |
|-----------------------|---|-------------------------------|
| DATA_OUTPUT | 1 Acq Ok | 0 Acq Error (Config 1) |
| DATA_OUTPUT +1 | Centro X (mm) Riferito al centro della camera (Config 1) | |
| DATA_OUTPUT +2 | Centro Y (mm) Riferito al centro della camera (Config 1) | |
| DATA_OUTPUT +3 | Raggio (mm) (Config 1) | |
| DATA_OUTPUT +4 | 1 Acq Ok | 0 Acq Error (Config 2) |
| DATA_OUTPUT +5 | Centro X (mm) Riferito al centro della camera (Config 2) | |
| DATA_OUTPUT +6 | Centro Y (mm) Riferito al centro della camera (Config 2) | |
| DATA_OUTPUT +7 | Raggio (mm) (Config 2) | |

..

Ecc.

G102.1 X304 (Gap Sezione)**Parametri Aggiuntivi - Nessuno**

| | | |
|-----------------------|-----------------------------------|-------------------------------|
| DATA_OUTPUT | 1 Acq Ok | 0 Acq Error (Config 1) |
| DATA_OUTPUT +1 | Valore Max (mm) (Config 1) | |
| DATA_OUTPUT +2 | Valore Min (mm) (Config 1) | |
| DATA_OUTPUT +3 | 1 Acq Ok | 0 Acq Error (Config 2) |
| DATA_OUTPUT +4 | Valore Max (mm) (Config 2) | |
| DATA_OUTPUT +5 | Valore Min (mm) (Config 2) | |

Ecc.

G102.1 X400 BarCode Reader

Possono essere letti **BarCode** e **QrCode**, prima di utilizzare questa funzione occorre creare una lista di associazione codici dal PlugIn **UsPxVisionAcq**. In pratica vengono definiti tutti i codici accettati con associazione ad un **INDICE** numerico, quindi il risultato di ritorno non è l' effettivo codice letto, ma l' indice nella lista

G102.1 X400**Parametri Aggiuntivi - Nessuno**

| | | |
|-----------------------|---|-------------------------------|
| DATA_OUTPUT | 1 Acq Ok | 0 Acq Error (Config 1) |
| DATA_OUTPUT +1 | Indice codice nella lista (-1) codice non codificato | |

G102.1 X401 Ocr Reader

Può essere letto una stringa di testo senza ritorni a capo.

Come per i BarCode, anche le stringhe devono essere codificate tramite il PlugIn **UsPxVisionAcq**.

Quindi viene associato un **INDICE** alla stringa

G102.1 X401**Parametri Aggiuntivi - Nessuno**

| | | |
|-----------------------|---|-------------------------------|
| DATA_OUTPUT | 1 Acq Ok | 0 Acq Error (Config 1) |
| DATA_OUTPUT +1 | Indice codice nella lista (-1) codice non codificato | |

17.11 PROGRAMMAZIONE DELLA Velocità INTERPOLAZIONE ASSI

La Velocità di interpolazione assi risulta essere un parametro molto importante al fine della lavorazione del pezzo. La Velocità viene espressa in Mt/min con risoluzione dipendente dal sistema utilizzato e dalla configurazione di Isous. Velocità di interpolazione programmata puo' essere variata in modo automatico dal sistema (autoriduzione F su archi), oppure in modo manuale con il parametro **F** o il potenziometro di override. Mentre il parametro **F** agisce solamente quando nel PartProgram viene invocata tale istruzione e comunque all' inizio del tratto successivo, il potenziometro di OVERRIDE agisce sempre in qualsiasi punto del percorso. Pertanto con questo sistema è possibile variare la Velocità di lavorazione in modo istantaneo.

17.11.1 **F – Velocità interpolazione assi**

Sintassi

Fval

Tipo funzione

MODALE

Parametri

Val **Valore in Mt/min della Velocità**

Descrizione

La funzione **F** imposta la Velocità vettoriale di interpolazione degli assi. Tale Velocità viene poi scomposta dall' interpolatore in base alla traiettoria programmata.

Gli spostamenti effettuati con **G0** non sono influenzati dalla **F** programmata.

Es:

G0 X75.45 Y157.46 // SPOSTA ALL F MAX DISPONIBILE

F10.3 // FEED ASSI 10.3 MT/MIN

G1 X107.84 Y167.43

17.11.2 **Autoriduzione Velocità su ARCHI**

Tipo funzione

Programmata dalla configurazione

Parametri

Vedi configurazione Isous

Descrizione

Isous utilizza uno speciale algoritmo che permette di ridurre in modo automatico la FEED assi durante la lavorazione degli archi. Questo permette di non preoccuparsi di dovere inserire nel PartProgram la **F** in base al raggio dell' arco effettuato (raggio piccolo **F** bassa – raggio grande **F** elevata).

Il calcolo che Isous esegue si basa sull' accelerazione centrifuga degli assi e quindi questo parametro va trovato sulla macchina.

17.12 TRASFORMAZIONI DEL PIANO DI LAVORO

Vengono spiegate tutte le funzioni relative alla manipolazione del piano di lavoro. Queste comprendono rotazione e varie specularità'.

17.12.1 G120 - Specularità verticale del piano

Sintassi

G120

Tipo funzione

MODALE

Revoca

G121

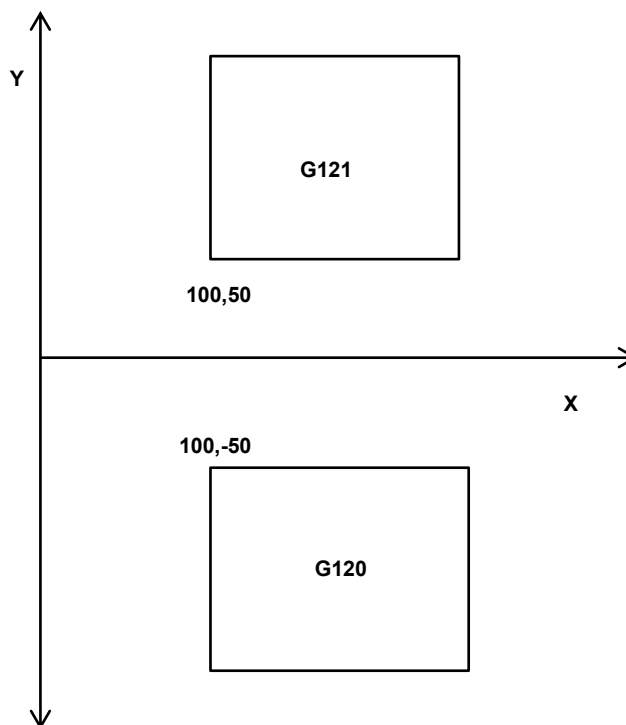
Descrizione

Abilita la specularità verticale del piano di lavoro (inversione delle coordinate Y). Tale funzione abilita una sorta di specchiatura sulla Y invertendo automaticamente il segno delle coordinate di questo asse.

Es:

```
G121 // DISABILITA G120
// ***** INIZIO PRIMO QUADRATO
G0 X100 Y100
G1 X150
Y50
X100
Y100
// ***** FINE PRIMO QUADRATO

G120 // ABILITA Specularità
VERTICALE
// ***** INIZIO SECONDO QUADRATO
G0 X100 Y100
G1 X150
Y50
X100
Y100
// ***** FINE SECONDO QUADRATO
```



17.12.2 G121 - Disabilita Specularità verticale del piano

Sintassi

G121

Tipo funzione

MODALE

Revoca

G120

Descrizione

Disabilita la Specularità verticale del piano di lavoro.

17.12.3 G24 - Specularità orizzontale del piano**Sintassi****G24****Tipo funzione****MODALE****Revoca****G25****Descrizione**

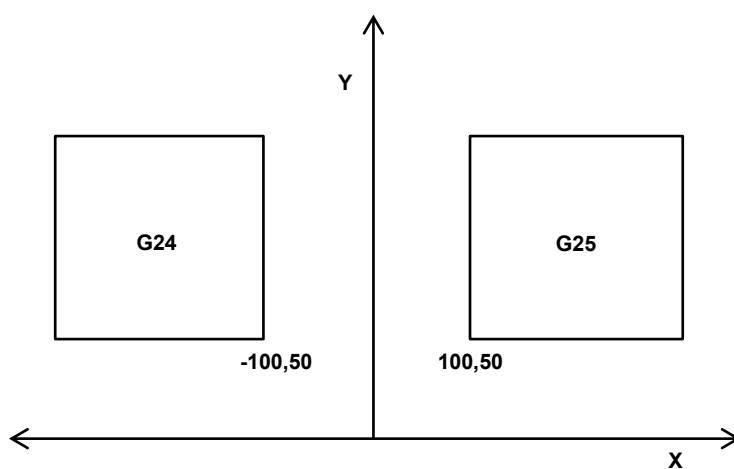
Abilita la speculata' orizzontale del piano di lavoro (inversione delle coordinate X). Tale funzione abilita una sorta di specchiatura sulla X invertendo automaticamente il segno delle coordinate di questo asse.

Es:

```

G25 // DISABILITA G24
// ***** INIZIO PRIMO QUADRATO
G0 X100 Y100
G1 X150
Y50
X100
Y100
// ***** FINE PRIMO QUADRATO
G24// ABILITA SPECULARITA'ORIZZ.
// ***** INIZIO SECONDO QUADRATO
G0 X100 Y100
G1 X150
Y50
X100
Y100
// ***** FINE SECONDO QUADRATO

```

**17.12.4 G25 - Disabilita Specularità orizzontale del piano****Sintassi****G25****Tipo funzione****MODALE****Revoca****G24****Descrizione**

Disabilita la Specularità orizzontale del piano di lavoro.

17.12.5 G22 - Scambia gli assi del piano di interpolazione**Sintassi****G22**

Tipo funzione

MODALE

Revoca

G23

Descrizione

Vengono automaticamente scambiati gli assi del piano di interpolazione (es. X con Y)

17.12.6 G23 - Ripristina gli assi del piano di interpolazione**Sintassi****G23**

Tipo funzione

MODALE

Revoca

G22

Descrizione

Ripristina gli assi originali del piano di interpolazione.

17.12.7 G26 - Scambia coppia di assi a piacere**Sintassi****G26 Asse1 Asse2**

Tipo funzione

MODALE

Revoca

G27

Descrizione

Vengono scambiati una coppia di assi a piacere

Es:

G26 YZ // SCAMBIA Y CON Z**G26 QYQZ // SCAMBIA CANALE 1 CON CANALE 2****17.12.8 G27 – Sospende G26****Sintassi****G27**

Tipo funzione

MODALE

Revoca

G26

Descrizione

Viene sospeso lo scambio assi effettuato con G26

17.12.9 G28 – Riprende G26

(Spostato su G1028 se USE_G80_CYCLES=TRUE)

Sintassi**G28**

Tipo funzione

MODALE

Revoca

*G27***Descrizione**

Viene ripreso lo scambio assi effettuato con G26

17.12.10 G1028 – Ritorno ad Home

(Spostato su G28 se USE_G80_CYCLES=TRUE)

Sintassi**G1028 Xval Yval Zval....****Descrizione**Ritorna al punto di **HOME** indicato dai parametri **G28HOME_X**, **G28HOME_Y** ecc.**Xval, Yval, Zval** ecc. parametri Opzionali.

Se vengono inseriti prima di tornare ai punti di HOME gli assi si spostano a questi valori

La velocità viene presa dalla precedente F o dal precedente G0 impostato

17.12.11 G51 - Sospende rotazione del piano di lavoro**Sintassi****G51**

Tipo funzione

MODALE

Revoca

*G50 G52***Descrizione**

Sospende la rotazione del piano di lavoro G50

17.12.12 G52 - Riprende rotazione del piano di lavoro**Sintassi****G52**

Tipo funzione

MODALE

Revoca

*G51***Descrizione**

Riprende la rotazione del piano di lavoro precedentemente programmata con G50

17.12.13 G50 - Rotazione del piano di lavoro**Sintassi****G50 X**centrox **Y**centroy **Z**angolo (DX,DY valori relativi)**Tipo funzione****DISATTIVATA AL RUN****Parametri**

X **Centro di rotazione asse X**
Y **Centro di rotazione asse Y**
Z **Angolo di rotazione in gradi**

Descrizione

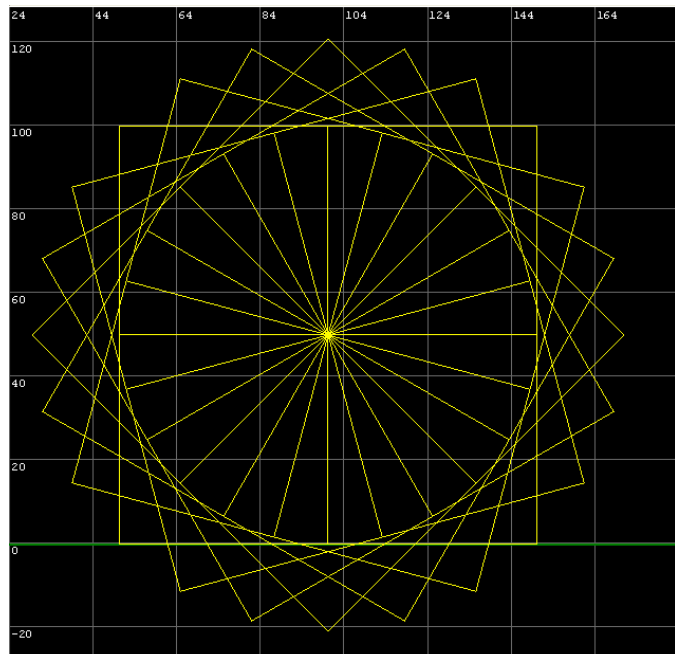
Effettua una rotazione del piano di lavoro. A tutte le nuove coordinate sarà applicata questa trasformazione.

G50 accetta come parametri il centro di rotazione lungo la X,Y e l'angolo in gradi di rotazione.

Ad ogni esecuzione del PartProgram il centro di rotazione viene disabilitato, oppure con angolo 0 gradi

Es:

```
// ROTAZIONE DI UN QUADRATO
$ROT=0
LOOP 24
G0 X100 Y100
G1 X150
Y50
X100
Y100
$ROT=$ROT+15// INCREMENTA ANGOLO
G50 X100 Y50 Z[$ROT]// ROTAZIONE
END_LOOP
```



17.12.14 G1050 – Disabilita Scaling**Sintassi****G1050****Tipo funzione****MODALE****Revoca****G1051****Parametri****nessuno****Descrizione**

G1050 disabilita la funzione Scaling G1051 riportando le condizioni originali

17.12.15 G1051 - Scaling**Sintassi****G1051 Xcentrox Ycentroy Zcentroz IratioX Jratioy Kratioz Pratioxyz****Tipo funzione****MODALE****Revoca****G105****Parametri**

| | |
|----------|--|
| X | Centro di scalatura asse X – se non inserito viene presa la posizione attuale dell' asse X |
| Y | Centro di scalatura asse Y – se non inserito viene presa la posizione attuale dell' asse Y |
| Z | Centro di scalatura asse Z – se non inserito viene presa la posizione attuale dell' asse Z |
| I | Rapporto scalatura Asse X – X<1 riduce X>1 aumenta se non inserito viene preso il valore precedente |
| J | Rapporto scalatura Asse Y – Y<1 riduce Y>1 aumenta se non inserito viene preso il valore precedente |
| K | Rapporto scalatura Asse Z – Z<1 riduce Z>1 aumenta se non inserito viene preso il valore precedente |
| P | Rapporto scalatura Asse XYZ – P<1 riduce P>1 aumenta esclude I,J,K |

SE NESSUN PARAMETRO INSERITO (solo G1051) vengono utilizzati tutti i precedenti valori**Descrizione**

G1051 effettua una scalatura del Gcode basandosi su un centro X,Y,Z

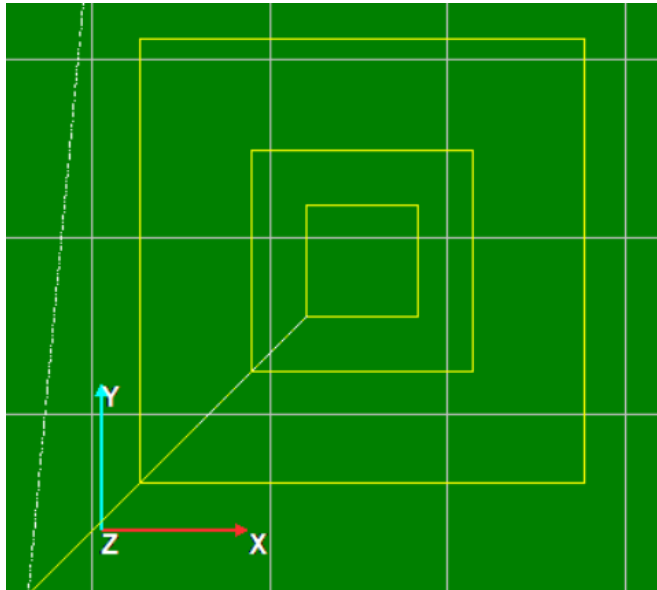
Tramite questa funzione, l' intero Gcode può essere disegnato con grandezze diverse.

Es:

```

//ORIGINAL
G0X0Y0
G1X10Y10
Y20
X20
Y10
X10
G0X0Y0
//SCALING
G1051 X15Y15I0.5J0.5
G1X10Y10
Y20
X20
Y10
X10
G0X0Y0
//SCALING
G1051 X15Y15I2J2
G1X10Y10
Y20
X20
Y10
X10

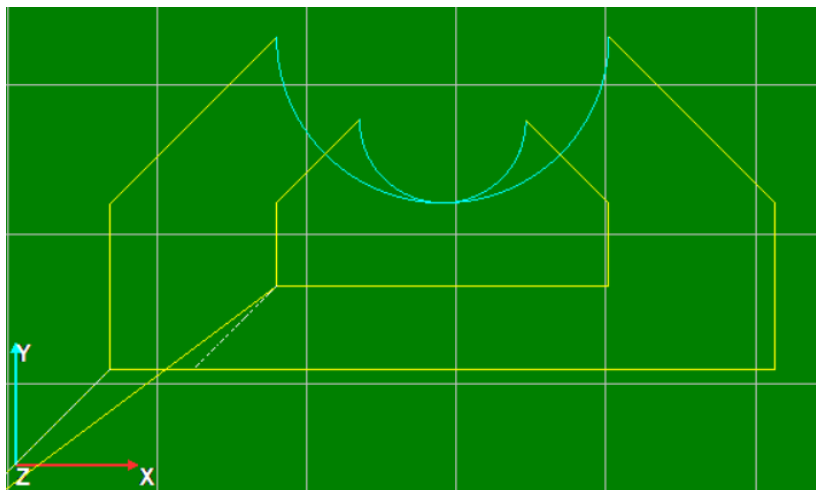
```

**Es:**

```

//ORIGINAL
G0X0Y0
G1X10Y10
Y20
X20Y30
G3X40Y30R10
G1X50Y20
X50Y10
X10Y10
G0X0Y0
//SCALING
G1051 X30Y20I0.5J0.5
G1X10Y10
Y20
X20Y30
G3X40Y30R10
G1X50Y20
X50Y10
X10Y10
G0X0Y0

```



17.12.16 G103 – Parametri RTCP**Sintassi****G103 Xval Yval Zval G103 QXval QYval QZval****Tipo funzione****MODALE****Parametri****X Valore abbinato al parametro X****Y Valore abbinato al parametro Y****Z Valore abbinato al parametro Z**

.....

Descrizione

Isous utilizza un **RTCP** (rotate tool center point) adattabile a qualsiasi tipo di geometria esistente sulla macchina. Pertanto i valore che si passano tramite G103 (max 9 valori X,Y,Z,A,B,C,U,V,W) determinano i giusti parametri per la rotazione del centro utensile. Questo viene deciso in fase di installazione e in base alla geometria (fulcri di rotazione pivot ecc.) presente sulla macchina.

Per questa funzione è necessario l' intervento di un tecnico specializzato.

G103 passa solamente i parametri al CN, ma non abilita la funzione RTCP, questa viene gestita da **G104-G105**.

17.12.17 G104 – Abilitazione RTCP**Sintassi****G104****Tipo funzione****MODALE****Parametri****nessuno****Revoca****G105****Descrizione**

G104 abilita la gestione dell' **RTCP** sul CN in base ai parametri passati con G103 senza alcuna riposizionamento degli assi X,Y,Z

17.12.18 G104.1 – Abilitazione RTCP**Sintassi****G104.1****Tipo funzione****MODALE****Parametri****nessuno****Revoca****G105****Descrizione**

G104.1 abilita la gestione dell' **RTCP** riposizionando X,Y,Z in base alle posizioni degli ASSI ROTATIVI

17.12.19 **G105 – Sospensione RTCP****Sintassi****G105****Tipo funzione****MODALE****Parametri****nessuno****Revoca****G104****Descrizione**

G105 sospende la gestione dell' **RTCP** sul CN riportando la normale interpolazione. La gestione RTCP può essere ripresa con **G104**.

17.13 CICLI FISSI

IsoUs Mette d disposizione 4 cicli Fissi relativi a:
G81,G82,G83,G84 (G1081,G1082,G1083,G1084)

17.13.1 G1080 – Disabilita Cicli Fisso in Corso

(Spostato su G80 se USE_G80_CYCLES=TRUE)

Sintassi

G1080

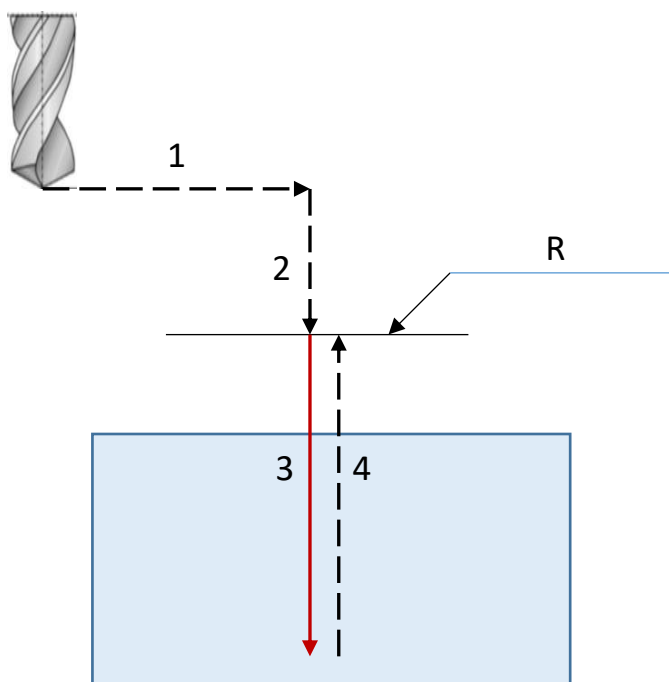
17.13.2 G1081 – Ciclo di Foratura

(Spostato su G81 se USE_G80_CYCLES=TRUE)

Sintassi

G1081 Xval Yval Zval Qval Rval Kval Fval

- X** Posizione in X prima foratura (opzionale) se non inserito prende punto attuale
- Y** Posizione in Y prima foratura (opzionale) se non inserito prende punto attuale
- Z** Profondità di foratura
- Q** Tipo di Ritorno (Opzionale) se **Q=1** ritorna al **punto 2** in **G1** con **F** programmata, altrimenti se diverso da **1** o **non inserito** ritorna nel **punto 2** in **G0**
- R** Quota di disimpegno
- K** Ripetizioni (opzionale, Attiva se siamo in G91 e se inserito parametro X o Y)
- F** Velocità Foratura (Opzionale) se non inserita prende quella attuale



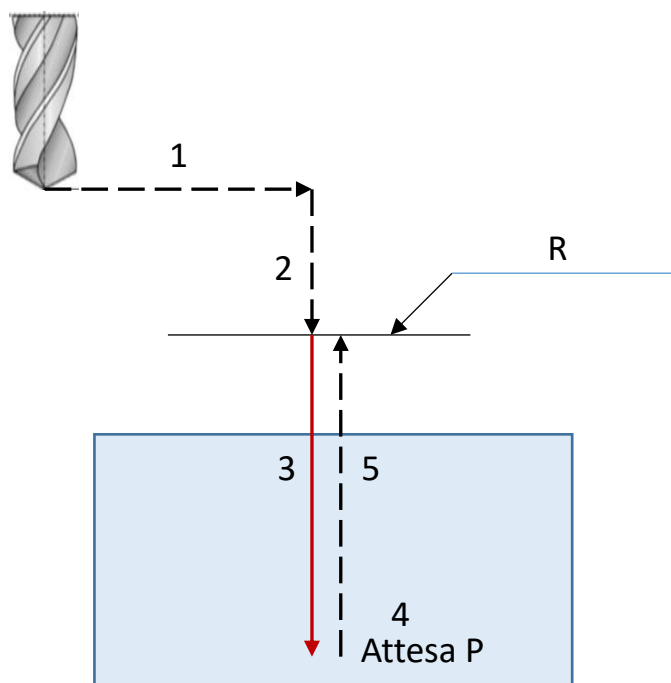
- 1) Spostamento in G0 X,Y
- 2) Spostamento in G0 ad R
- 3) Affondamento in G1 a Z
- 4) Ritorno in G0 ad R se Q diverso da 1 o NON INSERITO
Altrimenti ritorna in G1 alla F programmata

17.13.3 G1082 – Ciclo di Foratura

(Spostato su G82 se USE_G80_CYCLES=TRUE)

Sintassi**G1082 Xval Yval Zval Qval Rval Pval Kval Fval**

- X** Posizione in X prima foratura (opzionale) se non inserito prende punto attuale
- Y** Posizione in Y prima foratura (opzionale) se non inserito prende punto attuale
- Z** Profondità di foratura
- Q** Tipo di Ritorno (Opzionale) se **Q=1** ritorna al **punto 2** in **G1** con **F** programmata, altrimenti se diverso da **1** o **non inserito** ritorna nel **punto 2** in **G0**
- R** Quota di disimpegno
- P** Tempo di attesa fine foratura (Ms)
- K** Ripetizioni (opzionale, Attiva se siamo in G91 e se inserito parametro X o Y)
- F** Velocità Foratura (Opzionale) se non inserita prende quella attuale



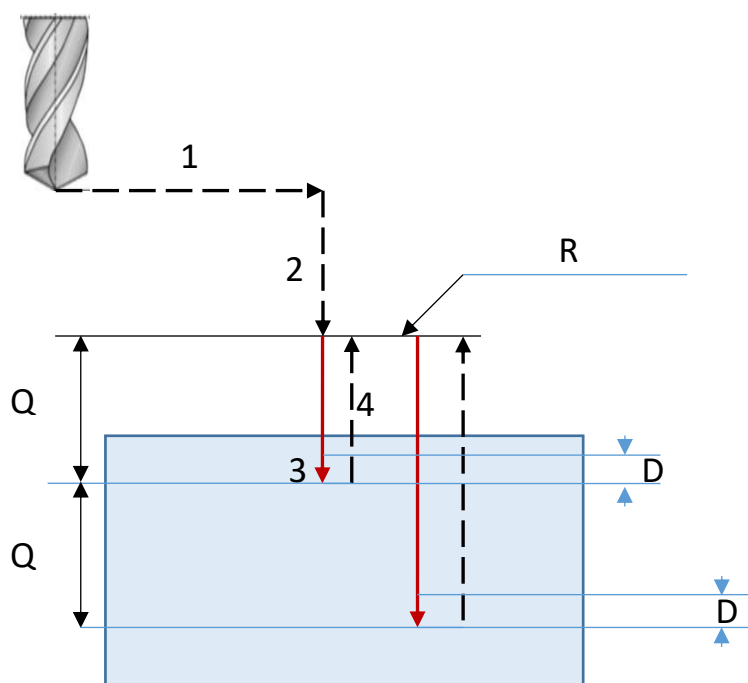
- 1) Spostamento in G0 X,Y
- 2) Spostamento in G0 ad R
- 3) Affondamento in G1 a Z
- 4) Attesa P Millisecondi
- 5) Ritorno in G0 ad R se Q diverso da 1 o NON INSERITO
Altrimenti ritorna in G1 alla F programmata

17.13.4 G1083 – Ciclo di Foratura

(Spostato su G83 se USE_G80_CYCLES=TRUE)

Sintassi**G1083 Xval Yval Zval Rval Qval Kval Fval**

- X** Posizione in X prima foratura (opzionale) se non inserito prende punto attuale
- Y** Posizione in Y prima foratura (opzionale) se non inserito prende punto attuale
- Z** Profondità di foratura
- P** Rottura truciolo (opzionale)
- R** Quota di disimpegno
- Q** Incremento Passata (solo positivo)
- K** Ripetizioni (opzionale, Attiva se siamo in G91 e se inserito parametro X o Y)
- F** Velocità Foratura (Opzionale) se non inserita prende quella attuale



- 1) Spostamento in G0 X,Y
- 2) Spostamento in G0 ad R
- 3) Affondamento in G1 a Q
- 4) Ritorno in G0 ad R
- 5) Affondamento in G0 ultima foratura + Parametro **DELTA_G83** (D)
- 6) Ripetizione fino a Fine ciclo quota Z

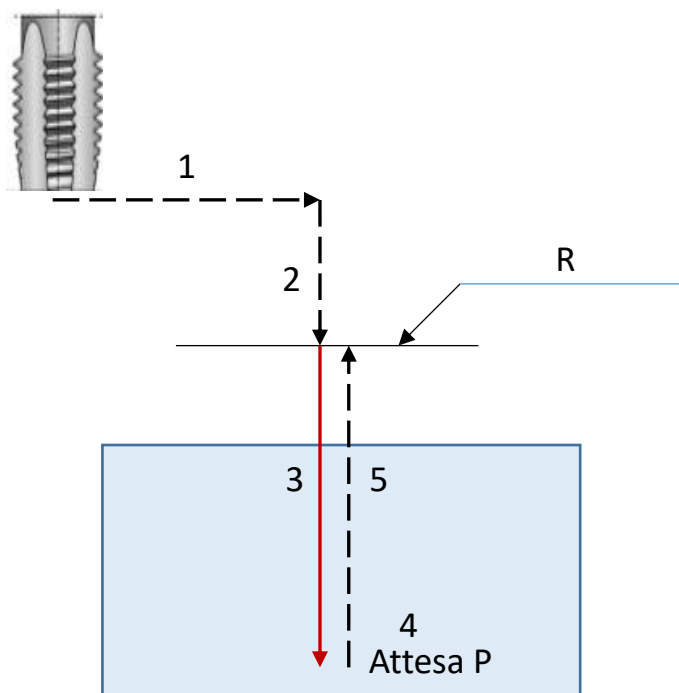
Se inserito parametro P (Rompi truciolo) ad ogni affondamento Q invece di ritornare ad R ritorna indietro del valore inserito in P

17.13.5 G1084 – Ciclo di Maschiatura

(Spostato su G84 se USE_G80_CYCLES=TRUE)

Sintassi**G1084 Xval Yval Zval Rval Pval Kval Qval Fval Aval Bval**

- X** Posizione in X prima foratura (opzionale) se non inserito prende punto attuale
- Y** Posizione in Y prima foratura (opzionale) se non inserito prende punto attuale
- Z** Profondità di Maschiatura
- R** Quota di disimpegno
- P** Tempo di attesa fine filettatura (Ms)
- K** Ripetizioni (opzionale, Attiva se siamo in G91 e se inserito parametro X o Y)
- Q** Passo della Maschiatura (mm) valido solo se siamo in INTERPOLAZIONE (**MODE_G84=TRUE**)
- F** Velocità Maschiatura (Opzionale) se non inserita prende quella attuale
- A** Senso di rotazione Maschiatura (Opzionale) se **A=1** viene invertito senso di rotazione
- B** Moltiplicatore Feed Ritorno (Opzionale) Moltiplica la FEED di ritorno (punto 5)
Es: B=2 la FEED (**F**) viene raddoppiata
Solo valori > 0



- 1) Spostamento in G0 X,Y
- 2) Spostamento in G0 ad R
- 3) Affondamento in G1 a Z interpolato con asse rotativo A (impostare la S correttamente alla F. Parametro ROT_G84 per definizione Senso)
- 4) Attesa P Millisecondi
- 5) Ritorno in G1 ad R

17.14 COMPENSAZIONE OFFSET UTENSILE

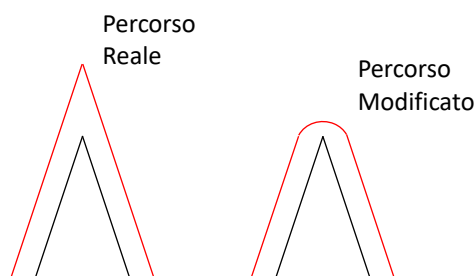
Isous dispone di una compensazione offset utensile attivabile con le funzioni G41 G42 iso. La compensazione prevede lo scostamento dell' utensile sul percorso in base al proprio diametro.

è normale che in alcuni casi possano verificarsi problemi di INTERFERENZA SUL PERCORSO UTENSILE. Questi sono dovuti al diametro utensile elevato.

Isous nella sua interfaccia mette a disposizione un anteprima grafica che segnala i punti di interferenza in modo da anticipare la correzione di questi prima della lavorazione.

La compensazione utensile prevede comunque una particolare attenzione sulla programmazione del percorso.

La compensazione può inserire in modo automatico degli archi (con raggio uguale a raggio utensile) di raccordo. Questi vengono inseriti quando l' intersezione dei tratti si trova particolarmente distante dal punto reale.



17.14.1 G41/G42 – Compensazione offset utensile Sinistra/Destra

Sintassi

G41

G42

Tipo funzione

MODALE

Revoca

G40

Descrizione

G41 Abilita la correzione utensile sinistra.

G42 Abilita la correzione utensile destra.

Il raggio dell' utensile viene impostato con la funzione **D** oppure con la selezione della testa **Tn**.

17.14.2 G40 – Disabilita Compensazione offset utensile

Sintassi

G40

Tipo funzione

MODALE

Revoca

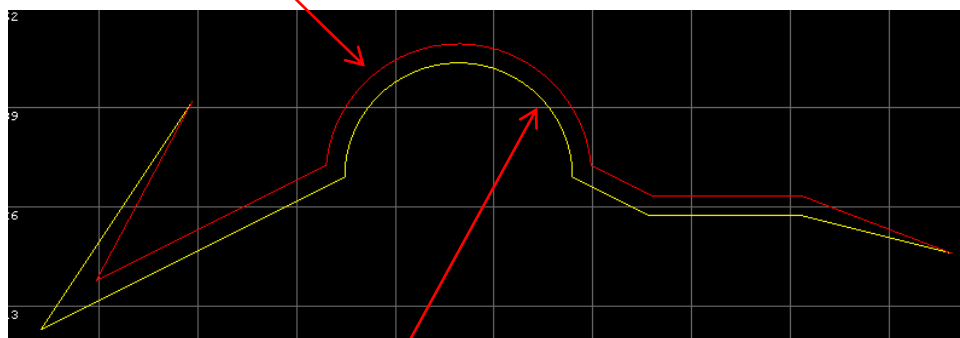
G41 - G42

Descrizione

G40 Disabilita la correzione utensile sia destra che sinistra.

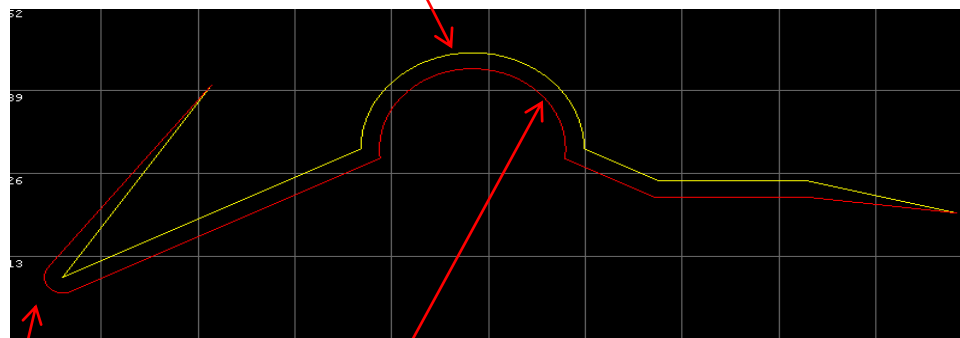
PERCORSO CON G41

Es:
 D5//COR.5 MM
 G41//COMP SINISTRA
 G0X30Y40
 G1X10Y10
 G1X50Y30
 G2X80Y30R15
 G1X90Y25
 G1X110
 G1X130Y20
 G40



PERCORSO CON G40

Es:
 D5//COR.5 MM
 G42//COMP DESTRA
 G0X30Y40
 G1X10Y10
 G1X50Y30
 G2X80Y30R15
 G1X90Y25
 G1X110
 G1X130Y20
 G40



PERCORSO CON G42

**ARCO
 INSERITO**

17.14.3 D - Correttore utensile

Sintassi

Dvalore

Tipo funzione

MODALE

Revoca

Tn (eventuale correzione impostata da Tabella Testa)

Parametri

Valore Diametro utensile in mm

Descrizione

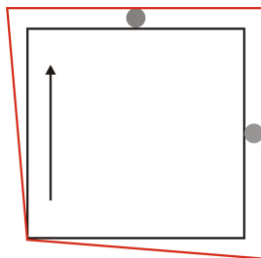
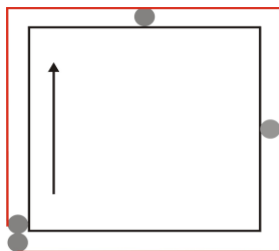
D permette di inserire il correttore utensile per la compensazione G41 G42.

Il suo valore è espresso in mm.

Il correttore D deve essere impostato prima di G41-G42 e non deve essere cambiato durante la fase di compensazione utensile.

17.14.4 G47 – Disabilita/Abilita impegno utensile**Sintassi****G47 Xval****Tipo funzione****MODALE****Descrizione**

G47 abilita/disabilita impegno/disimpegno utensile con tratto di entrata utensile

Di default **G47 X0** (impegno con tratto)Per **G47 X1** impegno utensile disabilitato è necessario che la figura sia chiusaEs lavorazione figura con impegno utensile **G47 X0**Es lavorazione figura con impegno utensile **G47 X1****NOTE:**

Con G47=1 è necessario che l' utensile sia portato dal partprogram su punto di inizio (quindi già offsetato del raggio utensile) mentre la fine della compensazione deve seguire il seguente schema:

G0 Z (ALTO) // alza Z**G40 //disabilita****G1 Z (ALTO)**

Esempio di compensazione di un quadrato:

```

G47 X1
F 2.5
G0 X 26.2332 Y 26.1708 // porta l' utensile sui punti di inizio con offset
F 5
G1 Z 0 // abbassa Z
D 3 // diametro utensile
G42 // abilita compensazione G42
G1 X 24.7332 Y 26.1708
G1 X 24.7332 Y 76.1708
G1 X 74.7332 Y 76.1708
G1 X 74.7332 Y 26.1708
G1 X 24.7332 Y 26.1708
G0 Z 20 // alza Z
G40 // disabilita compensazione
G1 Z 20 //alza Z (conferma)

```

17.15 COMPENSAZIONE LUNGHEZZA UTENSILE

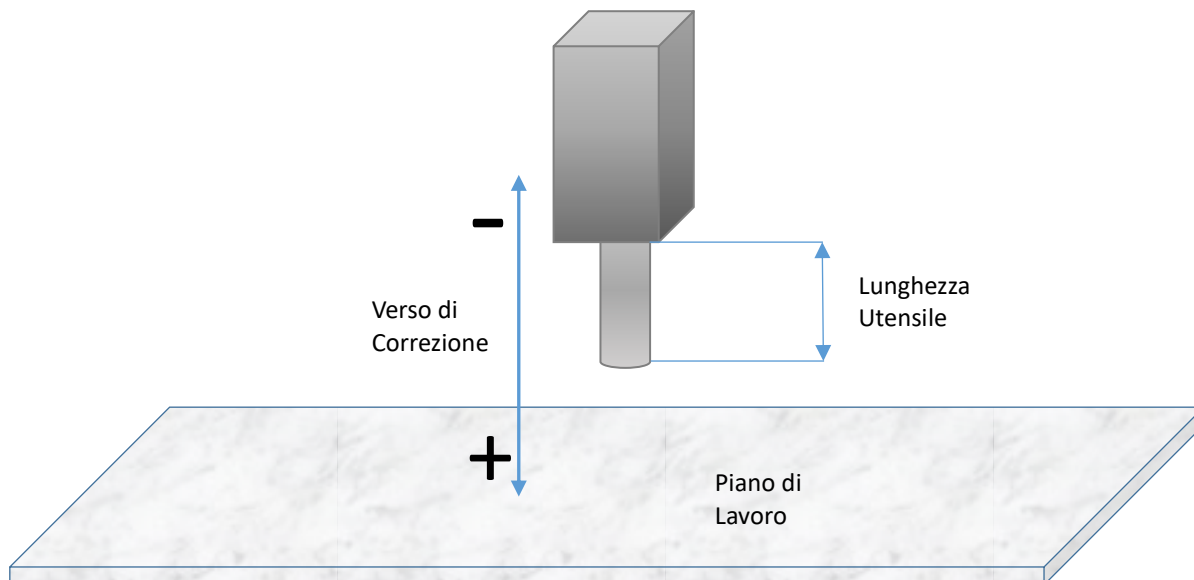
è possibile effettuare la compensazione della lunghezza utensile su qualsiasi asse e direzione.

La compensazione in lunghezza è utile in quei casi dove si voglia recuperare l' offset dovuto alla lunghezza utensile. La compensazione in lunghezza agisce solamente nel verso indicato dalla funzione G43.

L' asse su cui operare è selezionabile come la lunghezza di correzione.

La correzione in lunghezza fa in modo di compensare gli spostamenti sull' asse lasciando trasparente all' operatore la differenza di lunghezza utensile diversa.

La lunghezza effettiva dell' utensile deve essere quella che sporge realmente dal mandrino.



17.15.1 G43 – Attiva correzione lunghezza utensile da parametro

Sintassi

G43 Xlen Kmodo ASSEsenso(+/-) (DX valore relativo)

Tipo funzione

MODALE

Revoca

G45 (eventuale correzione impostata da Tabella Testa)

Parametri

Len lunghezza utensile in mm es: 12.35 (anche da espressione o variabile)

Se **LEN=0** viene presa la lunghezza utensile impostata dalla tabella Tn

Modo

Parametro **K** opzionale può avere i seguenti valori:

K0 Modo come se **K** fosse omesso

K1 Con questo parametro la lunghezza utensile viene considerata come un **OFFSET** vero e proprio, pertanto vengono aggiornate anche le quote assi **RELATIVE** e **ASSOLUTE**. In questo caso non viene più considerato il parametro **"SENSO"** in quanto questo è definito dal segno di **"LEN"**

K2 Come **K1** ma inverte il segno alla lunghezza in tabella utensile

ASSE

Asse sul quale va applicata la correzione in lunghezza

Senso

Senso di correzione

+ quote positive

- quote negative

Descrizione

G43 attiva la correzione utensile in lunghezza. La correzione viene abilitata solamente nel verso e sull' asse indicato

Es:

G43 X10.3 Z+ //ATTIVA CORREZIONE SU ASSE Z SENSO POSITIVO LEN=10.3 MM

17.15.2 G44 – disattiva correzione lunghezza utensile da parametro G43**Sintassi****G44**

Tipo funzione

MODALE

Revoca

G43

Descrizione

G44 disattiva la correzione utensile impostata con G43

17.15.3 G44.1(2) – Sospende/Riprende G43**Sintassi****G44.1 // sospende correzione G43****G44.2 // riprende correzione G43**

Tipo funzione

MODALE

Revoca

G43

Descrizione

G44.1 Sospende la correzione utensile impostata con G43

G44.2 Riprende la correzione utensile impostata con G43 con gli ultimi parametri immessi

17.15.4 G45 – Attiva correzione zero utensile da tabella T**Sintassi****G45 ASSEsenso(+/-)**

Tipo funzione

MODALE

Revoca

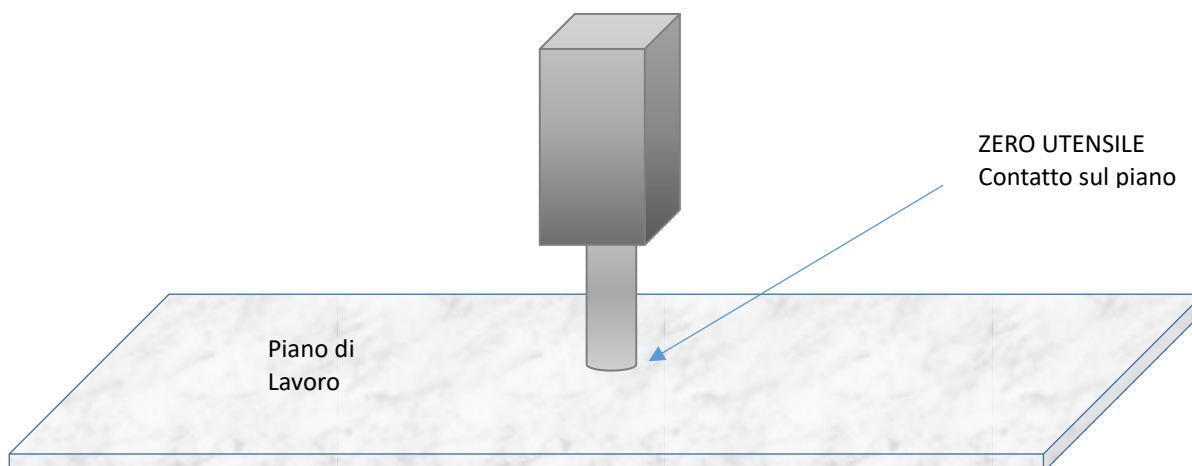
G43 (eventuale correzione impostata da parametro)

Parametri

ASSE Asse sul quale va applicata la correzione in lunghezza**Senso** Senso di correzione**+ quote positive****- quote negative****Descrizione**

G45 è una funzione simile a G43 ma che agisce in modo diverso. In pratica questa riporta lo ZERO UTENSILE sull' asse indicato rispettivamente sullo **ZERO DEL PIANO**. Praticamente fa in modo che con una quota = 0 l' utensile sfiori il piano di lavoro. In pratica per andare a sfiorare il pezzo occorre mettere come valore sull' asse l' altezza del pezzo con segno opposto al verso di correzione. La lunghezza dell' utensile viene presa dalla tabella **Tn** precedentemente selezionata.

Es:

G45 Z+ //ATTIVA CORREZIONE SU ASSE Z SENSO POSITIVO

17.15.5 ***G46 – disattiva correzione lunghezza utensile tabella G45***

Sintassi

G46

Tipo funzione

MODALE

Revoca

G45

Descrizione

G46 disattiva la correzione utensile impostata con G45

17.16 MODI DI INTERPOLAZIONE

Isous può utilizzare diversi metodi di interpolazione assi. Questi possono adattare la lavorazione del pezzo a diverse tipologie.

17.16.1 G60 – Interpolazione veloce senza fermata sul tratto

Sintassi

G60

Tipo funzione

MODALE

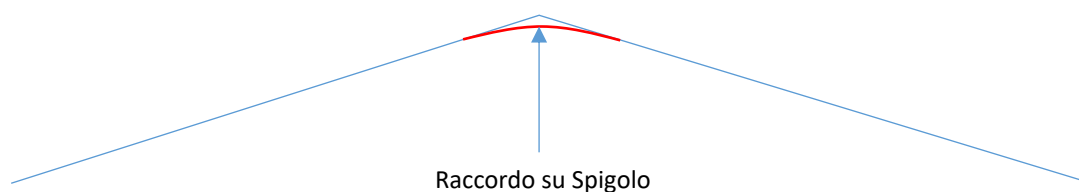
Revoca

G61

Descrizione

G60 abilita il tipo di interpolazione senza fermata sul tratto. La soglia dell'angolo sotto al quale l'asse non si ferma viene impostata da parametri macchina.

Nei punti dove l'asse non si ferma, viene generata una traiettoria ottimale che raccorda i due tratti con il minimo errore possibile. **G60** sfrutta un buffer di movimento all'interno del CN che può essere impostato ad un valore di **1024** su scheda NG35. Questo permette di lavorare in modo fluido percorsi formati da microrette. La traiettoria di fermata viene calcolata solo sugli assi del piano di lavoro



17.16.2 G61 – Interpolazione con fermata sul tratto

Sintassi

G61

Tipo funzione

MODALE

Revoca

G60

Descrizione

G61 abilita il tipo di interpolazione con fermata sul tratto. Questo fa in modo che ad ogni tratto eseguito, l'asse si fermi sul punto e riparta poi con il tratto successivo

17.16.3 G62 – Attesa assi fermi

Sintassi

G62

Tipo funzione

MODALE

Descrizione

G62 Attende che gli assi siano fermi e il buffer dei movimenti vuoto. Questo risulta utile quando il CN si trova in interpolazione veloce e abbiamo bisogno di attendere che tutti i movimenti siano finiti.

Ovviamente non ha senso quando il CN si trova in G61 (fermata sul tratto)

17.16.4 G63 – Interpolazione fuori dal piano di lavoro**Sintassi****G63****Tipo funzione****MODALE****Revoca****G64-G65****Descrizione**

G63 abilita il tipo di interpolazione fuori dal piano di lavoro. In questo modo, è possibile lavorare con traiettorie nello spazio indipendenti da quelle degli assi del piano di lavoro.

La fermata sul tratto viene decisa da G62. Particolare utilità di G63 è data nelle lavorazioni di pezzi 3D, dove la movimentazione comprende tre o più assi. Generalmente queste traiettorie sono definite da un programma CAM che inserisce automaticamente la fermata con G62 sui tratti dove richiesta.

17.16.5 G64 – Interpolazione sul piano di lavoro (default)**Sintassi****G64****Tipo funzione****MODALE****Revoca****G63-G65****Descrizione**

G64 abilita il tipo di interpolazione che considera il piano di lavoro. Questa è l' interpolazione di default selezionata sul CN. **G64** calcola le Velocità vettoriale solo sugli assi del piano impostato, spostando di conseguenza gli altri assi.

Isous verifica in modo automatico se ci sono o meno spostamenti sul piano, In casi contrario utilizza G63.

Possono verificarsi casi quali gli assi del piano abbiano movimenti ridotti rispetto agli assi fuori dal piano di lavoro. Questa è da considerarsi una situazione critica e pertanto in questi casi occorre utilizzare G63 in modo da evitare colpi fastidiosi sugli assi fuori dal piano.

DEVE ESSERE ABILITATA G68**Es:****G64****G17****G1X0.001Y0.001Z10**

Nell' esempio sopra riportato, gli assi X e Y del piano di lavoro si spostano di una percentuale irrilevante rispetto all' asse Z. Questo provoca una Velocità elevata su Z che può risultare dannosa per la macchina.

Tale situazione può essere corretta nel seguente modo:

G63**G17****G1X0.001Y0.001Z10****G64**

Attivando l' interpolazione fuori dal piano, Isous calcola la Velocità vettoriale su tutti gli ASSI scomponendo i giusti vettori per ogni asse.

17.16.6 G65 –Abilita interpolazione 3D – Px_moveto con fermata decisa dal CN**Sintassi****G65****Tipo funzione****MODALE****Revoca****G63-G64****Descrizione**

G65 Abilita l' interpolazione nello spazio con fermata decisa dal CN in base a parametri.

NON DEVE ESSERE ABILITATO G63-G64

Es:

G65

G1X100Y100Z10

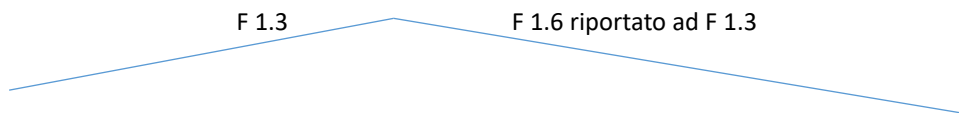
Nell' esempio viene utilizzata Px_Moveto con fermata decisa da CN

17.16.7 G75 –Abilita interpolazione 2D sul piano e 3D fuori dal piano – Px_moveto con fermata decisa dal CN**Sintassi****G75****Tipo funzione****MODALE****Revoca****G63-G64-G65****Descrizione**

G75 Per movimenti nel piano di lavoro viene utilizzata l' interpolazione tipo **G64** con soglia spigolo **SGLP**, per movimenti fuori dal piano di lavoro (o entrambi) viene utilizzata l' interpolazione tipo **G65** con soglia spigolo **SGL3D**

17.16.8 G66 – AFC – Adaptive Feed Control**Sintassi****G66 Xval Yvin Zval Aval****Tipo funzione***Disattivata al RUN***Parametri**

- Xval** **0 - AFC disabilitato 1-AFC Abilitato**
se maggiore di 1, viene effettuata una media mobile sul valore della F calcolato e il numero di campioni indicati in Xval. Questo rende più "morbida" la variazione della F calcolata.
- Yvin** *(opzionale) indica l'inerzia dell'asse*
Valori > 1 aumentano lo spazio di frenatura
Se non viene inserito viene considerato Vin=1 (nessuna inerzia di frenatura)
- Zval** *(opzionale) Indica i valori di Delta F al di sotto del quale, non vengono considerati. Es:*
Val=0.3, una delta F d 0.3 o minore non viene considerato e quindi la F viene lasciata quella precedentemente calcolata.



- Aval** **Valore minimo sotto al quale non viene diminuita la F**
Es: se Val=0.1 valori inferiori a 0.1 di F vengono limitato a 0.1

ATTENZIONE*Per il funzionamento di G66 occorre attivare G69 (profondità look ahead)***Descrizione**

G66 Abilita il filtro **AFC** sul calcolo automatico della velocità di interpolazione. Questo permette di adattare la **FEED** assi in base alla dinamica della macchina. In base ai parametri **SGLAFC_n** l' algoritmo **AFC** calcola la velocità del tratto nel modo ottimale. L' analisi della FEED viene effettuata sul buffer **look ahead** attivato con G69.. Ovviamente più profondo è il filtro, più accurata è la definizione. L' algoritmo **AFC** lavora solamente su tratti di retta **G1**, ed ha la sua reale valenza solamente quando si usano interpolazioni di **MICRO RETTE**. E' fondamentale capire che **AFC** può variare la **FEED** degli assi sui singoli tratti e quindi non otteniamo una velocità costante sul percorso. Il parametro **SGLAFC_** di ogni singolo asse definisce la variazione istantanea di velocità che l' asse può sopportare in modo corretto. Se nel tratto analizzato, anche un solo asse supera queste soglie, la **FEED** viene ridotta in modo da rientrare nei giusti parametri. Nel caso in cui si analizzino tratti che formano degli spigoli (definiti dal parametro **SGL3D_n**, gli assi vengono rallentati fino a fermarsi. **AFC** è utile per le lavorazioni dove non è necessario avere una velocità costante sul percorso. In questo caso **AFC** permette di andare veloce dove è possibile e rallentare invece nei punti critici (curve).

17.16.9 G66 X-100 – NEW AFC – Adaptive Feed Control**Sintassi****G66 X-100 Yarc Zaccs Afmin Bflevel Cval UaxisIndex VminLen WangDiff FnumSeg SradiusMax****Tipo funzione***Disattivata al RUN***Parametri**

| | |
|-------------------|---|
| X-100 | NAFC - Nuovo AFC abilitato 0 disabilitato -101 Disabilitato in PREVIEW Utilizzare X-101 per abilitare il nuovo AFC ma tenerlo disabilitato in fase di PREVIEW Questo poichè in alcuni casi può rallentare il PREVIEW |
| Ypeaks | <i>(opzionale)</i> Valido solo se parametro C bit 8=1 (nuovo algoritmo) indica il valore di riduzione della velocità sui i picchi (vedi sotto) Valore di default=0 |
| Zaccs | <i>(opzionale)</i> Inserisce un filtro per effettuare uno SMOOTHING sui picchi di ACCELERAZIONE Valori compresi tra 0 e 1 (es Z0.7) con valori più alti il filtro risulta più debole. Valore di default=0 |
| Afmin | <i>(opzionale)</i> Valore minimo sotto al quale non viene diminuita la F. Valore di disable Es: se Val=0.1 valori inferiori a 0.1 di F vengono limitato a 0.1 |
| BFastMode | <i>(opzionale)</i> Modo Fast per il calcolo Accelerazioni FastMode=0 Le accelerazioni vengono controllate anche sui segmenti di Find Arc FastMode=1 Le accelerazioni non sono controllare sui segmenti di FindArc Default FastMode=1 |
| Cval | <i>(opzionale)</i> Special Functions vedi descrizione. Valore di default=0 Il parametro C può essere inserito anche a BIT C<3,4> set bit 3 e 4 ed equivale a C24 |
| UAxisIndex | <i>(opzionale)</i> Nel caso si utilizzi assi tangenziali (lame, cutter) indicare l' indice dell' asse |
| VminLen | <i>(opzionale)</i> Indica la minima lunghezza del tratto in mm per essere considerato appartenente ad un ARCO. Segmenti di G1 maggiori di questa lunghezza non vengono considerati come appartenenti ad un arco. Utilizzato solo quando attiva la funzione FindArc parametro C bit 4=1. Valore di default=10 mm |
| WRadiusTol | <i>(opzionale)</i> Indica la tolleranza dei Raggi per l' algoritmo FindArc. I segmenti G1 che rientrano in questa tolleranza vengono considerati appartenenti allo stesso Arco. Valori più alti danno risultati meno precisi. Valore di default 5 (5 % di tolleranza) |
| FnumSeg | <i>(opzionale)</i> Indica il numero di segmenti G1 minimo continui che rispetta i parametri V e W per essere considerati appartenente ad un ARCO. Utilizzato solo quando attiva la funzione FindArc parametro C bit 4=1. Valore di default=3 segmenti G1 |
| SradiusMax | <i>(opzionale)</i> Indica il raggio limite massimo calcolato con i segmenti G1 dalla funzione FindArc. Raggi superiori a questo valore non vengono considerati. Utilizzato solo quando attiva la funzione FindArc parametro C bit 4=1. Valore di default=100 mm |

Descrizione**G66 X-100** abilita un nuovo calcolo della velocità sui file Gcode **NAFC**.Questo nuovo algoritmo è molto più preciso del normale **G66** ed utilizza i seguenti parametri:**ACC_MAX_X,Y,Z** ecc. Il parametro **ACC_MAX_** deve Essere impostato ad un valore maggiore o uguale ad **ACC_LAV**. In

pratica questo indica l'accelerazione massima che ogni asse può supportare, superata la quale, la velocità viene ridotta. **G66 X-100** opera con una profondità di look ahead pari a quella del CNC, pertanto non è necessario impostare la funzione **G69**. Il nuovo algoritmo di **G66 X-100** è molto più raffinato e reale del normale **G66**.

Normalmente non è necessario inserire nessun parametro opzionale per **G66 X-100**, in quanto basta impostare le varie **ACC_MAX_** a valori corretti, ma nel caso in cui si voglia raffinare meglio il calcolo, il parametro **Z** e **B** possono essere inseriti. **Z** con valori decimali compresi tra **0** (disattivato) e **5** (filtro minimo) effettua uno smoothing sulla variazione dell'accelerazione, eliminando i picchi. Un valore di riferimento può essere **Z0.5**.

Tramite questo si evitano riduzioni di velocità troppo aggressive.

B con valori interi compresi tra **0** (disattivato) e **5** (massimo effetto) raffina il calcolo della velocità.

Nel caso in cui si voglia escludere dall'algoritmo le **G2 G3**, impostare il parametro **Y** ad **1**.

La riduzione della velocità sulle **G2-G3** utilizzerà il parametro **ACC_RAGGIO_MAX**.

Impostare quest'ultimo ad un valore elevato (es. 90) se si includono le **G2-G3** nell'algoritmo.

Nel caso di assi tangenziali inserire nel parametro **U**, l'indice dell'asse tangenziale in uso (0-1-2 ecc.)

Il controllo di velocità sull'asse tangenziale viene effettuato quando il sistema interpola sul piano, portando l'eventuale terzo asse in posizione in tempo. Ad esempio, se si lavora con l'asse trasportato (**G68**), oppure sugli archi (**G2-G3**), nei quali la velocità è sempre e solo calcolata sugli assi del piano di interpolazione.

In questi casi, su ogni tratto verrà confrontata la velocità raggiunta dall'asse tangenziale con la massima inserita nei parametri (**VMAX_**). Nel caso che questa venga superata, verrà ricalcolata una nuova velocità di interpolazione, in modo da far muovere l'asse tangenziale alla sua velocità massima. In questo modo si cerca quindi di ottenere la massima velocità di percorrenza del tratto stesso, ma supportabile da tutti gli assi.

Abilitando il bit 5, questo controllo viene escluso.

FindArc

Abilitando il bit 4 del parametro C (valore decimale C=16), viene inserita anche l'algoritmo di ricerca archi da segmenti G1 consecutivi. L'algoritmo controlla che una serie di segmenti G1 consecutivi, soddisfi i parametri **V W F S**.

In caso positivo viene calcolato il raggio di curvatura e viene applicata la riduzione di velocità tramite parametro **ACC_RAGGIO_MAX**, come fosse un elemento G2-G3.

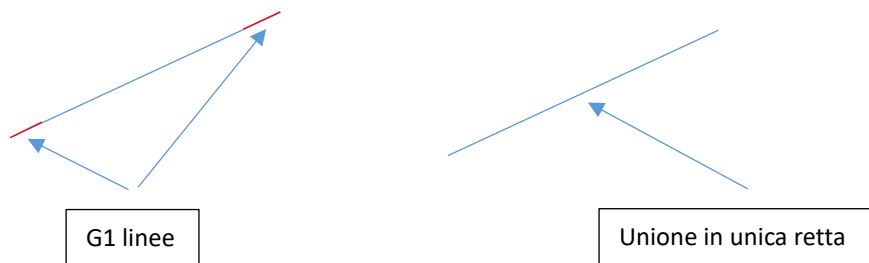


Questa funzione può operare insieme ai parametri di **ACC_MAX_** ma ha la precedenza sui valori calcolati dall'algoritmo **NAFC**

Parametro C Funzioni Speciali Mappato a Bit.

Bit 0=1 RLS Rimuove i tratti G1 che non possono essere lavorati con la velocità programmata

Bit 1=1 JOING1 Unisce i tratti in G1 appartenenti alla stessa retta



Bit 2=1 Esclude algoritmo di riduzione velocità

Solo RLS e JOING1

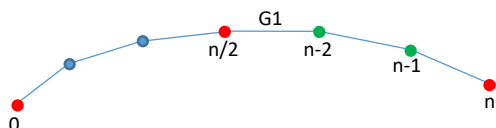
Bit 3=1 Esclude algoritmo NAFC. Calcolo feed in base ai parametri ACC_MAX_

Bit 4=1 Abilita algoritmo FINDARC

Bit 5=1 Esclude controllo velocità su asse tangenziale

Bit 6=1 Usa Primo Punto, Punto Mediano e ultimo punto per FindArc (0,n/2,n)

Bit 6=0 Usa ULTIMI TRE punti per FindArc (n-2,n-2,n)



Bit 7=1 G2-G3 al di fuori di ACC_RAGGIO_MAX vengono presi per FindArc come G1

Bit 8=1 Abilita i nuovi algoritmi per G66 X-100 (consigliata) in questo caso alcuni parametri assumono i seguenti significati

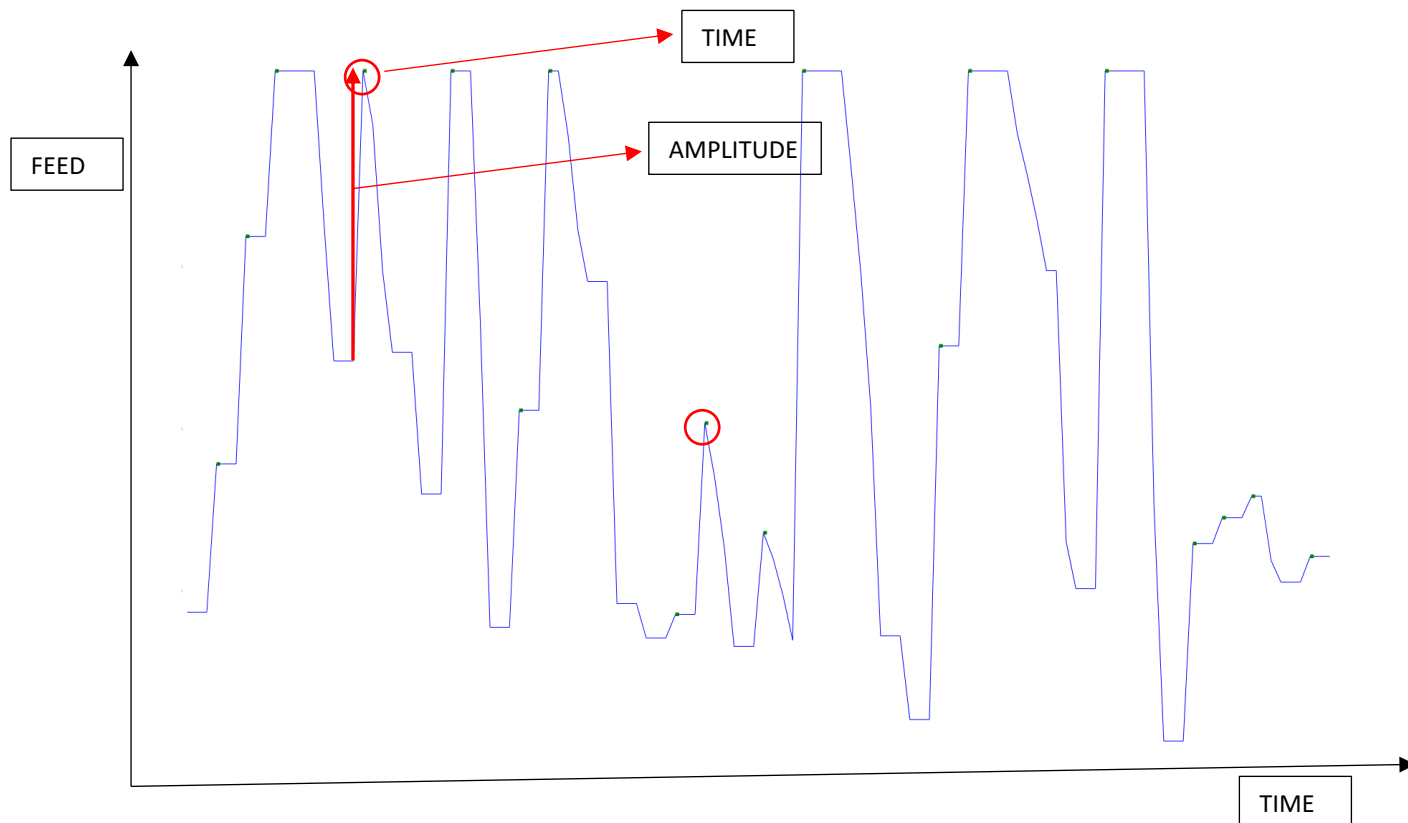
ZFdiv (opzionale) Fattore di divisione per riduzione FEED sui picchi di accelerazione. La velocità calcolata dall' algoritmo viene ulteriormente diviso per questo valore
Default 0 (disabilitato)

FPamp (opzionale) Valore di amplitudine per definire un picco di velocità (vedi sotto)

BPtime (opzionale) Valore di tempo per definire un picco di velocità (vedi sotto)

Riduzione dei picchi di velocità

Gli algoritmi che calcolano la riduzione delle velocità del percorso in modo automatico, possono causare dei picchi di velocità inutili. Cioè spostamenti alla massima velocità per un brevissimo tempo



Questi possono essere identificati e ridotti in modo di avere una velocità più omogenea.

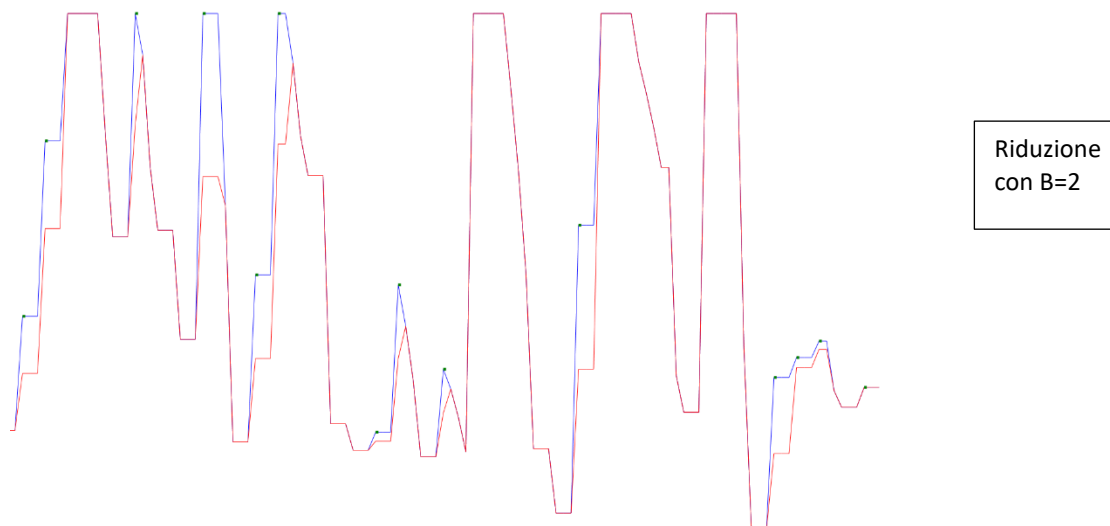
Il parametro **B** abilita la riduzione dei picchi di velocità indicando il tempo della durata del picco

B=1 vengono presi tutti i picchi di velocità e ridotti. **B=4** vengono considerati picchi solamente quelli che hanno un tempo ≤ 4 unità (default 0 disabilitato)

Il parametro **F** indica l' amplitudine (mt/min) del picco per essere considerato tale.

F=0 vengono presi tutti i picchi di velocità e ridotti. **F=8.1** vengono considerati picchi solamente quelli che hanno un valore ≥ 8.1 Mt/min (default 0 tutti i picchi)

Il parametro **Y** indica il fattore di divisione per la riduzione del picco. **Y=2** il picco viene ridotto del 50%. (default **2** 50%)



17.16.10 **G67 – Usa Px_Moveto per movimenti fuori dal piano – Px_Lineto per movimenti nel piano**

Sintassi

G67

Tipo funzione

MODALE

Revoca

G68

Descrizione

G67 Usa la combinazione Px_Moveto per movimenti fuori dal piano di lavoro e Px_Lineto per movimenti nel piano di lavoro. I movimenti fuori dal piano usano sempre la fermata sul tratto.

NON DEVE ESSERE ABILITATO G63-G65

17.16.11 **G68 – Usa solo– Px_Lineto per movimenti nel piano e fuori dal piano – ASSI TRASPORTATI con possibilità di combinare con px_moveto**

Sintassi

G68 Xval YTangAxis

Tipo funzione

MODALE

Revoca

G67

Parametri

Xval

Parametro opzionale

val>0 indica la velocità limite di un eventuale asse tangenziale

Se l' asse tangenziale supera questa velocità questo non viene più trasportato ma interpolato con gli altri. (**deve essere abilitato G69**)

Val=0 disabilitata

Val=-1 l' asse non viene mai trasportato, ma viene calcolata solamente la soglia spigolo sul 2D del piano di lavoro (come G65 ma con SGLP)

YTangAxis

Indice Asse Tangenziale se presente (Lama ecc.)

Descrizione

G68 Usa solamente Px_Lineto per movimenti nel piano di lavoro e fuori dal piano di lavoro. Questo permette di gestire alcuni tipi di interpolazioni per certe tipologie di macchine (es: tagli con lama tangente).

IN EFFETTI GLI ASSI FUORI DAL PIANO DI LAVORO DIVENTANO TRASPORTATI

Gli assi fuori dal piano di lavoro vengono portati alla sua quota finale con il calcolo delle velocità solo sul piano di lavoro, pertanto non viene scomposto su questi il vettore di velocità.

Un esempio dell' utilizzo di questo tipo di interpolazione può essere quello dei plotter da TAGLIO con lama che deve rimanere tangente al profilo di TAGLIO. La velocità degli assi viene calcolata scomponendo i vettori del piano di lavoro impostato, mentre l' asse tangenziale viene portato di conseguenza alla sua quota finale nello stesso tempo degli assi del PIANO. Questo permette il calcolo delle velocità sul piano di lavoro senza distorsione di questa dovuta all' asse tangenziale.

Utilizzando il parametro X è possibile ottenere interpolazioni miste tra assi trasportati e non.

Uno dei principali problemi per gli assi trasportati è che questi in alcuni casi possono raggiungere delle velocità limiti del sistema. Impostando queste velocità limite nel parametro X è possibile in quei punti, utilizzare un interpolazione su tutti gli assi, riportando la velocità dell' asse trasportato inferiore o uguale alla FEED programmata.

NON DEVE ESSERE ABILITATO G63-G65

17.16.12 G69 – LHK – Profondità buffer look ahead

Sintassi

G69 Xval ASSETANG

Tipo funzione

Disattivata al RUN

Parametri

val *Numero elementi di profondità buffer*
 Se VAL<20 AFC disattivato
 VAL viene limitato comunque al numero dei blocchi del PartProgram

ASSETANG *Eventuale asse tangenziale presente sulla macchina (Es Z A ecc.)*
 Parametro non obbligatorio, se non presente non viene considerato nessuna
 asse tangenziale

Descrizione

G69 definisce ed abilita la profondità del buffer **look ahead**. Questo permette alle varie funzioni che ne utilizzano (G66 – G72 – G73 – G74) di funzionare in modo ottimale.

Maggiore è la profondità migliore è il risultato definitivo di queste funzioni.

La profondità viene comunque limitata ad un valore <20 (LHK disabilitato ed ad un valore maggiore della lunghezza del part program.

Un valore tipo può essere 200 elementi.

Questa funzione definisce anche un eventuale asse tangenziale presente nella macchina (es. lama da taglio). Questo è necessario poichè le funzioni correlate a **G69** (G66 – G72 – G73 – G74) variano il profilo originale, aggiungendo o togliendo tratti e quindi risulta necessario ricalcolare il valore della tangenza dell' asse.

Es:

G69 X200 A

Nel precedente esempio la profondità del buffer viene impostata a 200 elementi e viene indicato l' asse tangenziale A.

17.17 FILTRI DI ISOUS

Isous comprende al suo interno una serie di filtri atti a manipolare il percorso originale degli assi proveniente da un part program. I filtri risultano utili quando si utilizzano part program derivati da CAM (es acquisizioni da scanner) che non hanno un percorso "PULITO", cioè che presentano spigolature che non devono essere riportate sulla lavorazione, o che comunque limiterebbero la velocità della lavorazione stessa.

I filtri agiscono solamente sui blocchi G1 e comunque deve essere abilitato la funzione G69 (profondità LHK).

I filtri possono avere funzionalità **2D** (solamente sugli assi del piano di lavoro impostato) oppure **3D** (che agiscono quindi sugli assi del piano e sull'asse di profondità).

I filtri **NON AGISCONO COMUNQUE SU PIU' DI TRE ASSI**.

AVVERTENZA

L'asse di profondità viene automaticamente preso dalla terna di assi XYZ quello rimanente fuori dal piano impostato.

Piano di lavoro X,Y Asse di profondità Z

Piano di lavoro X,Z Asse di profondità Y

Piano di lavoro Y,Z Asse di profondità X

Utilizzando i filtri le linee di lavorazione non corrispondono più al PART PROGRAM originale.

E' normale che alcune di queste non vengono lavorate, oppure che alcune di queste siano scomposte in più segmenti.

SUCCESSIONE DEI FILTRI

I filtri possono essere combinati insieme, comunque vengono eseguiti nella seguente successione:

| | | |
|-------------------|-------------|-----------------------|
| NOISE | G73 | (se abilitato) |
| NURBS | G72 | (se abilitato) |
| FINE NOISE | G73 | (se abilitato) |
| RLS | G74 | (se abilitato) |
| SMOOTHING | G106 | (se abilitato) |
| AFC | G66 | (se abilitato) |

17.17.1 G72 – N.U.R.B.S (Non Uniform Rational Bspline) (2D 3D)**Sintassi****G72 Xval Yminlen Zorder Aalseg****Tipo funzione***Disattivata al RUN***Parametri**

| | | |
|---------------|---|-----------|
| val | 0 – filtro NURBS <i>disabilitato</i> 1 - filtro NURBS <i>abilitato</i> | |
| minlen | Lunghezza minima del tratto G1 per essere considerata nel filtro NURBS. Il filtro agisce in un buffer che ha una successione continua di tratti di lunghezza minore del parametro MINLEN. Quando un tratto maggiore o uguale a questa lunghezza viene rilevato, il buffer NURBS viene elaborato effettuando interpolazione sui punti. | in |
| | Valore tipico di default 0.2 mm <u>Parametro non obbligatorio, se non viene inserito prende il valore precedente o quello di default</u> | |
| order | Ordine della curva NURBS (valori compresi tra 1 e 7) Questo definisce la precisione della curva risultante, più basso è il valore maggiore è la precisione della curva sui punti di controllo. Valore=1 vengono solamente riportati i punti originali Valore tipico di default 3. <u>Parametro non obbligatorio, se non viene inserito prende il valore precedente o quello di default</u> | |
| lenseg | Indica la lunghezza minima dei tratti in uscita della curva risultante dall' interpolazione dei punti. Più basso è il valore, minore è la lunghezza dei segmenti della curva, maggiore è il numero di segmenti risultanti (curva più accurata) Valore tipico di default 0.2mm <u>Parametro non obbligatorio, se non viene inserito prende il valore precedente o quello di default</u> | |

Descrizione

Le NURBS sono un tipo di curve atte a risolvere problemi che si hanno con le curve B-Spline; il più significativo è dato dall' impossibilità di disegnare figure semplici come il cerchio. Mediante questa evoluzione è infatti possibile rappresentare correttamente curve coniche, ed è possibile rappresentare curve molto complesse con un minor numero di punti di controllo.

Le N.U.R.B.S. (Non Uniform Rational B-Spline), sono curve razionali definite da punti di controllo e dai relativi pesi; aumentando il valore del peso di un punto di controllo avviciniamo la curva al punto stesso; viceversa il punto di controllo eserciterà una minor influenza sulla curva.

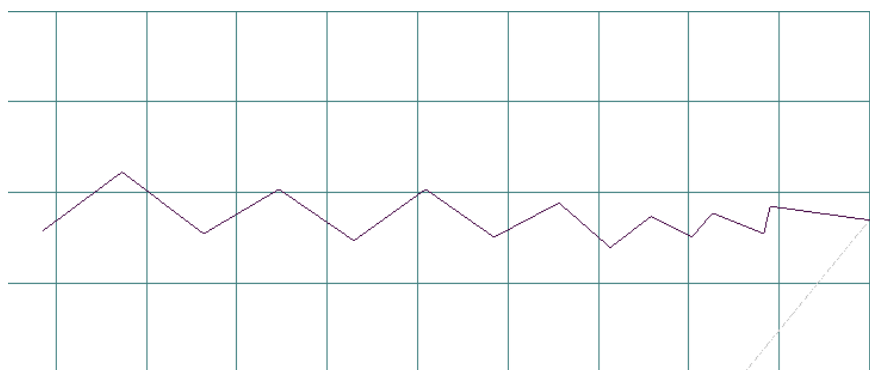
ISOUS utilizza questo tipo di interpolazione per **“ADDOLCIRE”** una serie di punti derivati da blocchi successivi del PartProgram. Il risultato porta ad una modellazione del profilo risultante

Di seguito vengono illustrati alcuni esempi per chiarire meglio il concetto delle NURBS.

Il filtro NURBS agisce anche su profili 3D**IL FILTRO NURBS E' IL SECONDO FILTRAGGIO AD ESSERE EFFETTUATO SUL PARTPROGRAM**

PROFILO ORIGINALE

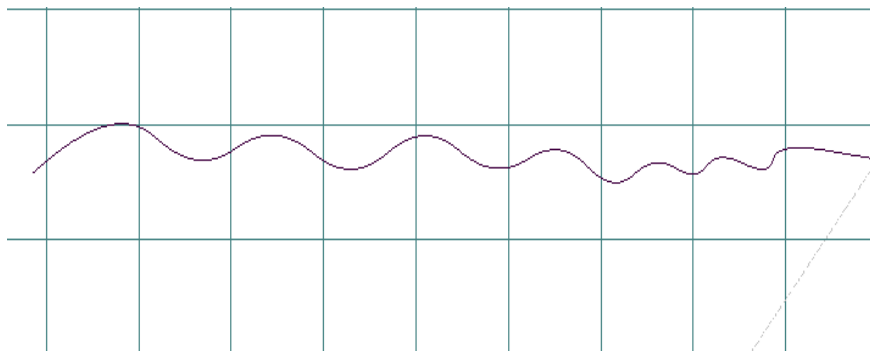
G0 X107.16 Y130.27
 G1 X100.59 Y131.18
 G1 X100.14 Y129.37
 G1 X96.74 Y130.73
 G1 X95.38 Y129.14
 G1 X92.66 Y130.5
 G1 X89.95 Y128.46
 G1 X86.55 Y131.41
 G1 X82.24 Y129.14
 G1 X77.71 Y132.31
 G1 X72.95 Y128.91
 G1 X67.97 Y132.31
 G1 X62.98 Y129.37
 G1 X57.55 Y133.45
 G1 X52.34 Y129.59



PROFILO FILTRATO CON NURBS MiLen=7mm Order=3 LenSeg=0.2mm

G69X200 // ABILITA LHK
 // ABILITA NURBS LEN 7 MM ORDER 3 SEGMENT 0.2

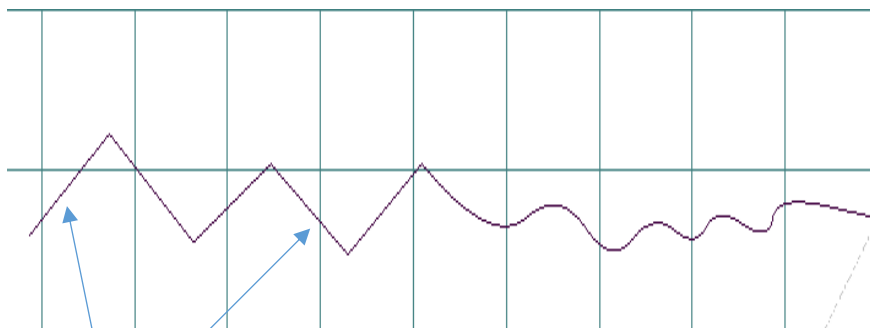
G72X1Y7
 G0 X107.16 Y130.27
 G1 X100.59 Y131.18
 G1 X100.14 Y129.37
 G1 X96.74 Y130.73
 G1 X95.38 Y129.14
 G1 X92.66 Y130.5
 G1 X89.95 Y128.46
 G1 X86.55 Y131.41
 G1 X82.24 Y129.14
 G1 X77.71 Y132.31
 G1 X72.95 Y128.91
 G1 X67.97 Y132.31
 G1 X62.98 Y129.37
 G1 X57.55 Y133.45
 G1 X52.34 Y129.59



PROFILO FILTRATO CON NURBS MiLen=5mm Order=3 LenSeg=0.2mm

G69X200 // ABILITA LHK
 // ABILITA NURBS LEN 5 MM ORDER 3 SEGMENT 0.2

G72X1Y5
 G0 X107.16 Y130.27
 G1 X100.59 Y131.18
 G1 X100.14 Y129.37
 G1 X96.74 Y130.73
 G1 X95.38 Y129.14
 G1 X92.66 Y130.5
 G1 X89.95 Y128.46
 G1 X86.55 Y131.41
 G1 X82.24 Y129.14
 G1 X77.71 Y132.31
 G1 X72.95 Y128.91
 G1 X67.97 Y132.31
 G1 X62.98 Y129.37
 G1 X57.55 Y133.45
 G1 X52.34 Y129.59

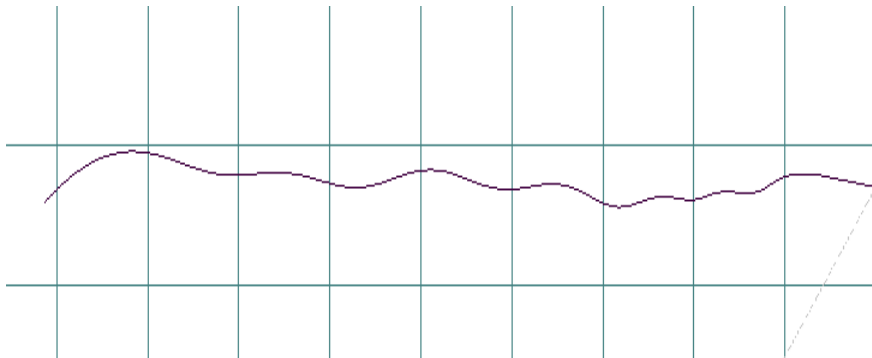


TRATTI DI LUNGHEZZA >= a 5 mm (non filtrati)

PROFILO FILTRATO CON NURBS MiLen=7mm Order=5 LenSeg=0.2mm

Con ordine NURBS più alto la curva è più e si discosta di più dai punti di controllo dati dal profilo originale

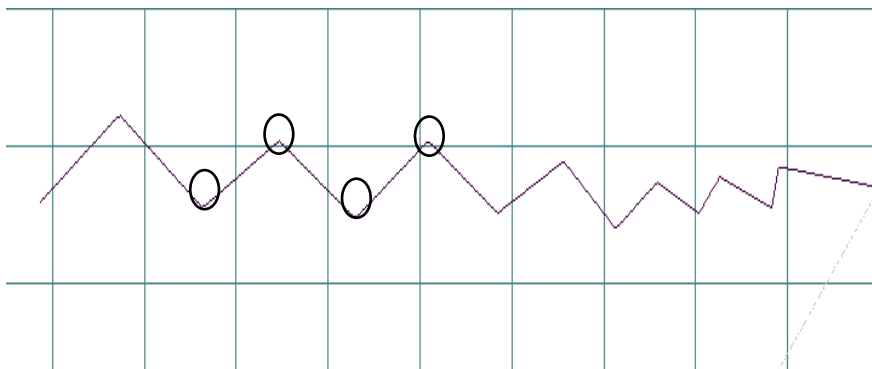
```
G69X200 // ABILITA LHK
G72X1Y7Z5
G0 X107.16 Y130.27
G1 X100.59 Y131.18
G1 X100.14 Y129.37
G1 X96.74 Y130.73
G1 X95.38 Y129.14
G1 X92.66 Y130.5
G1 X89.95 Y128.46
G1 X86.55 Y131.41
G1 X82.24 Y129.14
G1 X77.71 Y132.31
G1 X72.95 Y128.91
G1 X67.97 Y132.31
G1 X62.98 Y129.37
G1 X57.55 Y133.45
G1 X52.34 Y129.59
```



PROFILO FILTRATO CON NURBS MiLen=7mm Order=2 LenSeg=0.2mm

Con **Order=2** si rende quasi impercettibile i punti della spline. Anche se in effetti nei cerchi contrassegnati, il profilo originale viene leggermente modellato

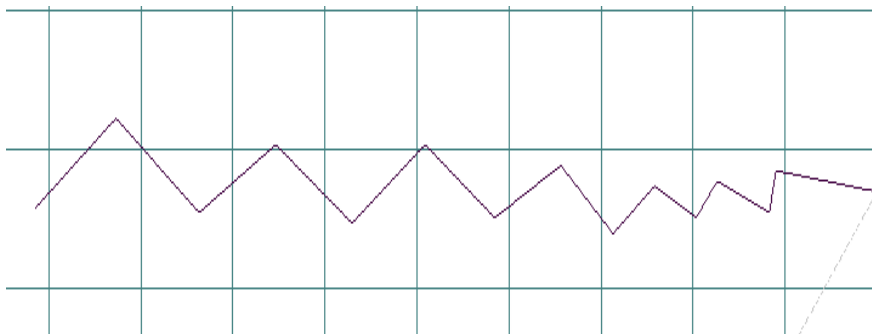
```
G69X200 // ABILITA LHK
G72X1Y7Z1
G0 X107.16 Y130.27
G1 X100.59 Y131.18
G1 X100.14 Y129.37
G1 X96.74 Y130.73
G1 X95.38 Y129.14
G1 X92.66 Y130.5
G1 X89.95 Y128.46
G1 X86.55 Y131.41
G1 X82.24 Y129.14
G1 X77.71 Y132.31
G1 X72.95 Y128.91
G1 X67.97 Y132.31
G1 X62.98 Y129.37
G1 X57.55 Y133.45
G1 X52.34 Y129.59
```



PROFILO FILTRATO CON NURBS MiLen=7mm Order=1 LenSeg=0.2mm

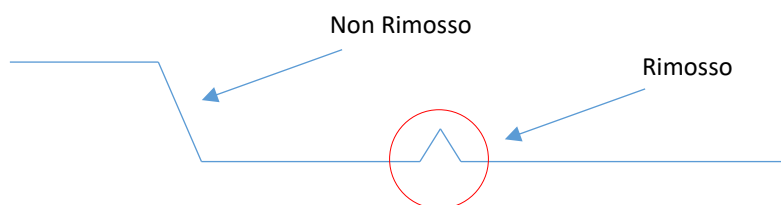
Profilo originale non filtrato

```
G69X200 // ABILITA LHK
G72X1Y7Z1
G0 X107.16 Y130.27
G1 X100.59 Y131.18
G1 X100.14 Y129.37
G1 X96.74 Y130.73
G1 X95.38 Y129.14
G1 X92.66 Y130.5
G1 X89.95 Y128.46
G1 X86.55 Y131.41
G1 X82.24 Y129.14
G1 X77.71 Y132.31
G1 X72.95 Y128.91
G1 X67.97 Y132.31
G1 X62.98 Y129.37
G1 X57.55 Y133.45
G1 X52.34 Y129.59
```



17.17.2 G73 – NOISE (2D - 3D)**Sintassi****G73 Xval Yminang Zminlen Aonlyvar****Tipo funzione***Disattivata al RUN***Parametri**

| | |
|----------------|---|
| val | <p>0 – filtro NOISE disabilitato FINE NOISE disabilitato</p> <p>1 - filtro NOISE abilitato FINE NOISE disabilitato</p> <p>2 - filtro NOISE disabilitato FINE NOISE abilitato</p> <p>3 - filtro NOISE abilitato FINE NOISE abilitato</p> |
| minang | <p>Angolo minimo in gradi dello spigolo che viene considerato come RUMORE</p> <p>Valore tipico di default 25 gradi</p> <p><u>Parametro non obbligatorio, se non viene inserito prende il valore precedente o quello di default</u></p> |
| minlen | <p>Lunghezza minima del tratto G1 per essere considerata nel filtro N.</p> <p>Il filtro agisce in un buffer che ha una successione continua di tratti di lunghezza minore del parametro MINLEN. Quando un tratto maggiore o uguale a questa lunghezza viene rilevato, il buffer NOISE viene elaborato effettuando il filtraggio sui punti.</p> <p>Valore tipico di default 0.4 mm</p> <p><u>Parametro non obbligatorio, se non viene inserito prende il valore precedente o quello di default</u></p> |
| onlyvar | <p>0 – rimuove tutti gli spigoli che rientrano nei parametri</p> <p>1 – rimuove solo cuspidi che rientrano nei parametri (variazioni di angolo improvvise)</p> <p>Valore tipico di default 0</p> <p><u>Parametro non obbligatorio, se non viene inserito prende il valore precedente o quello di default</u></p> |

**Descrizione**

Il filtro **NOISE** serve per rimuovere tratti di **G1** che formano piccoli angoli o cuspidi definiti come **RUMORE**.

Questo può risultare utile per eliminare tratti di profilo che anche passati al filtro NURBS sarebbero comunque inutili e fastidiosi.

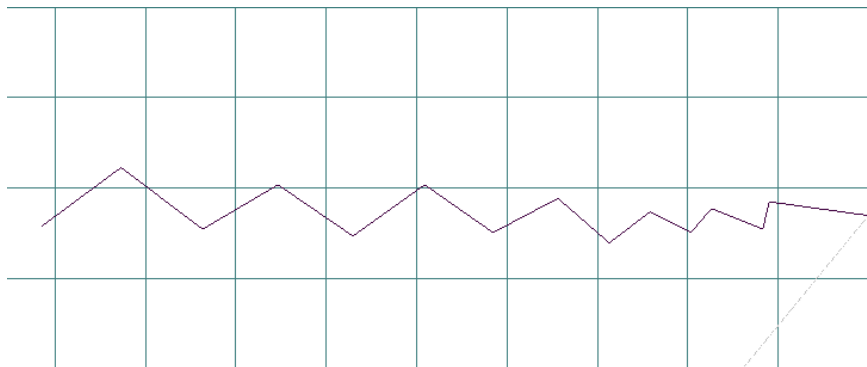
Di seguito vengono illustrati alcuni esempi per chiarire meglio il concetto del filtro NOISE

Il filtro NOISE agisce su profili 2De 3D

**IL FILTRO NOISE E' IL PRIMO FILTRAGGIO AD ESSERE EFFETTUATO SUL PARTPROGRAM
PROFILO ORIGINALE**

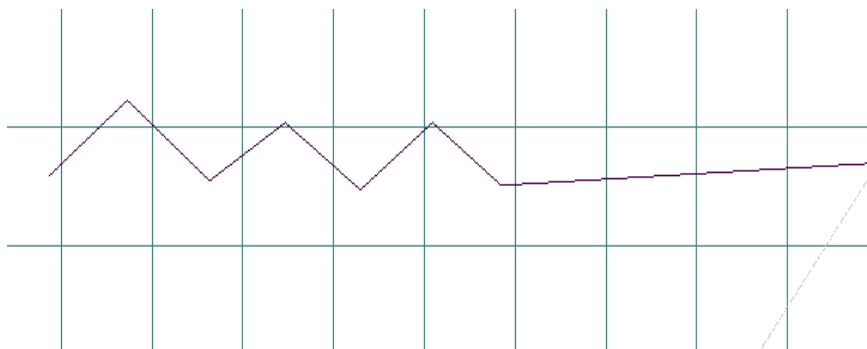
PROFILO ORIGINALE

G0 X107.16 Y130.27
 G1 X100.59 Y131.18
 G1 X100.14 Y129.37
 G1 X96.74 Y130.73
 G1 X95.38 Y129.14
 G1 X92.66 Y130.5
 G1 X89.95 Y128.46
 G1 X86.55 Y131.41
 G1 X82.24 Y129.14
 G1 X77.71 Y132.31
 G1 X72.95 Y128.91
 G1 X67.97 Y132.31
 G1 X62.98 Y129.37
 G1 X57.55 Y133.45
 G1 X52.34 Y129.59



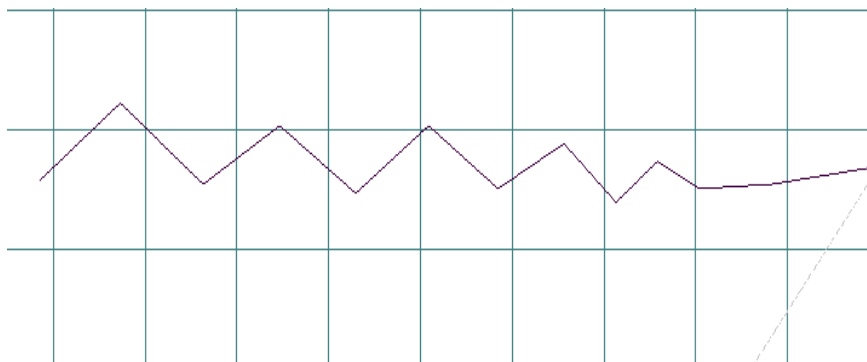
PROFILO FILTRATO MinAng=45 MinLen=5

G69X200 // ABILITA LHK
 G73X1Y45Z5
 G0 X107.16 Y130.27
 G1 X100.59 Y131.18
 G1 X100.14 Y129.37
 G1 X96.74 Y130.73
 G1 X95.38 Y129.14
 G1 X92.66 Y130.5
 G1 X89.95 Y128.46
 G1 X86.55 Y131.41
 G1 X82.24 Y129.14
 G1 X77.71 Y132.31
 G1 X72.95 Y128.91
 G1 X67.97 Y132.31
 G1 X62.98 Y129.37
 G1 X57.55 Y133.45
 G1 X52.34 Y129.59



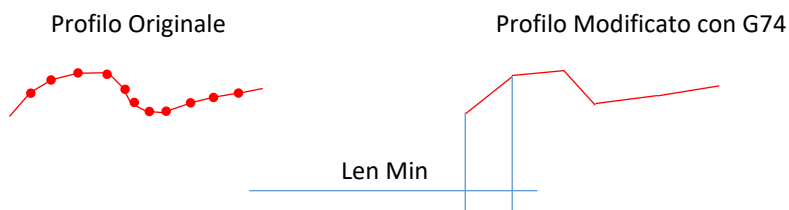
PROFILO FILTRATO MinAng=45 MinLen=3

G69X200 // ABILITA LHK
 G73X1Y45Z3
 G0 X107.16 Y130.27
 G1 X100.59 Y131.18
 G1 X100.14 Y129.37
 G1 X96.74 Y130.73
 G1 X95.38 Y129.14
 G1 X92.66 Y130.5
 G1 X89.95 Y128.46
 G1 X86.55 Y131.41
 G1 X82.24 Y129.14
 G1 X77.71 Y132.31
 G1 X72.95 Y128.91
 G1 X67.97 Y132.31
 G1 X62.98 Y129.37
 G1 X57.55 Y133.45
 G1 X52.34 Y129.59



17.17.3 G74 – RLS Remove Len Segment (2D 3D)**Sintassi****G74 Xval Ylenmin****Tipo funzione***Disattivata al RUN***Parametri**

| | |
|---------------|--|
| val | 0 – filtro RLS disabilitato 1 - filtro RLS abilitato |
| lenmin | Lunghezza minima del tratto G1 accettabile. Lunghezze inferiori a queste vengono eliminate con un unico tratto Valore tipico di default 0.5 mm Con lenmin=0 il valore di lunghezza viene calcolato in modo automatico in base alla FEED impostata. <u>Parametro non obbligatorio, se non viene inserito prende il valore precedente o quello di default</u> |

**Descrizione**

Il filtro **RLS** serve per rimuovere tratti di **G1 "CORTI"** che in qualche modo non verrebbero comunque considerati, poiché non rientrano nei termini di velocità richiesti. Questo in quanto la **FEED** impostata fa sì che venga comunque percorso uno spazio maggiore per campionamento **TAU** del **CN**.

Se il valore viene impostato a **ZERO**, **ISOUS** calcola automaticamente la lunghezza, ottimizzandola alla **FEED** attuale.

ATTENZIONE

In questo caso il potenziometro di override non viene considerato (potenziometro al valore 100%)

Il risultato porta ad un'esclusione di **TRATTI INUTILE**, liberando **BUFFER** di movimentazione e quindi permettendo **FEED** più elevate.

Il filtro RLS agisce solamente su profili 2D e 3D

IL FILTRO RLS E' IL QUARTO FILTRAGGIO AD ESSERE EFFETTUATO SUL PARTPROGRAM

17.17.4 G106 – SMOOTHING**Sintassi****G106.ax Xval ALevel Bminlen****Tipo funzione***Disattivata al RUN***Parametri**

.ax *Indice dell' asse su cui attivare lo smoothing (da 0 a n) es: G106.0 (asse X)*

val *0 – filtro Smoothing **disabilitato***
*1 - filtro Smoothing **abilitato***

Level *Livello di smoothing (da 0 a 1) Valori bassi smoothing più alto es. A0.3*

Parametro non obbligatorio, se non viene inserito prende il valore precedente o quello di default

minlen *Lunghezza minima del tratto G1 per essere considerata nel filtro. Valori maggiori a questa lunghezza vengono lasciati inalterati.*
La lunghezza è considerata come vettore del tratto, pertanto comprende tutti gli assi configurati.
E' necessario settarla una sola volta.
Se il parametro è omesso, viene preso l' ultima lunghezza inserita

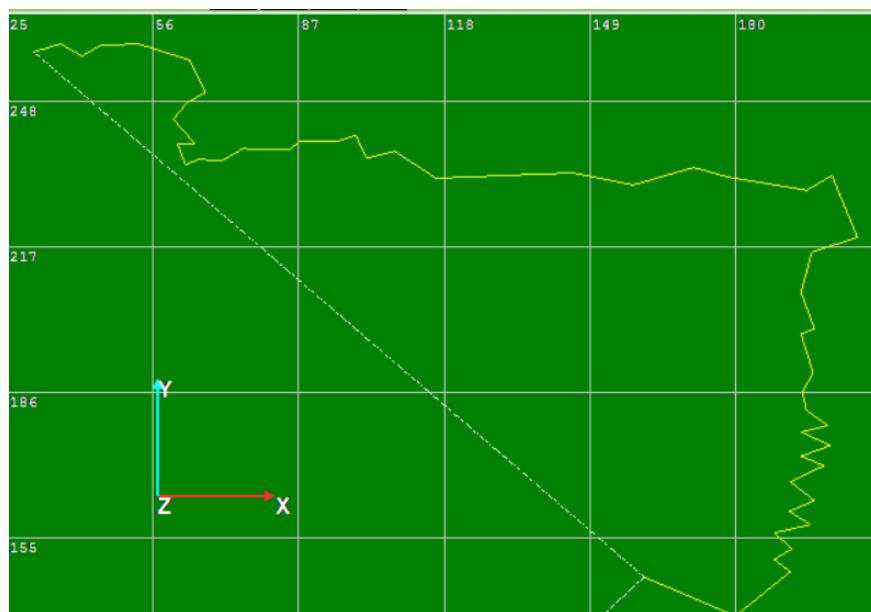
Descrizione

Il filtro **SMOOTHING** serve per “livellare” le variazioni di ogni singolo asse con elementi formati da **G1** che sono definiti come **RUMORE**.

Il profilo subirà un livellamento, favorendo la lavorazione come fluidità e velocità di esecuzione, ma con minor precisione rispetto all' originale.

Il filtro **SMOOTHING** viene attivato solo su singoli assi rendendo più flessibile il filtraggio.

IL FILTRO SMOOTHING E' IL QUINTO FILTRAGGIO AD ESSERE EFFETTUATO SUL PARTPROGRAMÙ

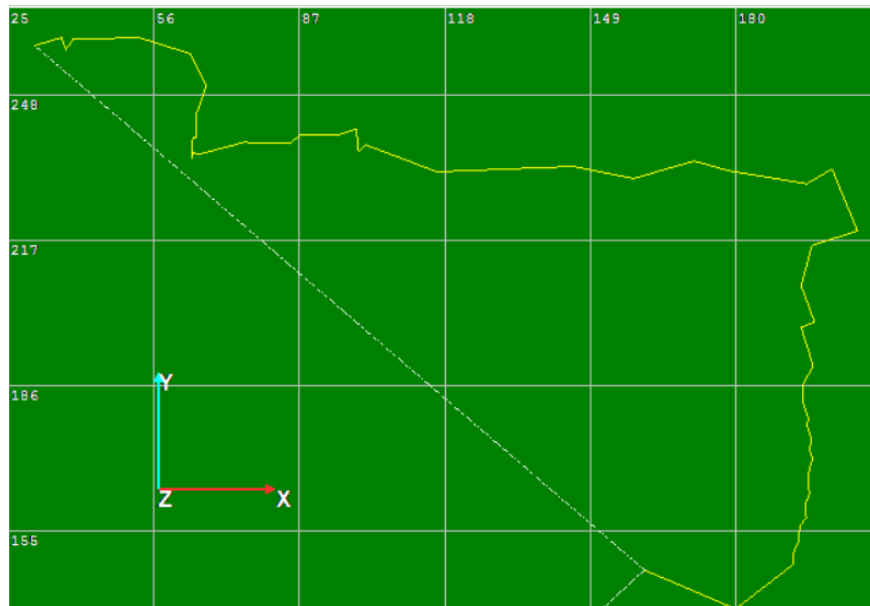
PROFILO ORIGINALE**G60****G0 Z3****G0X30.4845 Y258.703****F15****G01 Z-1****X36.1945 Y260.416 F15****X40.7625 Y257.8465****X44.7595 Y260.1305****X52.7535 Y260.416****X63.6025 Y256.99****X67.0285 Y250.138****X63.0315 Y247.854****X60.1765 Y244.428****X64.7445 Y239.289****X61.033 Y239.289****X62.746 Y234.721****X65.601 Y235.863****X70.4545 Y235.5775****X75.0225 Y238.147****X84.7295 Y237.8615****X87.299 Y239.86****X95.5785 Y239.86****X99.0045 Y241.002****X101.2885 Y236.1485****X107.284 Y237.576****X116.1345 Y231.866****X145.2555 Y233.008****X158.103 Y230.4385****X170.9505 Y234.15****X178.088 Y232.1515****X194.9325 Y229.2965****X200.357 Y232.437****X205.7815 Y219.304****X196.0745 Y216.1635****X193.7905 Y207.5985****X196.6455 Y199.89****X193.7905 Y198.748****X196.36 Y190.4685****X194.076 Y186.4715****X194.9325 Y182.4745****X199.5005 Y179.334****X193.7905 Y177.9065****X200.0715 Y175.0515****X193.7905 Y172.7675****X198.644 Y170.769****X191.5065 Y167.343****X196.6455 Y163.346****X191.221 Y161.062****X195.789 Y158.207****X188.0805 Y156.494****X191.792 Y153.068****X188.0805 Y150.784****X191.5065 Y148.2145****X179.801 Y138.793****X160.387 Y147.0725****G0 X160.387 Y147.0725**

PROFILO – Smoothing su X,Y livello A0.2 segmenti lunghezza minima di 8 mm B8

```

G60
G0 Z3
G69X200
G106.0 A0.2 B8
G106.1
G0X30.4845 Y258.703
F15
G01 Z-1
X36.1945 Y260.416 F15
X40.7625 Y257.8465
X44.7595 Y260.1305
X52.7535 Y260.416
X63.6025 Y256.99
X67.0285 Y250.138
X63.0315 Y247.854
X60.1765 Y244.428
X64.7445 Y239.289
X61.033 Y239.289
X62.746 Y234.721
X65.601 Y235.863
X70.4545 Y235.5775
X75.0225 Y238.147
X84.7295 Y237.8615
X87.299 Y239.86
X95.5785 Y239.86
X99.0045 Y241.002
X101.2885 Y236.1485
X107.284 Y237.576
X116.1345 Y231.866
X145.2555 Y233.008
X158.103 Y230.4385
X170.9505 Y234.15
X178.088 Y232.1515
X194.9325 Y229.2965
X200.357 Y232.437
X205.7815 Y219.304
X196.0745 Y216.1635
X193.7905 Y207.5985
X196.6455 Y199.89
X193.7905 Y198.748
X196.36 Y190.4685
X194.076 Y186.4715
X194.9325 Y182.4745
X199.5005 Y179.334
X193.7905 Y177.9065
X200.0715 Y175.0515
X193.7905 Y172.7675
X198.644 Y170.769
X191.5065 Y167.343
X196.6455 Y163.346
X191.221 Y161.062
X195.789 Y158.207
X188.0805 Y156.494
X191.792 Y153.068
X188.0805 Y150.784
X191.5065 Y148.2145
X179.801 Y138.793
X160.387 Y147.0725
G0 X160.387 Y147.0725

```



17.18 ASSI VIRTUALI

Gli assi virtuali sono comandi che vengono inseriti nell' interpolatore ma che non riguardano una asse vero e proprio. Questi in pratica sono dei comandi sincroni che possono abilitare degli eventi in punti precisi. Ovviamente il tutto ha ragione di esistere quando il CN si trova in interpolazione veloce e i tratti vengono inseriti nel buffer dei movimenti del CN. Il comando sincrono può specificare un' precisa operazione che il CN deve effettuare in un determinato punto.

17.18.1 G100 – Comando sincrono per asse virtuale

Sintassi

G100 Xcomando

Tipo funzione

immediata

Parametri

Comando **Comando da inviare al CN**

Descrizione

G100 invia il comando al CN da eseguire in modo sincrono. Tale comando deve essere poi gestito nell' applicazione del CN in VTB in modo effettuare la giusta operazione. (fare riferimento al manuale del CN per gestione comandi sincroni).

Esempio di gestione di un' uscita digitale sincrona con i movimenti assi:

G100 X0 Resetta uscita

G100 X1 Setta uscita

G60 //ATTIVA INTERPOLAZIONE VELOCE

G1X100Y100

X120Y110

X130

G100 X1 // SETTA USCITA

X200Y160

X210Y180

G100 X0 // RESETA USCITA

X300Y200

X310Y220

Applicazione in VTB inserita nel TaskPlc

'controlla set set/uscita

Uscita=nomeapplicazione.cmd

dove:

| | |
|-------------------------|---|
| Uscita | uscita digitale da aggiornare |
| nomeapplicazione | Nome dell' oggetto ISO utilizzato |
| .cmd | valore della variabile aggiornato da G100 |

17.19 Gestione Macro ad Interrupt

È possibile utilizzare 3 tipi di macro che vengono gestite in interruzione.

Le macro hanno una numerazione fissa e devono essere create con il relativo numero

| | |
|-------------------------------|------------------------|
| A Tempo | M50000 |
| Su ingresso Digitale | M50001 |
| Su Uscite Digitale | M50002 |
| Ad Ogni Movimento assi | M50003 e M50004 |

La gestione di queste macro viene effettuata dalla funzione G107

17.19.1 G107 – Gestione Macro ad Interrupt

Sintassi

G107 X0

Disabilita tutte le Macro ad Interrupt escluso M50003 e M50004

G107 X1

Abilita tutte le Macro ad Interrupt M50003 e M50004

G107 X2

Sospende tutte le Macro ad Interrupt M50003 e M50004

G107 X3

Riprende tutte le Macro ad Interrupt M50003 e M50004
(se erano disabilitate rimangono disabilitate)

G107.0 Xtime

Macro M50000 Attivazione a tempo

Time indica il tempo in millisecondi (X-1 Disabilitata)

Quando la macro è in esecuzione viene Disabilitata per essere ripresa alla fine dell' esecuzione

Esempio:

1) Creare la macro 50000

```
G62 // WAIT AXES STOP
$SAVEX=${Q0} // SAVE X
$SAVEY=${Q1} // SAVE Y
$SAVEZ=${Q2} // SAVE Z
G0X0Y0Z0 // MOVE AXES
G4F1 // PAUSE
G0X[$SAVEX]Y[$SAVEY] // RESUME AXES VALUES XY
G0Z[$SAVEZ] // RESUME AXIS VALUE Z
G4F1 // PAUSE
```

2) Attivazione nel Gcode ogni 5 sec

G107.0 X5000

La macro M50000 verrà eseguita ogni 5 secondi

G107.1 XDI

Macro M50001 Attivazione su ingresso digitale STATO ON

DI Indica il numero di ingresso digitale da 0 a 31 (-1 Disabilita)

Quando la macro è in esecuzione viene Disabilitata per essere ripresa alla fine dell' esecuzione

Esempio:

3) Creare la macro 50001

```
G62                // WAIT AXES STOP
$SAVEX=$[Q0]       // SAVE X
$SAVEY=$[Q1]       // SAVE Y
$SAVEZ=$[Q2]       // SAVE Z
G0X0Y0Z0           // MOVE AXES
G4F1               // PAUSE
G0X[$SAVEX]Y[$SAVEY] // RESUME AXES VALUES XY
G0Z[$SAVEZ]        // RESUME AXIS VALUE Z
G4F1               // PAUSE
```

4) Attivazione nel Gcode sul primo ingresso STATO ON

G107.1 X0

La macro M50001 verrà eseguita ogni qualvolta l' ingresso passa dallo stato logico 0 a quello 1

G107.2 XDO

Macro M50002 Attivazione su uscita digitale STATO ON

DO Indica il numero di uscita digitale da 0 a 31 (-1 Disabilita)

Quando la macro è in esecuzione viene Disabilitata per essere ripresa alla fine dell' esecuzione

Esempio:

1) Creare la macro 50002

```
G62                // WAIT AXES STOP
$SAVEX=$[Q0]       // SAVE X
$SAVEY=$[Q1]       // SAVE Y
$SAVEZ=$[Q2]       // SAVE Z
G0X0Y0Z0           // MOVE AXES
G4F1               // PAUSE
G0X[$SAVEX]Y[$SAVEY] // RESUME AXES VALUES XY
G0Z[$SAVEZ]        // RESUME AXIS VALUE Z
G4F1               // PAUSE
```

2) Attivazione nel Gcode sulla uscita STATO ON

G107.2 X0

La macro M50002 verrà eseguita ogni qualvolta l' uscita passa dallo stato logico 0 a quello 1

G107.3 Xp1 Yp2 Zp3 Ap4 Bp5 Cp6**G107.4 Xp1 Yp2 Zp3 Ap4 Bp5 Cp6**

Macro M50003 e M50004 Attivazione ad ogni movimento su blocco

Se il blocco contiene un movimento Assi, alla fine viene lanciata la macro M50003 o M50004

Parametri:

| | | |
|---------------|----------|--|
| P1 | 0 | Disattiva Macro M50003 o M50004 |
| | 1 | Attiva Macro M50003 o M50004 |
| P2..P6 | | Parametri che possono essere utilizzati nelle M50003 e M50004 per indirizzo dove: |
| | | P2 Indirizzo 20000 |
| | | P3 Indirizzo 20001 |
| | | P4 Indirizzo 20002 |
| | | P5 Indirizzo 20003 |
| | | P6 Indirizzo 20004 |

Esempio:**1) Creare la macro 50003**

```

$FEED1=${:20000]           // FEED1 SU P2 Y
$FEED2=${:20001]           // FEED2 SU P3 Z
F[$FEED1]
G1X0Y0Z0                   // MUOVE ASSI CON FEED1 (Y)
G4F1                       // PAUSA
F[$FEED2]
G1X100Y100Z100           // MUOVE ASSI CON FEED2 (Z)
G4F1                       // PAUSA

```

2) Attivazione nel Gcode

```

G0X0Y0Z0
G107.3 X1 Y10.2 Z5.3     // ABILITA M50003 CON FEED1=10.2 e FEED2=5.3
F2
G1X20Y20Z20             // ALLA FINE DEL BLOCCO LA M50003 VIENE CHIAMATA
G1X30Y30Z30             // ALLA FINE DEL BLOCCO LA M50003 VIENE CHIAMATA
G1X40Y40Z40             // ALLA FINE DEL BLOCCO LA M50003 VIENE CHIAMATA
G107.3 X0                // DISABILITA M50003

```

Le eventuali F impostate nel Gcode vengono salvate durante l' esecuzione della Macro e ripristinate alla fine

17.20 Gestione Macro di EMERGENZA in STOP M60000

La macro di **EMERGENZA** viene attivata quando **CN** rileva un **ALLARME DI EMERGENZA** e si trova in stato di **STOP**.

In caso di **RUN** la macro attivata è la **MACRO DI ERRORE** configurabile da sistema.

La macro di **EMERGENZA** ha un codice fisso **M60000**

Pertanto è necessario generare una macro **M60000** tramite il PlugIn **MHM** e questa è pronta.

Generalmente in questa macro viene inserito solo codice per disattivazione dello stato di alcune uscite digitali o analogiche, in quanto la movimentazione ASSI viene bloccata dal CN con la disabilitazione degli ASSI.

Per Disattivare la **MACRO DI EMERGENZA** è necessario rimuovere dalla cartella **DATA_M** il file **M60000.bin**

La cartella **DATA_M** è presente nella cartella di installazione di IsoUs.

17.21 G108 Gestione Assi Speciali

La funzione G108 serve per la gestione speciale degli assi. Questa permette di abilitare funzionamenti diversi sugli assi rispetto al normale uso.

17.21.1 G108.0 G108.1 G108.2 G108.3 – Gestione Assi Master Slave

La funzione G108.0 .1 .2 .3 permette di gestire assi MASTER-SLAVE.

Cioè permette di collegare uno o più assi ad un asse master. Quindi programmando il movimento solo su questo asse master, gli assi slaves seguiranno di conseguenza la traiettoria.

È possibile gestire fino a 4 Gruppi di assi Master Slave, programmati rispettivamente con G108.0 G108.1 G108.2 e G108.3

Il numero massimo di Slaves programmati su un singolo master è di 8

Sintassi

G108.x 0

Annulla tutti gli slaves relativi al gruppo

G108.x 1

Sospende tutti gli slaves relativi al gruppo

G108.x 2

Riprende tutti gli slaves relativi al gruppo precedentemente programmati

G108.x M,S1,S2,S3..(-)S4 ecc.

Programma Gli Slaves

M = Asse Master (è sempre il primo asse programmato nella funzione G108.x)

S1 = Slave 1

S2 = Slave 2

Ecc.

Il segno “-“ davanti allo slave, inverte la direzione del relative slave rispetto al master

Es:

G108.0 X,A,-C // X Master A Slave C slave con direzione invertita rispetto ad X

G108.1 Y,Z,B,U // Y Master Z,B,U Slaves

.

.

G108.0 0 // Annulla primo Gruppo master slaves

17.21.2 G108.4 G108.5 G108.6 Gestione Assi in velocità

La funzione **G108.4** permette di agganciare un asse dell' interpolatore in modo velocità, quindi non più controllato da una quota, ma da una velocità. La velocità può essere passata con la funzione Gcode S se è stato impostato il giusto canale di uscita (vedi [Cap. 11.18](#))

La funzione **G108.5** riporta l' asse in interpolazione

La funzione **G108.6** è analoga a **G108.5** ma l' asse viene AZZERATO.

Questo permette di evitare errore di limite asse quando viene riportato nel modo INTERPOLATORE

Lo stato dell' asse in velocità può essere letto dalla variabile speciale [\\$\[X16\]](#)

Sintassi**G108.4 X,Y,Z,A,B,C,U,V,W**

Aggancia l' asse indicato (solo uno) da interpolatore a modo velocità.

G108.5 X,Y,Z,A,B,C,U,V,W

Riporta l' asse indicato dal modo Velocità al modo interpolatore

G108.6 X,Y,Z,A,B,C,U,V,W

Riporta l' asse indicato dal modo Velocità al modo interpolatore AZZERANDO IL VALORE DELLA QUOTA

17.21.3 G108.7 G108.8 G108.9 Scambio fisico Assi

La funzione **G108.7** permette di scambiare a livello di **CNC** una coppia di assi.

Praticamente lo scambio degli assi viene effettuato in modo **FISICO** e ciascun asse mantiene i propri parametri come per esempio i limiti software.

Possono essere scambiate anche più di una coppia di assi

G108.8 Ripristina la coppia di assi precedentemente scambiata

G108.9 Ripristina **TUTTE** le coppie di assi scambiati

Sintassi**G108.7 XA**

Scambia l' asse X con l' asse A

G108.8 XA

Riporta la coppia XA nel modo originale

G108.9

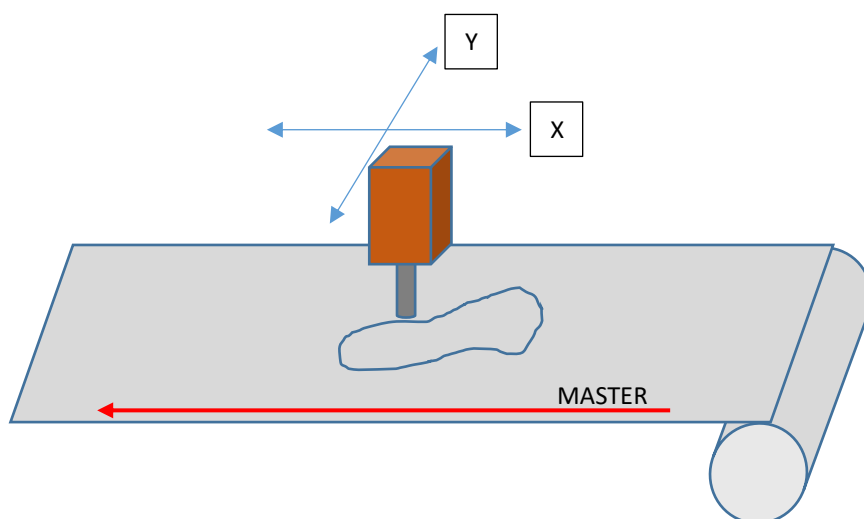
Riporta **TUTTE** le eventuali coppie di assi scambiati nel modo originale

17.21.4 G108.10 G108.11– Gestione Assi eGear

La funzione **G108.10 .11** permette di gestire assi **ELETTRICI**.

Questo può agganciare un **ASSE** interpolato **X,Y,Z** ecc. ad un **ASSE** esterno che si muove in modo **ASINCRONO** al processo Interpolato.

Per esempio, un nastro che trasporta l'oggetto da lavorare, può essere **INSEGUITO** da un asse interpolato ed effettuare la lavorazione considerando la velocità del nastro.



In questo esempio, l' asse MASTER si muove in una direzione trasportando il materiale da lavorare.

La funzione **G108.10** permette di agganciare l'**ASSE X** (slave) all' asse **MASTER** e di sommare la velocità di questo al profilo di interpolazione.

La funzione **G108.10** lavora insieme ai parametri [EGEAR KEM](#), [EGEAR KED](#), [EGEAR ACC](#)

Sintassi**G108.10 X,Y,Z,A,B,C,U,V,W**

AGGANCIA l' asse indicato (solo uno) al MASTER definito nell'applicazione VTB.

G108.11 X,Y,Z,A,B,C,U,V,W

SGANCA l' asse indicato al MASTER definito nell'applicazione VTB.

17.21.5 G108.12 G108.13 – Gestione velocità periferica assi rotativi

La G108.12 permette di armonizzare la velocità periferica relativa ad un asse rotativo, con la velocità degli altri assi presenti..

Questo algoritmo calcola il raggio “apparente”, ossia la distanza del punto di lavoro dal centro di rotazione dell’asse e mantiene costante la velocità del punto di lavoro al variare del suddetto raggio, dello spostamento degli altri assi e così via.

Il risultato, sarà un movimento con una velocità diversa da quella impostata nel codice Gcode, ma che mantiene costante la velocità del punto di lavoro.

Si possono impostare fino a due assi rotativi con il relativo asse di rotazione.

Sintassi

G108.12 Y,B

set assi da usare

G108.13 1

abilita gestione

G108.13 0

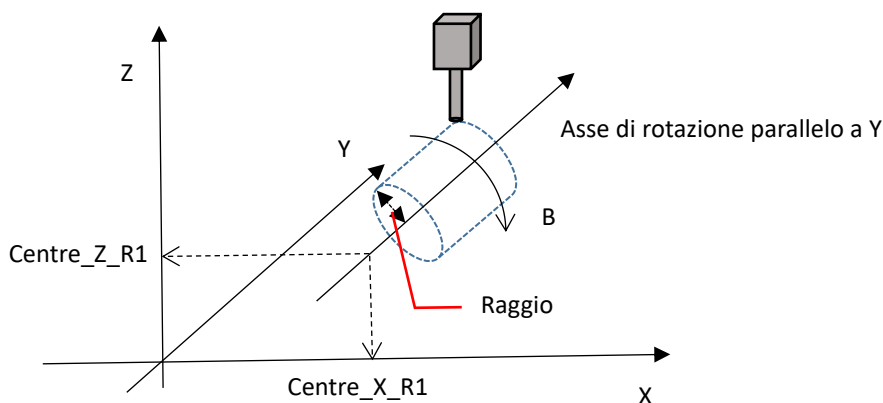
disabilita gestione

Nell’esempio, si imposta B come asse rotativo e Y come relativo asse di rotazione.

In questo caso il raggio “apparente” dell’asse rotativo è dato dalle posizioni X&Z rispetto al centro di rotazione impostato con i parametri [CENTRE X R1](#), [CENTRE Z R1](#).

Per chiarimenti sulla definizione della posizione dell’asse di rotazione, vedere la sezione relativa (vedi [Cap 17.34](#)).

La dichiarazione in esempio, da origine ad una gestione secondo questo schema. Gli altri assi, eventualmente presenti, non sono considerati nell’algoritmo.



17.22 FUNZIONI G DI UTILIZZO GENERICO

17.22.1 G4 – Pausa temporizzata

Sintassi

G4 Fvalore

Tipo funzione

immediata

Parametri

Valore *Tempo in sec. di pausa (risoluzione del centesimo di sec)*

Descrizione

G4 effettua una pausa del valore espresso nel parametro F. Il PartProgram viene arrestato per il tempo inserito.

ATTENZIONE

Il tempo di pausa può differire di una piccola percentuale del tempo impostato con tempi >= ad 1 sec
Per tempi minori di 1 Sec la percentuale aumenta.

Es:

G4F2.5 // PASUA DI 2.5 SEC

17.22.2 G4.1 – Addizionale Tempi su Calc Time

Sintassi

G4.1 Fvalore

Tipo funzione

immediata

Parametri

Valore *Tempo in sec di addizionale*

Descrizione

G4.1 Inserisce un addizionale del tempo per la funzione Calc Time (calcolo tempo di lavorazione Gcode). Questo serve per recuperare quei tempi che non possono essere previsti, tipo cicli PLC ecc. Normalmente viene inserite nelle varie M tipo M3,M4,M5 M6 ecc.

La funzione viene considerata solamente durante il calcolo tempo, ed esclusa al normale RUN

Es:

G4.1 F5.2 // Aggiorna 5,2 sec al calcolo tempo

17.22.3 G10 – Abilitazione potenziometro esterno di OVERRIDE assi

Sintassi

G10

Tipo funzione

modale - default

Revoca

G11

Descrizione

G10 abilita il controllo della Velocità degli assi da potenziometro esterno.

17.22.4 ***G11 – Disabilitazione potenziometro esterno di OVERRIDE assi******Sintassi*****G11****Tipo funzione***modale***Revoca***G10***Descrizione**

G11 disabilita il controllo della Velocità degli assi da potenziometro esterno.

può essere utile quando si devono effettuare spostamenti la cui Velocità non deve essere influenzata dal potenziometro di override.

17.22.5 G101 – Stop forzato degli assi**Sintassi****G101****Tipo funzione**

immediata

Descrizione

G101 comanda uno stop forzato degli assi nel punto attuale.

Nell' esempio sotto viene impostato un movimento all' asse X di 100.

Nella fase di movimento viene atteso un ingresso (I0) se questo si attiva termina il movimento in corso.

Viene anche controllato che se l' asse è arrivato in quota, il PartProgram salta ad una label di errore

Es:

```

G60
F1
G1X100
@INITLOOP
IF $[Q0]=100 // SE X=100
    GOTO ERR // ERRORE
END_IF
IF $[I0]=1 // ATTENDE SENSORE
    G101 // STOP ASSI
    GOTO EXIT
ELSE
    GOTO INITLOOP // CICLA
END_IF
@EXIT
.
.

@ERR
.
.

```

17.22.6 G80 – Pausa forzata da codice ISO

(Spostata su G1080 se USE_G80_CYCLES=TRUE)

Sintassi**G80 X**codice**Tipo funzione**

immediata

Parametri

| Codice | Valore numerico che indica il codice di pausa riferito alla configurazione |
|---------------|---|
|---------------|---|

Descrizione

G80 attiva una pausa forzata da codice ISO. Il parametro **CODICE** indica il messaggio associato alla PAUSA riferito ai codici PAUSA della configurazione di Isous. Se il codice non è stato configurato, viene visualizzato un messaggio "NO CODE PAUSE" ma la pausa procede comunque regolarmente. La PAUSA da codice segue tutte le procedure di una PAUSA da pulsante.

Es:

```
G80 X4 // ATTIVA LA PAUSA CON CODICE 4
```

17.22.7 G81 – Gestione LIMITI secondari ASSI

(Spostata su G1081 se USE_G80_CYCLES=TRUE)

Sintassi**G81** X_{mode}

Tipo funzione

immediata

Parametri

| | |
|-------------|---|
| <i>mode</i> | Valore numerico che indica il modo di gestione LIMITI |
| X0 | Reimposta i limiti PRIMARI positivi (default – LIMITE_P_) |
| X1 | Reimposta i limiti PRIMARI negativi (default – LIMITE_N_) |
| X2 | Imposta i limiti SECONDARI positivi (2ND_LIMITE_P_) |
| X3 | Imposta i limiti SECONDARI negativi (2ND_LIMITE_N_) |

Descrizione

Certe volte è necessario cambiare in modo temporaneo i limiti assi per poter effettuare alcune manovre (esempio Cambio Utensile) che sono al di fuori dei limitassi naturali. In Isous è possibile gestire i limiti assi SECONDARI positivi e negativi:

Parametri Macchina:

2ND_LIMITE_N_

2ND_LIMITE_P_

Questi possono essere attivati e disattivati dalla funzione G81. Nel caso in cui rimanessero attivati, al termine del PART PROGRAM, automaticamente vengono impostati i limiti PRIMARI.

Come si può notare, G81 è una funzione che spesso viene utilizzata nella MACRO DI CAMBIO UTENSILE (M6)

Es:

G81 X3 // ATTIVA I LIMITI SECONDARI NEGATIVI

.

G81 X1 // ATTIVA I LIMITI PRIMARI NEGATIVI**17.22.8 G20 – Programmazione quote assi in Pollici****Sintassi****G20**

Tipo funzione

modale

Revoca

G21**Descrizione**

G20 Abilita la programmazione delle quote Assi in Pollici.(G0-G1-G2-G3 ecc)

17.22.9 G21 – Programmazione quote assi in Millimetri**Sintassi****G21**

Tipo funzione

modale - Default

Revoca

G20**Descrizione**

G21 Abilita la programmazione delle quote Assi in Millimetri (G0-G1-G2-G3 ecc)

17.23 FUNZIONI PER GESTIONE VARIABILI PERMANENTI

Isous gestisce dei file che possono contenere lo stato di alcune variabili.

Questi file vengono salvati su HardDisk del PC e possono essere manipolati a piacere.

In pratica è possibile caricare questi file (uno alla volta) tramite Partprogram. Una volta caricato il file, viene riempita una lista contenente lo stato delle variabili precedentemente salvate. La lunghezza della lista dipende dal numero di variabili salvate. Non esiste in pratica una lunghezza massima di variabili salvate. Queste variabili possono essere lette e scritte nella lista sempre tramite istruzioni da PartProgram.

In pratica è possibile ottenere dei BackUp dello stato delle variabili desiderate per utilizzo generico.

17.23.1 **LOAD_VAR – carica un file di variabili precedentemente salvato**

Sintassi

LOAD_VAR nomefile

Parametri

Nomefile

Nome del file da caricare

Il nome può contenere l' estensione desiderata ma non può contenere SPAZI o caratteri di delimitazione (+- ecc)*

Descrizione

LOAD_VAR carica il file indicato inizializzando la lista interna con i valori contenuti nel FILE.

La lista si presenta come un ARRAY di valori double ai quali è possibile accedere tramite le funzioni **GET_VAR** e **WRITE_VAR**.

Viene generato un errore se il file non è disponibile.

Es:

```
LOAD_VAR MIOFILE.PNT // CARICA IL FILE
GET_VAR $VAR1 0 // CARICA VAR1 DALL' INDICE 0
GET_VAR $VAR2 1 // CARICA VAR2 DALL' INDICE 1
```

17.23.2 **GET_VAR – Legge una variabile dalla lista caricata**

Sintassi

GET_VAR \$var index

Parametri

\$var

Nome della variabile destinazione

Index

Espressione che indica l' indice sorgente della lista

Descrizione

GET_VAR Permette di leggere le variabili contenute nella lista.

L' accesso alle variabili è ottenuto tramite un indice che va da 0 a **MAXVAR-1**, dove **MAXVAR** è il numero di variabili contenute nella lista (vedi funzione **COUNT_VAR**).

Viene generato un errore se l' indice non è disponibile.

Es:

```
LOAD_VAR MIOFILE.PNT // CARICA IL FILE
$INDEX=0
LOOP 10
GET_VAR $VAR1 $INDEX // CARICA COORDINATA X
GET_VAR $VAR2 $INDEX+1 // CARICA COORDINATA Y
G1 X[$VAR1]Y[$VAR2] // MUOVE
$INDEX=$INDEX+2 // INCREMENTA INDICE
```

END_LOOP**17.23.3 WRITE_VAR – Scrive una variabile nella lista caricata****Sintassi****WRITE_VAR** \$var index**Parametri**

| | |
|--------------|--|
| \$var | Nome della variabile sorgente |
| index | Espressione che indica l' indice destinazione della lista |

ATTENZIONE la lista deve essere dimensionata per il parametro INDEX

Descrizione

WRITE_VAR Permette di scrivere le variabili della lista. La lista deve essere precedentemente inizializzata (tramite DIM_VAR) o comunque avere una dimensione tale da contenere il parametro INDEX.

Viene generato un errore se l' indice non è disponibile.

Es:

```

DIM_VAR 10           // DIMENSIONA LA LISTA CON 10 POSIZIONE VUOTE
WRITE_VAR $VAR1 0    // SALVA VAR1 IN POSIZIONE 0
WRITE_VAR $VAR1 1    // SALVA VAR2 IN POSIZIONE 1
SAVE_VAR MIOFILE.PNT // SALVA IL FILE

```

17.23.4 SAVE_VAR – salva un file di variabili dalla lista attuale**Sintassi****SAVE_VAR** nomefile**Parametri**

| | |
|-----------------|---------------------------------|
| Nomefile | Nome del file da salvare |
|-----------------|---------------------------------|

Il nome può contenere l' estensione desiderata ma non può contenere SPAZI o caratteri di delimitazione (+-* ecc)

Descrizione

SAVE_VAR salva il file indicato con i valori contenuti nella lista attuale

Es:

```

DIM_VAR 10           // DIMENSIONA LA LISTA CON 10 POSIZIONE VUOTE
WRITE_VAR $VAR1 0    // SALVA VAR1 IN POSIZIONE 0
WRITE_VAR $VAR1 1    // SALVA VAR2 IN POSIZIONE 1
SAVE_VAR MIOFILE.PNT // SALVA IL FILE

```

17.23.5 FILE_EXISTS – ritorna se il file indicato esiste**Sintassi****FILE_EXISTS** \$var nomefile**Parametri**

| | |
|-----------------|---|
| \$var | Variabile di ritorno del valore di interrogazione torna: 0 se il file non esiste - 1 se il file esiste |
| Nomefile | Nome del file da controllare |

Il nome può contenere l' estensione desiderata ma non può contenere SPAZI o caratteri di delimitazione (+-* ecc)

Descrizione

FILE_EXISTS permette di controllare l' esistenza di un file di variabili permanente

Es:

```

FILE_EXISTS $VAR MIOFILE.PNT // INTERROGAZIONE
IF $VAR=0                    // FILE NON ESISTE
....
END_IF

```

17.23.6 ADD_VAR – Aggiorna un valore alla lista attuale**Sintassi****ADD_VAR** valore**Parametri**

valore *Qualsiasi espressione o variabile da aggiungere alla lista*
Il valore viene addizionato in coda

Descrizione

ADD_VAR Permette di aggiungere dei valori alla lista attuale. La lista viene automaticamente dimensionata per contenere il nuovo valore.

Es:

```
ADD_VAR $VAR1            // ADDIZIONE VALORE
ADD_VAR 10.23           // ADDIZIONE VALORE
ADD_VAR $VAR2*($VAR3+$VAR4)    // ADDIZIONE VALORE
```

17.23.7 REMOVE_VAR – Rimuove un valore alla lista attuale**Sintassi****REMOVE_VAR** Index**Parametri**

index *Espressione che indica l' indice del valore da rimuovere*

Descrizione

REMOVE_VAR Permette di rimuovere dei valori dalla lista corrente.

Tutti gli indici al di sopra del valore rimosso verranno riassegnati in base alla nuova numerazione. Viene generato un errore se l' indice non è disponibile.

Es:

```
REMOVE_VAR 1            // RIMUOVE IL VALORE A INDICE 1
```

17.23.8 CLEAR_VAR – Rimuove tutti i valori dalla lista attuale**Sintassi****CLEAR_VAR****Descrizione**

CLEAR_VAR cancella tutta la lista attuale eliminando in modo permanente tutti i valori in essa contenuti. Il nuovo numero di variabili sarà quindi ZERO.

Es:

```
CLEAR_VAR                // CANCELLA TUTTA LA LISTA
```

17.23.9 DIM_VAR – Dimensiona la lista attuale al numero di elementi indicato**Sintassi****DIM_VAR** Nelementi**Parametri****Nelementi****Espressione che indica il numero di elementi dimensionati***(il valore degli elementi sarà ZERO)***Descrizione****DIM_VAR** Permette di ridimensionare la lista corrente ad un numero di elementi indicato nel parametro.

Tutti i valori in essa contenuti andranno persi e la lista assumerà la nuova dimensione indicata con tutti i relativi valori a ZERO.

Es:

DIM_VAR 10 // DIMENSIONA LA LISTA DI 10 ELEMENTI**17.23.10 COUNT_VAR – Ritorna il numero di elementi presenti nella lista****Sintassi****COUNT_VAR** \$var**Parametri****\$var****Variabile di ritorno del valore di interrogazione contiene il numero di elementi****Descrizione****COUNT_VAR** Permette di leggere il numero di elementi contenuti nella lista attuale

Es:

```

COUNT_VAR $VAR1 // LEGGE IN VAR1
IF $VAR1=0 // NESSUN ELEMENTO
    DIM_VAR 10 // DIMENSIONA LA LISTA
END_IF

```


17.24 FUNZIONI HM

Le funzioni HM rappresentano un'evoluzione delle funzioni semplici M. Queste possono utilizzare dei parametri che vengono passati durante la chiamata come le funzioni dei linguaggi più evoluti tipo C.

I parametri vengono poi prelevati dalla funzione HM inserendoli in delle variabili di tipo privato.

Le funzioni HM stanno residenti su PC e devono essere precedentemente create tramite l'utility dell'interfaccia operatore.

17.24.1 Chiamata di una funzione HM

Sintassi

HMnum par1 par2 par3

Tipo funzione

immediata

Parametri

Num Numero della funzione HM

PARn Variabile o costante. I parametri devono essere separati da uno SPAZIO

Descrizione

Viene invocata la funzione HM definita in **NUM** con i parametri definiti in **PARn**.

Le funzioni HM attendono sempre lo svuotamento del BUFFER MOVIMENTI (ASSI FERMI) prima di essere eseguite

Tutti i parametri, con i quali la funzione HM è stata dichiarata, devono essere passati altrimenti verrà generato un errore

Es:

HM10 \$VAR1 \$VAR2 12.3 24.125

// VIENE CHIAMATA HM 10 CON PARAMETRI

17.24.2 Costruzione di una funzione HM

Una funzione **HM** prima di essere utilizzata deve essere costruita e precompilata tramite l' opzione presente sull' interfaccia operatore.

Le funzioni **HM** sono costituite da codice ISO e pertanto fanno riferimento a tutte le istruzioni che questo supporta. Una volta precompilata, la funzione **HM** è utilizzabile all' interno del PartProgram.

Le **VARIABILI** e le eventuali **LABEL** che vengono dichiarate all' interno della funzione HM, sono di tipo **PRIVATO**, cioè visibili solamente dalla funzione stessa. Pertanto sia le **LABEL** che le **VARIABILI** possono avere lo stesso nome di quelle del PartProgram, in quanto Isous effettua un riconoscimento automatico.

è comunque possibile condividere delle VARIABILI con il Partprogram, dichiarando queste con il prefisso. **GLOBAL**. Da una funzione **HM** è possibile richiamare altre funzioni **HM** o **M** sia su CN che su PC.

Esempio di costruzione di un funzione HM

```

GET $PAR1 $PAR2 $PAR3 $PAR4 // LEGGE I PARAMETRI
GLOBAL $PAUSA // CONDIVIDE $PAUSA CON IL PART PROGRAM
F[$PAR3] // IMPOSTA LA FEED ASSI A $PAR3
G1 X[$PAR1] Y[$PAR2] Z[$PAR3] // INTERPOLAZIONE LINEARE
G4 F[$PAUSA] // PAUSA

```

Come è possibile vedere dall' esempio, una funzione HM è molto simile alle funzioni dei linguaggi più evoluti. Nel caso in cui la funzione HM debba tornare dei valori, questo è possibile utilizzando le variabili di tipo **GLOBAL**.

17.25 FUNZIONI M

Le funzioni M sono in parte simili alle funzioni HM. Queste si differenziano dal fatto che non possano avere uno scambio di parametri in modo diretto, ma possono comunque condividere le variabili di tipo **GLOBALE**.

Esistono due tipi di funzioni M:

- **Quelle interne al CN**
- **Quelle interne al PC**

Ovviamente le funzioni non possono avere lo stesso numero, Isous per prima controlla se la funzione **M** invocata è presente sul **PC**, in caso contrario richiama la funzione sul **CN**.

17.25.1 Funzioni M interne al CN

Le funzioni M interne al CN devono essere costruite con il compilatore VTB (vedi documentazione su applicazione VTB).

Queste possono ricevere dei parametri dal PartProgram o restituire dei valori a questo tramite le

VARIABILI PREDEFINITE

```

$_PARAM_1    parametro 1
$_PARAM_2    parametro 2
$_PARAM_3    parametro 3
.
.
$_PARAM_n    parametro n

```

Il numero di parametri che si possono scambiare dipende dalla configurazione, comunque è previsto un massimo di 10 parametri (default 5)

Ogni volta che il PartProgram scrive o legge una variabile di tipo \$_PARAM_n, questa viene trasferita o letta dalla memoria fisica del CN. Occorre porre particolare attenzione a questo tipo di parametri, in quanto questi sono solamente di tipo **INTERO** (non possono contenere una parte decimale in quanto il CN non la considera)

I parametri devono essere scritti prima della chiamata della funzione M e devono essere rilette dopo la fine della funzione M.

Le funzioni M attendono sempre lo svuotamento del BUFFER MOVIMENTI (ASSI FERMI) prima di essere eseguite

Es:

```

$_PARAM_1=134 // SCRIVE I PARAMETRO 1
$_PARAM_2=$VAR// SCRIVE I PARAMETRO 2
$_PARAM_3=12600 // SCRIVE I PARAMETRO 3
M100 // INVOCA LA FUNZIONE M100
$VAR1=$_PARAM_1 // RILEGGE IL PARAMETRO 1 MODIFICATO DALLA M100

```

17.25.2 Funzioni M interne al PC

Le funzioni **M** interne al PC sono del tutto simili alle funzioni **HM**, con la differenza che queste non possono avere parametri, pertanto lo scambio di questi deve essere attuato tramite le variabili di tipo **GLOBAL**.

Una funzione **M** prima di essere utilizzata deve essere costruita e precompilata tramite l'opzione presente sull'interfaccia operatore.

Le funzioni **M** sono costituite da codice ISO e pertanto fanno riferimento a tutte le istruzioni che questo supporta. Una volta precompilata, la funzione **M** è utilizzabile all'interno del PartProgram.

Le **VARIABILI** e le eventuali **LABEL** che vengono dichiarate all'interno della funzione **M**, sono di tipo **PRIVATO**, cioè visibili solamente dalla funzione stessa. Pertanto sia le **LABEL** che le **VARIABILI** possono avere lo stesso nome di quelle del PartProgram, in quanto Isous effettua un riconoscimento automatico.

È comunque possibile condividere delle **VARIABILI** con il Partprogram, dichiarando queste con il prefisso **GLOBAL**.

Da una funzione **M** è possibile richiamare altre funzioni **HM** o **M** sia su **CN** che su **PC**.

Esempio di costruzione di un funzione M

```
GLOBAL $VAR1           // CONDIVIDE $VAR1 CON IL PART PROGRAM
GLOBAL $VAR2           // CONDIVIDE $VAR2 CON IL PART PROGRAM
G1 X$VAR1 Y$VAR2       // INTERPOLAZIONE LINEARE
```

17.26 Configurazioni M principali

In questa sezione vengono descritte le varie M principali e la loro configurazione. Queste funzioni M sono necessarie al corretto funzionamento di della macchina.

17.26.1 M Di START

Questa funzione M viene eseguita ad ogni START PROGRAM. La sua programmazione non è importante e spesso viene disabilitata, in quanto le funzioni M preparatorie sono inserite sempre nel file ISO.

17.26.2 M Di END

Questa funzione M viene eseguita ad ogni END PROGRAM. La sua programmazione non è importante e spesso viene disabilitata, in quanto le funzioni M preparatorie sono inserite sempre nel file ISO.

17.26.3 M Di STOP

La funzione M STOP viene eseguita ogni volta che viene premuto il pulsante di STOP PROGRAMMA. Generalmente la sua configurazione prevede uno STOP IMMEDIATO degli assi e lo spegnimento delle varie utenze (mandrino, acqua ecc.)

Funzioni essenziali da inserire nella M di STOP nella giusta sequenza

- 1) G101 Stop assi
- 2) G62 Attesa assi fermi
- 3) Gestione delle varie utenze
- 4) Eventuale parcheggio assi a posizioni predefinite

Se nessuna M di STOP viene configurata, il CNC blocca solamente gli assi nel punto di STOP.

17.26.4 M Di PAUSA

La funzione M di PAUSA viene eseguita ogni volta che viene premuto il pulsante PAUSA. Questa funzione si preoccupa di gestire le varie utenze e predisporre il riposizionamento degli assi nel giusto punto di interruzione.

Funzioni essenziali da inserire nella M di PAUSA nella giusta sequenza (es 3 assi)

- 1) GLOBAL \$SAVEX Dichiarazioni variabile GLOBALE per ASSE X
- 2) GLOBAL \$SAVEY Dichiarazioni variabile GLOBALE per ASSE Y
- 3) GLOBAL \$SAVEZ Dichiarazioni variabile GLOBALE per ASSE Z
- 4) G62 Attesa assi fermi
- 5) \$SAVEX=\$[Q0] Salva quota asse X
- 6) \$SAVEY=\$[Q1] Salva quota asse Y
- 7) \$SAVEZ=\$[Q2] Salva quota asse Z
- 8) Gestione varie utenze
- 9) Eventuale parcheggio assi a posizioni predefinite

Se nessuna M di pausa viene configurata, il CNC si ferma sul punto in esecuzione

17.26.5 M Di RIPRESA DA PAUSA

La funzione M di Ripresa da PAUSA viene eseguita ogni volta che viene premuto il pulsante START dopo che il CNC è in PAUSA. Questa predispose la macchina per la ripresa della lavorazione.

Funzioni essenziali da inserire nella M di RIPRESA DA PAUSA nella giusta sequenza (es 3 assi)

- 1) GLOBAL \$SAVEX Dichiarazioni variabile GLOBALE per ASSE X
- 2) GLOBAL \$SAVEY Dichiarazioni variabile GLOBALE per ASSE Y
- 3) GLOBAL \$SAVEZ Dichiarazioni variabile GLOBALE per ASSE Z
- 4) G96 Sospende offset pezzo
- 5) G98 Sospende zero pezzo
- 6) G0 Z0 Asse Z verticale a quota di sicurezza
- 7) F1 Imposta F di posizionamento
- 8) G1 X[\$SAVEX] Y[\$SAVEY] posizionamento X Y ecc. a quote inizio (salvate da M di pause)
- 9) G62 Attesa assi fermi
- 10) G1 Z[\$SAVEZ] posizionamento Z (salvata da M di pausa)
- 11) G62 Attesa assi fermi
- 12) G97 Riprende Offset
- 13) G99 Riprende Zero
- 14) Accensione utenze varie

Se nessuna M di ripresa da pausa viene configurata, il CNC riporta gli assi al punto di inizio in interpolazione ad una F impostata nel parametro macchina "VRIPOS"

17.26.6 M Di RIPRESA DA BLOCCO

La funzione M di Ripresa da BLOCCO viene eseguita ogni volta che viene premuto il pulsante START del PLUG IN "GO BLOCK". Questa predispose la macchina per la ripresa della lavorazione.

Funzioni essenziali da inserire nella M di RIPRESA DA BLOCCO nella giusta sequenza (es 3 assi)

- 1) \$PO SX=\$[C0] Riprende quota contatore asse X
- 2) \$PO SY=\$[C1] Riprende quota contatore asse Y
- 3) \$PO SZ=\$[C2] Riprende quota contatore asse Z
- 4) G96 Sospende offset pezzo
- 5) G98 Sospende zero pezzo
- 6) G0 Z[0] Asse Z verticale a quota di sicurezza
- 7) F1 Imposta F di posizionamento
- 8) G1 X[\$PO SX] Y[\$PO SY] Posizionamento X Y ecc. a quote inizio
- 9) G62 Attesa assi fermi
- 10) G1 Z[\$PO SZ] Posizionamento Z
- 11) G62 Attesa assi fermi
- 12) G97 Riprende Offset
- 13) G99 Riprende Zero
- 14) Accensione utenze varie

E' necessario configurare sempre una M di ripresa da BLOCCO (se utilizzato il PlugIn)

17.26.7 M Di RIPRESA DA RETRACE

La funzione M di Ripresa da RETRACE viene eseguita ogni volta che viene premuto il pulsante START del PLUG IN "RETRACE". Questa predispose la macchina per la ripresa della lavorazione.

Funzioni essenziali da inserire nella M di RIPRESA DA RETRACE nella giusta sequenza accensione utenze varie

E' necessario configurare sempre una M di ripresa da RETRACE (se utilizzato il PlugIn)

17.27 DEFINIZIONE ASSE DI PROFONDITA'

L'asse di profondità è utilizzato per alcune funzioni di Isous. Normalmente tale asse è relativo al piano di lavoro selezionato:

| | |
|-----------------|-----------------------------|
| Piano XY | Asse di profondità Z |
| Piano XZ | Asse di profondità Y |
| Piano YZ | Asse di profondità X |

Nel caso in cui l'asse di profondità sia diverso da quelli scelti in automatico da Isous, occorre ridefinirlo con la funzione G48.

L'asse di profondità viene utilizzato dalla SIMULAZIONE PREVIEW della lavorazione e dal modo di compensazione G47.

17.27.1 G48 – Definisce asse di profondità**Sintassi**

G48 nome Asse

Tipo funzione

immediata

Parametri

Nome Asse

Nome dell'asse X,Y,Z,A,B ecc.

Descrizione

G48 definisce un asse di profondità diverso da quelli scelti in automatico da Isous.

Se l'asse scelto è uguale ad uno degli assi del piano di lavoro impostato, viene generato un errore di Run Time.

Es:

G48 A // ATTIVA L'ASSE A COME ASSE DI PROFONDITA'

17.28 MILD MODE – Smorzamento Spigoli

Isous può utilizzare uno speciale algoritmo indicato per SMORZAMENTO DEGLI SPIGOLI.

Tramite i PARAMETRI MILD_ (MILD_X – MILD_Y ecc.) gli spigoli possono essere livellati a piacere.

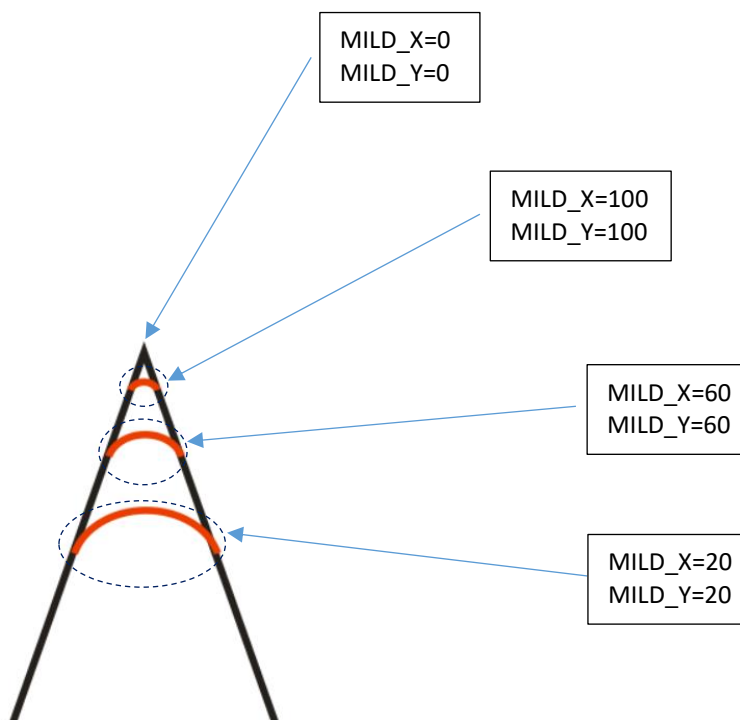
Il modo MILD può agire solamente sugli assi indicati.

Più basso è il valore del parametro MILD_, maggiore è lo SMORZAMENTO DELL' ASSE.

Valori indicati vanno da 200 a 20. La gestione del MILD MODE viene effettuata dal codice G49

Di seguito è riportato un riferimento di smorzamento di uno spigolo.

Il MILD agisce all' interno della soglia spigolo indicata per profili 2D da **SGLP_MILD** e per profili 3D (G65) da **SGL3D_MILD_**. Spigoli al di sopra di questi valori non sono smorzati dal MILD



17.28.1 G49 – Gestione MILD MODE

Sintassi

G49 nomi Assi dove abilitare il MILD MODE – Nessun asse indicati disabilita MILD MODE su tutti gli ASSI

G49.0 Sospende MILD MODE

G49.1 Riprende MILD MODE sugli assi precedentemente abilitati da G49

Tipo funzione

Immediata – disabilitata allo STOP PROGRAMMA

Parametri

Nomi Assi

Nome degli assi es: **G49 XY (QX,QY,QZ,QA,QB,QC,QU,QV,QW)**

Descrizione

G49 gestisce l' abilitazione o meno del MILD MODE.

G49 con il **nome degli assi** (es. G49 XY) viene attivato MILD MODE sugli assi indicati.

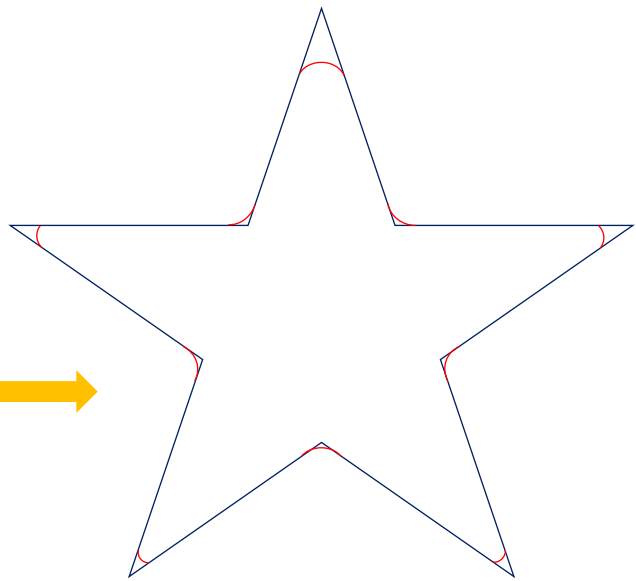
G49 senza alcun parametro, MILD MODE viene disattivato su tutti gli ASSI.

G49.0 senza parametri aggiuntivi **SOSPENDE MILD MODE** (utile nelle funzione M)

G49.1 senza parametri aggiuntivi **RIPRENDE MILD MODE** se questo era ATTIVATO da **G49 Nome Assi**.

Es:

```
G60
F15
G0 Z3
G49 XY // ABILITA MILD MODE SU XY
G0X63.7738 Y160.6052
F10
G01 Z0
X70.9693 Y140.2459 F15
X92.5556 Y139.694
X75.4163 Y126.5593
X81.5619 Y105.8589
X63.7738 Y118.1006
X45.9857 Y105.8589
X52.1313 Y126.5593
X34.992 Y139.694
X56.5783 Y140.2459
X63.7738 Y160.6052
G49 // DISABILITA MILD MODE
G0 Z3
G0 X56.5783 Y140.2459
```



17.29 TASK Paralleli

IsoUs può eseguire due task di Gcode che girano in modo parallelo all' esecuzione del Gcode principale. Questo permette di eseguire funzioni in modo asincrono e aumentare le potenzialità della macchina. I task paralleli **TASK1** e **TASK2** possono essere eseguiti in diverse modalità, e comunque non possono gestire tutte le funzionalità del Gcode principale (vedere singole istruzioni). Esistono 3 modi per eseguire il codice nei task paralleli.

17.29.1 USTASK - ENDUSTASK

Queste due istruzioni, definiscono un codice che viene caricato nel TASK1 in modo automatico dal compilatore. Queste DEVONO essere scritte nel Gcode PRINCIPALE

USTASK

```
LOOP 1000
  $[O1]=1
  G4F1
  $[O1]=0
  G4F1
```

END_LOOP

ENDUSTASK

TASK.RUN 1

```
LOOP 1000
  G1X100Y100F5
  G4F1
  XOY0
  G4F1
END_LOOP
```

Il codice racchiuso tra USTASK e ENDUSTASK viene caricato ma non eseguito al momento del RUN nel TASK1, mentre il codice esterno viene caricato ed eseguito al momento del RUN nel task principale.

L' esecuzione del codice del TASK1 è affidata in questo caso al TASK principale tramite l' istruzione **TASK.RUN 1** che mette in esecuzione il TASK1.

17.29.2 TASK.RUN

Avvia l' esecuzione del codice precedentemente caricato nel TASK indicato

Sintassi

TASK.RUN Ntask

Ntask Numero del TASK dove:

- 0** Task Principale (non usato)
- 1** TASK1
- 2** TASK2

Un task non può avviare se stesso

17.29.3 TASK.STOP

Interrompe l' esecuzione del codice precedentemente avviato nel TASK indicato

Sintassi

TASK.STOP Ntask

Ntask Numero del TASK dove:

- 0** Task Principale (non usato)
- 1** TASK1
- 2** TASK2

Un task non può interrompere se stesso

17.29.4 TASK.PAUSE

Mette in PAUSA l' esecuzione del codice precedentemente avviato nel TASK indicato

Sintassi**TASK.PAUSE Ntask**

Ntask Numero del TASK dove:
0 Task Principale (non usato)
1 TASK1
2 TASK2

Un task non può mettere in pausa se stesso

17.29.5 TASK.READVAR

Legge una variabile per indirizzo nel TASK indicato

Sintassi**TASK.READVAR Ntask AddrVar \$VARDEST**

Ntask Numero del TASK dove:
0 Task Principale
1 TASK1
2 TASK2
AddrVar Indirizzo della variabile da leggere nel TASK
\$VARDEST Variabile di destinazione

Es:

TASK.READVAR 1 2000 \$VAR // LEGGE LA VARIABILE 2000 DEL TASK 1 E LA DEPOSITA IN \$VAR

17.29.6 TASK.WRITEVAR

Scrive una variabile per indirizzo nel TASK indicato

Sintassi**TASK.WRITEVAR Ntask AddrVar \$VARSOURCE**

Ntask Numero del TASK dove:
0 Task Principale
1 TASK1
2 TASK2
AddrVar Indirizzo della variabile da scrivere nel TASK
\$VARSOURCE Variabile sorgente

Es:

TASK.WRITEVAR 1 2000 \$VAR // SCRIVE IL CONTENUTO DI \$VAR NELLA VARIABILE 2000 DEL TASK 1

17.29.7 TASK.STATUS

Legge lo stato del TASK indicato

Sintassi**TASK.STATUS Ntask \$VAR**

Ntask Numero del TASK dove:
0 Task Principale
1 TASK1
2 TASK2
\$VAR Variabile di deposito dello stato
0 Task Stop
1 Task Run
2 Task Pausa

Es:

TASK.STATUS 1 \$VAR // LEGGE LO STATO DEL TASK 1 E LO DEPOSITA IN \$VAR

17.29.8 TASK.LOADCMD

Carica un file CMD nel TASK indicato.

Il file CMD deve essere precedentemente creato con il PlugIn MHM

Il CMD non viene automaticamente eseguito ma è necessario utilizzare TASK.RUN

Sintassi**TASK.LOADCMD Ntask "CMDNAME"**

Ntask Numero del TASK dove:

- 0** Task Principale (non Usato)
- 1** TASK1
- 2** TASK2

"CMDNAME" Nome del file CMD (tra virgolette e senza estensione) da caricare ne TASK

Es:

TASK.LOADCMD 1 "TESTCMD" // CARICA IL CMD TESTCMD NEL TASK1

17.29.9 TASK.PRIORITY

Setta la priorità per il task indicato

Priorità più basse (valori alti di Pval) danno meno tempo di esecuzione al task.

Sintassi**TASK.PRIORITY Ntask Pval**

Ntask Numero del TASK dove:

- 0** Task Principale (non Usato)
- 1** TASK1
- 2** TASK2

Pval Priorità del TASK da 0 a 100

0 Priorità massima (default)

Es:

TASK. PRIORITY 1 2 // SETTA LA PRIORITA 2 AL TASK 1

17.30 PARAMETRI MACCHINA ESSENZIALI

In questo capitolo vengono descritti i parametri macchina ESSENZIALI che devono essere impostati in Isous per configurare la MACCHINA.

I parametri MACCHINA ESSENZIALI sono sempre presenti su qualsiasi applicazione, è comunque possibile che le applicazioni condividano più PARAMETRI (fare riferimento al manuale della macchina), in quanto Isous permette di personalizzare il parametri per adattare la macchina ad un corretto funzionamento.

NOTA PER CONVERSIONE ACCELERAZIONE DA UNITÀ ISOUS IN Mt/sec²

Le accelerazioni in Isous sono espresse in incremento per campionamento del CN, pertanto se si vuole ottenere il risultato in Mt/sec² è sufficiente applicare la seguente trasformazione:

$$ACCms = ACC(impostata) / TAU^2$$

dove TAU è il campionamento impostato sul CN espresso i millisecondi

pertanto se impostiamo un ACC=100 e il campionamento TAU=5 (millisecondi):

$$ACCms = 100 / (5 * 5) = 4 \text{ Mt/sec}^2$$

17.30.1 Parametri generali

17.30.1.1 FEEDMAX

Definisce la Velocità massima FEED accettata da sistema. La sua unità di misura dipende dalla risoluzione utilizzata (normalmente mm/min). Se impostata tramite F una Velocità superiore a questa, viene limitata a FEEDMAX.

Questo parametro è soggetto alle unità di misura impostate. Pertanto se lavoriamo i decimillesimi di millimetro, questo parametro va espresso non in mm/min ma in decimi/min.

17.30.1.2 FEEDMIN

Definisce la Velocità minima FEED accettata da sistema. La sua unità di misura dipende dalla risoluzione utilizzata (normalmente mm/min). Se impostata tramite F una Velocità inferiore a questa, viene limitata a FEEDMIN.

Questo parametro è soggetto alle unità di misura impostate. Pertanto se lavoriamo i decimillesimi di millimetro, questo parametro va espresso non in mm/min ma in decimi/min.

17.30.1.3 FEEDDEF

Definisce la Velocità di default impostata nel sistema. La sua unità di misura dipende dalla risoluzione utilizzata (normalmente mm/min).

Questo parametro è soggetto alle unità di misura impostate. Pertanto se lavoriamo i decimillesimi di millimetro, questo parametro va espresso non in mm/min ma in decimi/min.

17.30.1.4 FEEDRES

Definisce la risoluzione minima della Velocità impostata con F.

| | | |
|------|--------|------------------------|
| 1 | mm/min | es: F1500 = 1,5 Mt/Min |
| 1000 | mt/min | es: F1.5 = 1,5 Mt/min |

17.30.1.5 SPEEDMAX

Definisce la Velocità massima SPEED accettata da sistema. La sua unità di misura dipende dalla risoluzione utilizzata Se impostata tramite S una Velocità superiore a questa, viene limitata a SPEEDMAX.

17.30.1.6 SPEEDMIN

Definisce la Velocità minima SPEED accettata da sistema. La sua unità di misura dipende dalla risoluzione utilizzata Se impostata tramite S una Velocità inferiore a questa, viene limitata a SPEEDMIN.

17.30.1.7 SPEEDDEF

Definisce la SPEED' di default impostata nel sistema. La sua unità di misura dipende dalla risoluzione utilizzata.

17.30.1.8 VMAXGO

Definisce la Velocità massima ammessa per gli spostamenti in G0 in percentuale alla Velocità calcolata dal sistema.

Valori ammessi da 0 al 100 %. Questo parametro riduce in modo percentuale la Velocità calcolata dal sistema per spostamenti in G0.

Questo parametro è soggetto alle unità di misura impostate. Pertanto se lavoriamo i decimillesimi di millimetro, questo parametro va espresso non in mm/min ma in decimi/min.

17.30.1.9 ACC_G0

Definisce l' accelerazione per spostamenti in G0. L' unità di misura è in count macchina, quindi dipende dall' impostazione del sistema.

17.30.1.10 ACC_G0.1_

Definisce l' accelerazione per spostamenti in G0.1 del singolo asse. L' unità di misura è in count macchina, quindi dipende dall' impostazione del sistema.

17.30.1.11 ACC_MAX_

Definisce l' accelerazione massima consentita per l' asse utilizzata dalla funzione G66 X-100 per riduzione velocità sui punti.

17.30.1.12 ACC_LAV

Definisce l' accelerazione per spostamenti in in G1-G2-G3. L' unità di misura è in count macchina, quindi dipende dall' impostazione del sistema.

17.30.1.13 ACC_QSTOP

Definisce l' accelerazione per quick stop cioè quando viene intercettato un limite asse

17.30.1.14 VEL_GO_LINE_RETRACE

Definisce la Velocità per spostamenti GO LINE durante la funzione **RETRACE**. La sua unità di misura dipende dalla risoluzione utilizzata (normalmente mm/min).

Questo parametro è soggetto alle unità di misura impostate. Pertanto se lavoriamo i decimillesimi di millimetro, questo parametro va espresso non in mm/min ma in decimi/min.

17.30.1.15 ACC_RAGGIO_MAX

Definisce l' accelerazione centrifuga per autoriduzione Velocità su ARCHI. Il parametro va impostato in base alla meccanica della macchina.

Normalmente questo parametro ha dei valori compresi tra 1 e 15.

Valori più bassi definiscono una riduzione maggiore della velocità in RAGGI più piccoli.

Se il valore 1 non è sufficiente a ridurre le velocità in modo desiderato, si può passare all' utilizzo dei Decimali nel seguente modo:

Inserire il valore ACC_RAGGIO_MAX superiore a 100, per avere un decimale in più

Inserire il valore ACC_RAGGIO_MAX superiore a 1000, per avere due decimale in più

in questo modo il valore effettivo è dato:

ACC_RAGGIO_MAX=(ACC_RAGGIO_MAX-100)/10 (per valori superiori a 100)

ACC_RAGGIO_MAX=(ACC_RAGGIO_MAX-1000)/100 (per valori superiori a 1000)

Pertanto valori come:

109 = 0,9

1018 = 0,18

Ecc.

17.30.1.16**RED_ACC_RADIUS**

Questo parametro se >0 permette di ottenere un ulteriore riduzione del parametro **ACC_RAGGIO_MAX**. In pratica indica il raggio minimo espresso in millesimi di millimetro sotto al quale viene ridotto in modo proporzionale il parametro **ACC_RAGGIO_MAX**:

$$\text{ACC_RAGGIO_MAX} = (\text{RADIUS} / \text{RED_ACC_RADIUS}) * \text{ACC_RAGGIO_MAX}$$

Es:

$$\text{RED_ACC_RADIUS} = 3000$$

$$\text{ACC_RAGGIO_MAX} = 5$$

$$\text{Radius} = 2000 (< 3000)$$

$$\text{ACC_RAGGIO_MAX} = (2000 / 3000) * 5 = 3.33$$

17.30.1.17**MIN_ACC_RADIUS**

Indica il valore minimo di riduzione di **ACC_RAGGIO_MAX** tramite **RED_ACC_RADIUS**.

Valori al di sotto di questo non possono scendere.

MIN_ACC_RADIUS segue lo stesso principio di **VMAX**, cioè:

$$\text{MIN_ACC_RADIUS} = (\text{MIN_ACC_RADIUS} - 100) / 10 \quad (\text{per valori superiori a } 100)$$

$$\text{MIN_ACC_RADIUS} = (\text{MIN_ACC_RADIUS} - 1000) / 100 \quad (\text{per valori superiori a } 1000)$$

Valore=0 disabilitato

17.30.1.18**VRIPOSO**

Velocità assi di riposizionamento da PAUSA. Valore espresso in unità di sistema (normalmente mm/min). Nel caso in cui dopo in PAUSA gli assi vengano mossi e non riposizionati nel punto corretto dalla MACRO DI RIPRESA DA PAUSA, il CN riporta automaticamente gli assi nel punto di interruzione con questa Velocità.

17.30.1.19**SGLP**

Soglia in decimi di grado (default 200 – venti gradi) per calcolo spigolo per fermata assi in INTERPOLAZIONE VELOCE. La soglia agisce solo su gli assi del piano di lavoro.

Il CN ferma automaticamente gli assi quando tra la coppia di ASSI del piano di lavoro viene formato uno spigolo MAGGIORE al parametro SGLP.

17.30.1.20**SGLP_RED**

Percentuale di SGLP per iniziare il rallentamento graduale in base alla proporzione dello spigolo. Se **SGLP_RED=0** il parametro è disabilitato e viene effettuata solo la fermata sullo spigolo SGLP, altrimenti dalla percentuale impostata, il CNC inizia un rallentamento (vedi esempi sotto). Quando è impostata **SGLP_RED** ad un valore maggiore di 0, sarebbe opportuno aumentare il valore di SGLP.

17.30.1.21 **MAX_RED**

Massima Percentuale di riduzione della velocità causata dal parametro SGLP_RED. Questa viene limitata sempre ad un valore sotto il 100% (per evitare che la velocità vada a ZERO) (vedi esempi sotto)

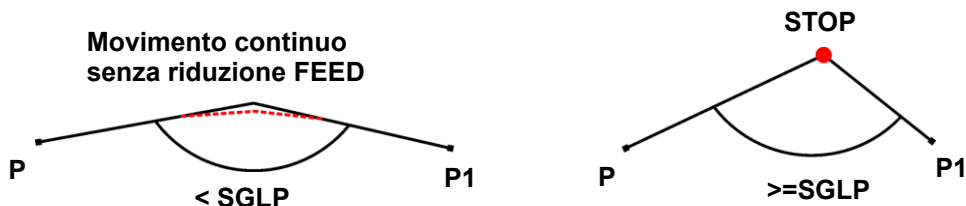
Esempi parametri SGLP_RED e MAX_RED

Se SGLP_RED è disabilitato (uguale a ZERO) entra in funzione solo il parametro SGLP o SGL3D nel seguente modo:

Angoli inferiori a **SGLP** o **SGL3D** il CNC prosegue senza fermarsi

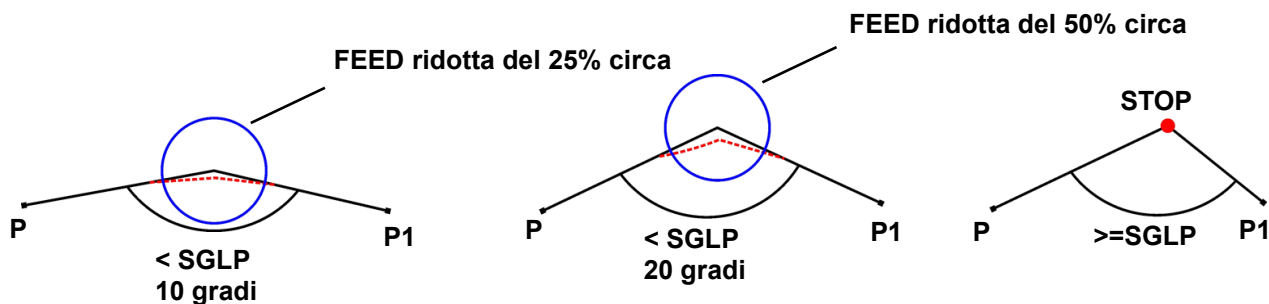
Angoli superiori a **SGLP** o **SGL3D** il CNC decelera e si ferma sul tratto

Es: **SGLP=200 (20 gradi)** e **SGLP_RED=0**



Es: **SGLP=450 (45 gradi)** e **SGLP_RED=10 (al 10% di 45 gradi – 4,5 gradi inizia a rallentare)**

MAX_RED=85

**17.30.1.22** **SGLR**

Soglia in unità di misura programmate (millesimi,decimillesimi ecc.) per errore su interpolazione circolare (default 10 millesimi).

Questo errore si verifica a livello di CN, in pratica viene effettuato un controllo sul RAGGIO nel punto iniziale e nel punto finale, Se la differenza supera questa soglia viene generato un errore. Questo può verificarsi esclusivamente se vengono passate le coordinate del centro I,J in modo non PRECISO.

Impostare questo parametro in base all' unità di misura scelta es:

Unità di misura in millesimi **10 = 10 millesimi di millimetro**

Unità di misura in decimillesimi **100= 10 millesimi di millimetro**

17.30.1.23 **ACQ_MODE**

Definisce il modo di acquisizione del **SENSORE** per la funzione **G102** o per il metodo **StartAcqSens** del Framework.

17.30.1.24 **ACQ_VEL**

Definisce la velocità per lo spostamento assi nella fase di acquisizione a sensore. Valore espresso in unità di sistema (normalmente mm/min).

Questo parametro NON è soggetto alle unità di misura impostate. Pertanto il suo valore deve essere sempre espresso in mm/min

17.30.1.25 **RESQUOTE**

Risoluzione utilizzata per quote assi. Dipendente dal parametro DSOF-asse.

Definisce l' unità minima per spostamento asse

10 → *Decimi di millimetro*
100 → *Centesimi di millimetro*
1000 → *Millesimi di millimetro*
10000 → *Decimillesimi di millimetro*
100000 → *Centimillesimi di millimetro*

17.30.1.26 **VISUAREAL**

Abilita o disabilita la visualizzazione delle quote REALI degli assi

-1 = *Lettura quote reali disabilitate*
0 = *Abilitazione da INTERFACCIA*
1 = *Lettura quote reali abilitate – QUOTA ASSE*
2 = *Lettura quote reali abilitate – ERRORE DI SPAZIO*

17.30.1.27 **ENABLE_OW_GO**

Abilita o disabilita il potenziometro di override sulle G0

NO = *Potenzimetro disabilitato*
SI = *Potenzimetro abilitato*
NO GO ACK = *Potenzimetro disabilitato con segnalazione uso G0 al CN (per app VTB)*
SI GO ACK = *Potenzimetro abilitato con segnalazione uso G0 al CN (per app VTB)*

17.30.2 **Parametri relativi all' asse**

17.30.2.1 **VMAX_**

Definisce la Velocità massima del singolo asse per spostamenti in G0. La sua unità di misura dipende dalla risoluzione utilizzata (normalmente mm/min)

Questo parametro è soggetto alle unità di misura impostate. Pertanto se lavoriamo i decimillesimi di millimetro, questo parametro va espresso non in mm/min ma in decimi/min.

17.30.2.2 **VOVERLIMIT_**

Definisce la Velocità limite per singolo asse utilizzata in G66 X-100 durante la fase di controllo limite di velocità per RTCP o Assi tangenziali. L' unità di misura segue quella di VMAX

In pratica questo valore di velocità limita la FEED di avanzamento negli assi gestiti da RTCP a negli assi tangenziali se questi superano il valore indicato in VOVERLIMIT.

17.30.2.3 **VMAX_Pn**

Definisce la Velocità massima per il posizionatore indicato

17.30.2.4 **VJOG**

Definisce la Velocità di JOG del singolo asse per spostamenti in manuale. La sua unità di misura dipende dalla risoluzione utilizzata (normalmente mm/min).

Questo parametro NON è soggetto alle unità di misura impostate. Pertanto il suo valore deve essere sempre espresso in mm/min

17.30.2.5 **ACC_JOG**

Accelerazione per spostamenti in JOG

L' unità di misura è in count macchina, quindi dipende dall' impostazione del sistema.

17.30.2.6 RZERO_MODE

La ricerca di zero asse (homing asse) viene eseguita in base al valore di questo parametro.
Questo è valido solamente se non vengono utilizzate le procedure di esecuzioni di homing interne ai driver.

Metodi di ricerca ammessi:

RZERO_MODE=0 - Indietro negativo sensore ATTIVO

- 1) Indietro veloce fino al sensore di ZERO con **VEL=RZERO_VEL**
- 2) Avanti lento per uscire dal sensore con **VEL=RZERO_VELF**
- 3) Indietro lento fino al sensore con **VEL=RZERO_VELF**
- 4) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 5) La quota asse assume il valore indicato in **RZERO_PRESET**

RZERO_MODE=1 - Avanti positivo sensore ATTIVO

- 1) Avanti veloce fino al sensore di ZERO con **VEL=RZERO_VEL**
- 2) Indietro lento per uscire dal sensore con **VEL=RZERO_VELF**
- 3) Avanti lento fino al sensore con **VEL=RZERO_VELF**
- 4) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 5) La quota asse assume il valore indicato in **RZERO_PRESET**

RZERO_MODE=2 - Indietro negativo sensore ATTIVO con ricerca tacca di zero encoder

- 1) Indietro veloce fino al sensore di ZERO con **VEL=RZERO_VEL**
- 2) Avanti lento per uscire dal sensore con **VEL=RZERO_VELF**
- 3) Indietro lento fino al sensore con **VEL=RZERO_VELF**
- 4) Proseguimento fino alla prima TACCA trovata con **VEL=RZERO_VELF**
- 5) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 6) La quota asse assume il valore indicato in **RZERO_PRESET**

RZERO_MODE=3 - Avanti positivo sensore ATTIVO con ricerca tacca di zero encoder

- 1) Avanti veloce fino al sensore di ZERO con **VEL=RZERO_VEL**
- 2) Indietro lento per uscire dal sensore con **VEL=RZERO_VELF**
- 3) Avanti lento fino al sensore con **VEL=RZERO_VELF**
- 4) Proseguimento fino alla prima TACCA trovata con **VEL=RZERO_VELF**
- 5) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 6) La quota asse assume il valore indicato in **RZERO_PRESET**

RZERO_MODE=4 - Indietro negativo sensore DISATTIVO

- 1) Indietro veloce fino al sensore di ZERO con **VEL=RZERO_VEL**
- 2) Avanti lento per uscire dal sensore con **VEL=RZERO_VELF**
- 3) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 4) La quota asse assume il valore indicato in **RZERO_PRESET**

RZERO_MODE=5 - Avanti positivo sensore DISATTIVO

- 1) Avanti veloce fino al sensore di ZERO con **VEL=RZERO_VEL**
- 2) Indietro lento per uscire dal sensore con **VEL=RZERO_VELF**
- 3) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 4) La quota asse assume il valore indicato in **RZERO_PRESET**

RZERO_MODE=6 - Indietro negativo sensore DISATTIVO con ricerca tacca di zero encoder

- 1) Indietro veloce fino al sensore di ZERO con **VEL=RZERO_VEL**
- 2) Avanti lento per uscire dal sensore con **VEL=RZERO_VELF**
- 3) Proseguimento fino alla prima TACCA trovata con **VEL=RZERO_VELF**
- 4) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 5) La quota asse assume il valore indicato in **RZERO_PRESET**

RZERO_MODE=7 - Avanti positivo sensore DISATTIVO con ricerca tacca di zero encoder

- 1) Avanti veloce fino al sensore di ZERO con **VEL=RZERO_VEL**
- 2) Indietro lento per uscire dal sensore con **VEL=RZERO_VELF**
- 3) Proseguimento fino alla prima TACCA trovata con **VEL=RZERO_VELF**
- 4) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 5) La quota asse assume il valore indicato in **RZERO_PRESET**

RZERO_MODE=8 - Indietro con solo ricerca tacca di zero encoder

- 1) *Indietro fino alla prima TACCA trovata con **VEL=RZERO_VELF***
- 2) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 3) La quota asse assume il valore indicato in **RZERO_PRESET**
- 4)

RZERO_MODE=9 - Avanti con solo ricerca tacca di zero encoder

- 1) *Avanti fino alla prima TACCA trovata con **VEL=RZERO_VELF***
- 2) Posizionamento alla quota di offset (**RZERO_OFFSET**) con **VEL=RZERO_VEL**
- 3) La quota asse assume il valore indicato in **RZERO_PRESET**

RZERO_MODE=32 – Ricerca disabilita ZERO su ENABLE DRIVE**RZERO_MODE=64 – Ricerca disabilita ZERO su ENCODER ASSOLUTO****RZERO_MODE=128+HOME_MODEx256 – Ricerca disabilita ZERO su ENCODER ASSOLUTO**

La ricerca è affidata al driver CanOpen con standard DS402. I Bit 8-15 rappresentano l' HOME-MODE

17.30.2.7**LIMITE_N_**

Limite primario quote asse in negativo **EXTRA CORSA SOFTWARE**.
Il valore dipende dalla risoluzione del sistema (normalmente millesimi di millimetro)

17.30.2.8**LIMITE_P_**

Limite primario quote asse in positivo **EXTRA CORSA SOFTWARE**.
Il valore dipende dalla risoluzione del sistema (normalmente millesimi di millimetro)

17.30.2.9**2ND_LIMITE_N_**

Limite secondario quote asse in negativo **EXTRA CORSA SOFTWARE**.
Il valore dipende dalla risoluzione del sistema (normalmente millesimi di millimetro)

17.30.2.10**2ND_LIMITE_P_**

Limite secondario quote asse in positivo **EXTRA CORSA SOFTWARE**.
Il valore dipende dalla risoluzione del sistema (normalmente millesimi di millimetro)

17.30.2.11**LIMITE_N_Pn**

Limite quote asse in negativo **EXTRA CORSA SOFTWARE** per posizionatori

17.30.2.12**LIMITE_P_Pn**

Limite quote asse in positivo **EXTRA CORSA SOFTWARE** per posizionatori

17.30.2.13**DSOFV**

Divisore software quota volantino elettronico. Riduce o aumenta la risoluzione del volantino elettronico. Un valore negativo inverte il senso di spostamento. Es:

Encoder volantino= 100 imp/giro

SCALA=1 (parametro impostato su applicazione VTB)

FILTRO=10 (parametro impostato su applicazione VTB)

DSOFV=10000 (impulsi x filtro x **10 millesimi**)

Lavorando con una risoluzione del millesimo di millimetro si ottiene un incremento del centesimo per ogni impulso encoder

- 17.30.2.14** **RZERO_OFFSET**
Movimento di riposizionamento do ricerca zero asse (vedi RZERO_MODE).
Il valore dipende dalla risoluzione del sistema (normalmente millesimi di millimetro)
- 17.30.2.15** **RZERO_PRESET**
Valore che deve assumere la quota dopo il riposizionamento a **RZERO_OFFSET** (vedi RZERO_MODE).
Il valore dipende dalla risoluzione del sistema (normalmente millesimi di millimetro)
- 17.30.2.16** **G62_TO_G40**
Se il valore è a 1 (SI) viene aggiunta una **G62** in modo automatico sulla fine compensazione **G40**.
Questo permette di svuotare tutti i movimenti rimasti anche se non si usa un disimpegno utensile
- 17.30.2.17** **LIMIT_G41G42**
Se il valore è a 1 (SI) i Limiti preventivi o in simulazione, vengono controllati solo sul profilo compensato **G41 G42** e non anche sul profilo **G40** (centro utensile).
ATTENZIONE
In questo caso, il centro utensile può rientrare nei limiti macchina, mentre l' ingombro dell' utensile può eccedere i limiti della macchina
- 17.30.2.18** **INVERT_G2G3**
Se il valore è a 1 (SI) le funzioni **G2** (Interpolazione circolare Oraria) vengono scambiare con **G3** (Interpolazione circolare Antioraria) quando è settato il piano di lavoro **G18 XZ**
- 17.30.2.19** **STANDARD_IJ**
Se il valore è a 1 (SI) usa la convenzione standard per i parametri **I,J,K**

17.30.2.20**RZERO_VEL**

Velocità **ALTA** di ricerca sensore ZERO ASSE(vedi RZERO_MODE).
Il valore dipende dalla risoluzione del sistema (normalmente millimetri al minuto)

17.30.2.21**RZERO_VELF**

Velocità **LENTA** di ricerca sensore ZERO ASSE(vedi RZERO_MODE).
Il valore dipende dalla risoluzione del sistema (normalmente millimetri al minuto)

17.30.2.22**RZERO_ACC**

Accelerazione assi per ricerca di ZERO.
L' unità di misura è in count macchina, quindi dipende dall' impostazione del sistema.

17.30.2.23**MSOF**

Impulsi a giro encoder ASSE. Nel caso di assi collegati ad ingresso encoder del CN, il valore deve essere moltiplicato 4, questo poichè la scheda encoder moltiplica x 4 in modo hardware gli impulsi.

17.30.2.24**DSOF**

Sviluppo asse a giro encoder. Il valore determina l' unità di misura di programmazione delle quote. Normalmente vengono inseriti i millesimi di millimetro. Occorre impostare di conseguenza il parametro **RESQUOTE** al giusto valore.

| | |
|---------------|---------------------------------------|
| 10 | → Decimi di millimetro |
| 100 | → Centesimi di millimetro |
| 1000 | → Millesimi di millimetro |
| 10000 | → Decimillesimi di millimetro |
| 100000 | → Centimillesimi di millimetro |

17.30.2.25**GANTRY**

Indica in quale asse viene replicato lo stesso asse (funzione Gantry). Tale parametro può assumere i seguenti valori (0-9):

| | |
|----------|-----------------|
| 0 | → ASSE X |
| 1 | → ASSE Y |
| 2 | → ASSE Z |
| 3 | → ASSE A |
| 4 | → ASSE B |
| 5 | → ASSE C |
| 6 | → ASSE U |
| 7 | → ASSE V |
| 8 | → ASSE W |

quindi:

GANTRY_X → (Se uguale a 0 disabilitato) altrimenti numero asse
GANTRY_Y → (Se uguale a 1 disabilitato) altrimenti numero asse
GANTRY_Z → (Se uguale a 2 disabilitato) altrimenti numero asse
GANTRY_A → (Se uguale a 3 disabilitato) altrimenti numero asse
GANTRY_B → (Se uguale a 4 disabilitato) altrimenti numero asse
GANTRY_C → (Se uguale a 5 disabilitato) altrimenti numero asse
GANTRY_U → (Se uguale a 6 disabilitato) altrimenti numero asse
GANTRY_V → (Se uguale a 7 disabilitato) altrimenti numero asse
GANTRY_W → (Se uguale a 8 disabilitato) altrimenti numero asse

ATTENZIONE!!!

La funzione Gantry per essere ABILITATA deve essere supportata dal codice VTB del CN

17.30.2.26**SGL3D**

Definisce la soglia **SPIGOLO** per interpolazioni nello spazio. Questa viene usata dall' interpolazione **G65 G67** o dal filtro **AFC G66**. Il valore è espresso in decimi di GRADO (effettivamente dipende dall' inclinazione degli assi). Il corretto valore viene determinato dalla dinamica della macchina. Durante l' interpolazione **G65** o **G67**, se uno degli assi supera questa soglia, il tratto decelera fino a fermarsi.

Valori di riferimento spigolo per SGL3D

| SOGLIA SPIGOLO IN GRADI | VALORE SGL3D (min-max) |
|-------------------------|------------------------|
| 5 | 60-90 |
| 10 | 125-175 |
| 20 | 250-350 |
| 30 | 300-500 |
| 45 | 400-700 |

17.30.2.27**SGLAFC**

Definisce la soglia di intervento per riduzione della velocità sul tratto. Questo viene utilizzato da **AFC G66**. Questo valore determina la variazione massima istantanea di velocità che può sopportare il singolo asse. Valori superiori a questa soglia non sono ammessi e pertanto viene effettuata una riduzione della velocità. Il suo valore deve essere calcolato in base alla dinamica del singolo asse (determinata dal motore, driver, massa , meccanica ecc.)

17.30.2.28**MILD**

Definisce il valore di ADDOLCIMENTO degli SPIGOLI ([vedi G49](#))

Valori consigliati da 200 a 20

0 o 1000 DISABILITATO

17.30.2.29**SGLP_MILD**

Soglia spigolo in decimi di grado **SOPRA LA QUALE il MILD** non agisce per profili 2D

17.30.2.30**SGL3D_MILD_**

Soglia spigolo **SOPRA LA QUALE il MILD** non agisce per profili 3D (per riferimenti in gradi vedi SGL3D)

17.30.2.31**BACKLASH**

Definisce il valore del GIOCO presente sul relativo ASSE. Il valore è espresso in unità di misura programmate.

17.30.2.32**TBCK**

Definisce il tempo (velocità) di recupero GIOCO. Se $TBCK \geq BACKLASH$ il gioco viene recuperato in un unico campionamento. Altrimenti il recupero viene effettuato in più campionamenti definiti dalla successiva formula.

$$NC = \text{BACKLASH} / TBCK$$

Eventuali decimali portano l' approssimazione ad un intero più alto

Es:

$$BACKLASH = 100$$

$$TBCK = 80$$

$$NC = 1.25 \rightarrow 2 \text{ campionamenti}$$

Per campionamento si intende il Sample programmato nell' applicazione VTB (tipico 2 Ms).

E' consigliabile distribuire il recupero GIOCO in più campionamenti quando questo risulta di valore elevato (sopra i 3-4 centesimi).

17.30.2.33**TSHF**

Definisce il tempo (velocità) di recupero incremento assi con metodo SetShiftAxis indica il numero di impulsi di incremento quota per campionamento TAU impostato.
Es:

```
TAU=2 Ms
ValShift=100
TSHF_=10
```

La quota viene raggiunta in 20 Ms

17.30.2.34**WR_SPD9**

Parametro che indica se abilitata la scrittura automatica della SPEED "S" sulla variabile USER GENERIC 10 del CN.

```
WR_SPD9=0   Disabilitata
WR_SPD9=1   Scrittura automatica abilitata
```

Se viene abilitata la scrittura automatica, ogni volta che viene incontrata l'istruzione S nel PartProgram, il relativo valore viene scritto sulla User Generic 10 (9 partendo da 0)

17.30.2.35**RFG**

Parametro che indica se abilitata la limitazione della FEED su G1

```
RFG=0   Nessuna limitazione della FEED
RFG=1   La FEED viene limitata se la velocità vettoriale di un singolo asse, supera quella
         impostata sui parametri ASSI (VMAX_)
```

17.30.2.36**JERK**

Valore del JERK per rampe ad S.

Se il valore è 0, vengono utilizzate le rampe trapezoidali.

Un valore maggiore di 0, abilita le rampe ad S.

Aumentando il JERK, si aumenta la curva della rampa ad S.

Valori tipici sono 10,20 etc fino ad un massimo di 100

17.30.2.37**CR_LIMIT**

Abilita o disabilita il controllo preventivo sui limiti assi

Se abilitato, ad ogni START, il PART PROGRAM non viene eseguito se gli assi eccedono dai limiti macchina impostati

```
CR_LIMIT=0   Controllo preventivo disattivato
CR_LIMIT=1   Controllo preventivo attivato su G0-G1 solo sul punto finale di G2-G3
CR_LIMIT=2   Controllo preventivo attivato su G0-G1-G2-G3 (con esplosione degli archi)
CR_LIMIT=3   Controllo preventivo attivato su G0-G1-G2-G3 solo in run time cioè prima di
             esecuzione del tratto ( G2-G3 viene controllato solo il punto finale)
```

17.30.2.38**ARC_REL**

Abilita o disabilita l'utilizzo dei parametri I,J (centro archi) in modo relativo

```
ARC_REL=0   I,J presi con valore assoluto (se siamo in G90)
```

```
ARC_REL=1   I,J presi con valore relativo (se siamo in G90)
```

17.30.2.39**NO_SHORT**

Abilita o disabilita la rimozione di tratti corti da parte del CNC

```
NO_SHORT=0   Nessuna funzione
```

```
NO_SHORT=1   I tratti corti vengono automaticamente eliminati dal CNC. Questo favorisce
             una lavorazione più fluida dei profili complessi.
```

```
NO_SHORT=2   Se vengono incontrati dei tratti corti viene dato uno Warning
```

- 17.30.2.40** **USE_G60**
Se >0, viene abilitata per default G60 e non G61
- 17.30.2.41** **TIME_OUT_CMD**
Time out comandi inviati al CNC. Valore in Ms (default 5000)
- 17.30.2.42** **TIME_OUT_M**
Time out M su CNC. Valore in Ms (default 15000)
- 17.30.2.43** **ACC_VMODE_X,Y,Z,A,B,C,U,V,W**
Indica l' accelerazione per l' asse in velocità quando viene attivata la funzione G108.4
- 17.30.2.44** **ENABLE_RFEED**
Se ad ON (1) abilita la gestione degli eventi (e la visualizzazione nell' interfaccia) della velocità reale degli assi. Disponibile solo dalla versione **ComSynk** 3.0.0.500 o superiori e **IsoVirtual** 2.5.0 o superiore
- 17.30.2.45** **NEW_ORIGINS**
Se ad ON (1) abilita la gestione avanzata del file delle origini "zeri.val".
Questo avrà un nuovo nome – "**Origins_n.val**" dove **n** è il numero di processo di IsoUs.
Pertanto con la nuova versione viene generato un file di origini per ogni processo in esecuzione e
Quindi tutte le origini sono **SEPARATE** per ogni processo
- 17.30.2.46** **USE_G80_CYCLES**
Se ad ON (1) abilita la migrazione dei codici:
 G1080 → G80
 G1081 → G81
 G1082 → G82
 G1083 → G83
 G1084 → G84
 G1028 → G28
 E viceversa
- 17.30.2.47** **G28HOME_**
Definisce la posizione di **HOME** degli assi per la funzione **G1028 (G28)**
Il valore dipende dalla risoluzione del sistema (normalmente millesimi di millimetro)
- 17.30.2.48** **DIVACC**
Divisore Accelerazione per interpolatore **MSL** (valore tipico 10)
Valori più alti definiscono rampe di accelerazione più basse
- 17.30.2.49** **AXES_Z**
Definisce indice **ASSE Z** (profondità) per cicli G81,82,83,84
- 17.30.2.50** **AXES_G84**
Definisce indice **ASSE MASCHIATURA** per G84
Solo per modalità **INTERPOLATA**
- 17.30.2.51** **MODE_G84**
Modalità G84
NORMALE (0) per Mandrino NON INTERPOLATO
INTERPOLATO (1) per Mandrino INTERPOLATO
Il passo per la Maschiatura viene calcolato tra il rapporto **S** e **F** impostati

- 17.30.2.52** **ROT_G84**
Senso di rotazione Asse maschiatura INTERPOLATO
POSITIVA (0)
NEGATIVA (1)
- 17.30.2.53** **MOLT_G84**
Moltiplicazione FEED asse rotativo di MASCHIATURA G84
- 17.30.2.54** **DELTA_G83**
Parametro **D** descritto nella funzione [G1083](#) (G83)
- 17.30.2.55** **PRIORITY_TASK1 - PRIORITY_TASK2**
Priorità dei task paralleli. Indica in che intervallo di tempo questi vengono eseguiti.
Valore **0** (default) priorità **MASSIMA**.
Inserire valori da 0 a 100
- 17.30.2.56** **USE_J_M6**
Se a SI (1) durante un ciclo M6 viene disabilitato PAUSA e STOP
- 17.30.2.57** **SH_FILTER_X,Y,Z,A,B,C,U,V,W**
Filtro per Shift assi.
0 Disabilitato
Valore bassi filtraggio maggiore
Da 0 a 200

17.31 PARAMETRI PID

I parametri PID sono usati per il controllo degli assi in velocità con retroazione da encoder. Questi vengono inseriti dal configuratore di Isous “Parametri Macchina→Default Par PID”:



17.31.1 **PID_KP**

Guadagno Proporzionale

17.31.2 **PID_KI**

Guadagno Integrativo

17.31.3 **PID_KV**

Feed Forward.

Per una corretta taratura procedure come segue:

- 1) Azzerare PID_KP e PID_KI
- 2) Inserire un valore in PID_KV
- 3) Muovere l'asse e controllare che la posizione reale si vicino alla posizione target (+/- 5 %)
- 4) Insert the PID_KP and PID_KI

ATTENZIONE

Questa operazione dovrebbe Essere effettuata con motore scollegato dalla meccanica

17.31.4 **PID_I_LIMIT**

Limite di integrazione

17.31.5 **PID_DIV**

Divisore dei parametric PID, aumentando questo valore aumenta la risoluzione dei parametri:

Es: PID_DIV=10 PID_KP=1
 È lo stesso
 PID_DIV=100 PID_KP=10

17.31.6 **PID_SERVO**

Errore di inseguimento massimo (micron).

Se l'asse eccede da questo valore viene generato un allarme

(deve eccedere questo valore per il tempo indicato in PID_TIME_SERVO)

17.31.7 **PID_TIME_SERVO**

Tempo Massimo di permanenza in errore di inseguimento (Millisecondi)

17.31.8 **PID_DIR**

Direzione uscita analogica (0 or 1)

Cambiare questo parametro per accordare la direzione asse con lettura encoder

ATTENZIONE

Per cambiare la direzione su un asse già accordato, è necessario invertire questo parametro e cambiare il segno anche al relative parametro MSOF: (es: 5000 in -5000)

17.31.9 **PID_OFFS_ANA**

Azzeramento offset uscita analogica

Con I parametri PID (KP,KI) a ZERO, settare questo parametro fine a che l'asse non risulta fermo.

(Quando l'asse è abilitato)

17.32 PARAMETRI SPINDLE

Questa tabella racchiude i parametri relativi al mandrino

17.32.1 SPEEDMAXSPINDLE

Valore Massimo in giri RPM del mandrino quando la tensione su uscita analogica è 10V

17.32.2 ANALOG_BIT_RES

Risoluzione in bit del canale analogico selezionato (vedi [Cap. 11.18](#))

17.32.3 SPEED_ANALOG_CH

Numero canale selezionato per mandrino (vedi [Cap. 11.18](#))

17.32.4 ENABLE_OW_SPEED

Gestisce il potenziometro di override per mandrino

DISABILITATO

Nessun potenziometro disponibile

ESTERNO

Potenziometro esterno gestito da applicazione VTB

INTERNO VIRTUALE

Abilita il potenziometro virtuale del Plugin **USSPINDLEMANAGER (IsoUs)**

17.32.5 SPEED_OW_MIN

Valore minimo in percentuale per override rispetto alla SPEED impostata (0-100%)

17.32.6 SPEED_OW_MAX

Valore massimo in percentuale per override rispetto alla SPEED impostata (0-100%)

17.33 Parametri EGEAR

17.33.1 *EGEAR_KEM*

Fattore moltiplicativo per taratura velocità **MASTER - SLAVE**

17.33.2 *EGEAR_KED*

Fattore divisivo per taratura velocità **MASTER - SLAVE**

17.33.3 *EGEAR_ACC*

Accelerazione di **AGGANCIO** asse **MASTER – SLAVE** (um/TAU)

Esempio

Asse **MASTER NASTRO** controllato da motore con lettura della posizione tramite **ENCODER 1000** i/g (consideriamo che gli impulsi sono moltiplicati x 4), asse **X SLAVE** motore brushless interpolato.

Spazio percorso **MASTER** ogni giro encoder = 100 mm.

Aggancio **SLAVE** in 500 msec (0,5 sec)

TAU=2 Msec

EGEAR_KEM =100000 um

EGEAR_KED = 4000 i/g

EGEAR_ACC = 100000/(500/TAU) = 100000/250=400

17.34 Parametri Velocità Periferica

17.34.1 *CENTRE_X_R1*

Quota X asse di rotazione primo asse rotativo in [um]

17.34.2 *CENTRE_Y_R1*

Quota Y asse di rotazione primo asse rotativo in [um]

17.34.3 *CENTRE_Z_R1*

Quota Z asse di rotazione primo asse rotativo in [um]

17.34.4 *CENTRE_X_R2*

Quota X asse di rotazione secondo asse rotativo in [um]

17.34.5 *CENTRE_Y_R2*

Quota Y asse di rotazione secondo asse rotativo in [um]

17.34.6 *CENTRE_Z_R2*

Quota Z asse di rotazione secondo asse rotativo in [um]

Per ognuno degli assi rotativi configurati, sono necessari due parametri per definirne l'asse di rotazione.

Riferendosi all'esempio presente nelle istruzioni sulla G108.12 (vedi [Cap.17.21.5](#)), trattandosi di un asse che ha Y come asse di riferimento, i parametri da impostare sono le quote centro X e Z.

Sommario

| | | |
|----------|---|-----------|
| 1 | PREFAZIONE | 3 |
| 2 | REGOLE DI ISOUS | 3 |
| 2.1 | BLOCCO | 3 |
| 2.2 | NUMERO DI LINEA O NUMERO DI BLOCCO | 3 |
| 2.3 | ESECUZIONE DEL PROGRAMMA | 3 |
| 2.4 | NOMI DEGLI ASSI | 3 |
| 2.5 | ORIGINE MACCHINA | 3 |
| 2.6 | ORIGINE PEZZO O ZERO PEZZO | 3 |
| 2.7 | OFFSET PEZZO | 3 |
| 2.8 | TESTE | 3 |
| 2.9 | FUNZIONI MODALI | 4 |
| 2.10 | CODICI RICONOSCIUTI | 4 |
| 2.11 | IMPOSTAZIONE DEI VALORI NUMERICI | 4 |
| 2.12 | COMMENTI AL PROGRAMMA | 4 |
| 2.13 | INIZIO DI UN PROGRAMMA | 4 |
| 2.14 | FINE DEL PROGRAMMA | 4 |
| 3 | ISTRUZIONI ISO US | 5 |
| 3.1 | ISTRUZIONI G RICONOSCIUTE | 5 |
| 3.2 | ISTRUZIONI ISO RICONOSCIUTE | 8 |
| 3.3 | ISTRUZIONI PER IL CONTROLLO DEL FLUSSO DI PROGRAMMA | 10 |
| 3.4 | ISTRUZIONI PER IL MULTI TASK | 10 |
| 3.5 | ISTRUZIONI PER UTILIZZO GENERICICO | 11 |
| 3.6 | ISTRUZIONI PER IL CONTROLLO REMOTO | 12 |
| 3.7 | ISTRUZIONI PER IL CONTROLLO DEL MULTIPROCESSO | 13 |
| 3.8 | OPERATORI LOGICI E MATEMATICI | 14 |
| 3.9 | MATEMATICA | 15 |
| 3.10 | VARIABILI E COSTANTI | 16 |
| 3.11 | VARIABILI PREDEFINITE | 16 |
| 3.12 | ISTRUZIONI DI LIBRERIA USFORMS | 17 |
| 3.13 | ISTRUZIONI ESTESE EXD | 18 |
| 3.14 | SWITCH E DIRETTIVE DI COMPILAZIONE | 19 |
| 4 | CONTROLLO DEL FLUSSO DEL PROGRAMMA | 20 |
| 4.1 | IF-ELSE-END_IF | 20 |
| 4.2 | LOOP - END_LOOP | 20 |
| 4.3 | FOR – NEXT – BREAK - CONTINUE | 21 |
| 4.4 | WHILE – END_WHILE – BREAK - CONTINUE | 22 |

| | | |
|------|---|----|
| 4.5 | GOTO | 23 |
| 4.6 | GOSUB - RETURN | 23 |
| 4.7 | LABEL (PUNTI DI ETICHETTA) | 24 |
| 4.8 | END_PROGRAM | 24 |
| 4.9 | WAIT_INPUT | 24 |
| 4.10 | TWAIT_INPUT | 25 |
| 4.11 | ERROR..... | 25 |
| 4.12 | RESUME_T | 26 |
| 4.13 | SAVE_T..... | 26 |
| 4.14 | SET_TABPAR..... | 26 |
| 4.15 | SELECT – CASE – END_SELECT | 27 |
| 4.16 | IF RUN..... | 28 |
| 4.17 | IF NOTRUN | 28 |
| 4.18 | USEFORM..... | 28 |
| 5 | ISTRUZIONI DI UTILIZZO GENERICO | 29 |
| 5.1 | SDO_DL..... | 29 |
| 5.2 | SDO_UL..... | 29 |
| 5.3 | GET | 29 |
| 5.4 | READ_PARMAC | 30 |
| 5.5 | WRITE_PARMAC | 30 |
| 5.6 | OPT | 30 |
| 5.7 | PAUSE_MODE | 31 |
| 5.8 | FILTER_MODE..... | 31 |
| 5.9 | IMPORT..... | 31 |
| 5.10 | END_IMPORT | 32 |
| 5.11 | STOP_MODE | 32 |
| 5.12 | DEBUG_INFO..... | 32 |
| 6 | ISTRUZIONI DI LIBRERIA GESTIONE DISPLAY REMOTE HANDWHELL..... | 34 |
| 6.1 | LIB.HMESSAGE | 34 |
| 7 | ISTRUZIONI DI LIBRERIA USFORMS | 35 |
| 7.1 | LIB.MESSAGE..... | 36 |
| 7.2 | LIB.SHOWFORM | 37 |
| 7.3 | LIB.CLOSEFORM | 37 |
| 7.4 | LIB.FORMPROP | 37 |
| 7.5 | LIB.FORMTEXT | 40 |
| 7.6 | LIB.ADDLABEL | 40 |
| 7.7 | LIB.LABELPROP..... | 40 |
| 7.8 | LIB.LABELTEXT..... | 41 |

| | | |
|------|----------------------------------|----|
| 7.9 | LIB.LABELPRINT | 42 |
| 7.10 | LIB.LABELF | 42 |
| 7.11 | LIB.ADDBUTTON | 42 |
| 7.12 | LIB.BUTTONPROP | 42 |
| 7.13 | LIB.BUTTONTEXT | 44 |
| 7.14 | LIB.BUTTONPRINT | 44 |
| 7.15 | LIB.BUTTONF | 45 |
| 7.16 | LIB.ADDINPUT | 45 |
| 7.17 | LIB.INPUTPROP | 45 |
| 7.18 | LIB.INPUTSETVALUE | 46 |
| 7.19 | LIB.ADDITEXT | 47 |
| 7.20 | LIB.ITEXTPROP | 47 |
| 7.21 | LIB.ITEXTSETVALUE | 48 |
| 7.22 | LIB.ADDCHECK | 48 |
| 7.23 | LIB.CHECKPROP | 48 |
| 7.24 | LIB.CHECKSETVALUE | 49 |
| 7.25 | LIB.CHECKTEXT | 49 |
| 7.26 | LIB.ADDCOMBO | 50 |
| 7.27 | LIB.COMBOPROP | 50 |
| 7.28 | LIB.COMBOSETVALUE | 51 |
| 7.29 | LIB.COMBOITEM | 51 |
| 7.30 | LIB.ADDSLIDER | 51 |
| 7.31 | LIB.SLIDERPROP | 52 |
| 7.32 | LIB.SLIDERSETVALUE | 53 |
| 7.33 | LIB.GETVAR | 53 |
| 7.34 | LIB.SETVAR | 53 |
| 7.35 | LIB.DEBUG | 54 |
| 7.36 | TABELLA RIFERIMENTO COLORI | 55 |
| 8 | ISTRUZIONI ESTESE | 57 |
| 8.1 | EXD.STL_LOAD | 57 |
| 8.2 | EXD.READ_TOOLPAR | 61 |
| 8.3 | EXD.READ_HEADPAR | 61 |
| 8.4 | EXD.WRITE_TOOLPAR | 61 |
| 8.5 | EXD.WRITE_HEADPAR | 62 |
| 8.6 | EXD.LOAD_LAST | 62 |
| 8.7 | EXD.SAVE_LAST | 62 |
| 8.8 | EXD.RESET_LAST | 62 |
| 8.9 | EXD.SET_OUT | 63 |

| | | |
|------|--|----|
| 8.10 | EXD.READ_OUT..... | 63 |
| 8.11 | EXD.READ_INP..... | 63 |
| 8.12 | EXD.WRITE_USER..... | 63 |
| 8.13 | EXD.READ_USER..... | 64 |
| 8.14 | EXD.READ_DEMAND..... | 64 |
| 8.15 | EXD.READ_REAL..... | 64 |
| 8.16 | EXD.MASK..... | 65 |
| 8.17 | EXD.WRITE_BIT..... | 65 |
| 8.18 | EXD.READ_BIT..... | 65 |
| 8.19 | EXD.PXV_ALL_DETECTORS..... | 66 |
| 8.20 | EXD.PXV_SINGLE_DETECTOR..... | 66 |
| 8.21 | EXD.PXV_READ_PROBE..... | 67 |
| 8.22 | EXD.PXV_RESET_PROBE..... | 67 |
| 8.23 | EXD.PXV_SET_DETECTOR..... | 67 |
| 8.24 | EXD.PXV_SAVE_IMAGE..... | 68 |
| 8.25 | EXD.PXV_GET_IMAGE..... | 68 |
| 8.26 | EXD.PXV_SET_JOB..... | 68 |
| 8.27 | EXD.RUN_SCRIPT..... | 69 |
| 8.28 | EXD.OPEN_DLL..... | 72 |
| 8.29 | EXD.CALL_DLL..... | 74 |
| 8.30 | EXD.CLOSE_DLL..... | 74 |
| 8.31 | EXD.SYNK_DLL..... | 74 |
| 8.32 | EXD.STL_SETVIS..... | 75 |
| 8.33 | EXD.STL_READVIS..... | 75 |
| 8.34 | EXD.STL_SETVISIDX..... | 75 |
| 8.35 | EXD.STL_READVISIDX..... | 76 |
| 9 | SWITCH E DIRETTIVE DI COMPILAZIONE..... | 90 |
| 9.1 | IFDEF..... | 90 |
| 9.2 | ELSEDEF..... | 90 |
| 9.3 | ENDIFDEF..... | 90 |
| 9.4 | NOAXESREADY..... | 91 |
| 9.5 | ONERROR..... | 91 |
| 9.6 | ONSTOP..... | 91 |
| 9.7 | ENDON..... | 91 |
| 9.8 | USET..... | 91 |
| 9.9 | USEH..... | 91 |
| 10 | ISTRUZIONI PER IL CONTROLLO MULTIPROCESSO..... | 92 |
| 10.1 | CNC.LOAD..... | 92 |

| | | |
|-------|--|-----|
| 10.2 | CNC.RUN | 92 |
| 10.3 | CNC.PREVIEW | 92 |
| 10.4 | CNC.STOP | 93 |
| 10.5 | CNC.PAUSE | 93 |
| 10.6 | CNC.STATUS | 93 |
| 10.7 | CNC.STATUSBIT | 94 |
| 10.8 | CNC.INFO | 94 |
| 10.9 | CNC.AXIS..... | 94 |
| 10.10 | CNC.GROUP | 95 |
| 10.11 | CNC.READVARADDR | 95 |
| 10.12 | CNC.READVARNAME | 95 |
| 10.13 | CNC.WRITEVARADDR..... | 96 |
| 10.14 | CNC.WRITEVARNAME | 96 |
| 10.15 | CNC.READPARMAC | 96 |
| 10.16 | CNC.WRITEPARMAC | 96 |
| 10.17 | CNC.ENABLEAXIS | 96 |
| 10.18 | CNC.HOMEAXIS | 97 |
| 10.19 | CNC.READGENERIC | 97 |
| 10.20 | CNC.WRITEGENERIC..... | 97 |
| 11 | ISTRUZIONI PER IL CONTROLLO REMOTO | 98 |
| 11.1 | REMOTE.LOAD | 98 |
| 11.2 | REMOTE.RUN | 98 |
| 11.3 | REMOTE.STOP | 98 |
| 11.4 | REMOTE.PAUSE | 99 |
| 11.5 | REMOTE.STATUS | 99 |
| 11.6 | REMOTE.MOVE | 99 |
| 11.7 | REMOTE.INFO | 100 |
| 11.8 | REMOTE.AXIS..... | 100 |
| 11.9 | REMOTE.GROUP | 101 |
| 11.10 | REMOTE.READISOVAR | 101 |
| 11.11 | REMOTE.READVARNAME | 101 |
| 11.12 | REMOTE.WRITEISOVAR | 102 |
| 11.13 | REMOTE.WRITENAMEVAR..... | 102 |
| 11.14 | REMOTE.READCNVAR..... | 102 |
| 11.15 | REMOTE.WRITECNVAR | 102 |
| 11.16 | REMOTE.READINPUT | 103 |
| 11.17 | REMOTE.READOUT | 103 |
| 11.18 | REMOTE.WRITEOUT | 103 |

| | | |
|-------|---|-----|
| 12 | CONFIGURAZIONE CNC REMOTI | 104 |
| 13 | ISTRUZIONI MATEMATICHE..... | 106 |
| 13.1 | SIN | 106 |
| 13.2 | COS | 106 |
| 13.3 | LOG..... | 106 |
| 13.4 | EXP | 106 |
| 13.5 | INT | 106 |
| 13.6 | FIX..... | 107 |
| 13.7 | ABS | 107 |
| 13.8 | DRG..... | 107 |
| 13.9 | RAD..... | 107 |
| 13.10 | SQR | 107 |
| 13.11 | TAN..... | 107 |
| 13.12 | ATAN..... | 108 |
| 13.13 | ASIN | 108 |
| 13.14 | ACOS | 108 |
| 14 | UTILIZZO DEGLI OPERATORI LOGICI..... | 109 |
| 14.1 | PARENTESI () | 109 |
| 14.2 | ESPRESSIONI IN ISO - PARENTESI [] | 109 |
| 14.3 | CONTROLLO VARIABILI | 109 |
| 14.4 | CONTROLLO DEI BIT | 110 |
| 15 | TIPI DI VARIABILI UTILIZZATE DA ISOUS | 110 |
| 15.1 | COSTANTI NUMERICHE..... | 110 |
| 15.2 | VARIABILI DI UTILIZZO GENERICO..... | 110 |
| 15.3 | VARIABILI PER INDIRIZZO | 111 |
| 15.4 | VARIABILI PER PUNTATORE | 111 |
| 15.5 | VARIABILI ARRAY - DIM | 111 |
| 15.6 | VARIABILI QUOTE ASSI TEORICHE/REALI | 111 |
| 15.7 | VARIABILI INGRESSI/USCITE DIGITALI | 112 |
| 15.8 | VARIABILI TIMER | 113 |
| 15.9 | VARIABILI TABELLA UTENSILE | 114 |
| 15.10 | VARIABILI TESTA UTENSILE | 115 |
| 15.11 | VARIABILI CONTATORI ASSI..... | 116 |
| 15.12 | VARIABILI LIMITI ASSI | 116 |
| 15.13 | STRUTTURE DATI | 116 |
| 15.14 | PARAMETRI PER PREVIEW | 117 |
| 15.15 | VARIABILI GLOBAL | 122 |
| 15.16 | VARIABILI PREDEFINITE PER FUNZIONI M..... | 122 |

| | | |
|--------------|---|------------|
| 15.17 | VARIABILI PARAMETRI SPECIALI | 123 |
| 15.18 | VARIABILI ZERO E OFFSET PEZZO | 125 |
| 15.19 | VARIABILI USER GENERIC..... | 125 |
| 15.20 | VARIABILI MARKER..... | 125 |
| 15.21 | VARIABILE USCITA ANALOGICA PER MANDRINO | 126 |
| 15.21.1 | <i>Canali Impostabili.....</i> | 126 |
| 15.21.2 | <i>Risoluzione dei Canali.....</i> | 126 |
| 15.22 | GESTIONE MACRO E PARAMETRI GCODE..... | 127 |
| 16 | GESTIONE DEI POSIZIONATORI..... | 128 |
| 16.1 | PA-PD(N,PAR)ESPR - SET POSIZIONE ASSOLUTA E CONTROLLO POSIZIONATORE | 128 |
| 16.2 | PF(N)ESPR - SET FEED POSIZIONATORE..... | 128 |
| 16.3 | PS(N) - STOP MOVIMENTO POSIZIONATORE..... | 128 |
| 16.4 | _PM(N,PAR) - LETTURA STATO POSIZIONATORE | 128 |
| 17 | IMPOSTAZIONE DELLE LAVORAZIONI | 129 |
| 17.1 | DEFINIZIONE DELLE QUOTE ASSI | 129 |
| 17.1.1 | <i>Definizione quote Assi per indirizzo Canale</i> | 129 |
| 17.1.2 | <i>G90 - PROGRAMMAZIONE CON QUOTE ASSOLUTE</i> | 130 |
| 17.1.3 | <i>G91 - PROGRAMMAZIONE CON QUOTE INCREMENTALI.....</i> | 130 |
| 17.1.4 | <i>Definizione delle coordinate incrementali assolute.....</i> | 130 |
| 17.2 | ORIGINI PEZZO | 131 |
| 17.2.1 | <i>Origine Pezzo da INDICE.....</i> | 131 |
| 17.2.2 | <i>G94 - Definizione di un origine pezzo a quote definite da parametro</i> | 131 |
| 17.2.3 | <i>G94.n – Salva le attuali origini nel file.....</i> | 132 |
| 17.2.4 | <i>G54-G55-G56-G57-G58-G59 Definizione di un origine PEZZO da FILE in memoria.....</i> | 132 |
| 17.2.5 | <i>G92 - Definizione di un origine pezzo nel punto attuale degli assi</i> | 133 |
| 17.2.6 | <i>G82 - Definizione di un origine pezzo nel punto attuale degli assi da sensore</i> | 133 |
| 17.2.7 | <i>G98 – G53 - Sospensione origine pezzo.....</i> | 133 |
| 17.2.8 | <i>G99 - Ripresa origine pezzo</i> | 134 |
| 17.2.9 | <i>G940 – Movimentazione assi escludendo ORIGINI e OFFSET solo nel blocco attuale</i> | 134 |
| 17.2.10 | <i>USER_ZERO - Indice del vettore ORIGINE ASSI</i> | 135 |
| 17.3 | OFFSET PEZZO | 136 |
| 17.3.1 | <i>G93 - Definizione di un offset pezzo a quote definite da parametro</i> | 136 |
| 17.3.2 | <i>G95 - Definizione di un offset pezzo nel punto attuale degli assi</i> | 136 |
| 17.3.3 | <i>G85 - Definizione di un offset pezzo nel punto attuale degli assi da sensore</i> | 137 |
| 17.3.4 | <i>G86 – Preset asse in modo hardware sul modulo 360 gradi</i> | 137 |
| 17.3.5 | <i>G96 - Sospensione offset pezzo</i> | 137 |
| 17.3.6 | <i>G97 - Ripresa offset pezzo</i> | 138 |
| 17.3.7 | <i>USER_OFFSET - Indice del vettore OFFSET ASSI.....</i> | 138 |
| 17.4 | SELEZIONE DELLA TESTA DI LAVORO – FUNZIONE H | 139 |
| 17.4.1 | <i>Hn – Seleziona testa utensile (mandrino).....</i> | 139 |
| 17.4.2 | <i>G87 – Sospensione offset testa</i> | 139 |
| 17.4.3 | <i>G88 – Ripresa offset testa</i> | 139 |
| 17.5 | GESTIONE ASSI ROTATIVI..... | 140 |
| 17.5.1 | <i>G36 - Definizione di un asse rotativo.....</i> | 140 |
| 17.6 | PRESET HARDWARE CONTATORI ASSI..... | 141 |

| | | |
|--------------|---|------------|
| 17.6.1 | <i>G89 – Preset contatore asse in modo HardWare</i> | 141 |
| 17.7 | PRESET CONTATORI ASSI DOPO PAUSA E PRIMA DELLE MACRO SPECIALI | 141 |
| 17.7.1 | <i>G84 – Preset contatori CPU</i> | 141 |
| 17.7.2 | <i>G83 – Preset contatori CPU 1</i> | 141 |
| 17.8 | SELEZIONE DEL PIANO DI LAVORO | 142 |
| 17.8.1 | <i>G17 - Piano di lavoro su X-Y</i> | 142 |
| 17.8.2 | <i>G18 - Piano di lavoro su X-Z</i> | 142 |
| 17.8.3 | <i>G18.1 - Piano di lavoro su X-Z Ma non in Preview</i> | 142 |
| 17.8.4 | <i>G19 - Piano di lavoro su Y-Z</i> | 142 |
| 17.8.5 | <i>G19.1 - Piano di lavoro su Y-Z Ma non in Preview</i> | 143 |
| 17.8.6 | <i>G70 - Piano di lavoro su coppia di assi a piacere</i> | 143 |
| 17.9 | RICERCA HOME ASSE DA PART PROGRAM | 144 |
| 17.9.1 | <i>G71 – Home asse</i> | 144 |
| 17.9.2 | <i>G71.1 – Abilita Asse</i> | 144 |
| 17.9.3 | <i>G71.2 – Disabilita Asse</i> | 144 |
| 17.10 | FUNZIONI DI MOVIMENTAZIONE ASSI | 145 |
| 17.10.1 | <i>G0 - Movimentazione con Velocità rapida</i> | 145 |
| 17.10.2 | <i>G0.1 - Movimentazione con Velocità rapida Asse Singolo con Accelerazione privata</i> | 145 |
| 17.10.3 | <i>G0.2 - Movimentazione con Velocità rapida – Per asse Tangenziale</i> | 146 |
| 17.10.4 | <i>G1 - Interpolazione lineare ad F programmata</i> | 147 |
| 17.10.5 | <i>G1.1 – Sospensione G1-G2-G3 e Impostazione G0</i> | 147 |
| 17.10.6 | <i>G1.2 – Riprende G1-G2-G3</i> | 147 |
| 17.10.7 | <i>G2/G3 - Interpolazione circolare ad F programmata</i> | 148 |
| 17.10.8 | <i>G30 - Inserimento automatico raccordi su spigoli</i> | 149 |
| 17.10.9 | <i>G31 - Sospensione Inserimento automatico raccordi su spigoli</i> | 150 |
| 17.10.10 | <i>G32 - Ripresa Inserimento automatico raccordi su spigoli</i> | 150 |
| 17.10.11 | <i>G33 - Inserimento automatico smussi su spigoli</i> | 151 |
| 17.10.12 | <i>G34 - Sospensione Inserimento automatico smussi su spigoli</i> | 152 |
| 17.10.13 | <i>G35 - Ripresa Inserimento automatico smussi su spigoli</i> | 152 |
| 17.10.14 | <i>G102 – Start acquisizione quote da sensore</i> | 153 |
| 17.10.15 | <i>G102.1 – Acquisizione da sistema PxVision</i> | 154 |
| 17.11 | PROGRAMMAZIONE DELLA VELOCITÀ INTERPOLAZIONE ASSI | 159 |
| 17.11.1 | <i>F – Velocità interpolazione assi</i> | 159 |
| 17.11.2 | <i>Autoriduzione Velocità su ARCHI</i> | 159 |
| 17.12 | TRASFORMAZIONI DEL PIANO DI LAVORO | 160 |
| 17.12.1 | <i>G120 - Specularità verticale del piano</i> | 160 |
| 17.12.2 | <i>G121 - Disabilita Specularità verticale del piano</i> | 160 |
| 17.12.3 | <i>G24 - Specularità orizzontale del piano</i> | 161 |
| 17.12.4 | <i>G25 - Disabilita Specularità orizzontale del piano</i> | 161 |
| 17.12.5 | <i>G22 - Scambia gli assi del piano di interpolazione</i> | 162 |
| 17.12.6 | <i>G23 - Ripristina gli assi del piano di interpolazione</i> | 162 |
| 17.12.7 | <i>G26 - Scambia coppia di assi a piacere</i> | 162 |
| 17.12.8 | <i>G27 – Sospende G26</i> | 162 |
| 17.12.9 | <i>G28 – Riprende G26</i> | 163 |
| 17.12.10 | <i>G1028 – Ritorno ad Home</i> | 163 |
| 17.12.11 | <i>G51 - Sospende rotazione del piano di lavoro</i> | 163 |
| 17.12.12 | <i>G52 - Riprende rotazione del piano di lavoro</i> | 163 |
| 17.12.13 | <i>G50 - Rotazione del piano di lavoro</i> | 164 |
| 17.12.14 | <i>G1050 – Disabilita Scaling</i> | 165 |
| 17.12.15 | <i>G1051 - Scaling</i> | 165 |

| | | |
|--------------|--|------------|
| 17.12.16 | G103 – Parametri RTCP | 167 |
| 17.12.17 | G104 – Abilitazione RTCP..... | 167 |
| 17.12.18 | G104.1 – Abilitazione RTCP..... | 167 |
| 17.12.19 | G105 – Sospensione RTCP..... | 168 |
| 17.13 | CICLI FISSI | 169 |
| 17.13.1 | G1080 – Disabilita Cicli Fisso in Corso | 169 |
| 17.13.2 | G1081 – Ciclo di Foratura | 169 |
| 17.13.3 | G1082 – Ciclo di Foratura | 170 |
| 17.13.4 | G1083 – Ciclo di Foratura | 171 |
| 17.13.5 | G1084 – Ciclo di Maschiatura | 172 |
| 17.14 | COMPENSAZIONE OFFSET UTENSILE..... | 173 |
| 17.14.1 | G41/G42 – Compensazione offset utensile Sinistra/Destra | 173 |
| 17.14.2 | G40 – Disabilita Compensazione offset utensile | 173 |
| 17.14.3 | D - Correttore utensile | 174 |
| 17.14.4 | G47 – Disabilita/Abilita impegno utensile..... | 175 |
| 17.15 | COMPENSAZIONE LUNGHEZZA UTENSILE | 176 |
| 17.15.1 | G43 – Attiva correzione lunghezza utensile da parametro..... | 176 |
| 17.15.2 | G44 – disattiva correzione lunghezza utensile da parametro G43 | 177 |
| 17.15.3 | G44.1(2) – Sospende/Riprende G43 | 177 |
| 17.15.4 | G45 – Attiva correzione zero utensile da tabella T | 178 |
| 17.15.5 | G46 – disattiva correzione lunghezza utensile tabella G45..... | 179 |
| 17.16 | MODI DI INTERPOLAZIONE | 180 |
| 17.16.1 | G60 – Interpolazione veloce senza fermata sul tratto..... | 180 |
| 17.16.2 | G61 – Interpolazione con fermata sul tratto | 180 |
| 17.16.3 | G62 – Attesa assi fermi..... | 180 |
| 17.16.4 | G63 – Interpolazione fuori dal piano di lavoro | 181 |
| 17.16.5 | G64 – Interpolazione sul piano di lavoro (default) | 181 |
| 17.16.6 | G65 –Abilita interpolazione 3D – Px_moveto con fermata decisa dal CN | 182 |
| 17.16.7 | G75 –Abilita interpolazione 2D sul piano e 3D fuori dal piano – Px_moveto con fermata decisa dal CN 182 | 182 |
| 17.16.8 | G66 – AFC – Adaptive Feed Control..... | 183 |
| 17.16.9 | G66 X-100 – NEW AFC – Adaptive Feed Control | 184 |
| 17.16.10 | G67 – Usa Px_Moveto per movimenti fuori dal piano – Px_Lineto per movimenti nel piano..... | 188 |
| 17.16.11 | G68 – Usa solo– Px_Lineto per movimenti nel piano e fuori dal piano – ASSI TRASPORTATI con possibilità di combinare con px_moveto | 188 |
| 17.16.12 | G69 – LHK – Profondità buffer look ahead | 189 |
| 17.17 | FILTRI DI ISOUS | 190 |
| 17.17.1 | G72 – N.U.R.B.S (Non Uniform Rational Bspline) (2D 3D) | 191 |
| 17.17.2 | G73 – NOISE (2D - 3D) | 194 |
| 17.17.3 | G74 – RLS Remove Len Segment (2D 3D) | 196 |
| 17.17.4 | G106 – SMOOTHING | 197 |
| 17.18 | ASSI VIRTUALI..... | 200 |
| 17.18.1 | G100 – Comando sincrono per asse virtuale | 200 |
| 17.19 | GESTIONE MACRO AD INTERRUPT | 201 |
| 17.19.1 | G107 – Gestione Macro ad Interrupt..... | 201 |
| 17.20 | GESTIONE MACRO DI EMERGENZA IN STOP M60000 | 204 |
| 17.21 | G108 GESTIONE ASSI SPECIALI | 205 |
| 17.21.1 | G108.0 G108.1 G108.2 G108.3 – Gestione Assi Master Slave..... | 205 |

| | | |
|--------------|---|------------|
| 17.21.2 | <i>G108.4 G108.5 G108.6 Gestione Assi in velocità</i> | 206 |
| 17.21.3 | <i>G108.7 G108.8 G108.9 Scambio fisico Assi</i> | 206 |
| 17.21.4 | <i>G108.10 G108.11– Gestione Assi eGear</i> | 207 |
| 17.21.5 | <i>G108.12 G108.13 – Gestione velocità periferica assi rotativi</i> | 208 |
| 17.22 | FUNZIONI G DI UTILIZZO GENERICO | 209 |
| 17.22.1 | <i>G4 – Pausa temporizzata</i> | 209 |
| 17.22.2 | <i>G4.1 – Addizionale Tempi su Calc Time</i> | 209 |
| 17.22.3 | <i>G10 – Abilitazione potenziometro esterno di OVERRIDE assi</i> | 209 |
| 17.22.4 | <i>G11 – Disabilitazione potenziometro esterno di OVERRIDE assi</i> | 210 |
| 17.22.5 | <i>G101 – Stop forzato degli assi</i> | 211 |
| 17.22.6 | <i>G80 – Pausa forzata da codice ISO</i> | 211 |
| 17.22.7 | <i>G81 – Gestione LIMITI secondari ASSI</i> | 212 |
| 17.22.8 | <i>G20 – Programmazione quote assi in Pollici</i> | 212 |
| 17.22.9 | <i>G21 – Programmazione quote assi in Millimetri</i> | 212 |
| 17.23 | FUNZIONI PER GESTIONE VARIABILI PERMANENTI | 213 |
| 17.23.1 | <i>LOAD_VAR – carica un file di variabili precedentemente salvato</i> | 213 |
| 17.23.2 | <i>GET_VAR – Legge una variabile dalla lista caricata</i> | 213 |
| 17.23.3 | <i>WRITE_VAR – Scrive una variabile nella lista caricata</i> | 214 |
| 17.23.4 | <i>SAVE_VAR – salva un file di variabili dalla lista attuale</i> | 214 |
| 17.23.5 | <i>FILE_EXISTS – ritorna se il file indicato esiste</i> | 214 |
| 17.23.6 | <i>ADD_VAR – Aggiunge un valore alla lista attuale</i> | 215 |
| 17.23.7 | <i>REMOVE_VAR – Rimuove un valore alla lista attuale</i> | 215 |
| 17.23.8 | <i>CLEAR_VAR – Rimuove tutti i valori dalla lista attuale</i> | 215 |
| 17.23.9 | <i>DIM_VAR – Dimensiona la lista attuale al numero di elementi indicato</i> | 216 |
| 17.23.10 | <i>COUNT_VAR – Ritorna il numero di elementi presenti nella lista</i> | 216 |
| 17.24 | FUNZIONI HM | 217 |
| 17.24.1 | <i>Chiamata di una funzione HM</i> | 217 |
| 17.24.2 | <i>Costruzione di una funzione HM</i> | 218 |
| | <i>ESEMPIO DI COSTRUZIONE DI UN FUNZIONE HM</i> | 218 |
| 17.25 | FUNZIONI M | 219 |
| 17.25.1 | <i>Funzioni M interne al CN</i> | 219 |
| 17.25.2 | <i>Funzioni M interne al PC</i> | 220 |
| | <i>ESEMPIO DI COSTRUZIONE DI UN FUNZIONE M</i> | 220 |
| 17.26 | CONFIGURAZIONI M PRINCIPALI | 221 |
| 17.26.1 | <i>M Di START</i> | 221 |
| 17.26.2 | <i>M Di END</i> | 221 |
| 17.26.3 | <i>M Di STOP</i> | 221 |
| 17.26.4 | <i>M Di PAUSA</i> | 221 |
| 17.26.5 | <i>M Di RIPRESA DA PAUSA</i> | 222 |
| 17.26.6 | <i>M Di RIPRESA DA BLOCCO</i> | 222 |
| 17.26.7 | <i>M Di RIPRESA DA RETRACE</i> | 222 |
| 17.27 | DEFINIZIONE ASSE DI PROFONDITA' | 223 |
| 17.27.1 | <i>G48 – Definisce asse di profondità</i> | 223 |
| 17.28 | MILD MODE – SMORZAMENTO SPIGOLI | 224 |
| 17.28.1 | <i>G49 – Gestione MILD MODE</i> | 224 |
| 17.29 | TASK PARALLELI | 226 |
| 17.29.1 | <i>USTASK - ENDUSTASK</i> | 226 |
| 17.29.2 | <i>TASK.RUN</i> | 226 |

| | | |
|--------------|--|------------|
| 17.29.3 | TASK.STOP | 226 |
| 17.29.4 | TASK.PAUSE | 227 |
| 17.29.5 | TASK.READVAR | 227 |
| 17.29.6 | TASK.WRITEVAR | 227 |
| 17.29.7 | TASK.STATUS | 227 |
| 17.29.8 | TASK.LOADCMD | 228 |
| 17.29.9 | TASK.PRIORITY | 228 |
| 17.30 | PARAMETRI MACCHINA ESSENZIALI | 229 |
| 17.30.1 | Parametri generali | 229 |
| 17.30.2 | Parametri relativi all' asse | 233 |
| 17.31 | PARAMETRI PID | 242 |
| 17.31.1 | PID_KP | 242 |
| 17.31.2 | PID_KI | 242 |
| 17.31.3 | PID_KV | 242 |
| 17.31.4 | PID_I_LIMIT | 242 |
| 17.31.5 | PID_DIV | 242 |
| 17.31.6 | PID_SERVO | 242 |
| 17.31.7 | PID_TIME_SERVO | 242 |
| 17.31.8 | PID_DIR | 242 |
| 17.31.9 | PID_OFFS_ANA | 242 |
| 17.32 | PARAMETRI SPINDLE | 243 |
| 17.32.1 | SPEEDMAXSPINDLE | 243 |
| 17.32.2 | ANALOG_BIT_RES | 243 |
| 17.32.3 | SPEED_ANALOG_CH | 243 |
| 17.32.4 | ENABLE_OW_SPEED | 243 |
| 17.32.5 | SPEED_OW_MIN | 243 |
| 17.32.6 | SPEED_OW_MAX | 243 |
| 17.33 | PARAMETRI EGEAR | 244 |
| 17.33.1 | EGEAR_KEM | 244 |
| 17.33.2 | EGEAR_KED | 244 |
| 17.33.3 | EGEAR_ACC | 244 |
| 17.34 | PARAMETRI VELOCITÀ PERIFERICA | 244 |
| 17.34.1 | CENTRE_X_R1 | 244 |
| 17.34.2 | CENTRE_Y_R1 | 244 |
| 17.34.3 | CENTRE_Z_R1 | 244 |
| 17.34.4 | CENTRE_X_R2 | 244 |
| 17.34.5 | CENTRE_Y_R2 | 244 |
| 17.34.6 | CENTRE_Z_R2 | 244 |