

Volume

3

AxesBrain

Configurazione e caratterizzazione



Configurazione e caratterizzazione

AxesBrain

Configurazione e caratterizzazione

AB&T Tecnologie Informatiche™
Via dell'About, 2/A • 10015 Ivrea
Fax 0125 234397
www.bausano.net
info@bausano.net

Informazioni legali

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

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

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

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

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

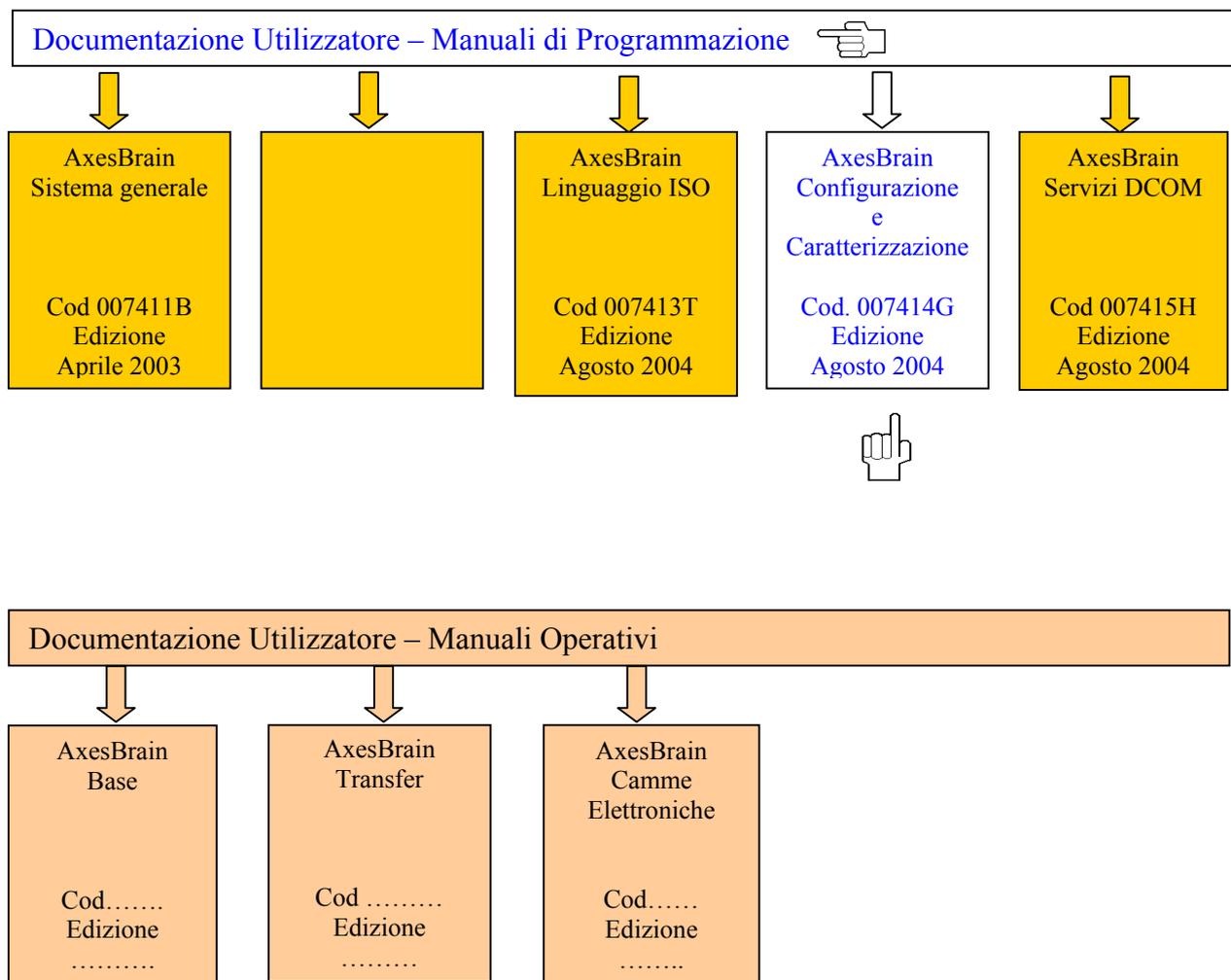
AxesBrain™, VisAlgo™, ScadaMERCURIO™ sono marchi registrati

©AxesBrain, ©VisAlgo sono tutelati da copyright

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

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

Panoramica della Documentazione Utilizzatore **Configurazione e caratterizzazione**



INDICE DEGLI ARGOMENTI

1	INTRODUZIONE.....	1-1
1.1	COME UTILIZZARE QUESTO MANUALE.....	1-1
1.2	SISTEMA TXT	1-2
2	CONFIGURAZIONE	2-1
2.1	PARAMETRI GENERALI	2-1
2.1.1	<i>Parametri Generali Tabella riassuntiva.....</i>	<i>2-8</i>
2.2	SEGNALI I/O (DIGITALI E ANALOGICI)	2-12
2.2.1	<i>Segnali I/O digitali</i>	<i>2-13</i>
2.2.2	<i>Segnali Analogici I/O</i>	<i>2-22</i>
2.2.3	<i>Parametri degli I/O digitali e analogici - Tabelle riassuntive.....</i>	<i>2-26</i>
2.3	ASSI E MANDRINI	2-41
2.3.1	<i>Assi</i>	<i>2-41</i>
2.3.2	<i>Mandrini</i>	<i>2-89</i>
2.3.3	<i>Assi virtuali.....</i>	<i>2-98</i>
2.4	PARAMETRI SPECIFICI DEGLI AMBIENTI	2-109
2.5	PREDEFINIZIONI ISO	2-110
3	CARATTERIZZAZIONE	3-127
3.1	PID	3-128
3.2	ANTICOLLISIONE	3-131
3.3	GANTRY	3-134
3.4	HOMING	3-138
3.5	DATI CAMPIONATI.....	3-144
3.6	READ AXES PARAMETERS.....	3-146
3.7	CHANGE AXES PARAMETERS	3-153
3.8	ASSI ROLLER	3-167
3.9	OVER TRAVEL	3-168
3.10	ASSI ASSERVITI SOLO IN MOVIMENTO.....	3-170
4	CONFIGURAZIONE NOME ASSI PER INTERFACCIA MANUALE	4-1
5	CICLI SEMPRE ATTIVI	5-1
6	CORREZIONE ASSI	6-1
6.1	LINEARE.....	6-2
6.2	GRIGLIA	6-4
7	TARATURE ASSI – PID (PROPORZIONALE, INTEGRATIVO, DERIVATIVO)	7-1
7.1	GENERALITÀ E PREMESSE	7-1
7.1.1	<i>CONCETTI PRELIMINARI SULLE MASCHERE.....</i>	<i>7-3</i>
7.2	CARATTERIZZAZIONE ASSI ANALOGICI IN VELOCITA'	7-5

7.2.1	TARATURE GENERALI DI SISTEMA RELATIVE ATUTTI GLI ASSI	7-5
7.2.2	CARATTERIZZAZIONE ASSI FISICI - PREDISPOSIZIONI.....	7-5
7.2.3	VALORI E COEFFICIENTI PER ASSERVIMENTO DINAMICO ASSI.....	7-12
7.2.4	CONTROLLO ED ERRORI ASSI	7-18
7.2.5	AZZERAMENTO ASSI, FINE CORSA E BACKLASH.....	7-22
7.2.6	GESTIONE DELLA FRENATURA ASSI	7-25
7.2.7	ANTICOLLISIONE	7-26
7.2.8	ASSI SLAVE – GANTRY	7-29
7.2.9	TARATURA MOTORI LINEARI	7-31
7.3	TARATURA IN CORRENTE	7-33
7.3.1	PREDISPOSIZIONI.....	7-33
7.3.2	COEFFICIENTI E SIGNIFICATI PER LA TARATURA IN CORRENTE	7-33
7.4	TARATURA STEP-MOTORS	7-37
7.4.1	TARATURE PER ASSERVIMENTO DINAMICO ASSI STEPPER.....	7-39
7.5	TRACCIA SU COME PROCEDERE NELLA TARATURA ASSI.....	7-2
7.5.1	Asservimento in velocità	7-3
7.5.2	Asservimento in coppia.....	7-5
7.5.3	Considerazioni finali.....	7-6
8	VISIONE.....	8-8
A.	ELENCO ERRORI.....	1
B.	ESEMPIO “SISTEMA.TXT”	1

CAPITOLO 1

1 INTRODUZIONE

1.1 Come utilizzare questo manuale

Questo manuale descrive come utilizzare le funzioni dedicate alla caratterizzazione del sistema AxesBrain.

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

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

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

1.2 Sistema TXT

*Il prodotto "AxesBrain" si configura attraverso il file "**sistema.txt**" che viene collocato nella sottodirectory "DAT" della directory dell'ambiente "AxesBrainStudio"*

Per utilizzare i servizi messi a disposizione da AxesBrain si devono configurare i parametri degli assi degli I/O digitali ed analogici caratterizzando i valori degli assi dei mandrini e degli I/O.

E' utile prima configurare il sistema in termini di assi e mandrini, di I/O sia digitali che analogici, poi caratterizzarli secondo le specifiche richieste.

Il sistema fornisce un "**sistema.ini**" con la configurazione fornita, che dovrà essere caratterizzata secondo le esigenze specifiche (nomi dei segnali, nome degli assi, parametri velocità eccetera.).

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

In particolare esso è così organizzato:

Il **Capitolo 2 Configurazione** descrive i parametri da configurare per poter gestire le risorse del sistema.

Il **Capitolo 3 Caratterizzazione** descrive i parametri da configurare per caratterizzare il sistema.

Il **Capitolo 4 Manuale** fornisce i rudimenti essenziali su come configurare gli assi e i relativi gruppi presenti nell'applicazione desiderata.

Il **Capitolo 5 Cicli sempre attivi** fornisce le informazioni necessarie alla configurazione delle stringhe di lancio per i parametri PLC e AUTOMAZIONE.

Il **Capitolo 6 Correzione assi** fornisce le informazioni necessarie alla configurazione dei parametri di correzione Lineare e Griglia.

Il **Capitolo 7 Tarature assi** illustra i concetti base inerenti alla taratura degli assi e mandrini.

Il **Capitolo 8 Visione** illustra i concetti base inerenti il pacchetto VisAlgo con funzione di analisi di immagine.

L' **Appendice A** è un elenco sottoforma di lista di tutti gli allarmi che possono essere generati dal sistema in fase di programmazione e caratterizzazione.

L' **Appendice B** descrive un esempio di file *sistema.txt*

CAPITOLO 2

2 Configurazione

Nel File “Sistema.txt” sono presenti tutti i parametri da caratterizzare suddivisi in:

Parametri Generali – Informazioni che riguardano tutta l’applicazione, quali ad esempio: Campionatura, n. di decimali, ecc

Parametri degli I/O digitali e analogici – Informazioni sul tipo di I/O, nomi assegnati ai segnali, n. del canale fisico utilizzato ecc.

Parametri relativi agli assi – Informazioni relative alla individuazione del tipo di scheda e di asse (analogico, sercos, ecc.), valori di taratura PID ecc

Parametri relativi agli assi virtuali – Informazioni relative alla caratterizzazione degli assi che necessitano di una trasformazione delle coordinate nel piano (Assi SCARA, Cilindrici ecc.)

Parametri Mandrini – Informazioni relative al canale di pilotaggio utilizzato (es. su scheda assi analogici, via CAN, ecc.), N. di giri, accelerazione, ecc.

Parametri specifici degli Ambienti – Informazioni sulle condizioni di lancio del part program di automazione

Definizione dei “processi ISO”- Predefinizioni ISO

2.1 Parametri generali

I parametri generali sono i primi contenuti nel file “Sistema.TXT” e riguardano dei registri interni utilizzati da “AxesBrain”.

Sono indicati sotto la voce: **[ParametriGenerali]**

Di seguito vi è un esempio per poterne definire il significato:

ESEMPIO

[ParametriGenerali]

Campionatura=1000
NumeriDecimali=3
NumeriCifre=10
TolleranzaCerchio=1000
KN=-1.0

```
NUMBER_SAMPLE_SWITCH_PORT=""
FALSETRUE_LOGIC_SAMPLE_SWITCH=1
SegnalePotenza=""
RitardoPotenza=5.0
NumeroLocali=32
TickMicro=1
GeneralStatus=0
TickCounter=1
FALSETRUE_LOGIC_SegnalePotenza=1
NumeriDecimaliDRT=3
NumeriCifreDRT=7
NumeriDecimaliFILE=4
NumeriCifreFILE=10
NumeroByteReadFieldBus=8
NumeroByteWriteFieldBus=8
Time_collision_latency=0.2
FileCorrezione=Correz.txt
IO_PLC_management=0
NumeroIstruzioniMax=128
PartProgramG89=G89.PP
VolantinoECS=1
GraficaCNC=1
ModuloFieldBusHilscher=1
ModuloFieldBusSofting=0
FlagFastWaitIO=0
GlobaleNumeroAllarmi=9999
TempoLatenzaErrori=1
FlagHoldG8183=1
ECSVolantinoSuRapido=0
WaitSEC=0.03
TempoScansione=0.010
SeGlobaleErroreNoDISM1=1
GlobaleBloccoMovimentazione=0
GlobaleLingua=0
```

Vediamoli in dettaglio:

Campionatura = 500 valore della campionatura assi in microsecondi
NumeriDecimali = 3 valore di default di decimali usato dal sistema
NumeriCifre = 10 valore di default di cifre usato dal sistema
TolleranzaCerchio = 1 il valore indica il massimo errore tra raggio iniziale del cerchio e quello finale.
KN = valore che di default è -1 in questo caso i valori interni del sistema si autotitano, altrimenti si determinano secondo il fattore indicato.

NUMBER_SAMPLE_SWITCH_PORT = "" se indicato il nome del segnale di output viene attivato (vedi **FALSETRUE_LOGIC_SAMPLE_SWITCH** se 1 o 0) quando il sistema è attivo, (potenza attiva con abilitazione degli assi).

FALSETRUE_LOGIC_SAMPLE_SWITCH = 1 attivo a uno, 0 attivo a zero

SegnalePotenza = "" se indicato come riferimento ad un segnale di input digitale, la potenza della macchina (abilitazione degli assi) verrà data dal sistema dopo un tempo indicato in secondi dal valore successivo

RitardoPotenza = 5.0 valore di ritardo come segnale automatico di potenza

NumeroLocali = 32 numero di VARIABILI LOCALI abbinato ad ogni part program di automazione (-DIM per cambiare in modo locale questo parametro)

TickMicro = Numero di Tick per la microcampionatura (solo il PID), normalmente 1 , se il sistema necessita un campionamento più fitto aumentare di qualche unità.

GeneralStatus = Maschera generale di sistema a bits

;Maschera generale di sistema a bits

;0x00000001 = Auto attivazione di USB MAVEL

;0x00000002 = Disattivazione del WATCH DOG con USB MAVEL

;0x00000004 = Gestione di DEADLOCK senza ERRORE (silente)

;0x00000008 = Gestione CFR a livello (modo nuovo)

;0x00000010 = Non viene resettato OFFSET RELATIVO

;0x00000020 = In emergenza i movimenti non sono resettati

;0x00000040 = Gestione a livello dell'input EXEC_MOV

;0x00000080 = Dopo cfr le rampe sono lineari sino a vel. costante
 ;0x00000100 = Il valore "TOUCH" con DCS DCT e' la posiz. reale
 (DEBUG)
 ;0x00000200 = CFR con cambio immediato della velocita'
 ;0x00000400 = CAMPIONATURA con sbordamento controllato
 ;0x00000800 = Utilizzo della memoria esterna (_AllocMemory)

**** ATTENZIONE non deve essere messo il commento // dopo il numero di
 GeneralStatus= perche' viene passato uno zero

TickCounter = Numero di tick della campionatura la gestione dei motori STEP usando i segnali I/O

NumeriDecimaliDRT = Numero decimali nella visualizzazione dei dati attivati dall'istruzione DRT

NumeriCifreDRT = Numero cifre nella visualizzazione dei dati attivati dall'istruzione DRT

NumeriDecimaliFILE = Numero decimali nella visualizzazione dei dati attivati dall'istruzione FWR

NumeriCifreFILE = Numero decimali nella visualizzazione dei dati attivati dall'istruzione FWR

NumeroByteReadFieldBus = Numero di byte letti nel gestione del BUS di campo

NumeroByteWriteFieldBus = Numero di byte scritti nel gestione del BUS di campo

FileCorrezione = Nome file di correzione assi sia lineare che di quadratura

Time_collision_latency = Tempo in secondi di ritardo per prima di disabilitare gli azionamenti, il comando di disabilitazione mette il riferimento degli azionamenti a zero,effettua il time_collision_latency, toglie le abilitazioni.

IO_PLC_management = Flag per l'abilitazione della gestione dell'acquisizione di tutti gli IO Digitali e Analogici con la gestione dei tre segnali per la Pagina segnali PLC.

NumeroIstruzioniMax = Per evitare che una sequenza di istruzioni logiche e di controllo condizionino il tempo dedicato dalla CPU, viene indicato il numero di istruzioni nel quale se il task non viene sospeso, il sistema forza una pausa di **TempoScansione** millisecondi per il part program in esame.
 Default = 64

TempoScansione = Tempo di sospensione automatico per esecuzione continua del numero di istruzioni indicato **NumeroIstruzioniMax**, senza sospensione.
 Default = 0.050

PartProgramG89 = Nome del part program che viene eseguito con le modalita' dei cicli fissi.
 Default = G89.pp

VolantinoECS	= Attiva il volantino da console ECS
GraficaCNC	= Attiva la segnalazione degli enti eseguiti
ModuloFieldBusHilscher	= Attiva la scheda Hilscher per la gestione del FieldBus
ModuloFieldBusTCP-IO	= Attiva la gestione del FieldBus TCP-IO
ModuloFieldBusSofting	= Attiva la scheda Softing per la gestione del FieldBus Can
FlagFastWaitIO	= Valore che attiva la modalità fast per le istruzioni “WDI” attive. Default = 0
GlobaleNumeroAllarmi	= Il numero indicato se diverso da zero indica la Globale nella quale il sistema carica il numero di allarmi presenti. Default = 0

- TempoLatenzaErrori** = Tempo in secondi che deve intercorrere tra due messaggi con lo stesso numero d'errore generati tramite "SEC/SET_MESSAGE,#", per attualizzarlo sul zona di video dedicata.
Default = 1
- FlagHoldG8183** = Flag che definisce, se a 1 che i cicli fissi di G81,G82 e G83 possono andare in HOLD
Default = 0
- ECSVolantinoSuRapido** = Abilita la gestione della manopola di R come volante
- WaitSEC** = Tempo di sospensione automatico che viene introdotto ad ogni istruzione di "SEC" e di "DIS/-1."
Default = 0.03
- SeGlobaleErroreNoDISM1** = Per evitare la sovrascrittura nella parte di segnalazione con l'istruzione -DIS/-1: se sono presenti segnalazioni, verificando la **GlobaleNumeroAllarmi**, il sistema non invia il messaggio.
Default = 1
- GlobaleBloccoMovimentazione** = Il numero indicato se diverso da zero indica la Globale nella quale il sistema verifica l'inibizione dei movimenti manuali (non programmati), nel caso caso che il valore sia diverso da zero 0.
Default = 0
- Time_collision_latency** = Tempo in secondi di permanenza asservimento con riferimento "asse fermo" per ogni SERVO OFF
- Time_default_lock_wait** = Tempo in secondi di permanenza asservimento a fine movimento con gli assi "DEFAULT LOCK"
- NoDisplayMachineGroupError** = Disabilita se è ad 1 la visualizzazione della segnalazione anomalia assi o mandrini abbinati ai Gruppi Macchina. Default =0 (Visualizza l'anomalia)

ReturnLocalFromEndTask = Abilita il ritorno dei valori delle Locali al part program chiamante alla fine dell'esecuzione del TASK attivato con l'istruzione: TSK/III,xxxx:L1,Lo.(Default 1 = Ritorno dei valori nelle Locali)

Time_NoCheckOverRunMicro = Tempo in secondi prima di segnalare l'anomalia quando si attivano gli assi (Enable agli azionamenti) per gli assi in oltre corsa. E' così possibile spostare l'asse dalla posizione di oltre corsa, con il verso nella direzione opposta e nel tempo indicato in questo parametro.

NumberRetryFieldbus=Numero di tentativi di riconnessione alla periferica di FieldBus prima di interrompere la comunicazione.

Newto_IOErrCnt=Globale su cui vengono incrementati gli errori di comunicazione con la scheda NEWTO o 0

Newto_WTDogEnable=Globale che se posta a 1 si arresta l'attualizzazione degli Output Digitali

GlobaleLingua = Numero della Globale su cui viene caricato il numero della lingua attiva ed in particolare:

1=Italiano

2=Inglese

3=Tedesco

4=Francese

5=Spagnolo

Se il numero è 0 non viene fornito tale servizio, Default 0

2.1.1 Parametri Generali Tabella riassuntiva

Voci da caratterizzare	Spiegazione
[ParametriGenerali]	Nome del tipo di Caratterizzazione
Campionatura=500	Espressa in microsecondi
NumeriDecimali=3	
NumeriCifre=10	
TolleranzaCerchio= 0,01	Tolleranza tra tipica tra raggio iniz.e finale
KN= -1.0	NON MODIFICARE
NUMBER_SAMPLE_SWITCH_PORT=""	Indicare il NOME dell'Output che si utilizza nella applicazione per permettere l'abilitazione della potenza - Segnala che il governo è attivo a regime.
FALSETRUE_LOGIC_SAMPLE_SWITCH=1	Stato logico del segnale NUMBER_SAMPLE_SWITCH_PORT
SegnalePotenza=""	Input di Servo ON che segnala "Cabina elettrica sotto potenza"
FALSETRUE_LOGIC_SegnalePotenza=1	Stato logico del Segnale Potenza
RitardoPotenza=2.0	Ritardo (sec.) prima di asservire gli Assi
NumeroLocali=32	N. di Variabili Locali assegnate per default ad ogni P.Program
TickMicro=1	NON MODIFICARE
TickCounter=1	NON MODIFICARE
GeneralStatus =0	Maschera di definizione degli stati generali del Sistema con i seguenti significati: GeneralStatus =0 Modo normale GeneralStatus =4 Dead-lock in anticollision_ non segnala errore GeneralStatus =16 (=0x10 in esadecimale) Gli offset relativi, dati dai Part Program non vengono annullati in seguito a Reset.
NumeriDecimaliDRT=3	N. di decimali (3) visualizzati nei valori delle quote e delle Globali sulla videata DRT (Display Real Time) della pagina Automazione dell'ambiente "Automazione"

	Valore di default = 3 NORMALMENTE NON MODIFICARE
NumeriCifreDRT=7	N. di cifre massimo della videata DRT. Valore di default = 7 NON MODIFICARE
NumeriDecimaliFILE=4	N. di decimali (4) del formato dei numeri scritti nei files tramite istruzioni Valore di default = 4 NORMALMENTE NON MODIFICARE
NumeriCifreFILE=10	N. di cifre del formato dei numeri scritti nei files tramite istruzioni . Valore di default = 10 NORMALMENTE NON MODIFICARE
NumeroByteReadFieldBus=32	Valore tipico del CAN
NumeroByteWriteFieldBus=32	Valore tipico del CAN
Time_collision_latency=0.2	Tempo di ulteriore tenuta in asservimento degli assi in caso di emergenza
FileCorrezione=Correz.txt	Nome del File in cui vengono registrate le correzioni di precisione degli assi
IO_PLC_management=0	NON MODIFICARE
NumeroIstruzioniMax=64	N. massimo di istruzioni eseguite in sequenza all'interno di ogni task prima di sospendere il Task per 1 ciclo – NORMALMENTE NON MODIFICARE
PartProgramG89=G89.PP	Nome del P.Program ISO lanciato dal ciclo fisso G89 (nell'esempio il P.Program si chiama G89.PP)
VolantinoECS=0	Se=0 il Volantino ECS non si collega tramite la Tastiera Se=1 il Volantino è connesso ma usa anche il codice del tasto F11 Valore di default = 0
GraficaCNC=1	Se=1 la grafica in ambienteCNC è abilitata Se=0 è disabilitata. Valore di default = 1 NORMALMENTE NON MODIFICARE
ModuloFieldBusHilscher=1	Se=1 è abilitata la scheda Field-bus Hilscher Se=0 è disabilitata Valore di default= 1
ModuloFieldBusTCP-IO =0	Se=1 è abilitata la scheda Field-bus Moduli TCP-IO Se=0 è disabilitata Valore di default= 0

ModuloFieldBusSofting=0	Se=1 è abilitata la scheda Field-bus Softing Se=0 è disabilitata Valore di default=0
FlagFastWaitIO=1	Se =1 abilita la gestione veloce dell'istruzione FWI Se=0 la FWI opera come la WDI Valore di default =1 Nota-Cautela nell'uso di FWI
GlobaleNumeroAllarmi=0	Numero della Globale in cui il sistema registra in modo puntuale quanti allarmi sono presenti Valore di default=0
TempoLatenzaErrori=1	Espresso in secondi.per dar tempo di visualizzare errori diversi. Valore di default =1(sec.)
FlagHoldG8183=0	Se=1 permette di abilitare l'andata in HOLD nei cicli fissi G81-82-83. Valore di default=0
ECSVolantinoSuRapido=0	Se=1 l'encoderino di tastiera ECS (quello di Rapido) è abilitato. Valore di default=0
WaitSEC=0.03	Tempo di ritardo automatico nelle trasmissioni dell'istruzione SEC. Valore di default=0.03 (secondi)
TempoScansione=0.050	Tempo di scansione degli I/O (in sec.) Valore di default=0.050 NON MODIFICARE
SeGlobaleErroreNoDISM1=1	Se=1 protegge la visualizzazione degli errori nella striscia DIS/-1:.....da sovrapposizioni- Valpore di default=1
GlobaleBloccoMovimentazione=0	Numero di una globale che, se vale zero non inibisce la movimentazione degli assi. Se invece=1 la inibisce Valore di default=0 (nessuna globale)
Lingua=ITA	E' la lingua di tutti i messaggi e le scritte del sistema. Default=ITA ITA= italiano ENG=Inglese ecc
AbilitaMinimizzazione=1	Se=1 la minimizzazione a icona è abilitata. Se=0 non sipuò minimizzare il quadro Valore di default=1

NoDisplayMachineGroupErro =0	Disabilita se è ad 1 la visualizzazione della segnalazione anomalia assi o mandrini abbinati ai Gruppi Macchina. Default =0 (Visualizza l'anomalia)
Time_wait_disabile_AxesDefaultLock= 0	Tempo(sec.) di permanenza della abilitazione per gli assi che vengono predisposti "NORMALMENTE DISABILITATI QUANDO SONO FERMI ". (Vedi taratura Kind_Axis_0=0x00000080) Valore di default=0
ReturnLocalFromEndTask = 1	Abilita il ritorno dei valori delle Locali al part program chiamante alla fine dell'esecuzione del TASK attivato con l'istruzione: TSK/III,xxxxx:L1,Lo.(Default 1 = Ritorno dei valori nelle Locali)
Time_NoCheckOverRunMicro = 0	Tempo in secondi prima di segnalare l'anomalia quando si attivano gli assi (Enable agli azionamenti) per gli assi in oltre corsa. E' così possibile spostare l'asse dalla posizione di oltre corsa, con il verso nella direzione opposta e nel tempo indicato in questo parametro. Valore di default=0
GlobaleLingua = 0	Numero della Globale su cui viene caricato numero della lingua attiva ed in particolare:

	<p>1=Italiano 2=Inglese 3=Tedesco 4=Francese 5=Spagnolo</p> <p>Se il numero è 0 non viene fornito tale servizio, Default 0</p>
--	--

2.2 Segnali I/O (digitali e analogici)

Segnali input ed output digitali ed analogici

Di seguito vi è un esempio per poterne definire il significato:

ESEMPIO

```
i=IO_NAME(501 , "R1" ) // A1
b={
Data_type="Word" // "Bit" "Byte" "Word" "Dword"
Description="LMInput"
TaskChangeFront=1 // 1=changefront 0=always
TaskChangeValue=0 // delta change value for analog signal
TaskModeRun=1 // 0=always 1=No if running 2=Error if running
TaskOnePP="InfoIO.pp"
TaskOneName="InOne"
TaskZeroPP="InfoIO.pp"
TaskZeroName="InZero"
Estension="1" // R1.1
Index_system=0
FalseTrue_logic=0
Kind_sensor="DIGITAL_INPUT"
Kind_board=0
Rack_number=0
Address_port=0
Physical_number=1
e=}
```

Vediamoli in dettaglio:

2.2.1 Segnali I/O digitali

Parte per la definizione del nome, numero e descrizione dell'IO

IO_NAME(indice NOME , "nome simbolico")

indice NOME deve essere compreso tra 1 e NUM_MAX_DI_BIT (INPUT)

indice NOME deve essere compreso tra 1 e NUM_MAX_DO_BIT (OUTPUT)

Data_type ="Word" ; "Bit" "Byte" "Word" "Dword"

Default Bit

Description ="LMInput"

Default ""

Estension ="1"

Default Non Attiva

Parte per l' attivazione automatica del cambiamento di stato può essere

TaskChangeFront=1 ; 1=changefront 0=always

TaskChangeValue=0 ; delta change value for analog signal

TaskModeRun=1 ; 0=always 1=No if running 2=Error if running

TaskOnePP="InfoIO.pp"

TaskOneName="InOne"

TaskZeroPP="InfoIO.pp"

TaskZeroName="InZero"

ZeroOnReset=1 viene posto a zero logico sul comando di RESET e le istruzioni

-RST/0 -RST/2

0 non effettua nessuna operazione

NoThreeState= 1 non viene messo in modalità threestate con l'istruzione -RST/4

Parte per la descrizione fisica dell'IO

Index_system = indice di sistema per usi futuri

FalseTrue_logic = 1 per bit in logica negata normalmente e' 0

Kind_sensor ="DIGITAL_OUTPUT" o "DIGITAL_INPUT"

Kind_board= tipo di digital I/O

- 0 ;VIRTUAL (PACCO K) 1 - 1024 BITS INPUT + 1024 BITS OUTPUT
- 1 ;PIASTRA I/O IP-UNIDIG-E-48
- 2 ;PIASTRA ASSI TIP102 I/O
- 3 ;USB MAVEL I/O
- 4 ;PIASTRA ASSI TPMC118 I
- 5 ;PIASTRA I/O FIELD BUS
- 6 ;PIASTRA I/O ETN PRIMA ELECTRONICS
- 7 ;PIASTRA ASSI YIDDISH MAVEL I/O
- 8 ;SCHEDA I/O pci 8255
- 9 ;SCHEDA I/O isa 8255
- 10 :(unused)
- 11 ;PIGGY BOARD I/O ETN su SERVO PRIMA ELECTRONICS
- 12 ;PIASTRA ASSI PRIMA ELECTRONICS ANALOGIC PASSIVE
- 13 ;PIASTRA ASSI PRIMA ELECTRONICS DIGITAL PASSIVE e ACTIVE
- 14 ;PIASTRA MLTA per DIGITAL I/O
- 15 ;PIASTRA ASSI MLTA per DIGITAL Input
- 17 ;PIASTRA I/O NEWTON (48 INPUT 48 OUTPUT)
- 18 ;PIASTRA ASSI NEWTON
- 20 ;PIASTRA ASSI SIX
- 21 ;PIASTRA I/O SIX
- 22 ;SERCOS I/O
- 23 ;YASKAWA
- 24 ;ETEL
- 25 ;PIASTRA I/O ETHAX (48 INPUT 48 OUTPUT)
- 26 ;PIASTRA ASSI ETHAX
- 27 ;PIASTRA ASSI UNIVERSALE
- 28 ;PIASTRA ASSI ETHERBOX
- 29 ;PIASTRA ASSI ETHERBOX (assi)
- 30 ;PIASTRA ASSI ETHERBOX_FRQ (assi)
- 100 ;VIRTUAL_MIX (PACCO K) 1 - 2048 BITS INPUT/OUTPUT
- 129 ; CanOpenDs401
- 528 ; Controller esterno

Board_number= numero piastra

Rack_number

- 0 ;numero non usato (0)
- 1 ;numero BOARD IP (0 - N)
- 2 ;numero BOARD IP (0 - N)
- 3 ;numero cestello (0 - 7)
- 4 ;numero non usato (0)
- 5 ;numero non usato (0)
- 6 ;numero non usato (0)
- 7 ;numero BOARD YIDDISH (0 - N)

8 ;numero BOARD I/O pci 8255
9 ;numero BOARD I/O isa 8255
10 ;unused
11 ;numero non usato (0)
12 ;numero BOARD PIASTRA ASSI PRIMA ELECTRONICS (0 - N)
13 ;numero BOARD PIASTRA ASSI PRIMA ELECTRONICS (0 - N)
14 ;numero BOARD MLTA (0 - N)
15 ;numero BOARD ASSI MLTA (0 - N)
17 ;numero BOARD NEWTON (0 - N)
18 ;numero BOARD ASSI NEWTON (0 - N)
20 ;numero BOARD ASSI SIX (0 - N)
21 ;numero BOARD I/O SIX (0 - N)
22 ;numero BOARD SERCOS (0 - N)
23 ;numero BOARD YASKAWA (0 - 3)
24 ;numero BOARD ETEL (0 - N)
25 ;numero BOARD ETHAX (0 - N)
26 ;numero BOARD ASSI ETHAX (0 - N)
27 ;numero BOARD ASSI UNIVERSALE (0 - N)
28 ;numero BOARD ASSI ETHERBOX (0 - 32)
29 ;numero BOARD ASSI (assi) ETHERBOX (0 - N)
30 ;numero BOARD ASSI (assi) ETHERBOX_FRQ (0 - N)
100 ;numero non usato (0)
129 ; numero non usato CanOpenDs401
528 ; numero non usato Controller esterno

Address_port= offset

0 ;Num WORD di 16 bit (0 - 128)
1 ;offset su Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
2 ;offset su Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
3 ;numero dispositivo USB MAVEL (0 - 7)
4 ;numero non usato (0)
5 ;numero dispositivo (0 - 767)
6 ;numero dispositivo (0 - 255)
7 ;numero dispositivo (0 - 15) (0 = locale)
8 ;numero dispositivo (0 - 5)
9 ;numero dispositivo (0 - 5)
10 ;unused
11 ;numero dispositivo (0 - 255)
12 ;numero asse PRIMA ELECTRONICS (1 - 12)
13 ;numero asse PRIMA ELECTRONICS (1 - 12)
14 ;numero dispositivo (0)
15 ;numero asse MLTA (1 - 6)
17 ;numero dispositivo (0)

18 ;numero asse NEWTON (1 - 4)
 20 ;unused
 21 ;unused
 22 ;numero asse SERCOS (1 - 8)
 23 ;numero asse YASKAWA (1 - 15)
 24 ;numero asse ETEL (1 - 4)
 25 ;numero dispositivo (0)
 26 ;numero asse ETHAX (1 - 4)
 26 ;numero asse ETHAX (1 - 4)
 27 ;numero asse UNIVERSALE (0 - N)
 28 : unused ETHERBOX (0)
 29 ;numero asse ETHERBOX (assi) (1-6)
 30 ;numero asse ETHERBOX_FRQ (assi) (1-6)
 100 ;Num WORD di 16 bit (0 - 256)
 129 ; numero dispositivo (0)
 528 ; numero dispositivo (0)

Physical_Number = posizione fisica del I/O su piastra

0 ;1 - 1024

1 ;1 - 48

2 ;1 - 16 INPUT (1-8 asse 1 1-4 OUTPUT (1-2 asse 1
 9-16 asse 2 3-4 asse 2)

3 ;1 - 64

4 ;1 - 6 solo INPUT

5 ;1 - 12288

6 ;1 - 1024

7 ;1 - 32 INPUT 1-32 OUTPUT (1-8 asse 1
 9-16 asse 2
 17-24 asse 3
 25-32 asse 4)

8 ;1 - 48

9 ;1 - 48

```

10 ;unused
-----
11 ;1 - 1024
-----
12 ;1 - 8 TheoSpeed_R      (1) Driver OK   input
    IOREG                 (3) ERRF         "
                            (4) OM           "
                            (5) FCA         "
                            (6) FCI         "
                            (7) FAULT        "
                            (8) ZEROF        "
;1 - 1 TargetSpeed_W      (1) Servo Enable output
-----
13 ;1 - 32 INPUT          (1) Driver enabled
                            (2) Driver in fault
                            (6) Ricerca Zero OK
                            (9) Driver is present
                            (10) Digital driver is phased
                            (11) Zero axis executed
                            (12) Zero micro signal
                            (13) Hardware end of stroke forward
                            (14) Hardware end of stroke backward
                            (15) Power is present on digital driver
                            (16) Hardware enable for driver not present
;1 - 32 OUTPUT            (1) Servo Enable
-----
14 ;1 - 16 ( 1-8 INPUT   9-16 OUTPUT )
-----
15 ;1 - 8 ( 1-8 INPUT )
-----
17 ;1 - 48 ( 1-48 INPUT   1-48 OUTPUT )
-----
18 ;1 - 3 ( 1-3 INPUT ) ( 1 OUTPUT )
-----
20 ;1 - 16   OUTPUT      1 Enable asse 1      85
                        2 Enable asse 3      86
                        3 Enable asse 5      87
                        4 Boost 1
                        5 Enable asse 2      35
                        6 Enable asse 4      36
                        7 Enable asse 6      37
                        8 Boost 2
                        9 DIR STEPPER 1
                       10 DIR STEPPER 2

```

- 11 DIR STEPPER 3
- 12 Boost 3
- 13 STEP STEPPER 1
- 14 STEP STEPPER 2
- 15 STEP STEPPER 3
- 16 Watch DOG

INPUT	1	Switch OMO asse 1	82
	2	Switch OMO asse 2	32
	3	Switch OMO asse 3	83
	4	Switch OMO asse 4	33
	5	Switch OMO asse 5	84
	6	Switch OMO asse 6	34
	7	Unused	
	8	Unused	
	9	Driver OK asse 1	88
	10	Driver OK asse 2	38
	11	Driver OK asse 3	89
	12	Driver OK asse 4	39
	13	Driver OK asse 5	90
	14	Driver OK asse 6	40
	15	Unused	
	16	Unused	

21 ;1 - 40	OUTPUT	1	Enable STEP 1	91	GND24
		2	Enable STEP 2	93	GND24
		3	Enable STEP 3	95	GND24
		4	Boost 1	90	GND24
		5	Unused		
		6	Unused		
		7	Unused		
		8	Boost 2	92	GND24
		9	DIR STEPPER 1	41	GND24
		10	DIR STEPPER 2	43	GND24
		11	DIR STEPPER 3	45	GND24
		12	Boost 3	94	GND24
		13	STEP STEPPER 1	40	GND5
		14	STEP STEPPER 2	42	GND5
		15	STEP STEPPER 3	44	GND5
		16	Watch DOG	70	GND24
		17	Output User 1	26	GND24
		18	Output User 2	76	GND24
		19	Output User 3	27	GND24

20 Output User 4	77	GND24
21 Output User 5	28	GND24
22 Output User 6	78	GND24
23 Output User 7	29	GND24
24 Output User 8	79	GND24
25 Output User 9	30	GND24
26 Output User 10	80	GND24
27 Output User 11	31	GND24
28 Output User 12	81	GND24
29 Output User 13	32	GND24
30 Output User 14	82	GND24
31 Output User 15	33	GND24
32 Output User 16	83	GND24
33 Output User 17	34	GND24
34 Output User 18	84	GND24
35 Output User 19	35	GND24
36 Output User 20	85	GND24
37 Output User 21	36	GND24
38 Output User 22	86	GND24
39 Output User 23	37	GND24
40 Output User 24	87	GND24

INPUT 1 - 16 Unused

17 Input User 1	1	GND24
18 Input User 2	51	GND24
19 Input User 3	2	GND24
20 Input User 4	52	GND24
21 Input User 5	3	GND24
22 Input User 6	53	GND24
23 Input User 7	4	GND24
24 Input User 8	54	GND24
25 Input User 9	8	GND24
26 Input User 10	58	GND24
27 Input User 11	9	GND24
28 Input User 12	59	GND24
29 Input User 13	10	GND24
30 Input User 14	60	GND24
31 Input User 15	11	GND24
32 Input User 16	61	GND24
33 Input User 17	15	GND24
34 Input User 18	65	GND24
35 Input User 19	16	GND24
36 Input User 20	66	GND24
37 Input User 21	17	GND24

38 Input User 22 67 GND24
 39 Input User 23 18 GND24
 40 Input User 24 68 GND24

 22 ;1 - 32 OUTPUT
 1 - 32 INPUT

23 ;0 Nessun OUTPUT YASKAWA
 1 - 16 INPUT YASKAWA
 1 Positive over travel
 2 Negative over travel
 3 Home
 4 Encoder Phase A
 5 Encoder Phase B
 6 Encoder Phase C
 7 External Latch 1
 8 External Latch 2
 9 External Latch 3
 10 Brake Status
 11
 12 Torque limited
 13 Warning
 14 Alarm
 15 MainPower
 16 Driver OK

 24 ;1 - 1 OUTPUT (1) Servo Enable
 1 - 16 INPUT (1) Servo Enable come input
 (16) Driver OK

25 ;1 - 48 (1-48 INPUT 1-48 OUTPUT)

26 ;1 - 3 (1-3 INPUT) (1 OUTPUT)

27 ;1 - 32 (1-4 INPUT) (0 OUTPUT) (Franciscono)

28 ;1 - 32 (1-32 INPUT) (1-32 OUTPUT)

29 ;1 - 32 (1-.. INPUT) (1-.. OUTPUT)

30 ;1 - 32 (1-.. INPUT) (1-.. OUTPUT)

100 ;1 - 2048

129 ; 1 - 12288
528 ; 1 - 64

2.2.2 Segnali Analogici I/O

IO_NAME(indice NOME , "nome simbolico")

indice NOME deve essere compreso tra 1 e NUM_MAX_AI_BIT (INPUT)

indice NOME deve essere compreso tra 1 e NUM_MAX_AO_BIT (OUTPUT)

Kind_sensor="ANALOG_OUTPUT" o "ANALOG_INPUT"

Kind_board= tipo di analog I/O

- 2 ;PIASTRA ASSI TIP102 ANALOG I/O
- 3 ;PIASTRA IP-FastDAC ANALOG OUTPUT
- 4 ;PIASTRA ASSI TPMC118 ANALOG OUTPUT
- 5 ;PIASTRA ANALOG FIELD BUS a 12 bit (con troncamento) +-10 Volt
- 6 ;PIASTRA ANALOG ETN PRIMA ELECTRONICS
- 7 ;PIASTRA ASSI YIDDISH MAVEL
- 11 ;PIGGY BOARD ANALOG ETN su SERVO PRIMA ELECTRONICS
- 12 :PIASTRA ASSI PRIMA ELECTRONICS ANALOGIC PASSIVE
- 13 :PIASTRA ASSI PRIMA ELECTRONICS DIGITAL PASSIVE
- 15 :PIASTRA ASSI MLTA
- 16 :PIGGY BOARD MLTA DAC ADC
- 17 :PIASTRA ASSI NEWTON
- 19 :BOARD NEWTON ADC
- 20 :PIASTRA ASSI SIX
- 22 :SERCOS ANALOG
- 25 ;ANALOG FIELD BUS A 16 BIT (gestione Gamberro 12 -> 16 bit)
- 26 ;ANALOG FIELD BUS A 16 BIT (0 - 10 Volt)
- 27 :BOARD ETHAX ADC
- 28 :PIASTRA ASSI ETHAX (DAC)
- 29 :PIASTRA ASSI ETHERBOX (ADC)
- 30 :PIASTRA ASSI ETHERBOX (assi) (DAC)

Board_number= numero piastra

Rack_number

- 2 ;numero BOARD IP (0 - N)
- 3 ;numero BOARD IP (0 - N)
- 4 ;numero non usato (0)
- 5 ;numero non usato (0)
- 6 ;numero non usato (0)
- 7 ;numero BOARD YIDDISH (0 - N)
- 11 ;numero non usato (0)
- 12 ;numero BOARD ASSI PRIMA ELECTRONICS ANALOG PASSIVE (0 - N)
- 13 ;numero BOARD ASSI PRIMA ELECTRONICS DIGITAL PASSIVE (0 - N)
- 15 ;numero BOARD PIASTRA ASSI MLTA (0 - N)

- 16 ;numero BOARD PIASTRA MLTA (0 - N)
- 17 ;numero BOARD PIASTRA NEWTON (0 - N)
- 19 ;numero BOARD PIASTRA NEWTON (0 - N)
- 20 ;numero BOARD PIASTRA SIX (0 - N)
- 22 ;numero BOARD SERCOS (0 - N)
- 25 ;numero non usato (0)
- 26 ;numero non usato (0)
- 27 ;numero BOARD PIASTRA ETHAX (0 - N)
- 28; numero BOARD PIASTRA ETHAX (0 - N)

Address_port= offset

- 2 ;offset su Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
- 3 ;offset su Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
- 4 ;numero non usato (0)
- 5 ;numero dispositivo (0 - 767)
- 6 ;numero dispositivo (0 - 255)
- 7 ;numero dispositivo (0 - 15) (0 = locale)
- 11 ;numero dispositivo (0 - 255)
- 12 ;numero asse PRIMA ELECTRONICS ANALOG PASSIVE (1 - 12)
- 13 ;numero asse PRIMA ELECTRONICS DIGITAL PASSIVE (1 - 12)
- 15 ;unused
- 16 ;offset PIGGY BOARD su MLTA (0 - 3)
- 17 ;unused
- 19 ;unused
- 20 ;unused
- 22 ;unused
- 25 ;numero dispositivo (0 - 767)
- 26 ;numero dispositivo (0 - 767)
- 27 ;unused
- 28 ;unused
- 29 ;unused
- 30 ;numero asse (1 - 6)

Physical_Number= posizione fisica del canale analogico I/O su piastra

- 2 ;1 - 2 INPUT 1 - 2 OUTPUT
- 3 ;1 - 8
- 4 ;1 - 6 solo INPUT
- 5 ;1 - 12288/16
- 6 ;1 - 10
- 7 ;1 - 4
- 11 ;1 per dispositivo
- 12 ;unused
- 13 ;1 -1
 - (1) = SERVO in Corrente
 - (-1) = SERVO in Tensione
- 15 ;numero asse MLTA (1 - 6)
- 16 ;numero canale MLTA DAC ADC (1 - 2)
- 17 ;numero asse NEWTON (1 - 4)
- 19 ;numero canale NEWTON ADC (1 - 4)
- 20 ;numero asse SIX DAC = (1 - 6+3) ADC = (1 - 3)

DAC	1	DAC1	91 (caldo)	41 (freddo)
	2	DAC2	92 (caldo)	42 (freddo)
	3	DAC3	93 (caldo)	43 (freddo)
	4	DAC4	94 (caldo)	44 (freddo)

	5	DAC5	95 (caldo)	45 (freddo)
	6	DAC6	96 (caldo)	46 (freddo)
	7	DAC2	22 (caldo)	97 (freddo)
	8	DAC2	71 (caldo)	98 (freddo)
	9	DAC2	72 (caldo)	99 (freddo)
ADC	1	ADC1	47 (caldo)	97 (freddo)
	2	ADC2	48 (caldo)	98 (freddo)
	3	ADC3	49 (caldo)	99 (freddo)
	22	;		
	25	;	1 - 12288/16	
	26	;	1 - 12288/16	
	27	;	numero canale ETHAX ADC (1 - 4)	
	28	;	numero canale ETHAX ASSI DAC (1 - 4)	
	29	;	numero canale ETHERBOX (1 - 3)	
	30	;	numero canale ETHERBOX (assi) (1 - 6+3)	

2.2.3 Parametri degli I/O digitali e analogici - Tabelle riassuntive

Gli I/O digitali e analogici della applicazione che si sta realizzando si trovano su più schede HW, spesso di tipo diverso.

Nel File sistema.txt, la caratterizzazione di tutti gli I/O inizia dopo l'indicazione **[IO]** scritta una sola volta in testa a tutti gli I/O

Se si tratta di I/O virtuali (sorta di flags per situazioni logiche interne, non collegate a segnali esterni, ma comunque gestite come se fossero degli I/O), caratterizzare il **Kind_board=0**

Se invece si tratta di I/O reali, il **Kind_board=** valore indicato per ogni tipo di scheda (es. 20) Vengono caratterizzati tutti in questa fase che, come esempio, riporta alcuni casi con le indicazioni atte a descrivere la metodologia più che la semplice regolazione.

Input ed Output si trovano sulle schede:

- PCI 2001 6 Assi Analogici + 3 Mandrini (20)
- PCI 2001 24+24 I/O (21)
- 48 I/O TTL (definibili a ottetti) (8) PCI (9) ISA
- SCHEDA FIELD-BUS (5)
- SERCOS (22)

2.2.3.1 Output Digitali - Esempio di caratterizzazione della scheda PCI 2000

[IO] ← Scritto una sola volta prima della caratterizzazione di tutti gli I/O (Digitali e Analogici)

<i>Voci da caratterizzare</i>	<i>Informazioni per la definizione di un Output digitale</i>
i=IO_NAME(1, "EnableX1")	Numero logico = 1 (di norma si assegna 1 al primo Output, 2 al secondo ecc.) – Vedi nota 1 Nome logico = EnableX1
b={	
Description="Abilitazione Asse X1"	Descrizione in chiaro, da inserire, come si vuole compaia nella videata di sistema degli I/O
Index_system= 0	Riservato ad AWL - NON MODIFICARE
FalseTrue_logic= 0	Se = 0 i livelli logico e fisico coincidono Se = 1 sono invertiti
Kind_sensor="DIGITAL_OUTPUT"	DIGITAL_OUTPUT definisce che è un Output digitale - NON MODIFICARE
Kind_board= 20	20 è il codice di riconoscimento obbligatorio di tutti gli I/O della scheda PCI 2000
Rack_number= 0	Indicare la posizione della scheda nel Back-plane (la scheda "0" è la più vicina alla CPU, "1" la seconda dello stesso tipo ecc.)
Address_port= 0	In questo caso rimane =0. (perché la scheda Kind_board= 20 non ha I/O dedicati a priori ad ogni asse) Se invece ad esempio, si utilizzano schede con Kind_board=22 (per assi SERCOS con 8 sezioni), verrà scritto un numero da 1 a 8, a seconda della sezione utilizzata per gli I/O di ogni singolo Asse.
Physical_number= 1	1 è il numero con il quale il Sistema riconosce questo output sulla scheda (Kind_board=20). Come mostrato nelle Tabella delle Corrispondenze , se si cambia il numero si cambia fisicamente l'output utilizzato.
e=}	

NOTE –

1) Il Numero Logico deve essere univoco nell'ambito dello stesso tipo di I/O. Potranno pertanto esistere l'Input Digitale =1, l'Analogico =1, L'Output Digitale =1, l'Analogico =1 e così via. Non è accettata la situazione di due o più I/O dello stesso tipo con lo stesso numero.

2) gli altri Output della scheda, previsti per i comandi Enable degli altri 5Assi e dei segnali di Enable, Dir, Boost dedicati al pilotaggio degli Step-motors si caratterizzano con lo stesso criterio di quello mostrato nell'esempio.

3) Gli Output non utilizzati per lo scopo originale (se ad esempio non si hanno Step-Motors), possono essere utilizzati come dei normali Output, assegnando opportunamente il numero logico ed il nome logico (prima riga).

4) La caratterizzazione di un Output digitale della scheda 24+24I/O PCI 2001 si effettua con le stesse modalità, assegnando però Kind_board=21 anziché 20 ed il Physical_number indicato nella tabella delle corrispondenze.

Stesso principio per gli Output gestiti da Field-bus o altro.

2.2.3.2 Output Analogici - Esempio di caratterizzazione della scheda PCI 2000

NOTA - La Caratterizzazione degli Output Analogici è uguale a quella di quelli Digitali, in quanto serve ad indicare al Sistema di gestione quale via si intende utilizzare tra quelle predisposte, e quale nome si vuole dare al segnale.

Nell'esempio seguente infatti si caratterizza l'Output Analogico (Kind_sensor="ANALOG_OUTPUT")

- denominato (2, "DACM2")
- che la descrizione chiarifica come ("DAC Mandr. 2")
- residente sulla 2^a Scheda a partire dalla CPU (Rack_number=1)
- la quale scheda è del tipo 6Assi analogici + 3Mandrini, PCI 2000 (Kind_board=20)
- e, non avendo sezioni multiple, (Address_port=0)
- è riconosciuta dal Sistema con il (Physical_number=8 – Vedi Tabella delle Corrispondenze)

```
i=IO_NAME(2 , "DACM2" )  
b={  
Description="DAC Mandr. 2"  
Kind_sensor="ANALOG_OUTPUT"  
Kind_board=20  
Rack_number=1  
Address_port=0  
Physical_number=8  
e=}
```

2.2.3.3 Input Digitali - Esempio di caratterizzazione della scheda PCI 2000

<i>Voci da caratterizzare</i>	<i>Informazioni per la definizione di un Output digitale</i>
i=IO_NAME(1 , "AUX_ON")	Numero logico = 1 (di norma si assegna 1 al primo Output, 2 al secondo ecc.) Nome logico = AUX_ON
b={	
Description=" Auxiliary ON "	Descrizione in chiaro, da inserire, come si vuole compaia nella videata di sistema degli I/O
Index_system=1	Riservato ad AWL - NON MODIFICARE
FalseTrue_logic=0	Se = 0 i livelli logico e fisico coincidono Se = 1 sono invertiti
Kind_sensor=" DIGITAL_INPUT "	DIGITAL_INPUT definisce che è un Input digitale - NON MODIFICARE
Kind_board=21	21 è il codice di riconoscimento obbligatorio di tutti gli I/O della scheda PCI 2001
Rack_number=0	Indicare la posizione della scheda nel Back-plane (la scheda "0" è la più vicina alla CPU, "1" la seconda dello stesso tipo ecc.)
Address_port=0	In questo caso rimane =0. (L'indicazione serve ad individuare l'eventuale sezione di scheda)
Physical_number=17	17 è il numero con il quale il Sistema riconosce questo output sulla scheda (Kind_board=20). Come mostrato nelle Tabella delle Corrispondenze, se si cambia il numero si cambia fisicamente l'output utilizzato.
e=}	

NOTA – 1) Con lo stesso criterio si caratterizzano gli altri Input della scheda.

Altro esempio di Caratterizzazione dell'Input di Homing sulla scheda assi PCI 2000 (Kind_board=20)

```

i=IO_NAME( 403 , "HomeY1" )
b={
Description="Home Y1"
Index_system=1
FalseTrue_logic=0
Kind_sensor="DIGITAL_INPUT"
Kind_board=20
Rack_number=0
Address_port=0
Physical_number=2
e=}

```

2.2.3.4 Input Analogici - Esempio di caratterizzazione della scheda PCI 2000

NOTA - La Caratterizzazione degli Input Analogici è uguale a quella di quelli Digitali, in quanto serve ad indicare al Sistema di gestione quale via si intende utilizzare tra quelle previste, e quale nome si vuole dare al segnale. Ovviamente non si caratterizzano le voci `Index_system=` e `FalseTrue_logic=` in quanto trattasi di segnali analogici

- Nell'esempio seguente infatti si caratterizza l'Input Analogico (`Kind_sensor="ANALOG_INPUT"`)
- denominato (**3, "AD3"**) – con AD3 nell'esempio, si intende "Analogico-Digitale 3"
- che la descrizione chiarifica come ("Analog Input 3")
- residente sulla 1^a Scheda a partire dalla CPU (**Rack_number=0**)
- la quale scheda è del tipo 6Assi analogici + 3Mandrini, PCI 2000 (**Kind_board=20**)
- e, non avendo sezioni multiple, (`Address_port=0`)
- è riconosciuta dal Sistema con il (**Physical_number=3 – Vedi Tabella delle Corrispondenze**)

```
i=IO_NAME(3 , "AD3" )  
b={  
Description="Analog Input 3"  
Kind_sensor="ANALOG_INPUT"  
Kind_board=20  
Rack_number=0  
Address_port=0  
Physical_number=3  
e=}
```

2.2.3.5 CARATTERIZZAZIONE DI I/O DIGITALI CONSECUTIVI A GRUPPI

Creazione della struttura

Per minimizzare il numero di tabelle quando si hanno parecchi Input o Output consecutivi (virtuali o reali) come ad esempio tipicamente organizzati dalla gestione Field Bus, è possibile usufruire dell'automatismo qui descritto, che permette di definire un intero gruppo omogeneo di Input oppure Output con una sola tabella.

Nel file di taratura tipico di un Input (ma la stessa logica si applica al file di un Output), si inseriscono le due seguenti voci da caratterizzare, che definiscono la struttura di Input (oppure di Output) multiplo:

Data_type="Word"
Extension="n"

Dove **"Word"**, valore scelto qui come esempio, può assumere i seguenti nomi e valori:

"Bit"	= 1 Input singolo (Caso senza senso. Per un solo Input si compila la struttura.)
"Nibble"	= 4 Input consecutivi
"Byte"	= 8 Input consecutivi
"Word"	=16 Input consecutivi
"Dword"	=32 Input consecutivi

E dove **"n"** rappresenta l'estensione del nome data automaticamente dal sistema.

Ciò è dovuto al fatto che con questa struttura, la voce `i=IO_NAME(...,...)` non è più il nome di un singolo segnale ma il "Cognome della famiglia di segnali". I singoli Nomi sono assegnati automaticamente ed in sequenza ai vari Input della famiglia, a partire dal numero indicato in Extension con **"n"**

NOTE IMPORTANTI

- Poiché il sistema non deve mai trovare Numeri Logici ripetuti, è buona norma assegnare un numero Logico sicuramente maggiore di quelli utilizzati nelle tabelle precedenti, singole o multiple.
- Ricordarsi che ogni tabella multipla utilizza automaticamente i numeri da quello assegnato, più quanto definito con Data Type= ("Bit", "Nibble", "Byte", "Word", "Dword")
Esempio: `i=IO_NAME(501, "I_011")` occuperà automaticamente il campo 501÷532 se `Data_type="Dword"`
Il prossimo numero logico utilizzabile sarà in questo caso 533.
- Come Physical Number dovrà essere assegnato quello del primo I/O della famiglia di segnali, come risulta dalla "Tabella delle Corrispondenze", ad esempio del CAN (`Kind_board=5`). Si noti ancora che, se ad esempio si intende creare la struttura a

tabella multipla a partire dal Physical number=7, in questo esempio verranno gestiti automaticamente gli Input (oppure di Output) dal 7° al 38°.

La possibilità di definire diverse lunghezze della tabella multipla (tramite **Data_type=""**) permette di minimizzare il numero di tabelle di taratura, semplicemente riscrivendo nuove strutture.

Assegnazione di Nomi e Descrizioni

Nella struttura multipla viene assegnata una unica descrizione nel campo **Description=""** che vale per tutta la famiglia di Input (oppure di Output), e quindi tutti i segnali della famiglia hanno la stessa descrizione.

Per assegnare descrizioni diverse è necessario aggiungere dopo la struttura multipla, i 2 seguenti statement per ogni segnale da descrivere:

Come dall'esempio su citato `i=IO_NAME(501 , "I_011")`,

[I_011.1] ←———— Name.Extension
Description="Descrizione in lingua del segnale"

[I_011.32] ←———— 32 perché : **Data_type="Dword"** (quindi l'Extension assegna 32)
Description="Descrizione in lingua del segnale"

Esempio di Struttura Multipla di Input

<i>Voci da caratterizzare</i>	<i>Informazioni per la definizione di un Output digitale</i>
<code>i=IO_NAME(501 , "I_011")</code>	Numero logico = 501 Nome logico ="I_011"
<code>b={</code>	
<code>Description="LM Input"</code>	Descrizione in chiaro, da inserire, come si vuole compaia nella videata di sistema degli I/O. Sarà uguale per tutti gli Input
Data_type="Dword"	VOCE DA AGGIUNGERE ALLA STRUTTURA BASE, con tabella per 32 Input "Dword"
Extension=1	Numero da cui l'Extension parte a contare (nell'esempio fino a 36)
<code>Index_system=0</code>	Riservato ad AWL - NON MODIFICARE
<code>FalseTrue_logic=0</code>	Se = 0 i livelli logico e fisico coincidono Se = 1 sono invertiti
<code>Kind_sensor="DIGITAL_INPUT"</code>	DIGITAL_INPUT definisce che è un Input digitale - NON MODIFICARE
<code>Kind_board=5</code>	5 è il codice di riconoscimento obbligatorio di tutti gli I/O della scheda Field Bus
<code>Rack_number=0</code>	Indicare la posizione della scheda nel Back-plane (la scheda "0" è la più vicina alla CPU, "1" la seconda dello stesso tipo ecc.)
<code>Address_port=0</code>	In questo caso rimane =0. (L'indicazione serve ad

	individuare l'eventuale sezione di scheda)
Physical_number=7	7 è il numero con il quale il Sistema riconosce il primo Input della famiglia di segnali consecutivi sull'interfaccia CAN (Kind_board=5). Come mostrato nelle Tabella delle Corrispondenze, se si cambia il numero si cambia fisicamente l'Input utilizzato.
e=}	

2.2.3.6 LANCIO DI TASK E P.PROGRAM SUL CAMBIAMENTO DI STATO DI I/O

Creazione della struttura

Per lanciare dei P.Program sulla commutazione di un segnale di Input (oppure di un Output), è possibile inserire le voci, qui di seguito descritte, all'interno della struttura tipica di caratterizzazione degli Input (oppure degli Output).

Le voci possono essere inserite in una posizione qualsiasi, ma si suggerisce di metterle in fondo, subito prima della chiusura e=}

Nota - se queste voci vengono inserite in una struttura multipla, è sufficiente la commutazione di uno qualsiasi dei segnali della struttura per attivare il lancio del P.Program.

Definizioni della struttura di lancio P.Program sullo stato di un I/O

a	TaskChangeFront=1	<p><u>Se=1 sente il fronte di commutazione del segnale</u> Al cambio di livello da 0 a 1 lancia il Part Program "InOne" indicato in riga "e" che risiede nel File "LM_011.pp" indicato in riga "d". Al cambio di livello da 1 a 0 lancia il Part Program "InZero" indicato in riga "g" che risiede nel File "LM_011.pp" indicato in riga "f".</p> <p><u>Se=0 sente il livello del segnale.</u> Se livello 1 lancia il Part Program "InOne" indicato in riga "e" che risiede nel File "LM_011.pp" indicato in riga "d". Se livello 0 lancia il Part Program "InZero" indicato in riga "g" che risiede nel File "LM_011.pp" indicato in riga "f".</p> <p>Nota-I nomi dei P.Program e dei Files sono dati come esempio.</p>
c=b	TaskModeRun=1	<p>Se =1 i P.Program delle righe "e" oppure "g" (in accordo con la taratura di riga "a") vengono lanciati solo se non sono attivi. Se =0 i P.Program vengono comunque rilanciati Se =2 segnala errore se un task attivo viene rilanciato</p> <p>Valore di default=1</p>
d=c	TaskOnePP="LM_011.pp"	Nome del File di P.Program tipo .pp
e=d	TaskOneName="InOne"	Nome de P.Program nel File indicato in "d"

f=e	TaskZeroPP="LM_011.p p"	Nome del File di P.Program tipo .pp
g=f	TaskZeroName="InZero"	Nome de P.Program nel File indicato in "g"
h= g	ZeroOnReset=0	Se=0 Il comando di Reset dato in Manuale e/o da P.Program non agisce su questo Output. (Nota - Il comando non agisce se la taratura è relativa a un Input) Se=1 Il comando di Reset è abilitato ad agire portando comunque l'output a zero
i	NoThreeState=0	Se =0 questo Input/Output va nella condizione logica di ThreeState. Se= 1 questo Input/Output non va nella condizione logica di ThreeState. Valore di default=0

NOTE

- 1) Per Tristate si intende.....
- 2) Le ultime 2 voci "h" e "i" possono essere inserite anche da sole in una struttura, in quanto non legate allo stato della commutazione dell'Output.

2.2.3.7 TABELLA DELLE CORRISPONDENZE TRA PHISICAL NUMBER E I/O FISICO PER I VARI TIPI DI SCHEDA

(Serve per seguire fisicamente il percorso di un segnale logico fino al pin di interfacciamento con la cabina elettrica)

SCHEDA FIELD-BUS ISA (CANopen Master)

Kind_board=5

Esempio di numerazione degli Input digitali di 2 nodi CAN



Input Digitali (per Kind_board=5)

Physical_number	Connessioni verso cablaggio				
	N. PIN	N. Modulo	N. Nodo		
1					
2					
3					
4					
5					
....					
....					
1024					

ESEMPIO PER moduli 8In+8Out e in sequenza modulo da 16 IN

Output Digitali (per Kind_board=5)

Physical_number	Connessioni verso cablaggio			Significato del nome	
	N. PIN	N. Modulo	N. Nodo		
1					
2					
3					
4					
5					
6					
7					
....					
....					
1024					

ESEMPIO PER moduli 8In+8Out e in sequenza modulo da 16 OUT

SCHEMA PCI 2000 (6 Assi Analogici + 3 Mandrini)
Kind_board=20
Input Digitali (per Kind board=20)

Physical number	N.pin del Connettore		Nome Segnale	Significato del nome
	Caldo	Rif. Freddo		
1	82	GND	HOME1	Micro di Zero (homing) asse 1
2	32	GND	HOME2	Micro di Zero (homing) asse 2
3	83	GND	HOME3	Micro di Zero (homing) asse 3
4	33	GND	HOME4	Micro di Zero (homing) asse 4
5	84	GND	HOME5	Micro di Zero (homing) asse 5
6	34	GND	HOME6	Micro di Zero (homing) asse 6
7				<i>Non Usato</i>
8				<i>Non Usato</i>
9	88	GND	DROK1	Ingresso Drive 1 OK
10	38	GND	DROK2	Ingresso Drive 2 OK
11	89	GND	DROK3	Ingresso Drive 3 OK
12	39	GND	DROK4	Ingresso Drive 4 OK
13	90	GND	DROK5	Ingresso Drive 5 OK
14	40	GND	DROK6	Ingresso Drive 6 OK
15				<i>Non Usato</i>
16				<i>Non Usato</i>

Output Digitali (per Kind board=20)

Physical number	N.pin del Connettore		Nome Segnale	Significato del nome
	Caldo	Rif. Freddo		
1	85	GND	DREN1	Uscita Enable Drive 1
2	86	GND	DREN3	Uscita Enable Drive 3
3	87	GND	DREN5	Uscita Enable Drive 5
4	////////	////////	<i>Boost 1</i>	<i>Vedi Tabella dispositivo Assi Supplementari</i>
5	35	GND	DREN2	Uscita Enable Drive 2
6	36	GND	DREN4	Uscita Enable Drive 4
7	37	GND	DREN6	Uscita Enable Drive 6
8	////////	////////	<i>Boost 2</i>	<i>Vedi Tabella dispositivo Assi Supplementari</i>
9	////////	////////	<i>Dir 1</i>	<i>Vedi Tabella dispositivo Assi Supplementari</i>
10	////////	////////	<i>Dir 2</i>	<i>Vedi Tabella dispositivo Assi Supplementari</i>
11	////////	////////	<i>Dir 3</i>	<i>Vedi Tabella dispositivo Assi Supplementari</i>
12	////////	////////	<i>Boost 3</i>	<i>Vedi Tabella dispositivo Assi Supplementari</i>
13	////////	////////	<i>PULSE 1</i>	<i>Vedi Tabella dispositivo Assi Supplementari</i>
14	////////	////////	<i>PULSE 2</i>	<i>Vedi Tabella dispositivo Assi Supplementari</i>
15	////////	////////	<i>PULSE 3</i>	<i>Vedi Tabella dispositivo Assi Supplementari</i>
16	21	GND	WDOG	Uscita segnalazione di Watch-Dog

Output Digitali relativi al dispositivo Assi Supplementari (per Kind_board=20)

Physical_number	N.pin del Connettore		Nome Segnale	Significato del nome
	Caldo	Rif. Freddo		
1	////////	////////	DREN1	Vedi Tabella precedente (Output Dgitali)
2	////////	////////	DREN3	Vedi Tabella precedente (Output Dgitali)
3	////////	////////	DREN5	Vedi Tabella precedente (Output Dgitali)
4			Boost 1	(preliminary)
5	////////	////////	DREN2	Vedi Tabella precedente (Output Dgitali)
6	////////	////////	DREN4	Vedi Tabella precedente (Output Dgitali)
7	////////	////////	DREN6	Vedi Tabella precedente (Output Dgitali)
8			Boost 2	(preliminary)
9			Dir 1	(preliminary)
10			Dir 2	(preliminary)
11			Dir 3	(preliminary)
12			Boost 3	(preliminary)
13			PULSE 1	(preliminary)
14			PULSE 2	(preliminary)
15			PULSE 3	(preliminary)
16	////////	////////	WDOG	Vedi Tabella precedente (Output Dgitali)

Input Analogici (per Kind_board=20)

Physical_number	N.pin del Connettore		Nome Segnale	Significato del nome
	Caldo	Rif. Freddo		
1	47	97	ADC1	Input Analogico 1
2	48	98	ADC2	Input Analogico 2
3	49	99	ADC3	Input Analogico 3

Output Analogici (per Kind_board=20)

Physical_number	N.pin del Connettore		Nome Segnale	Significato del nome
	Caldo	Rif. Freddo		
1	91	41	DAC1	Uscita riferimento analogico Asse 1
2	92	42	DAC2	Uscita riferimento analogico Asse 2
3	93	43	DAC3	Uscita riferimento analogico Asse 3
4	94	44	DAC4	Uscita riferimento analogico Asse 4
5	95	45	DAC5	Uscita riferimento analogico Asse 5
6	96	46	DAC6	Uscita riferimento analogico Asse 6
7	22	97	DACM1	Uscita riferimento analogico Mandrino 1
8	71	98	DACM2	Uscita riferimento analogico Mandrino 2
9	72	99	DACM3	Uscita riferimento analogico Mandrino 3

SCHEMA PCI 2001 (24+24 I/O)

Kind_board=21

Input Digitali (per Kind_board=21)

Physical number	N.pin del Caldo	Connettore Rif. Freddo	Nome Segnale	Significato del nome	
<i>1÷16</i>				<i>Physical number non usati</i>	
17	1	GND 24	INPUT 1	Input Digitale 1	
18	51	GND 24	INPUT 2	Input Digitale 2	
19	2	GND 24	INPUT 3	Input Digitale 3	
20	52	GND 24	INPUT 4	Input Digitale 4	
21	3	GND 24	INPUT 5	Input Digitale 5	
22	53	GND 24	INPUT 6	Input Digitale 6	
23	4	GND 24	INPUT 7	Input Digitale 7	
24	54	GND 24	INPUT 8	Input Digitale 8	
25	8	GND 24	INPUT 9	Input Digitale 9	
26	58	GND 24	INPUT 10	Input Digitale 10	
27	9	GND 24	INPUT 11	Input Digitale 11	
28	59	GND 24	INPUT 12	Input Digitale 12	
29	10	GND 24	INPUT 13	Input Digitale 13	
30	60	GND 24	INPUT 14	Input Digitale 14	
31	11	GND 24	INPUT 15	Input Digitale 15	
32	61	GND 24	INPUT 16	Input Digitale 16	
33	15	GND 24	INPUT 17	Input Digitale 17	
34	65	GND 24	INPUT 18	Input Digitale 18	
35	16	GND 24	INPUT 19	Input Digitale 19	
36	66	GND 24	INPUT 20	Input Digitale 20	
37	17	GND 24	INPUT 21	Input Digitale 21	
38	67	GND 24	INPUT 22	Input Digitale 22	
39	18	GND 24	INPUT 23	Input Digitale 23	
40	68	GND 24	INPUT 24	Input Digitale 24	

Output Digitali (per Kind board=21)

Physical number	N.pin del Connettore		Nome Segnale	Significato del nome
	Caldo	Rif. Freddo		
1	91	GND 24	ENABLE 1	Abilitazione movimento Step Motor 1
2	93	GND 24	ENABLE 2	Abilitazione movimento Step Motor 2
3	95	GND 24	ENABLE 3	Abilitazione movimento Step Motor 3
4	90	GND 24	BOOST 1	Segnale di BOOST per lo Step Motor 1
5				<i>Non Usato</i>
6				<i>Non Usato</i>
7				<i>Non Usato</i>
8	92	GND 24	BOOST 2	Segnale di BOOST per lo Step Motor 2
9	41	GND 24	DIR 1	Direzione positiva per lo Step Motor 1
10	43	GND 24	DIR 2	Direzione positiva per lo Step Motor 2
11	45	GND 24	DIR 3	Direzione positiva per lo Step Motor 3
12	94	GND 24	BOOST 3	Segnale di BOOST per lo Step Motor 3
13	40	GND 5	PULSE 1	Impulso di avanzamento Step Motor 1
14	42	GND 5	PULSE 2	Impulso di avanzamento Step Motor 2
15	44	GND 5	PULSE 3	Impulso di avanzamento Step Motor 3
16	70	GND 24	W-DOG	Uscita segnalazione di Watch-Dog (5V.TTL)
17	26	GND 24	OUTPUT 1	Output Digitale 1
18	76	GND 24	OUTPUT 2	Output Digitale 2
19	27	GND 24	OUTPUT 3	Output Digitale 3
20	77	GND 24	OUTPUT 4	Output Digitale 4
21	28	GND 24	OUTPUT 5	Output Digitale 5
22	78	GND 24	OUTPUT 6	Output Digitale 6
23	29	GND 24	OUTPUT 7	Output Digitale 7
24	79	GND 24	OUTPUT 8	Output Digitale 8
25	30	GND 24	OUTPUT 9	Output Digitale 9
26	80	GND 24	OUTPUT 10	Output Digitale 10
27	31	GND 24	OUTPUT 11	Output Digitale 11
28	81	GND 24	OUTPUT 12	Output Digitale 12
29	32	GND 24	OUTPUT 13	Output Digitale 13
30	82	GND 24	OUTPUT 14	Output Digitale 14
31	33	GND 24	OUTPUT 15	Output Digitale 15
32	83	GND 24	OUTPUT 16	Output Digitale 16
33	34	GND 24	OUTPUT 17	Output Digitale 17
34	84	GND 24	OUTPUT 18	Output Digitale 18
35	35	GND 24	OUTPUT 19	Output Digitale 19
36	85	GND 24	OUTPUT 20	Output Digitale 20
37	36	GND 24	OUTPUT 21	Output Digitale 21
38	86	GND 24	OUTPUT 22	Output Digitale 22
39	37	GND 24	OUTPUT 23	Output Digitale 23
40	87	GND 24	OUTPUT 24	Output Digitale 24

2.3 Assi e mandrini

Parametri inerenti agli assi e mandrini

2.3.1 Assi

[YS]

PHYSICAL_AXIS_NAME(1, "YS")
;Numero e Nome dell'asse

{

Kind_axis="ARM"
;Definisce un asse

Kind_axis_0=0x00005001

;Maschera 0 a bits dell'asse

;0x00000001 = ASSERVIMENTO IN CORRENTE (0 = IN VELOCITA')

;0x00000002 = GESTIONE "SQUARE" di KP

;0x00000004 = GESTIONE "MEDIATA" di KD

;0x00000008 = GESTIONE CON LETTURA CORRETTA DA SAMPLE

;0x00000010 = GESTIONE "SQUARE" di KI

;0x00000020 = ASSE CON PARTENZA DOLCE DEL SERVO

;0x00000040 = ASSE CHE DECELERA SU MICRO DI FINE CORSA

;0x00000080 = ASSE NORMALMENTE FRENATO (QUANDO E' FERMO)

;0x00000100 = ASSE CON CONTROLLO OVER TRAVEL ANCHE IN ASSERV.

;0x00000200 = ASSE CON AGGIUSTAMENTO SU MARKER IN 1 SOLA DIREZ.

;0x00000800 = ESECUZIONE OMO "A BATTUTA"

;0x00001000 = ESECUZIONE OMO SOLO SUL MARKER

;0x00002000 = ESECUZIONE OMO SOLO SUL MICRO

;0x00004000 = ASSE CHE SI MUOVE SOLO DOPO OMO

;0x00010000 = SISTEMA VIRTUALE SCARA

;0x00020000 = SISTEMA VIRTUALE SCARA PANTOGRAFO COMP. POSIT.

;0x00040000 = SISTEMA VIRTUALE SCARA PANTOGRAFO COMP. NEGAT.

;0x00080000 = SISTEMA VIRTUALE CILINDRICO TIPO CANOTTO

;0x00100000 = SISTEMA VIRTUALE CILINDRICO TIPO TRASLANTE

;0x00200000=SISTEMA VIRTUALE"SAURO"

Kind_axis_1=0x00070200

;Maschera 1 a bits dell'asse
;0x00000002 = ASSE TIPO POTENZIOMETRICO
;0x00000004 = ASSE CON ABILITAZIONE SERVO DOPO MOV.
;0x00000008 = ASSE ROLLOVER CON VALORI tra 0 e +MODULO
;0x00000010 = ASSE ROLLOVER CON VALORI tra -MODULO e +MODULO
;0x00000020 = ASSE PER PLOTTER
;0x00000040 = ASSE "TRASLO"
;0x00000080 = RAMPE CON ACCELERAZIONE S_CURVE
;0x00000100 = RAMPE CON DECELERAZIONE S-CURVE
;0x00000200 = ASSE CON FASATURA (MAVEL)
;0x00000400 = ASSE FUORI SERVIZIO
;0x00010000 = COLLISION ERROR
;0x00020000 = SERVO ERROR (ASSE FERMO)
;0x00040000 = POSITION ERROR (ASSE IN MOVIMENTO)
;0x00080000 = STOP COUNTER ERROR
;0x00100000 = DRIVER OK ERROR
;0x00800000 = ASSE MASTER CON SLAVE CARAT. GANTRY
;0x01000000 = ASSE MASTER CON SLAVE CHE DEVE SEGUIRE
;0x02000000 = ASSE MASTER CON SLAVE CHE SEGUE IL TEORICO
;0x04000000 = ASSE MASTER CON SLAVE INTERPOLANTE
;0x08000000 = PID INTEGRATIVO ANCHE CON MOVIMENTO
;0x10000000 = BOOST ATTIVATO IN ACC. (SOLO CON ASSE STEPPER)

Kind_board_TransducerPosition=1

;Tipo di piastra trasduttore

- ; 1 = ASSE TIP102 (ENCODER+DAC)
- ; 2 = ASSE TPMC118 (ENCODER+DAC)
- ; 10 = ASSE USB MAVEL (ENCODER+DAC)
- ; 11 = ASSE YIDDISH MAVEL (ENCODER+DAC)
- ; 20 = ASSE MLTA (ENCODER+DAC)
- ; 30 = ASSE NEWTON (ENCODER+DAC)
- ; 40 = ASSE SIXN (ENCODER+DAC)
- ; 50 = ASSE ETHAX (ENCODER+DAC)
- ; 60 = ASSE ETHERBOX Encoder + Analogico
- ; 61 = ASSE ETHERBOX Frequenza + Counter
- ; 100 = ASSE PRIMA ELECTRONICS (POSIZIONE)
- ; 101 = ASSE PRIMA ELECTRONICS (ANALOGICO PASSIVO)
- ; 102 = ASSE PRIMA ELECTRONICS (DIGITALE PASSIVO)
- ; 103 = ASSE ENCODER CON RSM304 P.E. (ETN)
- ; 129 = ASSE CANOPEN DS402
- ; 200 = ASSE SERCOS (POSIZIONE)
- ; 201 = ASSE SERCOS (VELOCITA')
- ; 202 = ASSE SERCOS (CORRENTE) - NON IMPLEMENTATO
- ; 210 = ASSE YASKAWA MECHATROLINK (POSIZIONE)
- ; 220 = ASSE ETEL (POSIZIONE)
- ; 300 = ASSE UNIVERSALE
- ; 1000 = ASSE PER STEPPER A BITS
- :1002 = ASSE PER STEPPER A CONTATORE SIX
- :1003 = ASSE PER STEPPER A CONTATORE SIX_IO
- :1100 = ASSE PER STEPPER RSM304 ETN SERVŌ

Board_number_TransducerPosition=1

;Numero di piastra per trasduttore

- ; 1 0 - n numero piastra TIP102
- ; 2 0 - n numero piastra TPMC118
- ; 10 0 - 7 numero cestello USB MAVEL
- ; 11 0 - n numero piastra YIDDISH
- ; 20 0 - n numero piastra MLTA
- ; 30 0 - n numero piastra NEWTON
- ; 40 0 - n numero piastra SIX
- ; 50 0 - n numero piastra ETHAX
- ; 60 0 - n numero piastra ETHERBOX Encoder + Analogico
- ; 61 0 - n numero piastra ETHERBOX Frequenza + Counter
- ; 100 0 - n numero piastra PRIMA ELECTRONICS
- ; 101 0 - n numero piastra PRIMA ELECTRONICS
- ; 102 0 - n numero piastra PRIMA ELECTRONICS

```

; 103 unused      ASSE ENCODER CON RSM304 P.E. (ETN)
; 129 unused
; 200 0 - n numero piastra SERCOS
; 201 0 - n numero piastra SERCOS
; 202 0 - n numero piastra SERCOS
; 210 0 - n numero piastra YASKAWA (POSIZIONE)
; 220 0 - n numero piastra ETEL (POSIZIONE)
; 300 0 - n numero piastra ASSE UNIVERSALE
;1000 unused      STEPPER A BITS
;1002 0 - n        numero piastra SIX
;1003 0 - n        numero piastra SIX_IO
:1100 unused      ASSE PER STEPPER RSM304 ETN SERVO
    
```

Address_port_TransducerPosition=0x00004000

```

;Indirizzo asse trasduttore
; 1 offset su Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
; 2 unused
; 10 unused
; 11 0 - 15 numero piastra ( 0 = locale)
; 20 unused
; 30 unused
; 40 unused
; 50 unused
; 60 unused
; 61 unused
; 100 unused
; 101 unused
; 102 unused
; 103 6 - n indirizzo DRIVER CONTROL modulo (M-1)*2
      4 moduli consecutivi con M>=4

; 129 unused
; 200 unused
; 201 unused
; 202 unused
; 210 unused
; 220 unused
; 300 unused
;1000 unused
;1001 unused
;1002 unused
;1003 unused
:1100 6 - n indirizzo DRIVER CONTROL modulo (M-1)*2
      4 moduli consecutivi con M>=4
    
```

Physical_number_Transducerposition=2

```

;Numero fisico asse trasduttore
; 1 1 - 2 numero assi piastra TIP102
; 2 1 - 6 numero assi piastra TPMC118
; 10 0 - 7 numero asse USB MAVEL
; 11 1 - n numero asse piastra YIDDISH (n = 4 locale 2 remota)
; 20 1 - n numero asse piastra MLTA (n = 2 o 4 o 6)
; 30 1 - n numero asse piastra NEWTON (n = 4)
; 40 1 - n numero asse piastra SIX (n = 6)
; 50 1 - n numero asse piastra ETHAX (n = 4)
; 60 1 - n numero asse piastra ETHERBOX (n = 6)
; 61 1 - n numero asse piastra ETHERBOX (n = 6)
; 100 1 - 16 numero assi piastra PRIMA ELECTRONICS
; 101 1 - 16 numero assi piastra PRIMA ELECTRONICS
; 102 1 - 16 numero assi piastra PRIMA ELECTRONICS
: 103 1 - 3 Numero asse
: 129 1 - 64 Numero asse
; 200 1 - 8 numero assi piastra SERCOS
; 201 1 - 8 numero assi piastra SERCOS
; 202 1 - 8 numero assi piastra SERCOS
; 210 1 - 15 numero assi piastra YASKAWA (POSIZIONE)
; 220 1 - 4 numero assi piastra ETEL (POSIZIONE)
; 300 1 - n numero assi piastra ASSE UNIVERSALE
;1000 unused STEPPER A BITS
;1002 1 - 3 STEPPER A CONTATORE SIX (pulse 1 - 3)
;1003 1 - 3 STEPPER A CONTATORE SIX_IO (pulse 1 - 3)
:1100 1 - 3 Numero motore

```

Kind_board_DAC=1

```

;Tipo di piastra asse convertitore
; 1 = ASSE TIP102 (ENCODER+DAC)
; 2 = ASSE TPMC118 (ENCODER+DAC)
; 10 = ASSE USB MAVEL (ENCODER+DAC)
; 11 = ASSE YIDDISH MAVEL (ENCODER+DAC)
; 20 = ASSE MLTA (ENCODER+DAC)
; 30 = ASSE NEWTON (ENCODER+DAC)
; 40 = ASSE SIX (ENCODER+DAC)
; 50 = ASSE ETHAX(ENCODER+DAC)
; 60 = ASSE ETHERBOX Encoder + Analogico
; 61 = ASSE ETHERBOX Frequenza + Counter
; 100 = ASSE PRIMA ELECTRONICS (POSIZIONE)
; 101 = ASSE PRIMA ELECTRONICS (ANALOGICO PASSIVO)
; 102 = ASSE PRIMA ELECTRONICS (DIGITALE PASSIVO)
; 103 = Tipo illegale

```

- ; 129 = ASSE CANOPEN DS402
- ; 200 = ASSE SERCOS (POSIZIONE)
- ; 201 = ASSE SERCOS (VELOCITA')
- ; 202 = ASSE SERCOS (CORRENTE)
- ; 210 = ASSE YASKAWA (POSIZIONE)
- ; 220 = ASSE ETEL (POSIZIONE)
- ; 300 = ASSE UNIVERSALE
- ;1000 = ASSE PER STEPPER A BITS
- ;1002 = ASSE PER STEPPER A CONTATORE SIX
- ;1003 = ASSE PER STEPPER A CONTATORE SIX_IO
- :1100 = ASSE PER STEPPER RSM304 ETN SERVŌ

Board_number_DAC=1

```

;Numero di piastra asse convertitore
; 1 0 - n numero piastra TIP102
; 2 0 - n numero piastra TPMC118
; 10 0 - 7 numero cestello USB MAVEL
; 11 0 - n numero piastra YIDDISH
; 20 0 - n numero piastra MLTA
; 30 0 - n numero piastra NEWTON
; 40 0 - n numero piastra SIX
; 50 0 - n numero piastra ETHAX
; 60 0 - n numero piastra ETHERBOX Encoder + Analogico
; 61 0 - n numero piastra ETHERBOX Frequenza + Counter
; 100 0 - n numero piastra PRIMA ELECTRONICS
; 101 0 - n numero piastra PRIMA ELECTRONICS
; 102 0 - n numero piastra PRIMA ELECTRONICS
; 103 unused Tipo illegale
; 129 unused
; 200 0 - n numero piastra SERCOS
; 201 0 - n numero piastra SERCOS
; 202 0 - n numero piastra SERCOS
; 210 0 - n numero piastra YASKAWA (POSIZIONE)
; 220 0 - n numero piastra ETEL (POSIZIONE)
; 300 0 - n numero piastra ASSE UNIVERSALE
;1000 unused STEPPER A BITS
;1002 0 - n numero piastra SIX
;1003 0 - n numero piastra SIX_IO
;1100 unused ASSE PER STEPPER RSM304 ETN SERVO

```

Address_port_DAC=0x00004000

```

;Indirizzo asse convertitore
; 1 offset su Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
; 2 unused
; 10 0 - 7 numero dispositivo
; 11 0 - 15 numero piastra ( 0 = locale)
; 20 unused
; 30 unused
; 40 unused
; 50 unused
; 60 unused
; 61 unused
; 100 unused
; 101 unused

```

```
; 102 unused
; 103 unused      Tipo illegale
; 129 unused
; 200 unused
; 201 unused
; 202 unused
; 210 unused
; 220 unused
; 300 unused
;1000 unused
;1001 unused
;1002 unused
;1003 unused
:1100 6 - n indirizzo DRIVER CONTROL modulo (M-1)*2
      4 moduli consecutivi con M>=4
```

Physical_number_DAC=2

- ; Numero fisico asse convertitore
- ; 1 1 - 2 numero assi piastra TIP102
- ; 2 1 - 6 numero assi piastra TPMC118
- ; 10 0 - n numero assi cestello USB MAVEL
- ; 11 1 - n numero assi piastra YIDDISH (n = 4 locale 2 remota)
- ; 20 1 - n numero assi piastra MLTA (n = 2 o 4 o 6)
- ; 30 1 - n numero assi piastra NEWTON (n = 4)
- ; 40 1 - n numero assi piastra SIX (n = 6)
- ; 50 1 - n numero assi piastra ETHAX(n = 4)
- ; 60 1 - n numero asse piastra ETHERBOX (n = 6)
- ; 61 1 - n numero asse piastra ETHERBOX (n = 6)
- ; 100 1 - 16 numero assi piastra PRIMA ELECTRONICS
- ; 101 1 - 16 numero assi piastra PRIMA ELECTRONICS
- ; 102 1 - 16 numero assi piastra PRIMA ELECTRONICS
- ; 103 unused Tipo illegale
- ; 129 1 - 64 Numero asse
- ; 200 1 - 8 numero assi piastra SERCOS
- ; 201 1 - 8 numero assi piastra SERCOS
- ; 202 1 - 8 numero assi piastra SERCOS
- ; 210 1 - 15 numero assi piastra YASKAWA (POSIZIONE)
- ; 220 1 - 4 numero assi piastra ETEL (POSIZIONE)
- ; 300 1 - n numero assi piastra ASSE UNIVERSALE
- ; 1000 unused STEPPER A BITS
- ; 1002 1 - 3 STEPPER A CONTATORE SIX (pulse 1 - 3)
- ; 1003 1 - 3 STEPPER A CONTATORE SIX_IO (pulse 1 - 3)
- ; 1100 1 - 3 Numero motore

Level_TCH=1

Ogni asse puo' staticizzare in modo indipendente dagli altri.
 Parallelizzando con l'hardware l'input di touch degli assi voluti,
 la staticizzazione e' contemporanea, ma deve essere definita al
 software mettendo LEVEL_TCH_ASSE con un numero uguale per ognuno
 di questi assi che hanno il segnale di touch in parallelo.
 Il valore di LEVEL_TCH_ASSE e' compreso tra 0 a 32;
 0 significa asse senza touch.

MachineGroup=2

Raggruppa gli assi a gruppi, cio' permette a questi assi di non
 influenzare gli altri assi (gruppi diversi) in certi stati
 (servo error servo off servo on reset ...).

StepsEncoder=4000

impulsi per giro encoder. usato per aggiustare la posizione
se nel giro si perdono steps

MaxStepsLost=40

impulsi per giro encoder che possono essere persi

MarkerSteps=2

Larghezza in impulsi del MARKER

Con 0x200 in Kind_axis_0 (aggiustamento in una sola direzione) il
valore >=0 significa in direzione positiva; <0 in direzione negativa.

NotifyEnabled=0

Deve essere = 0

ANNULLATO

ANNULLATO

KP=3000.0

guadagno proporzionale (errore di posizione)

KI=16.0

guadagno integrativo (sommatoria degli errori di posizione)

KD=20000.0

guadagno derivativo (errore di velocità)

Per assi stepper definisce la frequenza di asservimento (Hz)

KD2=10000.0

guadagno accelerativo (errore di accelerazione)

Per assi stepper definisce la frequenza di scavalco risonanza (Hz)

VoltMaxIntegratedError=2.0

massima tensione (Volt) ottenuta dall'integrativo KI

Molt_AFF=100.0

Acceleration Feed Forward

Percentuale di accelerazione/decelerazione usata nel PID

E' tipico per driver in corrente (analogo al VFF) ma può usarsi
anche per driver in velocità

K_DAC_1Volt=3276.7

valore digitale del DAC per ottenere 1 Volt

mettendo 0 il valore è rilevato automaticamente dal codice asse.

Con asse ETHERBOX_FRQ il valore va impostato ed è il valore di
frequenza / 10 (il KC_parameter deve essere un decimo di Vel_max)

Con asse sercos il valore va impostato.

Molt_VFF=0.0

Velocity Feed Forward

da usarsi tipicamente per driver in velocita'

E' tipico per driver in velocita' (analogo ad AFF); per driver in corrente puo' servire a ridurre l'effetto della

Forza Elettro Motrice indotta (a patto che la tensione di BUS del driver sia sufficiente)

KS=0.0

Valore di troncamento per il PID. Di solito viene posto a 0.0005

KF=0.0

Valore in VOLT di ricupero "FRICTION". Questo valore e' di solito positivo e agisce se l'asse e' in movimento.

KT=0.0

Tempo di campionatura in percentuale corrispondente al ritardo tra lettura posizione e corrispondente comando attivo

KM=0.001

Per assi stepper misti ("Kind_board_TransducerPosition=..." e "Kind_board_DAC=1000") questo campo deve rappresentare il valore di uno step del motore, perche' il campo Step= e' usato dall'encoder
Con assi stepper che hanno "Kind_board_TransducerPosition=1000" e "Kind_board_DAC=1000" viene usato il solo valore di "Step=.." e quindi "KM=.." non e' usato

KN=0

Per assi stepper a contatore difinisce la frequenza di base del contatore in Hz // NON PIU' USATO

Volt_max=9.0

massima tensione data al DAC (Volt)

Per assi stepper definisce la frequenza massima (Hz)

Volt_vel=-8.0

tensione (Volt) necessaria per muoversi a Vel_max per driver in velocita'

tensione (Volt) necessaria per accelerare a Acc_max per driver in corrente

KC_Parameter=0

Questo parametro permette di definire il rapporto di conversione per l'asservimento, in alternativa a Volt_vel Vel_max o Volt_vel Acc_max.

Se e' zero vengono utilizzati i due campi suddetti altrimenti viene usato questo.

Il significato fisico del campo e' la velocita' (u/min) per 1 volt nel caso di asservimenti in velocita'; l'accelerazione (u/sec²) per 1 volt per gli asservimenti in corrente.

Con gli assi di tipo ETEL SERCOS ETHERBOX_FRQ, a parte il segno il valore impostato serve per visualizzare l'ipotetica tensione di asservimento, che di norma e' 10, percio' il valore impostato deve essere la velocita' (Vel_max=) diviso 10 (velocita' con un volt)

Volt_offset=+0.0

offset di tensione necessaria per fermare l'asse (Volt)

Step=+0.001

moltiplicatore impulso -> coordinata calcolo

Backslash=0.0

Valore del gioco meccanico da recuperare

Revolution=0.0

MODULO di rotazione per assi ROLLOVER

Vel_max=50000.0

Velocita' massima dell'asse
per asservimenti in velocita' e' la velocita' ottenuta con "Volt_vel"

Acc_max=20000.0

Accelerazione massima dell'asse
per asservimenti in corrente e' l'accelerazione ottenuta con "Volt_vel"

Dec_max=15000.0

Decelerazione massima dell'asse

ServoError_standby=2.0

Errore di posizione con asse fermo
gestito se Kind_axis_1 ha il settato bit 0x00020000 = SERVO ERROR

ServoError_motion=10.0

Errore di posizione con asse in movimento
gestito se Kind_axis_1 ha il settato bit 0x00010000 = COLLISION ERROR

Position_error=0.03

Errore di posizione dell'asse a fine movimento
gestito se Kind_axis_1 ha il settato bit 0x00040000 = POSITION ERROR

Cycles_positioning=1.0

Tempo massimo di posizionamento in secondi per entrare in "Position_error"
(Timeout di posizionamento)
Se Kind_axis_0 ha il settato bit 0x00000020 (PARTENZA DOLCE DEL SERVO)
questo e' il tempo in secondi che impiega il PID per funzionare a
regime
Se Kind_axis_1 ha il settato bit 0x00000200 (Asse con FASATURA)
definisce il tempo massimo di fasatura

Cycles_wait=0.002

Tempo di permanenza in secondi al valore di Position_error=
(finestra di posizionamento)

Offset_axis=-534.4

Offset asse (zero voluto rispetto a zero MARKER)

Positive_over_travel=572.0

Campo positivo di lavoro rispetto allo zero offsettato
viene controllato dopo OMO

Negative_over_travel=-2.0

Campo negativo di lavoro rispetto allo zero offsettato
viene controllato dopo OMO

DelayStopCounter=0.2

Finestra in secondi per controllo STOP COUNTER

VoltStopCounter=0.5

Tensione in Volt; con valori => viene attivato il controllo STOP COUNTER

VelStopCounter=1.0

Velocita' in mm/min; con valori <= viene attivato il controllo STOP COUNTER
gestito se Kind_axis_1 ha il settato bit 0x00080000 = STOP COUNTER ERROR

TimeOutDriverOk=1.0

Time Out in secondi DRIVER OK senza segnalare errore
gestito se Kind_axis_1 ha il settato bit 0x00100000 = DRIVER OK ERROR

Time_acc_nla=0.0

tempo di variazione accelerazione rispetto a rampa lineare caratterizzata
> 0 = JERK
= 0 = RAMPA LINEARE
< 0 = RAMPA SINUSOIDALE (acc media = acc max * 2 / PI)

Time_dec_nla=0.0

tempo di variazione decelerazione rispetto a rampa lineare caratterizzata
> 0 = JERK
= 0 = RAMPA LINEARE
< 0 = RAMPA SINUSOIDALE (dec media = dec max * 2 / PI)

Axis_1_name_anticollision=""

Distance_1_anticollision=0.0

Kind_axis_1_anticollision=0

Axis_2_name_anticollision=""

Distance_2_anticollision=0.0

Kind_axis_2_anticollision=0

Axis_name_slave=""

nome dell'asse slave

K_follower_slave=0.0

coefficiente percentuale d'inseguimento dello slave
con assi GANTRY percentuale di correzione incrociata

VoltGantry=5.0

massimo squilibrio di tensione per GANTRY rigidi

DeltaOffset_SlaveGantry=0.640

valore di spostamento dello zero slave rispetto al master (GANTRY)

Number_SERVO_switch_port=""

nome dell'output di SERVO

con nome nullo ("") nei seguenti casi viene usato l'output di default

Kind_board_DAC=1 (TIP102) 1 ;Servo Amplifier Enable

Kind_board_DAC=10 (USB MAVEL) 1 ;Servo Amplifier Enable

Kind_board_DAC=11 (YIDDISH MAVEL) 1 ;Servo Amplifier Enable

Kind_board_DAC=102 (PRIMA ELECTRONICS DIGITALE PASSIVO) 1 ;Flag_W

nel caso YIDDISH vengono usati 8 bits per asse, in ordine di asse

FalseTrue_logic_SERVO_switch=1

stato dell'output con SERVO ON

= 0 output messo a zero

= 1 output messo a uno

<> 0 o 1 output non gestito

Number_DRVOK_switch_port="INP13"

con nome nullo ("") nei seguenti casi viene usato l'input di default

Kind_board_TransducerPosition=1 (TIP102) 16 ;General Input Axis

Kind_board_TransducerPosition=10 (USB MAVEL) 1 ;Driver OK

Kind_board_TransducerPosition=11 (YIDDISH MAVEL) 1 ;Driver OK

Kind_board_TransducerPosition=102 (P. E. DIGITALE PASSIVO) 2 ;Device_R + 1
se l'asse deve essere fasato

Kind_board_TransducerPosition=102 (P. E. DIGITALE PASSIVO) 1 ;Device_R
se l'asse e' senza fasatura

nel caso YIDDISH vengono usati 8 bits per asse, in ordine di asse

FalseTrue_logic_DRVOK_switch=1

stato dell'input con DRIVER OK

= 0 input a zero con DRIVER OK

= 1 input a uno con DRIVER OK

<> 0 o 1 input non gestito

Number_PHASE_switch_port="OUT13"

con nome nullo ("") nei seguenti casi viene usato l'output di default

Kind_board_DAC=1 (TIP102) 2 ;Phase axis (LED)

Kind_board_DAC=10 (USB MAVEL) 2 ;Phase axis

Kind_board_DAC=11 (YIDDISH MAVEL) 2 ;Phase axis

nel caso YIDDISH vengono usati 8 bits per asse, in ordine di asse

FalseTrue_logic_PHASE_switch=1

stato dell'output con PHASE

= 0 output messo a zero

= 1 output messo a uno

<> 0 o 1 output non gestito

Delay_PHASE=2.0

tempo di fasatura in secondi (salita tensione in meta' tempo)

per assi senza fase tempo di OMO a battuta

Volt_PHASE=3.0

tensione di fasatura

per assi senza fase tensione di OMO a battuta

Number_OMO_switch_port=""

con nome nullo ("") nei seguenti casi viene usato l'input di default

Kind_board_TransducerPosition=1 (TIP102) 4 ;Reference Switch Input

Kind_board_TransducerPosition=10 (USB MAVEL) 2 ;Reference Switch Input

Kind_board_TransducerPosition=11 (YIDDISH MAVEL) 2 ;Reference Switch Input

Kind_board_TransducerPosition=102 (P. E. DIGITALE PASSIVO) 8 ;Device_R + 1
nel caso YIDDISH vengono usati 8 bits per asse, in ordine di asse

FalseTrue_logic_OMO_switch=1

stato dell'input di OMO

= 0 input a zero con micro OMO schiacciato

= 1 input a uno con micro OMO schiacciato

◆ 0 o 1 input non gestito

◆

Space_OMO=-1000.0

dimensione e verso di ricerca micro di omo

Space_OMO_Micro_OFF=0

dimensione di rilascio micro di omo (con 0 pari a Space_OMO_Micro)

Space_OMO_Marker=0

dimensione di ricerca marker (con 0 pari a Space_OMO_Micro)

Vel_OMO=5000.00

velocita' in mm/minuto di ricerca micro

Vel_MARKER=5000.00

velocita' in mm/minuto di ricerca marker

Number_MARKER_switch_port=""

nome dell'input usato per leggere la tacca di marker (STEPPER)

FalseTrue_logic_MARKER_switch=0

stato dell'input con tacca di marker trovata (STEPPER)

Number_DIRECTION_switch_port=""

nome dell'output che indica direzione (STEPPER)

FalseTrue_logic_DIRECTION_switch=0

stato dell'output per avere direzione positiva (STEPPER)

Number_STEP_switch_port=""

nome dell'output che abilita l'esecuzione di 1 step (STEPPER)

FalseTrue_logic_STEP_switch=0

stato dell'output per avere 1 step (STEPPER)

Number_BOOST_switch_port=""

nome dell'output che abilita l'aumento di corrente (STEPPER)

FalseTrue_logic_BOOST_switch=0

Con bit 0x10000000 di Kind_axis_1 = 0 (STEPPER)
stato dell'output per avere l'aumento di corrente
con asse in movimento

Con bit 0x10000000 di Kind_axis_1 = 1 (STEPPER)
stato dell'output per avere l'aumento di corrente con
l'asse in accelerazione

Number_BRAKE_switch_port=""

nome dell'output che comanda il freno asse

FalseTrue_logic_BRAKE_switch=0

stato dell'output per asse frenato

Time_BRAKE_Lock=0

tempo in secondi di frenatura asse (servo e freno attivi)

Time_BRAKE_Unlock=0

tempo in secondi di sfrenatura asse (servo e freno attivi)

Time_BRAKE_Enable=0

tempo in secondi di asservimento abilitato prima di iniziare a sfrenare

Number_PositiveOverTravel_switch_port=""

nome dell'input usato per sentire OVER TRAVEL POSITIVE

FalseTrue_logic_PositiveOverTravel_switch=0

stato dell'input con asse in OVER TRAVEL POSITIVE

Number_NegativeOverTravel_switch_port=""

nome dell'input usato per sentire OVER TRAVEL NEGATIVE

FalseTrue_logic_NegativeOverTravel_switch=0

stato dell'input con asse in OVER TRAVEL NEGATIVE

Number_MOV_READY_switch_port=""

nome dell'output che segnala che il movimento e' pronto

FalseTrue_logic_MOV_READY_switch=1

stato dell'output per movimento pronto

Number_MOV_EXEC_switch_port=""

nome dell'input usato per autorizzare il movimento

FalseTrue_logic_MOV_EXEC_switch=1
stato dell'input per movimento autorizzato

}

;Termine parametri asse

Con asservimento in velocita' viene calcolato il KC (trasforma i valori interni in volt digit) usando Volt_vel e Vel_max.

Con asservimenti in corrente KC viene calcolato usando Volt_vel e Acc_max.

Per non cambiare i valori di KP KI KD KD2 KS bisogna cambiare i valori usati mantenendo invariato il rapporto tra loro. e Acc_max.

Per non cambiare i valori di KP KI KD KD2 KS bisogna cambiare i valori usati mantenendo invariato il rapporto tra loro.

2.3.1.1 Parametri relativi agli assi Tabella riassuntiva

La caratterizzazione di ogni Asse inizia con il nome sintetico dell'asse racchiuso tra 2 parentesi quadre (nel nostro caso =[X1])

Note Importanti

- 1) I caratteri scritti in questo campo possono essere maiuscoli o minuscoli, ma nei due casi sono riconosciuti come assi diversi. (ad esempio gli Assi **X1** e **x1** sono due assi diversi)
- 2) Il nome dell'asse può essere al massimo di **7 caratteri**
- 3) Il PHYSICAL_AXIS_NAME deve essere lo stesso indicato all'inizio tra parentesi quadre
- 4) Per la scelta dei valori da inserire in caratterizzazione, si veda il capitolo "PID", nel quale sono trattate tutte le varie possibilità di taratura per Assi Analogici, Digitali, con motori lineari, con motori Stepper ecc.
- 5) Nella struttura seguente sono indicati i valori di default.

[X1]	
i=PHYSICAL_AXIS_NAME(1,"X1")	Numero e nome
b={	
Description="Asse X1"	
Kind_axis="ARM"	
Kind_axis_0=0x00000000	
Kind_axis_1=0x001F0000	
Kind_board_TransducerPosition=40	N° di riconoscimento del tipo di scheda
Board_number_TransducerPosition=0	Posizione fisica del Back-plane in cui si trova la scheda su cui risiede il canale Encoder di questo Asse
Address_port_TransducerPosition=0	
Physical_number_TransducerPosition=1	
Kind_board_DAC=40	
Board_number_DAC=0	
Address_port_DAC=0	
Physical_number_DAC=1	
Number_SERVO_switch_port="EnableX1"	
FalseTrue_logic_SERVO_switch=1	
Number_DRVOK_switch_port="DrOKX1"	
FalseTrue_logic_DRVOK_switch=1	
Number_MOV_READY_switch_port=""	
FalseTrue_logic_MOV_READY_switch=2	
Number_MOV_EXEC_switch_port=""	

FalseTrue logic MOV EXEC switch=2	
K DAC 1Volt=0	
Level TCH=0	
Revolution=0.0	
KP=10.0	
KI=0.0	
VoltMaxIntegratedError=1.0	
KD=0.0	
KD2=0.0	
Molt VFF=0.0	
Molt AFF=0.0	
KS=0.0005	
Step=-0.001	
Volt max=10	
Volt offset=0	
KC Parameter=3000/10*(20/1)	
Vel max=45000.0	
Acc max=1000.0	
Dec max=1000.0	
Time acc nla=-1.0	
Time dec nla=-1.0	
ServoError standby=2.0	
ServoError motion=15.0	
Position error=0.5	
Cycles positioning=2.0	
Cycles wait=0.02	
VelStopCounter=1.0	
VoltStopCounter=0.5	
DelayStopCounter=0.2	
TimeOutDriverOk=1.0	
Number_OMO_switch_port="HomeX1"	
FalseTrue logic OMO switch=1	
Space OMO=-10000.0	
Vel_OMO=1000.0	
Vel MARKER=500.0	
Offset_axis=0.0	
Positive over travel=0.0	
Negative over travel=0.0	
Number PositiveOverTravel switch_port=""	
FalseTrue logic PositiveOverTravel switch=2	
Number NegativeOverTravel switch_port=""	
FalseTrue logic NegativeOverTravel switch=2	

Backslash=0.0	
Number BRAKE_switch_port=""	
FalseTrue logic BRAKE_switch=2	
Time BRAKE_Enable=1.0	
Time BRAKE_Lock=1.0	
Time BRAKE_Unlock=1.0	
Axis_1_name_anticollision=""	
Distance_1_anticollision=0.0	
Kind_axis_1_anticollision=0	
Axis_2_name_anticollision=""	
Distance_2_anticollision=0.0	
Kind_axis_2_anticollision=0	
Axis_name_slave=""	
K_follower_slave=0.0	
VoltGantry=0.0	
DeltaOffset_SlaveGantry=0.0	
Number PHASE_switch_port=""	
FalseTrue logic PHASE_switch=2	
Delay_PHASE=15.0	
Volt_PHASE=0.0	
Number MARKER_switch_port=""	
FalseTrue logic MARKER_switch=2	
Number DIRECTION_switch_port=""	
FalseTrue logic DIRECTION_switch=2	
Number STEP_switch_port=""	
FalseTrue logic STEP_switch=2	
Number BOOST_switch_port=""	
FalseTrue logic BOOST_switch=2	
StepsEncoder = 0	
MaxStepsLost =0	
MarkerSteps =0	
Space OMO_Micro_OFF=0	
Space OMO_Marker=0	
e=}	

2.3.1.2 Parametri relativi agli assi virtuali

Gli assi virtuali servono a trasformare delle coordinate cartesiane (come espresse dal programmatore) in movimenti degli assi fisici (non cartesiani) tali da realizzare gli spostamenti lungo e sui punti dello spazio cartesiano.

Pertanto la programmazione di un Robot SCARA o Cilindrico viene effettuata come se fosse un Robot Cartesiano ed il Controllo provvede in questi casi a correlare automaticamente le caratteristiche fisiche del ROBOT alle quote programmate.

Tramite la struttura seguente, si comunicano al Sistema le caratteristiche fisiche (Lunghezza dei singoli bracci, ecc) per dargli le informazioni per eseguire correttamente le trasformazioni.

Anche tramite programmazione di Assi Virtuali si possono effettuare interpolazioni Lineari e Circolari e Spline.

La tabella seguente serve a raggruppare gli assi reali che realizzano il braccio del Robot, nelle tipologie :

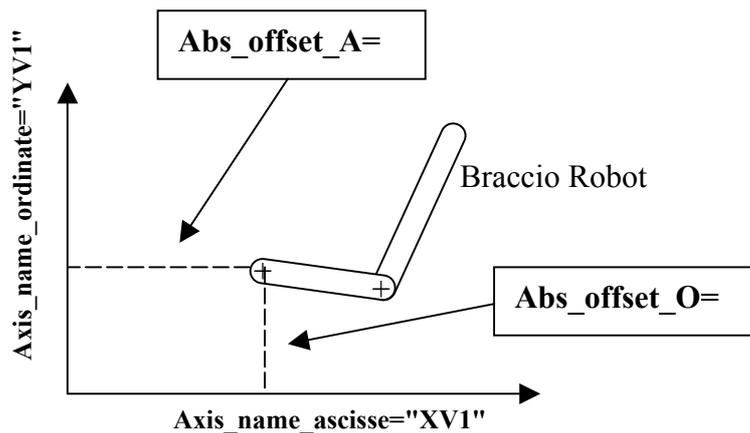
- Braccio SCARA
- Braccio SCARA Pantografo
- Braccio Cilindrico Canotto
- Braccio Cilindrico Traslante

TABELLA DI DEFINIZIONE DEGLI ASSI VIRTUALI

[XV1]	Nome del primo asse Virtuale (XV1)
i=VIRTUAL_AXES_NAME(10,"XV1","YV1")	Numero e nomi dei due assi virtuali
b={	
Kind_axis="ARM"	NON MODIFICARE
Kind_axis_0=0x00000000	0x00010000 Sistema SCARA (*) 0x00020000 SCARA Pantografo (compensato in positivo) 0x00040000 SCARA Pantografo (compensato in negativo) 0x00080000 Cilindrico Canotto 0x00100000 Cilindrico Traslante
Kind_axis_1=0x00000000	NON MODIFICARE
Axis_name_ascisse="XV1"	Nome dell'Asse Virtuale scelto come Ascisse
Axis_name_ordinate="YV1"	Nome dell'Asse Virtuale scelto come Ordinate
Axis_name_1="A1"	1° Asse fisico Es: [A1] (*)
Axis_name_2="B1"	2° Asse Fisico Es: [B1] (*)
Axis_name_3="C1"	3° Asse Fisico Es: [C1] (*)
Axis_name_4="X1"	4° Asse Fisico Es: [X1] (*)

Lenght arm 1=200.0	Prima parte lunghezza Braccio (**)
Lenght arm 2=200.0	Seconda parte lunghezza Braccio (**)
Abs offset A=100.0	Valore dell'Ascissa dell'Offset Assoluto (***)
Vel max axis A=20000.0	Velocità max dell'Asse virtuale lungo le Ascisse
Acc max axis A=200.0	Accelerazione max dell'Asse virtuale lungo le Ascisse
Dec max axis A=200.0	Decelerazione max dell'Asse virtuale lungo le Ascisse
Abs offset O=100.0	Valore dell' Ordinata dell'Offset Assoluto (***)
Vel max axis O=20000.0	Velocità max dell'Asse virtuale lungo le Ordinate
Acc max axis O=1000.0	Accelerazione max dell'Asse virtuale lungo le Ordinate
Dec max axis O=1000.0	Decelerazione max dell'Asse virtuale lungo le Ordinate
e=}	

- (*) A1, B1, C1 ,X1 sono esempi di nomi dati agli Assi fisici. Ognuno di questi assi fisici deve essere tarato nella sua Struttura Assi che inizia ad esempio con [A1] ecc.
 Gli assi rotanti (ad esempio per lo SCARA = A1, B1, C1) devono essere caratterizzati in gradi nella voce **Step=-0.001** (Il valore di default =-0,001 è tipico di assi lineari. Per gli assi rotanti occorrerà inserire un valore tale che, tenendo conto dei rapporti di trasmissione ecc, faccia corrispondere 1 giro di braccio con 360°. Il verso di conta dell'Encoder del motore deve essere positivo quando il braccio ruota in senso antiorario, come da norme della trigonometria.)
- (**) Gli assi lineari del Robot devono avere tutti la stessa unità di misura, sia per quanto riguarda le Lunghezze, gli Offset, le Velocità, le Accelerazioni.
- (***) Valore dell'Ascissa e dell'Ordinata dell'Offset Assoluto (Unità di misura degli Assi Lineari)
 L'origine degli assi Virtuali che realizzano il Robot, può essere shiftato rispetto all'origine tramite le due voci **Abs_offset_A=** e **Abs_offset_O=**



2.3.1.3 NOTE VARIE SUI SISTEMI VIRTUALI

Massima velocità: corrisponde alla massima velocità voluta sull'asse virtuale; se questa velocità fosse impossibile da effettuare (occorre tener presente che i due assi reali abbinati all'asse virtuale hanno ognuno una massima velocità) per default viene utilizzata la massima possibile.

Massima accelerazione e massima decelerazione: corrispondono alla massima accelerazione e decelerazione volute sull'asse virtuale; anche per questi dati se i valori indicati fossero superiori al possibile, per default vengono utilizzati i massimi possibili.

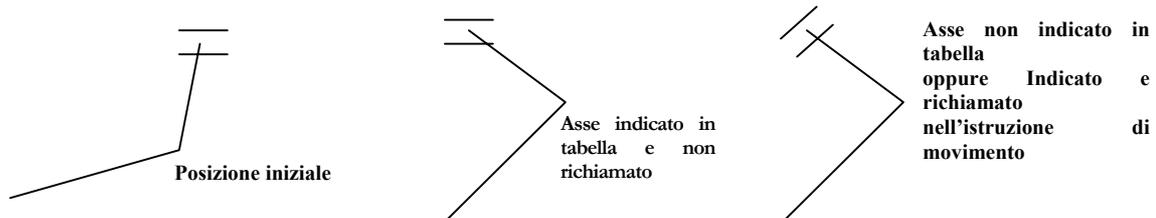
Terzo Asse - Nel caso sia presente il terzo asse (nell'esempio C1 dello SCARA), l'asse pinza deve essere concentrico con l'asse di rotazione del terzo asse (in questo caso la lunghezza del secondo asse è fissa).

Se invece l'asse pinza non fosse concentrico, la lunghezza del secondo asse (rispetto all'asse pinza) sarebbe variabile, essendo dipendente dalla posizione angolare della pinza. Di conseguenza, poiché si imposta una sola lunghezza, il sistema cartesiano ottenuto sarà valido solo nelle condizioni di concentricità.

Nome del terzo asse: indica l'eventuale asse rotante del polso, posto all'estremità del secondo asse; (nell'esempio C1).

L'indicazione o meno di questo asse in tabella, influisce solo sulla geometria del sistema agli effetti della compensazione automatica di detto rotante e precisamente:

La compensazione automatica consiste nel mantenere la pinza del polso automaticamente parallela alla sua posizione di partenza. Ciò permette di equiparare lo spostamento della pinza a quello di un sistema cartesiano.



3° Asse non indicato nella struttura Virtuale: in caso di richiesta movimento assi tale asse rotante viene ignorato. (se comandato, come asse reale, esegue il movimento richiesto con riferimento al proprio zero)

3°Asse indicato nella struttura Virtuale: in caso di richiesta movimento assi virtuali tale asse rotante, se non richiesto nel movimento, viene automaticamente compensato e la pinza posta su tale asse rimane quindi sempre parallela a se stessa. Se richiesto nel

movimento, l'asse ruoterà, in modo coordinato col movimento degli altri assi, fino alla posizione angolare programmata.

Vincoli e criteri di programmazione per assi virtuali

- Con un sistema di Assi Virtuali, nella stessa istruzione di movimento, i primi due Assi (A1 e B1) devono sempre essere entrambi virtuali oppure fisici, ma mai uno virtuale e l'altro fisico.

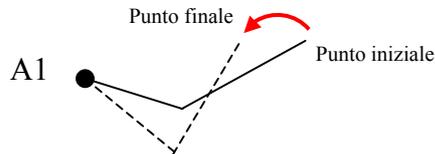
Esempio di istruzione errata : **MOV/XV,50,B1,20** (XV è virtuale mentre B1 è fisico)

E' viceversa possibile richiedere assi virtuali unitamente al 3° e/o 4° asse reale del sistema. Es: **MOV/XV,50,YV,100,C1,25**

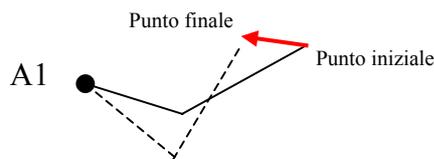
- Il 4° asse (traslante), viene utilizzato nelle interpolazioni virtuali ma deve essere programmato con valori espliciti (indicati nell'istruzione di movimento)
- Programmando l'istruzione di MOV senza esplicitare la velocità, il movimento degli assi virtuali per raggiungere la quota finale non segue una traiettoria rettilinea, ma quella naturale .

Esplicitando invece la velocità, il movimento segue una traiettoria lineare sul piano Virtuale

MOV senza velocità esplicita Es: **MOV/XV1,80,YV1,40** (coordinate del punto finale)

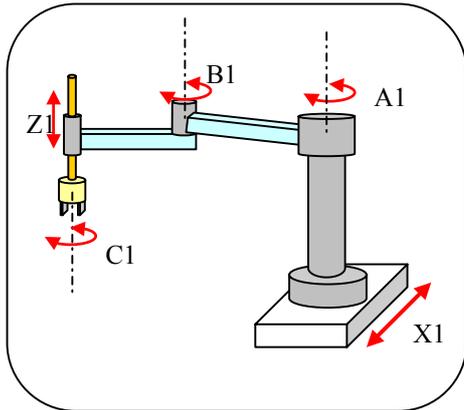


MOV con velocità esplicita Es: **MOV/1000:XV1,80,YV1,40** (coordinate del punto finale)



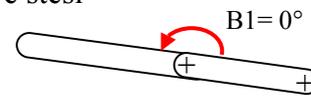
2.3.1.4 Sistema SCARA

Il sistema SCARA viene definito caratterizzando **Kind_axis_0=0x00010000**

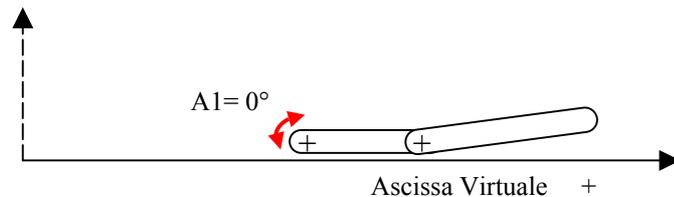


Vincoli di Taratura

- Con l'asse rotante B1 a Zero gradi (0°) i due bracci devono essere allineati e stesi



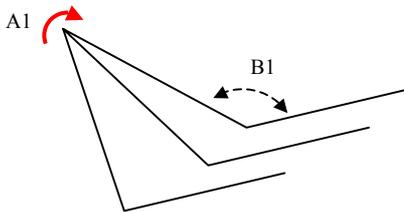
- Con l'Asse rotante A1 a Zero gradi (0°) si definisce la direzione ed il verso della Ascissa Virtuale



- La taratura **Axis_name_1="A1"** definisce l'asse fisico A1 (1° rotante-obbligatorio per SCARA)
- La taratura **Axis_name_2="B1"** definisce l'asse fisico B1 (2° rotante-obbligatorio per SCARA)
- La taratura **Axis_name_3="C1"** definisce l'asse fisico rotante del polso, se esiste.
- La taratura **Axis_name_4="X1"** definisce l'asse traslante, se esiste
Nota - Se è usato l'asse traslante X1, deve essere parallelo e concorde alla Ascissa Virtuale.
- Il valore **Lenght_arm_1=200.0** è la lunghezza del braccio tra il centro di rotazione di A1 e di B1
- Il valore **Lenght_arm_2=200.0** è la lunghezza dell'avambraccio tra il centro di rotazione di B1 e l'asse della mano (ovvero, se esiste, il centro di rotazione di C1)

2.3.1.5 SCARA PANTOGRAFO

Costruttivamente lo SCARA pantografo, al muovere dell'asse A1 (1° rotante) mantiene meccanicamente l'avambraccio sempre parallelo a se stesso. In realtà l'asse B1 (2° rotante) ruota, trascinato meccanicamente, in senso opposto dello stesso angolo comandato per A1. In questo modo il movimento è quello schematicamente rappresentato in figura



Note sulla gestione SCARA pantografo

Tutta la struttura della gestione degli assi virtuali è quella dello SCARA.

L'unico campo differente è quello relativo al **Kind_axis_0=** che deve essere caratterizzato come segue:

Kind_axis_0=0x00020000 SCARA Pantografo (compensato in positivo)

Kind_axis_0=0x00040000 SCARA Pantografo (compensato in negativo)

Il verso della compensazione va inteso nel modo seguente:

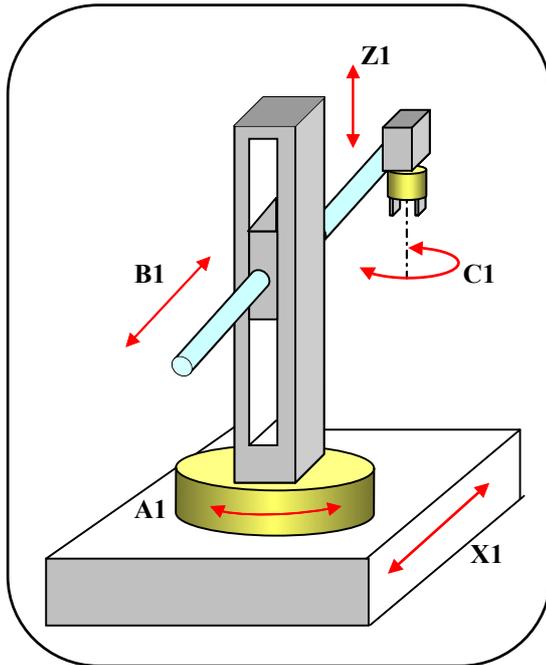
Con compensazione positiva si intende compensare l'effetto pantografo sommando B1 l'angolo effettuato dall'asse A1.

Con compensazione negativa si intende compensare l'effetto pantografo sottraendo all'asse B1 l'angolo effettuato dall'asse A1.

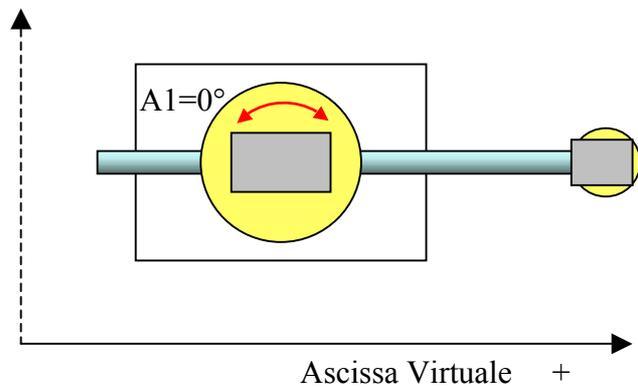
2.3.1.6 Sistema CILINDRICO CANOTTO

Il sistema CILINDRICO CANOTTO viene definito caratterizzando

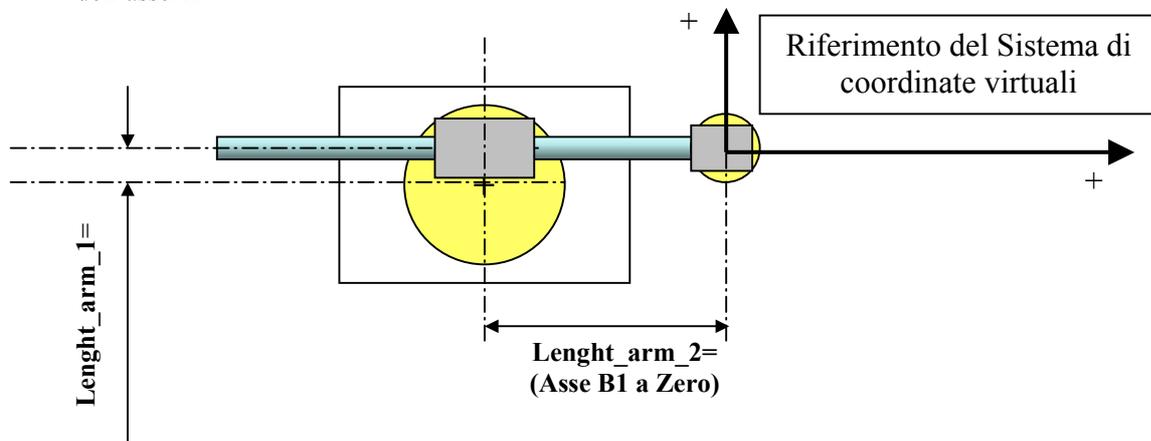
Kind_axis_0=0x00080000



- Con l'Asse rotante A1 a Zero gradi (0°) si definisce la direzione ed il verso della Ascissa Virtuale



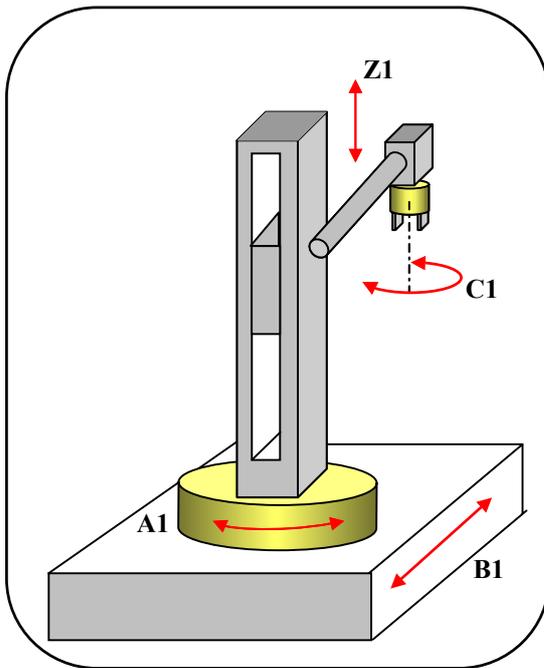
- La taratura **Axis_name_1="A1"** definisce l'asse fisico A1 (1° rotante-obbligatorio per CILINDRICO)
 - La taratura **Axis_name_2="B1"** definisce l'asse fisico B1 (Canotto-obbligatorio per CILINDRICO)
 - La taratura **Axis_name_3="C1"** definisce l'asse fisico rotante del polso, se esiste.
 - La taratura **Axis_name_4="X1"** definisce l'asse traslante, se esiste
- Nota - Se è usato l'asse del carro traslante X1, deve essere parallelo e concorde alla Ascissa Virtuale.
- Il valore **Lenght_arm_1=0** è il valore del **disassamento** tra il centro di rotazione di A1 e l'asse canotto.
 - Il valore **Lenght_arm_2=200.0** è la distanza tra lo Zero dell'asse canotto ed il centro di rotazione dell'asse A1



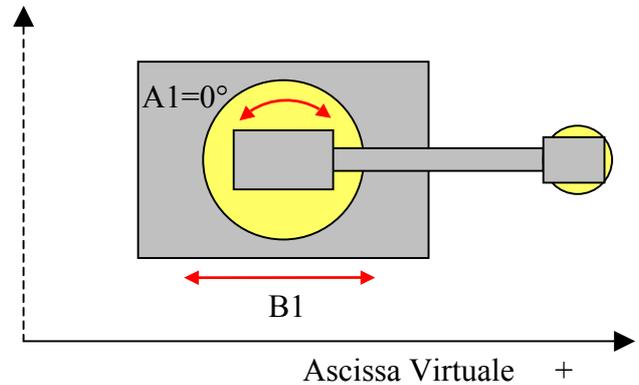
2.3.1.7 Sistema CILINDRICO TRASLANTE

Il sistema CILINDRICO TRASLANTE viene definito caratterizzando

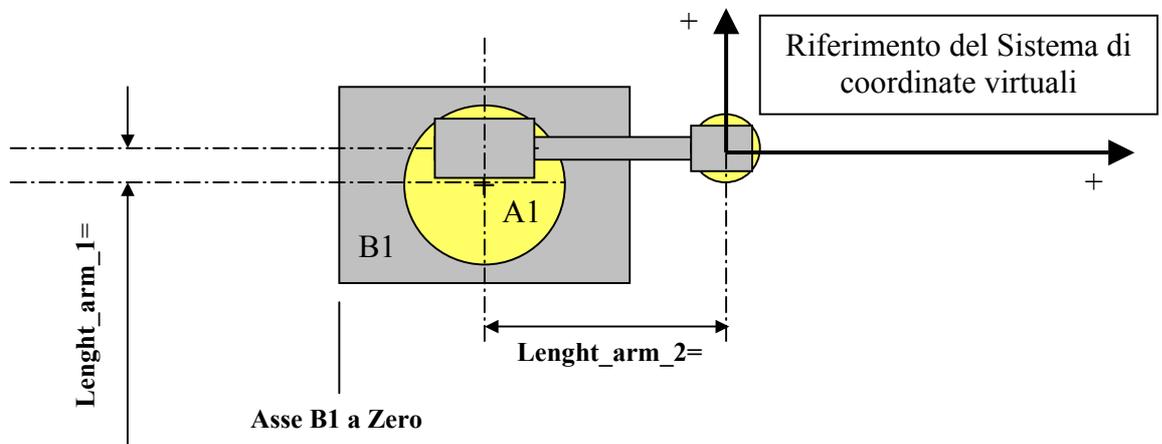
Kind_axis_0=0x00100000



- Con l'Asse rotante A1 a Zero gradi (0°) si definisce la direzione ed il verso della Ascissa Virtuale



- La taratura **Axis_name_1="A1"** definisce l'asse fisico A1 (1° rotante-obbligatorio per CILINDRICO)
 - La taratura **Axis_name_2="B1"** definisce l'asse fisico B1 (Carro-obbligatorio per CILINDRICO)
 - La taratura **Axis_name_3="C1"** definisce l'asse fisico rotante del polso, se esiste.
 - La taratura **Axis_name_4="X1"** IN QUESTO CASO NON ESISTE
- Nota - L'asse del carro traslante è B1 e deve essere parallelo e concorde alla Ascissa Virtuale.
- Il valore **Lenght_arm_1=0** è il valore del **disassamento** tra il centro di rotazione di A1 e l'asse canotto.
 - Il valore **Lenght_arm_2=200** è la distanza tra il centro dell'asse di rotazione del polso ed il centro di rotazione dell'Asse A1



2.3.1.8 Note sugli assi stepper

Gli assi stepper vengono definiti in caratterizzazione utilizzando i seguenti campi di caratterizzazione asse

Kind_axis_1=0x10070000
 ;Maschera 1 a bits dell'asse
 ;0x10000000 = BOOST ATTIVATO IN ACC. (SOLO CON ASSE STEPPER)

Kind_board_TransducerPosition=
 ;Tipo di piastra trasduttore
 ;1000 = ASSE PER STEPPER A BITS
 ;1002 = ASSE PER STEPPER A CONTATORE PCI2000
 ;1003 = ASSE PER STEPPER A CONTATORE PCI2001
 ;1100 = ASSE PER STEPPER RSM304 ETN SERVO P.E.

Board_number_TransducerPosition=0
 ;Numero di piastra per trasduttore
 ;1000 unused STEPPER A BITS
 ;1002 0 - n numero piastra PCI2000
 ;1003 0 - n numero piastra PCI2001
 ;1100 unused ASSE PER STEPPER RSM304 ETN SERVO P.E.

Address_port_TransducerPosition=0x0
 ;Indirizzo asse trasduttore
 ;1000 unused
 ;1001 unused
 ;1002 unused
 ;1003 unused
 ;1100 6 - n indirizzo DRIVER CONTROL modulo (M-1)*2
 4 moduli consecutivi con M>=4

Physical_number_Transducerposition=
 ;Numero fisico asse trasduttore
 ;1000 unused STEPPER A BITS
 ;1002 1 - 3 ASSE PER STEPPER A CONTATORE PCI2000(pulse 1 - 3)
 ;1003 1 - 3 ASSE PER STEPPER A CONTATORE PCI2001 (pulse 1 - 3)
 ;1100 1 - 3 Numero motore

Kind_board_DAC=

;Tipo di piastra asse convertitore
 ;1000 = ASSE PER STEPPER A BITS
 ;1002 = ASSE PER STEPPER A CONTATORE PCI2000
 ;1003 = ASSE PER STEPPER A CONTATORE PCI2001
 ;1100 = ASSE PER STEPPER RSM304 ETN SERVO

Board_number_DAC=0

;Numero di piastra asse convertitore
 ;1000 unused STEPPER A BITS
 ;1002 0 - n numero piastra PCI2000
 ;1003 0 - n numero piastra PCI2001
 ;1100 unused ASSE PER STEPPER RSM304 ETN SERVO

Address_port_DAC=0

;Indirizzo asse convertitore
 ;1000 unused STEPPER A BITS
 ;1002 unused ASSE PER STEPPER A CONTATORE PCI2000
 ;1003 unused ASSE PER STEPPER A CONTATORE PCI2001
 ;1100 6 - n indirizzo DRIVER CONTROL modulo (M-1)*2
 4 moduli consecutivi con M>=4

Physical_number_DAC=2

;Numero fisico asse convertitore
 ;1000 unused STEPPER A BITS
 ;1002 1 - 3 STEPPER A CONTATORE PCI2000 (pulse 1 - 3)
 ;1003 1 - 3 STEPPER A CONTATORE PCI2001 (pulse 1 - 3)
 ;1100 1 - 3 Numero motore

KP=10.0

guadagno proporzionale (errore di posizione usato ad asse fermo)

KD=100

Per assi stepper definisce la frequenza di asservimento (Hz)

KD2=400

Per assi stepper definisce la frequenza di scavalco risonanza (Hz)

KS=0.01

Valore di troncamento per il PID. Deve essere uguale o maggiore del passo del motore a stepper

KM=0.01

Per assi stepper misti ("Kind_board_TransducerPosition=..." e "Kind_board_DAC=1000") questo campo deve rappresentare il valore di uno step del motore, perche' il campo Step= e' usato dall'encoder
 Con assi stepper che hanno "Kind_board_TransducerPosition=1000" e "Kind_board_DAC=1000" viene usato il solo valore di "Step=.." e quindi "KM=.." non e' usato. (Quando e' usato viene comunque sempre automaticamente reso assoluto)

KN=0

Per assi stepper a contatore definisce la frequenza di base del contatore in Hz // ATTUALMENTE NON SERVE RISERVATO AD ALTRI USI

FUTURI

Volt_max=12000

massima "frequenza" in Hz. Con gli assi stepper non viene controllata la tensione, ma la frequenza massima che puo' sopportare il motore

Volt_vel = 10

"frequenza" necessaria per muoversi a Vel_max
 Usato per le plottature del comando (frequenza) che e' visualizzato in volts. Se il valore e' la frequenza necessaria per andare a vel_max il valore visualizzato e' la frequenza. Se il valore e' diverso tutte le frequenze sono visualizzate in proporzione. Cio' permette di impostare valori che possano essere plottati.

KC_Parameter = 0

Questo parametro permette di definire il rapporto di conversione per l'asservimento, in alternativa a Volt_vel Vel_max.

Se e' zero vengono utilizzati i due campi suddetti altrimenti viene usato questo.

Il significato fisico del campo e' la velocita' (u/min) per 1 Hz o per valori proporzionali alla frequenza. Se per esempio si vogliono avere 10 con una velocita' 600 il valore sara' 600/10. Questo campo serve per mettere il valore voluto per la "plottatura"; il segno puo' servire a invertire la direzione dell'asse.

Step = +0.001

moltiplicatore impulso -> coordinata calcolo (In stepper viene forzato automaticamente in assoluto)

StepsEncoder

= Numero impulsi dell'Encoder (Solo se si desidera il recupero delle perdite di conteggio)

MaxStepsLost

= Massimo numero di impulsi Encoder persi nel giro (non cumulativi) prima di segnalare errore

MarkerSteps = Numero impulsi della larghezza del marker (Default 2 se attiva la verifica della perdita passi encoder)

Vel_max=3000

Velocita' massima dell'asse

La velocita' deve essere $\leq a \text{ KM} * \text{max_Hz} * 60$

Number_MARKER_switch_port=""

nome dell'input usato per leggere la tacca di marker

FalseTrue_logic_MARKER_switch=0

stato dell'input con tacca di marker trovata

Number_DIRECTION_switch_port=""

nome dell'output che indica direzione

FalseTrue_logic_DIRECTION_switch=0

stato dell'output per avere direzione positiva

Number_STEP_switch_port=""

nome dell'output che abilita l'esecuzione di 1 step

Nel caso di STEPPER PCI2000 non viene usato , ma la corrispondente uscita PULSE

Nel caso di STEPPER RSM304 non viene usato , ma il corrispondente bit di uscita.

FalseTrue_logic_STEP_switch=0

stato dell'output per avere 1 step

Number_BOOST_switch_port=""

nome dell'output che abilita la riduzione di corrente

FalseTrue_logic_BOOST_switch=0

stato dell'output per avere la riduzione di corrente

Per il giusto segno delle "QUOTE" agire cambiando il segno a "Step="

Per cambiare la direzione al motore esistono due modi; quando la direzione e' data da un output digitale si puo' invertire lo stato dell'output "FalseTrue_logic_DIRECTION_switch=". Oppure cambiare il segno a "Volt_vel=" o a "KC_Parameter". Questo e' il solo modo possibile quando la direzione non e' data da un output digitale (cod 1100 con ASSE PER STEPPER RSM304 ETN SERVO P.E.).

ALGORITMI STEPPER "BITS"

TC = tempo campionatura in millisecondi

TickCounter = cicli di counter (1 - n)

Frequenza in Hz = $1000 * \text{TickCounter} / (\text{TC} * 2 * \text{Pulse})$

KM = incremento per passo del motore STEPPER

Vel Reg. in SAMPLE = $\text{KM} * \text{Frequenza} * \text{TC} / 1000$

Vel Reg. in SAMPLE = $\text{KM} * \text{TickCounter} / (2 * \text{Pulse})$

KN = Vel Max in SAMPLE = $\text{KM} * \text{TickCounter} / (2 * 1)$

Vel Reg. in SAMPLE = KN / Pulse

Pre.Max = $(\text{KM} * \text{TC} * \text{Freq.Max}) / 1000$

Pulse = $(1000 * \text{TickCounter} / (\text{Frequenza} * \text{TC} * 2)$

ALGORITMI STEPPER "PCI2000" E "RSM304"

La gestione della frequenza e uguale; Tclk e' sempre a 1 con RSM304

Periodo in microsecondi = $\text{Tclk} * 2 * (\text{Pulse} + 1)$

Frequenza in Hz = $1000000 / (\text{Tclk} * 2 * (\text{Pulse} + 1))$

TC = tempo campionatura in millisecondi

KM = incremento per passo del motore STEPPER

KN = Vel Max in SAMPLE = $(1000 * \text{KM} * \text{TC}) / (\text{Tclk} * 2 * (0 + 1))$

Vel Reg. in SAMPLE = $(1000 * \text{KM} * \text{TC}) / (\text{Tclk} * 2 * (\text{Pulse} + 1))$

Vel Reg. in SAMPLE = $\text{KN} / (\text{Pulse} + 1)$

Pre.Max = $(\text{KM} * \text{TC} * \text{Freq.Max}) / 1000$

Pulse = $(1000000 / (\text{Frequenza} * \text{Tclk} * 2)) - 1$

2.3.1.9 NOTE SUGLI ASSI SERCOS

Gli assi SERCOS in velocita' vengono caratterizzati ponendo il valore 200 nei campi "Kind_board_TransducerPosition=" e "Kind_board_DAC="

Nella gestione di questi assi il valore di tensione da dare al DAC non e' reale, percio' e' buona norma considerare di lavorare con 10 Volt ipotetici, in modo che i grafici per la messa a punto diano delle indicazioni nel range che normalmente e' usato (10 Volt).

Cio' comporta il caricamento dei seguenti campi nel modo che sara' descritto:

Il campo "K_DAC_1Volt=" deve essere caricato con il valore "OUTPUT" che SERCOS utilizza per muovere l'asse ad un decimo della velocita' massima.

"Volt_max=11" verra' caricato con 11 Volt. Il valore non deve limitare la velocita' massima che si intende ottenuta con 10 Volt.

In "KC_Parameter=" verra' messa la velocita' ottenuta con 1 Volt, che e' la velocita' massima diviso 10.

In "Step=" il rapporto tra le unita' SERCOS e quelle usate in AxesBrain. Di norma il valore e' 0.0001; ossia se si caratterizza l'asse in mm lo step e' un decimo di micrometro.

"Vel_max=" e' il valore di velocita' massima, che e' ipoteticamente ottenuta con 10 Volt.

E.G.

Immaginiamo di tarare un asse con velocita' massima di 3000 mm/min, con unita' SERCOS pari a 0.0001 mm e OUTPUT SERCOS di 30000000 step sercos/min (3000*10000), i campi verranno caricati con i seguenti valori:

```

[X1]
i=PHYSICAL_AXIS_NAME( 1,"X1" )
b={
Description="Asse X1"
Kind_axis="ARM"
Kind_axis_0=0x00000000
Kind_axis_1=0x001F0000
Kind_board_TransducerPosition=200 // ** SERCOS
Board_number_TransducerPosition=0 // ** SERCOS
Address_port_TransducerPosition=0 // ** SERCOS
Physical_number_TransducerPosition=1 // ** SERCOS
Kind_board_DAC=200 // ** SERCOS
Board_number_DAC=0 // ** SERCOS
Address_port_DAC=0 // ** SERCOS
Physical_number_DAC=1 // ** SERCOS
Number_SERVO_switch_port="" // ** SERCOS
FalseTrue_logic_SERVO_switch=1 // ** SERCOS
Number_DRVOK_switch_port="DrOKX1"
FalseTrue_logic_DRVOK_switch=1
K_DAC_1Volt=3000000 // 3000*10000/10 // ** SERCOS
Level_TCH=0
Revolution=0.0
KP=10.0
KI=0.0
VoltMaxIntegratedError=1.0
KD=0.0
KD2=0.0
Molt_VFF=100.0
Molt_AFF=0.0
KS=0.00005 // ** SERCOS
Step=-0.0001 // ** SERCOS
Volt_max=11.0 // ** SERCOS
Volt_offset=0.0
KC_Parameter=3000/10 // ** SERCOS
Vel_max=3000.0 // ** SERCOS
Acc_max=1000.0
Dec_max=1000.0
Time_acc_nla=-1.0
Time_dec_nla=-1.0
ServoError_standby=2.0
ServoError_motion=15.0
Position_error=0.5
Cycles_positioning=2.0
Cycles_wait=0.02

```

```
VelStopCounter=1.0
VoltStopCounter=0.5
DelayStopCounter=0.2
TimeOutDriverOk=1.0
Number_OMO_switch_port="HomeX1"
FalseTrue_logic_OMO_switch=1
Space_OMO=-10000.0
Vel_OMO=1000.0
Vel_MARKER=500.0
Offset_axis=0.0
Positive_over_travel=0.0
Negative_over_travel=0.0
Number_PositiveOverTravel_switch_port=""
FalseTrue_logic_PositiveOverTravel_switch=2
Number_NegativeOverTravel_switch_port=""
FalseTrue_logic_NegativeOverTravel_switch=2
Backslash=0.0
Number_BRAKE_switch_port=""
FalseTrue_logic_BRAKE_switch=2
Time_BRAKE_Enable=1.0
Time_BRAKE_Lock=1.0
Time_BRAKE_Unlock=1.0
Axis_1_name_anticollision=""
Distance_1_anticollision=0.0
Kind_axis_1_anticollision=0
Axis_2_name_anticollision=""
Distance_2_anticollision=0.0
Kind_axis_2_anticollision=0
Axis_name_slave=""
K_follower_slave=0.0
VoltGantry=0.0
DeltaOffset_SlaveGantry=0.0
Number_PHASE_switch_port=""
FalseTrue_logic_PHASE_switch=2
Delay_PHASE=15.0
Volt_PHASE=0.0
Number_MARKER_switch_port=""
FalseTrue_logic_MARKER_switch=2
Number_DIRECTION_switch_port=""
FalseTrue_logic_DIRECTION_switch=2
Number_STEP_switch_port=""
FalseTrue_logic_STEP_switch=2
Number_BOOST_switch_port=""
FalseTrue_logic_BOOST_switch=2
```

```
Number_MOV_READY_switch_port=""  
FalseTrue_logic_MOV_READY_switch=2  
Number_MOV_EXEC_switch_port=""  
FalseTrue_logic_MOV_EXEC_switch=2  
e=}
```

2.3.1.10 NOTE SUGLI ASSI ROLLOVER

Per definire un asse come ROLLOVER e' necessario settare Kind_axis_1= il bit 0X00000008 (ROLLOVER 0 +) e contemporaneamente caricare con il valore del modulo il campo Revolution= (che non deve essere zero).

La visualizzazione dell'asse e' sempre compresa tra 0 e +MODULO; per visualizzare la quota assoluta usare -RAV/42:

Per default i movimenti vanno nella direzione comandata, sino a raggiungere la posizione finale, che e' data in coordinate rollover.

Usando l'istruzione -CAP/100:A1,32 tutti i movimenti che seguono vengono fatti percorrendo lo spazio minore. Non viene percorso un valore > di MODULO anche dando un movimento molto grande.

Questa istruzione rimane attiva sino a che non venga annullata da -CAP/100:A1,-32 o dal rilancio di AxesBrain. Anche il reset non ne cambia lo stato.

Es. asse con modulo 360 visualizzante 60; comandando 3670 (360*10+70) l'asse si porta a 70 percorrendo 10.

Percio' se viene letta la quota con -RAV/42: e poi mosso l'asse a questa quota esso rimane fermo. Questo solo utilizzando la funzione "percorso minore" (-CAP/100:A1,32)

Rammentare che non utilizzando la funzione "percorso minore" (-CAP/100:A1,32) i movimenti percorrono tutto lo spazio comandato.

Es. asse con modulo 360 visualizzante 60; comandando 3670 (360*10+70) l'asse si porta a 70 percorrendo 3610.

Esiste anche l'asse ROLLOVER SEGNATO, che si puo' attivare con Kind_axis_1= il bit 0X00000010 (ROLLOVER +- MODULO) In questo caso le quote variano tra +- modulo e non funziona -CAP/100:A1,32. Inoltre non essendo veramente utile, potrebbe essere una prestazione da annullare.

2.3.1.11 NOTE SUI GANTRY

Sono gestiti due tipi di GANTRY:

Quello con struttura rigida che non e' ingallonabile;
vengono adattati i motori ad avere il medesimo carico.

Quello classico che deve essere guidato dai due motori,
rispettando il piu' possibile per ogni asse la posizione.

Col tipo rigido l'asse master e comandato alla quota voluta,
l'altro asse si adatta.

Viene caratterizzato mettendo sull'asse master in
Kind_axis_1=il bit 800000H (SLAVE_CHA_GANTRY)
Axis_name_slave="nome asse slave"
K_follower_slave=100.0 (100 significa gestione mediata , 0 non mediata)

Il tipo classico ha le stesse informazioni, con l'aggiunta in
Kind_axis_1=il bit 1000000H (SLAVE_CHA_HARD)
Con questo tipo bisogna anche assegnare la massima tensione di
scompenso tra i due assi che e' il campo di caratterizzazione

VoltGantry=5.0

Se il valore di VoltGantry e' ≤ 0 lo scompenso tra i due assi
non e considerato:

Inoltre bisogna definire lo scostamento dello zero marker slave
rispetto allo zero marker del master con il seguente campo
di caratterizzazione

DeltaOffset_SlaveGantry=0.640

Questi due campi sono messi sull'asse master

Nota per "DeltaOffset_SlaveGantry"

asse master su zero marker con lettura della quota reale = 0

asse slave con lettura della quota reale = +2.403

Il valore da caratterizzare deve essere -2.403 che e' la
posizione dello zero marker asse slave rispetto al corrispettivo
del master.

Mettendo anche il

Kind_axis_1=il bit 2000000H (SLAVE_CHA_THEOR)
il gantry assegna tutto l'errore di incrocio sull'asse slave; in
tal modo il master segue integralmente il teorico e l'asse slave
accelera o decelera per annullare l'errore tra gli assi.
Usando questo bit, anche per gantry rigido, l'asse slave segue
il teorico e non si adatta.

L'azzeramento degli assi GANTRY e' bene farlo con un part-program, che e' diverso a secondo dei tipi fisici di gantry.

Asse gantry non rigido con assi che non si liberano

Caratterizzazione asse master gantry YM:

```
Kind_axis_1=0x01800000H (SLAVE_CHA_HARD + SLAVE_CHA_GANTRY)
Axis_name_slave="YS"
K_follower_slave=0.0
VoltGantry=1.0
DeltaOffset_SlaveGantry=-0.640
```

Come calcolare "DeltaOffset_SlaveGantry="

Lanciare il programma seguente, attendere che l'azzeramento sia fatto e alla richiesta di movimento incrementale di asse master dare i valori che mettono gli assi gantry in posizione di lavoro. Leggere sulla riga 2 il valore da introdurre in caratterizzazione.

```
-TSK/READ_GANTRY ;Lancio del task di lettura assi gantry
-CAP/100:YM,1024 ;Disabilitazione di SLAVE_CHA_HARD
-OMO/YM ;Ricerca micro e marker asse master
-MOV/YM,100 ;Posizionamento asse per azzeramento asse slave
-CAP/0:YM,262144 ;0x40000 revers master slave del gantry
-OMO/YS ;Ricerca micro e marker asse slave
-CAP/0:YM,-262144 ;-(0x40000) ritorno al master slave del gantry
-CAP/100:YM,2048 ;Disabilitazione temporanea del gantry
REP -KYB/MOV INCREM. YM,L1
-INQ/YM
-MOV/500:YM,L1
-ABQ/YM
-JMP/REP
-END
```

```
PartProgram[READ_GANTRY]
LOOP -RAV/42:L1,YM,L2,YS
-DIS/1:1:YM = ,L1
-DIS/1:31:YS = ,L2
-DIS/2:(YM - YS) = ,L1-L2
-TMM/100
-JMP/LOOP
-RET
```

Come eseguire l'azzeramento del sistema di assi gantry
Lanciare il programma seguente

```
-OMO/YM          ;Ricerca micro e marker asse master  
-MOV/YM,100      ;Posizionamento asse per azzeramento asse slave  
-CAP/0:YM,262144 ;0x40000 revers master slave del gantry  
-OMO/YS          ;Ricerca micro e marker asse slave  
-CAP/0:YM,-262144 ;-(0x40000) ritorno al master slave del gantry  
-MOV/YM,200      ;Posizionamento asse per fine azzeramento  
-END
```

Come togliere gli assi gantry dall'ingallonamento
Lanciare il programma seguente e muovere il master
sino ad avere gli assi mobili

```
-CAP/100:YM,4096 ;L'asse YM muove anche senza HOME  
-CAP/100:YS,4096;L'asse YS muove anche senza HOME  
-CAP/100:YM,2048 ;Disabilitazione temporanea del gantry  
REP -KYB/MOV INCREM. YM,L1  
-INQ/YM  
-MOV/500:YM,L1  
-ABQ/YM  
-JMP/REP  
-END
```

Mettendo anche il
Kind_axis_0=il bit 0x00000020 (ASSE CON PARTENZA DOLCE DEL SERVO)
con il GANTRY di tipo non rigido, asservendo i due assi si portano
in posizione piu' dolcemente e il controllo di collision non viene
fatto per tutto il tempo di "Cycles_positioning".
Durante il movimento, se viene superato uno spostamento tra
i due assi del valore di servo error viene segnalato errore.

2.3.1.12 NOTE SULLA GESTIONE DI OVER TRAVEL

L'OVER TRAVEL puo' essere definito in caratterizzazione mettendo in Positive_over_travel= e Negative_over_travel= i valori di fine corsa riferiti allo zero assoluto. Quindi se viene cambiato Offset_axis= e' necessario cambiare anche i limiti di over travel perche' viene cambiato l'offset assoluto.

Questi limiti sono attivati dall'azzeramento assi; muovendo in manuale l'asse si ferma sul limite che si vuole superare.

Gli altri movimenti non partono se si supera il limite di corsa, ma danno una segnalazione di errore.

Se entrambi i limiti sono zero, non viene considerato il limite di OVER TRAVEL.

Movimenti "sino a OVER TRAVEL"

NOTE per i movimenti lanciati da HANDLING_MOV
Mettendo nel campo MODIFYER_MOV della struttura dei movimenti il valore 0x4000 (or con gli altri bit del campo) i movimenti sono eseguiti sino al limite se questi viene superato. Non vengono trattati in questo modo i movimenti in continuo e circolari e HOM.

Utilizzando i campi :

Number_PositiveOverTravel_switch_port=
FalseTrue_logic_PositiveOverTravel_switch=
Number_NegativeOverTravel_switch_port=
FalseTrue_logic_NegativeOverTravel_switch=

vengono anche sentiti i micro di fine corsa. Essi sono gestiti nel seguente modo:

- 1) Caso normale che a prescindere dall'azzeramento assi viene segnalato errore con caduta di asservimento se vogliono superarli (la direzione del movimento viene testata con il corrispondente micro di oltre corsa)
- 2) Caso con Kind_axis_0= che ha il bit 0x00000040 settato; a prescindere dall'azzeramento assi il movimento in manuale quando sente il micro decelera e si ferma senza nessuna segnalazione.

La stessa cosa succede con i movimenti "sino a OVER TRAVEL"

Per tutti gli altri movimenti vale il controllo OVER TRAVEL con i limiti in caratterizzazione; se viene incontrato il micro di oltre corsa viene segnalato errore con caduta di asservimento.

2.3.1.13 NOTE SULLA GESTIONE DI ASSI NORMALMENTE DISABILITATI

L'asse normalmente disabilitato e' un asse che quando e' fermo non e' asservito e se ha il freno e' frenato. Con movimento esso si autoasserve liberando l'eventuale freno sino a movimento concluso.

L'asse puo' essere caratterizzato tale settando in Kind_axis_0= il bit 0x00000080

Anche da part-program si puo' attivare/disattivare sull'asse la funzione di normalmente disabilitato con le seguenti istruzioni:

- 1) CAP/100:A1,512 in questo modo l'asse e' messo normalmente disabilitato, ma dopo di un reset ritorna nello stato di caratterizzazione
- 2) CAP/100:A1,-512 in questo modo l'asse non e' normalmente disabilitato, ma dopo di un reset ritorna nello stato di caratterizzazione
- 3) CAP/258:A1,512 in questo modo l'asse e' messo normalmente disabilitato, ma anche dopo un reset resta in questo stato
- 4) CAP/258:A1,-512 in questo modo l'asse non e' normalmente disabilitato, ma anche dopo un reset resta in questo stato

2.3.1.14 NOTE SULLA GESTIONE DEGLI ASSI CON FRENO

I freni sugli assi sono usati per il loro bloccaggio con asservimento disabilitato.

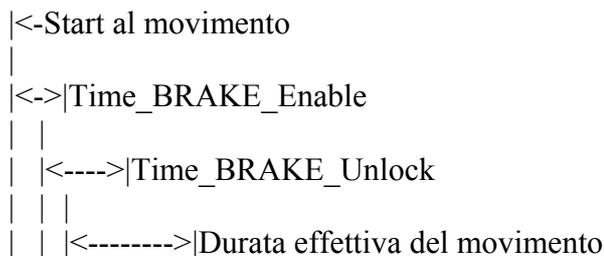
Essi vengono caratterizzati ponendo sulla struttura asse di sistema.txt le seguenti informazioni:

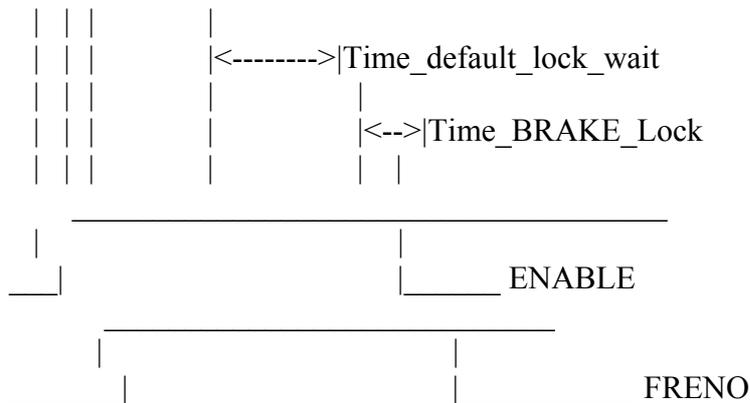
```
Number_BRAKE_switch_port=""
    nome dell'output che comanda il freno asse
FalseTrue_logic_BRAKE_switch=0
    stato dell'output per asse frenato
Time_BRAKE_Lock=0
    tempo in secondi di frenatura asse (servo e freno attivi )
    ovvero tempo necessario al freno per bloccare l'asse
Time_BRAKE_Unlock=0
    tempo in secondi di sfrenatura asse (servo e freno attivi)
    ovvero tempo necessario al freno per sbloccare l'asse
Time_BRAKE_Enable=0
    tempo in secondi di asservimento abilitato prima di iniziare a sfrenare
    ovvero tempo necessario all'asservimento per essere attivo
```

Nel caso che l'asse sia un asse normalmente disabilitato viene anche usata il seguente comando generale di sistema.txt

```
Time_default_lock_wait=0
    ;Tempo in secondi di permanenza asservimento a fine movimento
    con gli assi normalmente disabilitati
```

SCHEMA DI SFRENATURA / FRENATURA PER ASSE NORMALMENTE DISABILITATO





Gli assi si suddividono nei seguenti 2 tipi:

- a) Asse che ha l'asservimento sempre funzionante quando viene abilitato, ossia asse normalmente abilitato
- b) Asse che si attiva solo durante i movimenti anche quando viene abilitato, ossia che e' normalmente disabilitato. Questo asse e' caratterizzabile ponendo 0x80 in Kind_axis_0= o tramite CAP/100:X,512 (per "normalmente disabilitato") CAP/100:X,-512 (per "normalmente abilitato")
Se vengono usate queste CAP , un reset rimette l'asse nelle condizioni di caratterizzazione; un reset non agisce se al posto di CAP/100: viene usato CAP/228:

L'asse abilitato e' in SERVO ON, con PID funzionante e se ha il freno con FRENO SBLOCCATO

L'asse disabilitato e' in SERVO OFF, con PID escluso e se ha il freno con FRENO BLOCCATO

L'asse normalmente abilitato puo' essere diabilitato con SERVO OFF o con CPL/-2:... . La movimentazione non e' possibile in questo stato e viene segnalato errore.

Anche se l'asse ha il freno non ne viene controllata la posizione con asservimento disabilitato.

CPL/-1:... e CPL/0:... non sono accettate se sull'asse e' caratterizzato il freno.

Con l'asse normalmente disabilitato, mettendolo in SERVO OFF il funzionamento e' paragonabile all'asse normale, ma in SERVO on si hanno le seguenti funzionalita':

Se l'asse ha il freno caratterizzato, viene controllata la posizione anche durante la disabilitazione (SERVO ERROR ASSE FERMO), e riportato

in posizione di bloccaggio appena riabilitato. Un SERVO OFF o CPL/-2:.. fanno perdere queste funzionalita'

Movimentazione non in continuo, l'asse viene automaticamente abilitato con eventuale sfrenatura e riposizionamento al punto di bloccaggio. A movimento finito, dopo il tempo di attesa in abilitazione, l'asse viene automaticamente disabilitato sino ad un nuovo movimento.

Movimentazione in continuo, l'asse viene automaticamente abilitato con eventuale sfrenatura e riposizionamento al punto di bloccaggio. A movimento finito l'asse non viene disabilitato sino al termine del continuo. Quindi gli assi vengono abilitati (e sbloccati) man mano che i movimenti li richiamano, e il tutto e' accettato se la velocita' del movimento e zero.

L'asse normalmente disabilitato con freno, oltre a controllare l'errore di posizione a freno inserito, puo' essere comandato al recupero di questo errore con rampa lineare pari a accelerazione decelerazione massima della'asse e velocita massima controllata con change feedrate. Questo comando che vale per un solo recupero puo' essere dato nel seguente modo:

```

-
-LSA -CAP/100:Z,0x800000 ;Attiva 1 recupero dell'errore di posizione
      -RAV/100:L1,Z ;Legge stati dell'asse
      -JEQ/and(L1,0x800000),0,RF ;Recupero in corso ???
      -TMM/100
      -JMP/LSA
-RF - ;Recupero fatto e freno bloccato
-

```

Il recupero viene fatto se l'errore supera position error (caratterizzato o cambiato con CAP/27:Z,0.05).

Se CAP/100:Z,0x800000 viene dato non nelle condizioni di recupero, questo naturalmente non e'fatto e il comando e' perso; ossia CAP/100:Z,0x800000 vale solo per una volta ed agisce se l'asse deve recuperare . Le condizioni di recupero sono:

- 1) Asse normalmente disabilitato
- 2) Asse con freno
- 3) Freno attivo
- 4) Asse non in movimento.

Usando il comando -CAP/100:Z,0x2000000 il recupero dell'errore di servo e' fatto sempre, se ne sussistono le condizioni. La funzione e' disabilitata dal comando -CAP/100:Z,-0x2000000 o da un RESET. Si noti che in questo modo il recupero viene fatto autonomamente,

cio' comporta la possibilita' di movimento dell'asse in modo casuale; tener conto di cio' nel suo utilizzo.

2.3.1.15 NOTE SULLA MOVIMENTAZIONE CON GESTIONE A BIT

Per la gestione con bit dei movimenti occorre mettere su SISTEMA.TXT i seguenti campi nelle informazioni ASSI.

```
Number_MOV_READY_switch_port=""  
FalseTrue_logic_MOV_READY_switch=1  
Number_MOV_EXEC_switch_port=""  
FalseTrue_logic_MOV_EXEC_switch=1
```

Number_MOV_READY_switch_port= definisce l'output usato come segnalatore di asse in movimento o pronto a muovere.

Number_MOV_EXEC_switch_port= definisce l'input che permette la movimentazione vera di quell'asse. Il movimento rimane in wait sino a che l'input sia settato come FalseTrue_logic_MOV_EXEC_switch.
Il movimento eseguito termina, segnalando gli stati di asse fermi, ma non notifica la sua fine se il bit input non viene resettato.

RIASSUMENDO:

Number_MOV_READY_switch_port= permette di sapere quando un asse e' in movimento, ma non agisce sulla sua esecuzione.
Number_MOV_EXEC_switch_port= permette di gestire l'evolversi della movimentazione; il movimento deve trovare tutti gli assi del movimento (con bit input definito) settati per partire, ed a movimento finito tutti devono essere resettati per notificarne la fine.
Quando il movimento e' in continuo la gestione di Number_MOV_READY_switch_port= e' invariata.
Number_MOV_EXEC_switch_port= invece non permette i movimenti del continuo solo la prima volta che viene lanciato un movimento con almeno un asse con l'input definito. Poi a bit settato, tutti i movimenti sono lanciati senza alcun controllo. Solo l'istruzione di chiusura continuo (HLC) attende che tutti i bit, di tutti gli assi che hanno fatto parte di quel continuo, siano resettati prima di notificare continuo finito.

La gestione MOV_EXEC puo' essere disabilitata con la seguente istruzione modale:

```
CAP/100:X1,0x40000
```

Si puo' riabilitare con la funzione inversa:
CAP/100:X1,-0x40000

2.3.2 Mandrini

```
i=IO_NAME(1 , "MAND1" )  
b={  
  Kind_sensor="ANALOG_OUTPUT"  
  Kind_board=4  
  Rack_number=0  
  Address_port=0  
  Physical_number=6  
e=}
```

```
[M1]  
      ;Nome del mandrino  
SPINDLE_NAME(1, "M1" )  
      ;Numero e Nome del mandrino  
{
```

```
Kind_spindle="SPINDLE"  
      ;Definisce un mandrino
```

```
Kind_spindle_0=0x00000000  
      ;Tipo di mandrino  
      ;Maschera a bits del mandrino  
      ;0x00000001 = MANDRINO CON COMANDO A ONDA QUADRA
```

```
Number_DAC_switch_port=""  
1)  
  nome dell'output analogico usato per comandare il mandrino  
2)  
  Se il nome non esiste ("" o 0) viene usato come mandrino  
  l'asse definito con "Axis_SERVO_with_move="
```

```
Number_ACD_switch_port=""  
  nome dell'input analogico usato per leggere la velocita' del mandrino  
(unused)
```

```
NotifyEnabled=0  
(unused)
```

Axis_SERVO_with_move=""

1)

Nome asse che permette la rotazione se e' asservito

Se l'asse non e' definito la rotazione e' sempre permessa

"Number_DAC_switch_port=" con il valore di un output analogico

2)

nome dell'asse usato come mandrino

"Number_DAC_switch_port=" con il valore nullo ("" o 0)

K_DAC_1Volt=3276.7

valore digitale del DAC per ottenere 1 Volt

mettendo 0 il valore e rilevato automaticamente dal codice mandrino

Volt_max=10.0

massima tensione data al DAC (Volt)

Volt_turn=10.0

tensione (Volt) necessaria per ruotare a "Turn_max=" giri

KC_parameter=0

Questo parametro permette di definire il rapporto di conversione per il mandrino, in alternativa a Volt_turn Turn_max.

Se e' zero vengono utilizzati i due campi suddetti altrimenti viene usato questo.

Il significato fisico del campo sono i giri (giri/min) per 1 volt.

Volt_offset=+0.0

tensione di shift (Volt) necessaria per far ruotare il mandrino a Turn_min; sotto la quale il mandrino e' fermo

Turn_max=110000.0

giri/min ottenuti con "Volt_turn="

Acc_max=300.0

Acc. in giri/sec²

Dec_max=300.0

Dec. in giri/sec²

Turn_min=0.0

Giri min del mandrino; al di sotto di questi giri viene messo 0

Delay_spindle=1

Tempo di attesa in secondi per vel. regime raggiunta o per fermata

Number_ENABLE_switch_port=""
nome dell'output per l'abilitazione del mandrino

FalseTrue_logic_ENABLE_switch=1
stato dell'output per l'abilitazione del mandrino

Number_DRVOK_switch_port=""
nome dell'input per driver mandrino OK

FalseTrue_logic_DRVOK_switch=0
stato dell'input per driver mandrino OK

Number_REGIM_switch_port=""
nome dell'input per mandrino a regime (velocita' raggiunta)

FalseTrue_logic_REGIM_switch=0
stato dell'input con mandrino a regime

Number_STOP_switch_port=""
nome dell'input che segnala il mandrino fermo

FalseTrue_logic_STOP_switch=1
stato dell'input con il mandrino fermo

Number_LOCK_switch_port=""
nome dell'output che frena il mandrino (MLTA)

FalseTrue_logic_LOCK_switch=1
stato dell'output per frenare il mandrino (MLTA)
}

I Volts da dare al DAC del mandrino sono calcolati considerando la rotazione come funzione lineare del comando.

$$\text{VOLT} = \text{Volt_offset} + (\text{VELOCITA' VOLUTA} - \text{Turn_min}) * (\text{Volt_turn} - \text{Volt_offset} / (\text{Turn_max} - \text{Turn_min}))$$

Se la VELOCITA' VOLUTA e' minore di Turn_min allora VOLT = 0

Con l'asse usato come mandrino bisogna mettere i parametri asse e mandrino in modo che siano correlati:

K_DAC_1Volt= e K_DAC_1Volt= devono essere uguali

Volt_vel= e Volt_turn= devono essere uguali

Vel_max= e Turn_max= devono essere nello stesso rapporto delle unita' di misura rappresentate.

es) con asse in gradi, per avere i giri bisogna dividere la velocita' asse con 360; quindi 360000 gradi/min diventano 1000 giri/min

Acc_max= lo stesso ragionamento di Vel_max

Dec_max= lo stesso ragionamento di Vel_max

Tutti gli input/output della struttura mandrino non sono usati

Volt_offset= e Turn_min= vengono forzati a zero

2.3.2.1 NOTE SULLA GESTIONE DEL MANDRINO

La rotazione del mandrino e' lanciata con SPD/M1, giri
dove M1 e' il nome simbolico del mandrino e giri sono giri/min.
Ci sono alcuni I/O che gestiscono la rotazione.

```

Number_ENABLE_switch_port="Enable Spindle ZB"           // Output per abilit. mandrino
"OUT11"
FalseTrue_logic_ENABLE_switch=1                         // Output per abilitazione mandrino

Number_DRVOK_switch_port=""                             // Input per mandrino OK
FalseTrue_logic_DRVOK_switch=2                         // Input per mandrino OK

Number_STOP_switch_port=""                              // Input per mandrino fermo
FalseTrue_logic_STOP_switch=2                          // Input per mandrino fermo

Number_LOCK_switch_port=""                              // Output per mandrino BLOCCATO
FalseTrue_logic_LOCK_switch=2                          // Output per mandrino BLOCCATO

Number_RUN_switch_port=""                               // Input per mandrino a REGIME
FalseTrue_logic_RUN_switch=2                           // Input per mandrino a REGIME

```

CAMBIO GAMMA

Utilizzando CAP/57:nome mandrino,rapporto mandrino/motore
e CAP/58:nome mandrino,giri massimi del mandrino viene permessa
una gestione CAMBIO GAMMA permettendo di programmare in giri mandrino
Il mandrino nasce con un "CAP/57" pari a 1 e un "CAP/58" uguale al
massimo giri del motore caratterizzati.

I cambiamenti di "CAP/57" agisce nella visualizzazione del mandrino
anche se e' in rotazione; il "CAP/58" solo con un cambio di SPD.

La gestione corretta dovrebbe comunque avvenire a mandrino fermo
Con RAV/57 e RAV/58 si possono leggere i valori precedentemente
impostati.

Mettendo CAP/57:S,0 viene messo il rapporto 1 e con CAP/58:S,0 vengono
messi i massimi giri del motore che sono definiti in caratterizzazione del mandrino

```

-CAP/57:S,2           ;2 = rapporto GIRI MANDRINO / GIRI MOTORE
-CAP/58:S,10000      ;10000 = GIRI MASSIMI DEL MANDRINO
-SPD/S,5000          ;Il mandrino fa 5000 giri e il motore 2500

```

2.3.2.2 Parametri Mandrini Tabella riassuntiva

La caratterizzazione di ogni Mandrino inizia con il nome sintetico del mandrino stesso racchiuso tra 2 parentesi quadre (nel nostro caso =[M2])

Note Importanti

- 1) I caratteri scritti in questo campo possono essere maiuscoli o minuscoli, ma nei due casi sono riconosciuti come mandrini diversi. (ad esempio i Mandrini **M2** e **m2** sono **due mandrini diversi**)
- 2) Il nome del mandrino può essere al massimo di **7 caratteri**
- 3) Lo SPINDLE_NAME deve essere lo stesso indicato all'inizio tra parentesi quadre
- 4) Se il Mandrino è a LOOP APERTO, (senza Encoder) è sufficiente caratterizzare tutta la struttura seguente, dedicata ai Mandrini.
- 5) Se invece si tratta di un Mandrino a LOOP CHIUSO, (con Encoder) per la gestione del quale si utilizza il canale DAC ed Encoder di un Asse, occorrerà tarare tutta la struttura Asse e poi la struttura Mandrino seguente limitatamente alle voci indicate (*), lasciando i valori di default nelle rimanenti voci di tabella.

Un mandrino gestito come un Asse dovrà perciò essere tarato come Asse ed anche come Mandrino.

Esempio di caratterizzazione del Mandrino M2

[M2]	(*)
i=SPINDLE_NAME(2,"M2")	(*)
b={	NON MODIFICARE
Description="Mandrino M2"	(*)
Kind_spindle="SPINDLE"	NON MODIFICARE
Number_DAC_switch_port="DACM2"	A seconda che il mandrino sia a loop aperto (senza Encoder) oppure a loop chiuso (con Encoder), agire come segue: Loop aperto – dare il nome dell'output analogico scelto nella taratura degli Output analogici Loop chiuso – indicare "" (senza nome perché si usa un canale di un asse con Encoder, definito in Axis SERVO_with_move= (*)
Number_ACD_switch_port=""	NON UTILIZZATO
Axis_SERVO_with_move=""	A seconda che il mandrino sia a loop aperto (senza Encoder) oppure a loop chiuso (con Encoder), agire come segue: Loop aperto – Se l'asse indicato col nome in

	questa taratura non è sotto controllo, il Mandrino non ruota. (se non indicato un nome di asse, il mandrino non è condizionato) Loop chiuso – indicare in "" il nome dell'asse usato per il mandrino (*)
K_DAC_1Volt=0	Inserire il peso legato al n. di bit del convertitore DAC utilizzato, così calcolato: $2^{(n.bit-1)}/10.$ Esempio: con DAC a 12 bit, comprensivo di segno, il valore sarà: $2^{(12-1)}/10 = 2^{11}/10 = 2048/10 = 204,8$ Valore di default = 0 (Il sistema è predisposto per le schede 6 Assi +3 Mandrini PCI 2000 (*))
KC_parameter=300	Per Mandrini a loop aperto comandati in modo Analogico, indicare il <u>N. di giri del Motore mandrino per 1Volt di uscita del DAC.</u> (per mandrini con comando digitale vedi le tarature specifiche.) Per Mandrini a loop chiuso vedi NOTE a piè di pagina (*)
Volt_max=10	Massima tensione in uscita dal DAC Valore di default = 10 (*)
Volt_turn=0	NON MODIFICARE
Volt_offset=-0	Tensione di offset per tener fermo il mandrino Valore di default = 0
Turn_min=0	Di norma = 0 Per mandrini speciali che non possono scendere sotto un N. di giri/min. predefinito, indicare questo limite in N.di giri (es.2000) Valore di default = 0
Turn_max=3000	N. di giri/minuto max a cui può girare il mandrino. Valore di default = 3000 (*)
Acc_max=50	Accelerazione max espressa in N. di giri /sec ² Valore di default = 50 (*)
Dec_max=50	Decelerazione max espressa in N. di giri /sec ² Valore di default = 50 (*)
Delay_spindle=0	Tempo supplementare di attesa dopo che il Mandrino è arrivato al regime programmato. Valore bdi default = 0 (Taratura poco usata)
Number_ENABLE_switch_port=""	Nome dell'Output digitale utilizzato per dare l'Enable al DRIVE

FalseTrue_logic_ENABLE_switch=1	Livello logico del segnale Enable Valore di default = 1
Number_DRVOK_switch_port=""	Nome dell'Input digitale utilizzato per la segnalazione di Drive OK
FalseTrue_logic_DRVOK_switch=2	Livello logico del segnale Drive OK Valore di default = 2 (nessun controllo)
Number_REGIM_switch_port=""	Nome dell'input digitale che segnala Mandrino a regime. (Controllo poco usato)
FalseTrue_logic_REGIM_switch=2	Livello logico del segnale REGIM Valore di default = 2 (nessun controllo)
Number_STOP_switch_port=""	Nome dell'input digitale che segnala Mandrino fermo. (Controllo poco usato)
FalseTrue_logic_STOP_switch=2	Livello logico del segnale STOP Valore di default = 2 (nessun controllo)
Number_LOCK_switch_port=""	Nome dell'output digitale usato per bloccare il Mandrino. (Comando poco usato)
FalseTrue_logic_LOCK_switch=2	Livello logico del segnale LOCK Valore di default = 2 (nessun controllo)
e=}	NON MODIFICARE

NOTE SULLA GESTIONE DEI MANDRINI (*)

Mandrini a loop chiuso (utilizzano il DAC e l'Encoder di un Asse).

Nella taratura della struttura Asse relativa a questo Mandrino, (ad esempio chiamata C1) essendo la struttura relativa ad un Mandrino e non a un asse, si tarerà di norma

Kind_axis_0=0x00000008

Revolution=360 (In questo caso l'asse verrà gestito con reset della quota ad ogni giro)

Step= Valore tale che 1 giro di mandrino valga 360° (In accordo all'Encoder usato)

Da notare che un mandrino a loop chiuso, se non tarato come sopra, si comporterà come un asse, la cui quota non si resetta ad ogni giro. Pertanto, in questo caso, programmando la quota Zero, il mandrino la raggiungerà tornando indietro di tutti i giri fatti all'andata.

In questo caso si utilizzano tutte le tarature di un Asse tenendo conto dei seguenti legami:

K_DAC_1Volt della struttura Mandrino deve essere uguale al valore scritto in **K_DAC_1Volt** della taratura Asse (C1)

KC_parameter= (della struttura Mandrino) deve essere uguale a 1/3600 del valore di **Vel_max** definito per la struttura Asse (C1)

Turn_max= (della struttura Mandrino) deve essere uguale a 1/360 del valore di
Vel_max= (della struttura Asse C1)

Acc_max= (della struttura Mandrino) deve essere uguale a 1/360 del valore di
Acc_max= (della struttura Asse C1)

Dec_max= (della struttura Mandrino) deve essere uguale a 1/360 del valore di
Dec_max= (della struttura Asse C1)

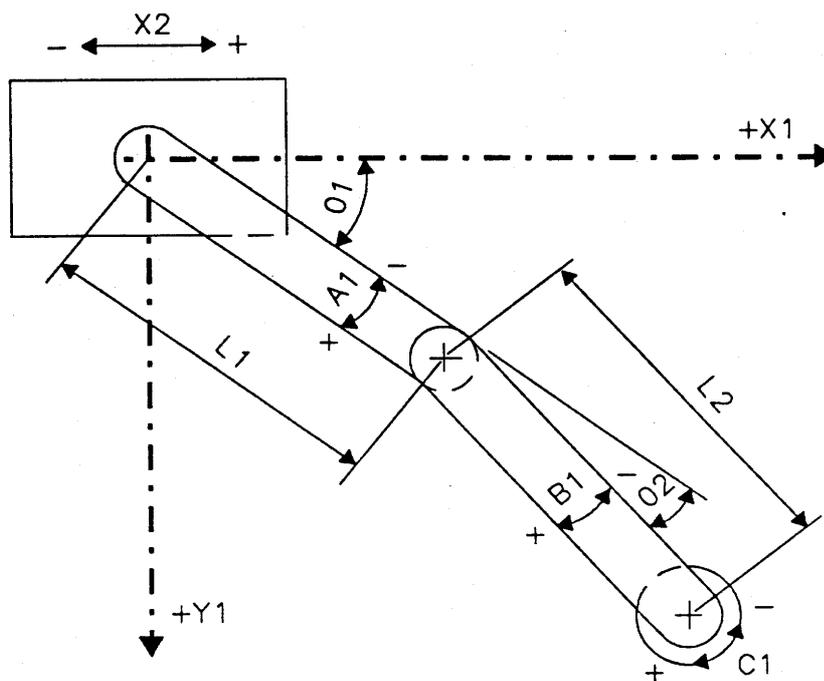
2.3.3 Assi virtuali

Gli assi virtuali sono quelli che necessitano di una trasformazione nel piano delle coordinate.

Gli assi virtuali, come dice la parola stessa, non sono assi fisici ma assi che per comodità utente vengono caratterizzati su sistemi a geometria SCARA o CILINDRICA in modo che oltre ad effettuare movimenti sugli assi fisici si possa anche effettuare movimenti secondo un sistema cartesiano.

Con tale sistema, inoltre si possono effettuare interpolazioni lineari e circolari.

2.3.3.1 Sistema SCARA



Numero di assi SCARA: indica il numero di assi da prendere in considerazione; minimo 2 (i primi 2); tre (1,2,3) oppure 4 (tutti)

Nell'esempio riportato, vengono presi in considerazione tutti e 4 (i 2 snodi A1 e B1, il 3° asse rotante C1 posto all'estremità del secondo snodo e il 4° asse traslante X2)

Nome del primo asse: indica il primo asse reale del sistema (1° snodo); nell'esempio l'asse A1

Lunghezza del primo asse: indica la lunghezza del primo asse (distanza fra il primo e secondo snodo); nell'esempio L1

Nome del secondo asse: indica il secondo asse reale del sistema (2° snodo); nell'esempio B1

Lunghezza del secondo asse: indica la lunghezza del secondo asse (distanza fra snodo centrale e asse pinza); nell'esempio L2.

Nel caso sia presente il terzo asse (nell'esempio C1), l'asse pinza deve essere concentrico con l'asse di rotazione del terzo asse (in questo caso la lunghezza del secondo asse è fissa). Se invece vi è eccentricità, la lunghezza del secondo asse (rispetto all'asse pinza) è variabile, essendo dipendente dalla posizione angolare della pinza. Di conseguenza, poiché si imposta una sola lunghezza, il sistema cartesiano ottenuto sarà valido solo in quelle condizioni.

In base a queste considerazioni, in caso di eccentricità, occorre tener presente quanto segue:

- non è possibile ottenere traiettorie rettilinee e circolari e l'errore di traiettoria sarà tanto maggiore quanto maggiore è l'eccentricità.

Nome del terzo asse: indica l'eventuale asse rotante posto all'estremità del secondo asse; nell'esempio C1.

L'indicazione o meno di questo asse in tabella, influisce solo sulla geometria del sistema agli effetti della compensazione automatica di detto rotante e precisamente:

- Asse non indicato in tabella: in caso di richiesta movimento assi (reali o virtuali) tale asse rotante viene ignorato (naturalmente se non richiesto nel movimento)
- Asse indicato in tabella: in caso di richiesta movimento assi (reali o virtuali) tale asse rotante (naturalmente se non richiesto nel movimento) viene automaticamente compensato; la pinza posta su tale asse rimane quindi sempre parallela a se stessa.
E' certo che se questo asse viene richiesto nel movimento, l'indicazione o meno dell'asse nella tabella non ha più alcun significato essendo il movimento richiesto determinante.

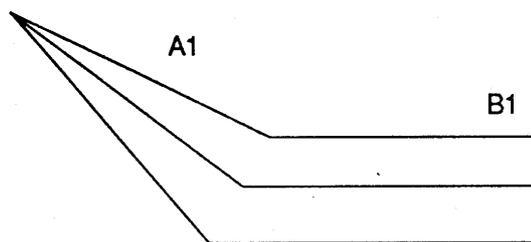
Nome del quarto asse: indica l'eventuale asse traslante (traslazione di tutto lo SCARA); nell'esempio X2

Compensazione meccanica: questo dato serve per individuare il tipo di robot secondo la convenzione seguente:

0 = robot di tipo SCARA tradizionale

± 1 = robot di tipo SCARA ma con sistema a pantografo

dove muovendo un solo asse (nell'esempio sottostante A1) il secondo asse (nell'esempio B1) si mantiene parallelo a se stesso.



Il segno algebrico va inteso nel modo seguente:

Con il segno positivo si intende la compensazione automatica sull'asse B1 sommando l'angolo effettuato dall'asse A1.

Con il segno negativo si intende la compensazione automatica sull'asse B1 sottraendo l'angolo effettuato dall'asse A1.

Offset assoluto: questo dato viene utilizzato ogni volta si desidera ottenere uno zero asse pratico diverso da quello teorico.

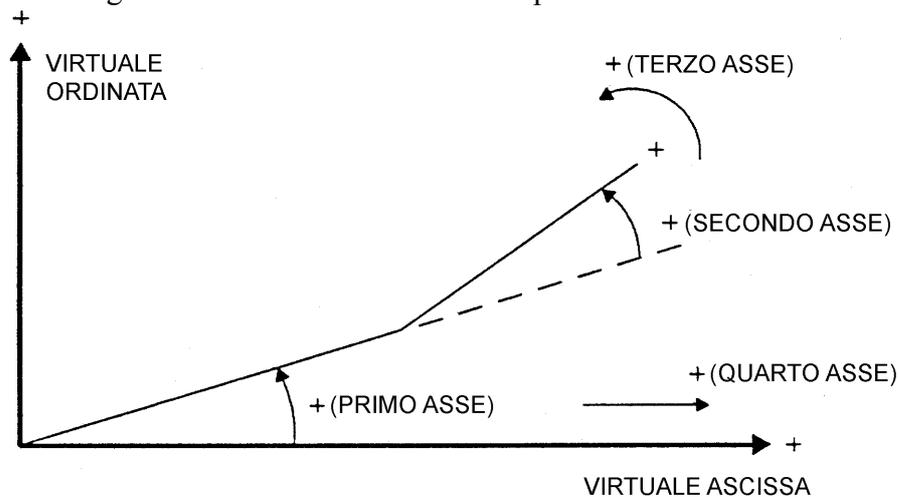
Massima velocità: corrisponde alla massima velocità voluta sull'asse virtuale; se questa velocità fosse impossibile da effettuare (occorre tener presente che i due assi reali abbinati all'asse virtuale hanno ognuno una massima velocità) per default viene utilizzata la massima possibile.

Massima accelerazione e massima decelerazione: corrispondono alla massima accelerazione e decelerazione volute sull'asse virtuale; anche per questi dati se i valori indicati fossero superiori al possibile, per default vengono utilizzati i massimi possibili.

Note:

a) Poiché tutti i calcoli per gli assi virtuali vengono eseguiti secondo la geometria classica operando su seni e coseni di angoli espressi in gradi, occorre tenere presente quanto segue:

- 1 - L'unità di misura dei primi 3 assi reali deve essere il grado.
- 2 - L'unità di misura del quarto asse, può essere il mm o pollici a seconda che le lunghezze primo e secondo asse vengano date rispettivamente in mm o inc.
- 3 - L'offset assoluto del primo asse reale deve essere dato (se è presente il 4° asse traslante) in modo che l'asse virtuale ascissa sia parallelo al quarto asse SCARA.
- 4 - L'offset assoluto del secondo asse reale deve essere dato in modo che i primi due assi risultino allineati (braccio tutto teso).
- 5 - Le direzioni di movimento (positiva, negativa) degli assi del sistema devono essere tutte concordi: per i primi 3 assi rotanti direzione positiva significa andare da ascissa positiva a ordinata positiva; l'eventuale quarto asse deve avere la stessa direzione della ascissa. La figura successiva fornisce un esempio di dette direzioni.

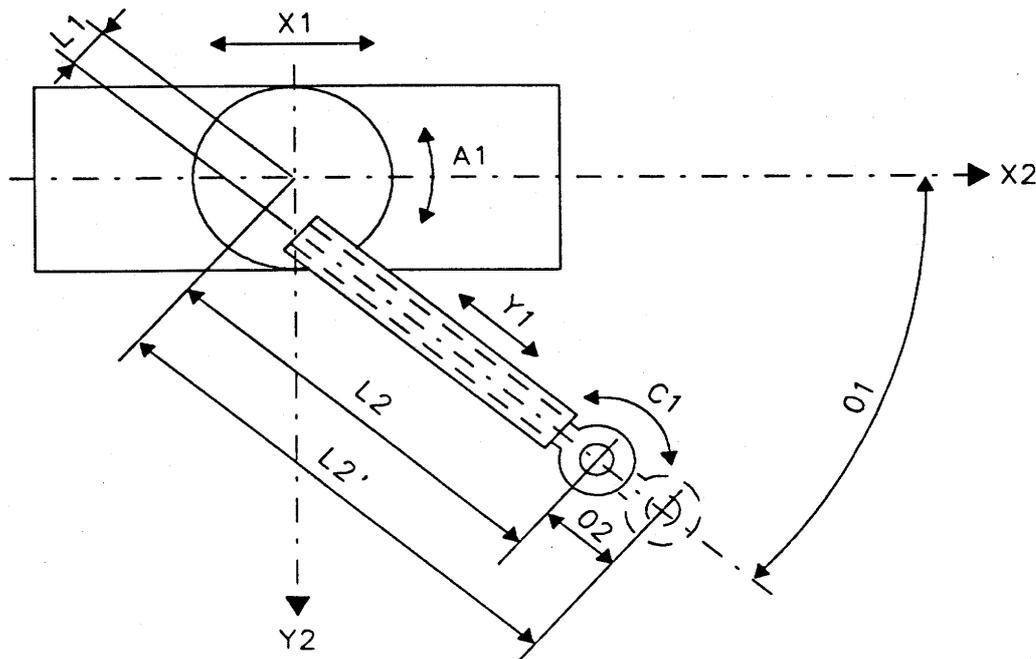


6 - In base alle considerazioni viste nei punti 1-5 precedenti, si avrà che dopo un movimento a zero di tutti e 4 gli assi reali SCARA, l'asse virtuale ascissa si troverà ad una coordinata equivalente alla somma delle lunghezze del primo e secondo asse; l'asse virtuale ordinata a coordinata zero.

b) Con un sistema SCARA, non è possibile richiedere tramite una stessa istruzione di movimento, assi virtuali unitamente ai primi due assi reali del sistema (uno dei due o entrambi).

E' viceversa possibile richiedere assi virtuali unitamente al 3° e/o 4° asse reale del sistema.

2.3.3.2 Sistema CILINDRICO



Numero di assi del sistema CILINDRICO: indica il numero di assi da prendere in considerazione; $1^{\circ}+2^{\circ}$, $1^{\circ}+2^{\circ}+3^{\circ}$, $1^{\circ}+2^{\circ}+3^{\circ}+4^{\circ}$. Si consiglia di indicare sempre 4 e se qualche asse non è presente (2° , 3° , 4°) non indicarlo nelle righe apposite. Nell'esempio riportato, vengono presi in considerazione tutti e 4 gli assi.

Nome del primo asse: indica il primo asse reale del sistema; nell'esempio A1. Il primo asse è sempre l'asse rotante generale e deve essere sempre presente

Lunghezza del primo asse: indica la distanza (può anche essere 0) fra l'asse di rotazione del primo asse e l'asse pinza; tale distanza va data perpendicolarmente all'asse canotto (se canotto mancante, al braccio porta pinza); nell'esempio L1.

Nome del secondo asse: indica il secondo asse reale del sistema; nell'esempio Y1.

Tale asse è sempre l'asse canotto e potrebbe anche non essere presente; in tale caso deve però essere necessariamente presente il quarto asse

Lunghezza del secondo asse: indica la distanza fra l'asse di rotazione del primo asse e l'asse pinza; tale distanza va data parallelamente all'asse canotto (se canotto mancante, al braccio porta pinza); nell'esempio, L2 se il secondo asse (canotto) non ha offset assoluto; 12', se questo asse ha un suo offset (02).

Nel caso sia presente il terzo asse (nell'esempio C1), l'asse pinza deve essere concentrico con l'asse di rotazione del terzo asse (in questo caso la lunghezza del secondo asse è fissa). Se invece vi è eccentricità, la lunghezza del secondo asse (rispetto all'asse pinza) è variabile, essendo indipendente dalla posizione angolare della pinza. Di conseguenza, poiché si imposta una sola lunghezza, il sistema cartesiano ottenuto sarà valido solo in quelle condizioni.

In base a queste considerazioni, in caso di eccentricità, occorre tener presente quanto segue:

- non è possibile ottenere traiettorie rettilinee e circolari e l'errore di traiettoria sarà tanto maggiore quanto maggiore è l'eccentricità.

Nome del terzo asse: indica l'eventuale asse rotante posto all'estremità del canotto (o braccio porta pinza se canotto mancante); nell'esempio C1.

L'indicazione o meno di questo asse in tabella, influisce solo sulla geometria del sistema agli effetti della compensazione automatica di detto rotante e precisamente:

- Asse non indicato in tabella: in caso di richiesta movimento assi (reali o virtuali) tale asse rotante viene ignorato (naturalmente se non richiesto nel movimento)
- Asse indicato in tabella: in caso di richiesta movimento assi (reali o virtuali) tale asse rotante (naturalmente se non richiesto nel movimento) viene automaticamente compensato; la pinza posta su tale asse rimane quindi sempre parallela a se stessa.
E' certo che se questo asse viene richiesto nel movimento, l'indicazione o meno dell'asse nella tabella non ha più alcun significato essendo il movimento richiesto determinante.

Nome del quarto asse: indica l'eventuale asse traslante (traslazione di tutto il CILINDRICO); nell'esempio X1. Tale asse potrebbe anche non essere presente purché sia presente il secondo asse;

Compensazione meccanica: tale dato va sempre dato con valore 0.

Offset assoluto: questo dato viene utilizzato ogni qualvolta si desidera ottenere uno zero asse pratico diverso da quello teorico.

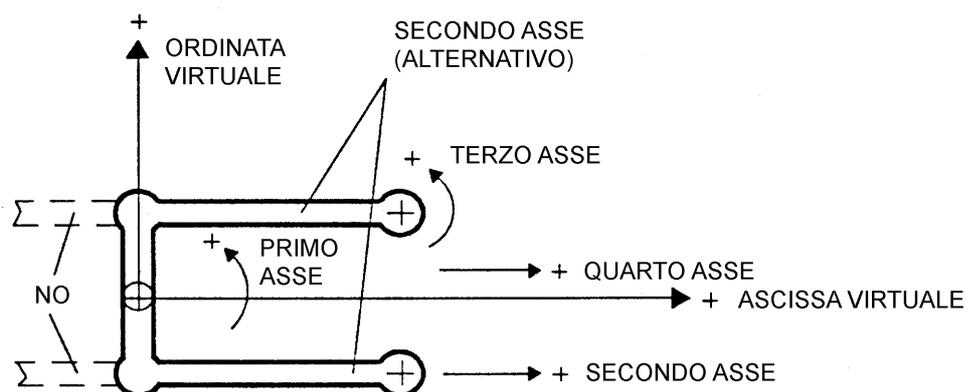
Massima velocità: corrisponde alla massima velocità voluta sull'asse virtuale; se questa velocità fosse impossibile da effettuare, (ogni asse reale appartenente al sistema ha una propria massima velocità) per default viene utilizzata la massima possibile.

Massima accelerazione e decelerazione: corrispondono alla massima accelerazione e decelerazione volute sull'asse virtuale; anche per questi dati se i valori indicati fossero superiori al possibile, per default vengono utilizzati i massimi possibili.

Note:

- a) Poiché tutti i calcoli per gli assi virtuali vengono eseguiti secondo la geometria classica, operando su seni e coseni di angoli espressi in gradi, occorre tenere presente quanto segue:
- 1 - L'unità di misura degli assi rotanti (primo e terzo asse) reali deve essere il grado.
 - 2 - L'unità di misura del quarto asse reale può essere il mm o l'inc. a seconda che la lunghezza secondo asse venga data rispettivamente in mm o inc.
 - 3 - L'offset assoluto del primo asse reale deve essere dato (se è presente il 4 asse traslante) in modo che l'asse virtuale ascissa sia parallelo al quarto asse CILINDRICO.
 - 4 - Il secondo asse (cannotto) può o meno avere offset; tale offset implica chiaramente una diversa lunghezza del secondo asse.
 - 5 - Le direzioni di movimento (positive, negative) degli assi del sistema devono essere tutte concordi: per gli assi rotanti (primo e terzo asse) direzione positiva significa andare da ascissa positiva a ordinata positiva; gli assi lineari (secondo e quarto) devono avere la stessa direzione delle ascisse.
 - 6 - Il secondo asse (cannotto) non può trovarsi in ascissa negativa in caso contrario ruotare il primo asse in modo che si trovi in ascissa positiva.

La figura successiva fornisce un esempio di come devono essere messi gli assi su di un sistema cilindrico



- 7 - In base alle considerazioni viste nei punti 1-6 precedenti, si avrà che dopo un movimento a zero di tutti e 4 gli assi reali CILINDRICI, l'asse virtuale ascissa si

troverà ad una coordinata pari alla lunghezza del secondo braccio e l'asse virtuale ordinata ad una coordinata pari alla lunghezza del primo braccio. (Per quest'ultimo asse la coordinata sarà positiva o negativa)

b) Con un sistema CILINDRICO, non è possibile richiedere tramite una stessa istruzione di movimento, assi virtuali unitamente ai due assi reali (uno dei due o entrambi) costituenti il sistema cilindrico e precisamente: 1° asse del sistema (Rotante generale), 2° asse del sistema (cannotto) o 4° asse del sistema (traslante) nel caso di assenza dell'asse cannotto. E' invece possibile richiedere assi virtuali unitamente al 3° e/o 4° asse (se è presente il 2° asse)

Riepilogo degli assi reali non abbinabili ai virtuali in una stessa istruzione di movimento:

- Sistema cilindrico composto dal 1° e 2° asse (rotante generale + cannotto)
Non sono ammessi virtuali +1° e/o 2° asse reale
- Sistema cilindrico composto dal 1° e 4° asse (rotante generale + traslante)
Non sono ammessi virtuali + 1° e/o 4° asse reale
- Sistema cilindrico completo di 1°, 2°, e 4° asse

Non sono ammessi virtuali + 1° e/o 2° asse (da solo o con il 4°).

```
[X3]
i=VIRTUAL_AXES_NAME( "X3" , "Y3" )
b={
Kind_Axis="ARM"
Kind_Axis_0=0x00010000 // SCARA STANDARD
Kind_Axis_1=0x00000000
Axis_name_ascisse="X3"
Axis_name_ordinate="Y3"
Axis_name_1="A1" // ASSE ANGOLARE BRACCIO
Axis_name_2="B1" // ASSE ANGOLARE AVAMBRACCIO
Axis_name_3="" // ASSE ROTANTE (POLSO)
Axis_name_4="" // ASSE TRASLANTE SUPPLEMENTARE
Lenght_arm_1=200.0
Lenght_arm_2=200.0
Abs_offset_A=100.0
Vel_max_axis_A=20000.0
Acc_max_axis_A=200.0
Dec_max_axis_A=200.0
Abs_offset_O=100.0
Vel_max_axis_O=20000.0
Acc_max_axis_O=1000.0
Dec_max_axis_O=1000.0
e=}
```

```
[X2]
i=VIRTUAL_AXES_NAME("X2" , "Y2" )
b={
Kind_Axis="ARM"
Kind_Axis_0=0x00080000 // CILINDRICO CON CANOTTO
Kind_Axis_1=0x00000000
Axis_name_ascisse="X2"
Axis_name_ordinate="Y2"
Axis_name_1="C1" // ASSE ANGOLARE
Axis_name_2="D1" // ASSE LINEARE (CANOTTO)
Axis_name_3="" // ASSE ROTANTE (POLSO)
Axis_name_4="" // ASSE TRASLANTE SUPPLEMENTARE
Lenght_arm_1=000.0
Lenght_arm_2=200.0
Abs_offset_A=150.0
Vel_max_axis_A=50000.0
Acc_max_axis_A=1000.0
Dec_max_axis_A=1000.0
Abs_offset_O=100.0
Vel_max_axis_O=50000.0
Acc_max_axis_O=1000.0
Dec_max_axis_O=1000.0
e=}
```

2.4 Parametri specifici degli Ambienti

Informazioni sulle condizioni di lancio del part program

[ParametriAUTOMAZIONE]

StringaLancio=

NomePartProgramLancio=via.pp

Note:

Con la caratterizzazione standard qui mostrata, il sistema si avvia automaticamente lanciando il Part program “via.pp” (oppure con altro nome dato dallo sviluppatore del Sistema)

Essendo questo Part Program editabile come un normale Part program, se si vuole che il Sistema si avvii con un percorso diverso e non accessibile dall’ambiente , occorre definire questo percorso particolare nella “StringaLancio=”

Esempio: **StringaLancio=**\programmi\application\partprog\start.pp

2.5 Predefinizioni ISO

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

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

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

esempio 5 processi avremo

Base_xxxx

Testa1_xxxxx

Testa2_xxxxx

Testa3_xxxxx

Testa4_xxxxx

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

Le sessioni sono:

a- Sessione identificativo nome del processo

[AbbinamentoNomeProcesso]
Base_NOME_PROCESSO=[nome alfanumerico]
Testa1_NOME_PROCESSO=[nome alfanumerico]
..
Testa31_NOME_PROCESSO=[nome alfanumerico]

Esempio

[AbbinamentoNomeProcesso]
Base_NOME_PROCESSO=Unita011
Testa1_NOME_PROCESSO= Unita012
Testa2_NOME_PROCESSO= Unita021

b- Sessione identificativo ID del processo

[AbbinamentoNumeroTesta]
Base_NumeroTesta=[numero identificativo ID del processo]
Testa1_NumeroTesta=[numero identificativo ID del processo]
..
Testa31_NumeroTesta=[numero identificativo ID del processo]

Esempio

[AbbinamentoNumeroTesta]
Base_NumeroTesta=11
Testa1_NumeroTesta=12
Testa2_NumeroTesta=21

c- Sessione abbinamento assi macchina ad assi ISO

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

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

[AbbinamentoAssiTeste]

Base_X=[nome asse fisico o virtuale]
Base_Y=[nome asse fisico o virtuale]
Base_Z=[nome asse fisico o virtuale]
Base_A=[nome asse fisico o virtuale]
Base_B=[nome asse fisico o virtuale]
Base_C=[nome asse fisico o virtuale]
Base_U=[nome asse fisico o virtuale]
Base_V=[nome asse fisico o virtuale]
Base_W=[nome asse fisico o virtuale]
Testa1_X=[nome asse fisico o virtuale]
Testa1_Y=[nome asse fisico o virtuale]
Testa1_Z=[nome asse fisico o virtuale]
Testa1_A=[nome asse fisico o virtuale]
Testa1_B=[nome asse fisico o virtuale]
Testa1_C=[nome asse fisico o virtuale]
Testa1_U=[nome asse fisico o virtuale]
Testa1_V=[nome asse fisico o virtuale]
Testa1_W=[nome asse fisico o virtuale]
...
...
Testa31_X=[nome asse fisico o virtuale]
Testa31_Y=[nome asse fisico o virtuale]
Testa31_Z=[nome asse fisico o virtuale]
Testa31_A=[nome asse fisico o virtuale]
Testa31_B=[nome asse fisico o virtuale]
Testa31_C=[nome asse fisico o virtuale]
Testa31_U=[nome asse fisico o virtuale]
Testa31_V=[nome asse fisico o virtuale]
Testa31_W=[nome asse fisico o virtuale]

Esempio

[AbbinamentoAssiTeste]

Base_X=X011

Base_Y=Y011

Base_Z=Z011

Base_A=-999

Base_B=-999

Base_C=C011

Base_U=-999

Base_V=-999

Base_W=-999

Testa1_X=X012

Testa1_Y=Y012

Testa1_Z=Z012

Testa1_A=-999

Testa1_B=-999

Testa1_C=-999

Testa1_U=-999

Testa1_V=-999

Testa1_W=-999

Testa2_X=X021

Testa2_Y=Y021

Testa2_Z=Z021

Testa2_A=-999

Testa2_B=-999

Testa2_C=-999

Testa2_U=-999

Testa2_V=-999

Testa2_W=-999

d- Sessione abbinamento mandrino S ISO a mandrino fisico

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

[AbbinamentoMandrinoTeste]

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

Esempio

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

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

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

[AbbinamentoDirectoryPP]

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

Esempio

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

f- Sessione abbinamento file tabella origini al processo ISO

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

[AbbinamentoORIGINI]

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

Esempio

[AbbinamentoORIGINI]

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

g- Sessione abbinamento file tabella TOOL al processo ISO

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

[AbbinamentoTOOL]

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

Esempio

[AbbinamentoTOOL]

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

h- Sessione abbinamento file tabella parametri P al processo ISO

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

[AbbinamentoPARAMETRI]

Base_FILE_PARAMETRI =[percorso e nome file]

Testa1_FILE_PARAMETRI =[percorso e nome file]

...

Testa31_FILE_PARAMETRI =[percorso e nome file]

Esempio

[AbbinamentoPARAMETRI]

Base_FILE_PARAMETRI=C:\Transfer\Staz01\Unita011_MS\SlotParam.dat

Testa1_FILE_PARAMETRI=C:\Transfer\Staz02\Unita012_MS\SlotParam.dat

Testa2_FILE_PARAMETRI=C:\Transfer\Staz03\Unita021_MS\SlotParam.dat

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

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

[AbbinamentoFUNZIONI_MG]

Base_FILE_FUNZIONI_MG =[percorso e nome file]

Testa1_FILE_FUNZIONI_MG =[percorso e nome file]

...

Testa31_FILE_FUNZIONI_MG =[percorso e nome file]

Esempio

[AbbinamentoFUNZIONI_MG]

Base_FILE_FUNZIONI_MG=C:\Transfer\Staz01\Unita011_MS\SlotGM.dat

Testa1_FILE_FUNZIONI_MG=C:\Transfer\Staz02\Unita012_MS\SlotGM.dat

Testa2_FILE_FUNZIONI_MG=C:\Transfer\Staz03\Unita021_MS\SlotGM.dat

l- Sessione abbinamento file tabella Correttori Utensili al processo ISO

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

[AbbinamentoCORREZ]

Base_FILE CORREZIONI =[percorso e nome file]

Testa1_FILE_CORREZIONI =[percorso e nome file]

...

Testa31_FILE_CORREZIONI =[percorso e nome file]

Esempio

[AbbinamentoCORREZ]

Base_FILE_CORREZIONI= C:\Transfer\Staz01\Unita011_MS\Correz.dat

Testa1_FILE_CORREZIONI= C:\Transfer\Staz01\Unita011_MS\Correz.dat

Testa2_FILE_CORREZIONI= C:\Transfer\Staz01\Unita011_MS\Correz.dat

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

Se la logica di macchina acquisisce il valore di coppia e lo fornisce come valore percentuale in una variabile globale G, quest'ultima può venire indicata al processo ISO abbinato.

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

[AbbinamentoCoppia]

Base_Globale_Coppia =[numero Globale]

Testa1_Globale_Coppia =[numero Globale]

...

Testa31_Globale_Coppia =[numero Globale]

Esempio

[AbbinamentoCoppia]

Base_Globale_Coppia=11011

Testa1_Globale_Coppia=11012

Testa2_Globale_Coppia=-999

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

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

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

[AbbinamentoG_Disabilitazione]

Base_Globale_Disabilitazione =[numero Globale]

Testa1_Globale_Disabilitazione =[numero Globale]

...

Testa31_Globale_Disabilitazione =[numero Globale]

Esempio

[AbbinamentoG_Disabilitazione]

Base_Globale_Disabilitazione=-999

Testa1_Globale_Disabilitazione=-999

Testa2_Globale_Disabilitazione=-999

o- Sessione abbinamento default sistema misura in mm o pollici

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

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

[AbbinamentoInch]

Base_INCH=[flag 1 per sistema in pollici]

Testa1_INCH=[flag 1 per sistema in pollici]

..

Testa31_INCH=[flag 1 per sistema in pollici]

Esempio

[AbbinamentoInch]

Base_INCH=-999

Testa1_INCH=-999

Testa2_INCH=-999

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

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

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

[AbbinaIncrementaleG0302]

Base_INCREMENTALEG0302=[flag 1 per sistema in incrementale]

Testa1_INCREMENTALEG0302=[flag 1 per sistema in incrementale]

..

Testa31_INCREMENTALEG0302=[flag 1 per sistema in incrementale]

Esempio

[AbbinaIncrementaleG0302]

Base_INCREMENTALEG0302=-999

Testa1_INCREMENTALEG0302=-999

Testa2_INCREMENTALEG0302=-999

q- Sessione abbinamento modalità di esecuzione G01

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

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

[AbbinaNonContinuo]

Base_NONCONTINUO =[flag 1 per modalità con arresto assi]

Testa1_NONCONTINUO =[flag 1 per modalità con arresto assi]

..

Testa31_NONCONTINUO =[flag 1 per modalità con arresto assi]

Esempio

[AbbinaNonContinuo]

Base_NONCONTINUO=-999

Testa1_NONCONTINUO=-999

Testa2_NONCONTINUO=-999

r- Sessione abbinamento output digitale per M08 refrigerante primario

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

[AbbinamentoRefrigerantePrincipale]

Base_RefrigerantePrimario =[nome output digitale]

Testa1_RefrigerantePrimario =[nome output digitale]

..

Testa31_RefrigerantePrimario =[nome output digitale]

Esempio

[AbbinamentoRefrigerantePrincipale]

Base_RefrigerantePrimario=EV_RP011

Testa1_RefrigerantePrimario EV_RP012

Testa2_RefrigerantePrimario EV_RP021

s- Sessione abbinamento output digitale per M07 refrigerante secondario

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

[AbbinamentoRefrigeranteSecondario]

Base_RefrigeranteSecondario =[nome output digitale]

Testa1_RefrigeranteSecondario =[nome output digitale]

..

Testa31_RefrigeranteSecondario =[nome output digitale]

Esempio

[AbbinamentoRefrigeranteSecondario]

Base_RefrigeranteSecondario=EV_RS011

Testa1_RefrigeranteSecondario=EV_RS012

Testa2_RefrigeranteSecondario=EV_RS021

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

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

Ed in particolare:

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

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

Vengono scritti dal processo ISO

[AbbinamentoOutputDigitaleControllo]

Base_OutputDigitaleControllo=[nome input od output digitale bit 0]
Testa1_OutputDigitaleControllo=[nome input od output digitale bit 0]
..
Testa31_OutputDigitaleControllo=[nome input od output digitale bit 0]

Vengono letti dal processo ISO

[AbbinamentoInputDigitaleControllo]

Base_InputDigitaleControllo=[nome input od output digitale bit 0]
Testa1_InputDigitaleControllo=[nome input od output digitale bit 0]
..
Testa31_InputDigitaleControllo=[nome input od output digitale bit 0]

Numero di bit 16 o 32

[AbbinamentoNumeroBitControllo]

Base_NumeroBitControllo=[NumeroBit 16 0 32]
Testa1_NumeroBitControllo=[NumeroBit 16 0 32]
..
Testa31_NumeroBitControllo=[NumeroBit 16 0 32]

Funzionalità future definite automaticamente, ma non interpretate dal sistema.

[AbbinamentoOutputDigitaleCicloFisso]

Base_OutputDigitaleCicloFisso=[nome input od output digitale]

Testa1_OutputDigitaleCicloFisso=[nome input od output digitale]

..

Testa31_OutputDigitaleCicloFisso=[nome input od output digitale]

[AbbinamentoInputDigitaleCicloFisso]

Base_InputDigitaleCicloFisso=[nome input od output digitale]

Testa1_InputDigitaleCicloFisso=[nome input od output digitale]

..

Testa31_InputDigitaleCicloFisso=[nome input od output digitale]

[AbbinamentoOutputAnalogicoCicloFisso]

Base_OutputAnalogicoCicloFisso=[nome output analogico]

Testa1_InputDigitaleCicloFisso=[nome input od output digitale]

..

Testa31_InputDigitaleCicloFisso=[nome input od output digitale]

[AbbinamentoInputAnalogicoCicloFisso]

Base_InputAnalogicoCicloFisso=[nome input analogico]

Testa1_InputAnalogicoCicloFisso=[nome input analogico]

..

Testa31_InputAnalogicoCicloFisso=[nome input analogico]

[AbbinamentoChiaveInputControllo]

Base_ChiaveInputControllo=[nome input od output digitale]

Testa1_ChiaveInputControllo=[nome input od output digitale]

..

Testa31_ChiaveInputControllo=[nome input od output digitale]

Parametri globali a tutti i processi ISO

[ParametriCNC]

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

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

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

X_offsetTavola=offset asse X TAVOLA , verrà sostituito dalla prima O trovata nel part program

Y_offsetTavola= offset asse Y TAVOLA , verrà sostituito dalla prima O trovata nel part program

Z_offsetTavola= offset asse Z TAVOLA , verrà sostituito dalla prima O trovata nel part program

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

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

BLOCCA_STATI -1 se letto dal sistema a -1 blocca gli stati macchina

NO_STATI 0

AUTOMATICO 1

MANUALE 3

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

BLOCCA_COMANDI -1 se letto dal sistema a -1 blocca i comandi macchina

NO_CICLO 0

VIA_CICLO 3

BREAK 10

HOLD 11

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

BLOCCA_SERVIZI -1 se letto dal sistema a -1 blocca l'attivazione dei servizi

NO_SERVIZI_ATTIVI 0

MDI 1

COMM 2

FUNZIONI_MANUALI 3

IMPOSTA_PEZZO 4

MODALITA_CICLO 5

GlobaleProtezioni=NON utilizzata dal sistema, futuri sviluppi

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

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

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

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

NomeIOFeedHold= Nome IO

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

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

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

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

NomeIOHold=Nome IO

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

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

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

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

Esempio

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

CAPITOLO 3

3 Caratterizzazione

Per poter gestire le risorse di sistema nella loro globalità con i cicli di lavorazione è necessario configurare i seguenti parametri.

. 1-PID	Proporzionale Integrativo Derivativo
. 2-Anticollisione	Anticollisione tra assi vincolati
. 3-Gantry	Assi cooperanti
. 4-Homing	Modalità azzeramento asse
. 5-Dati campionati	Buffer acquisizione dati campionati
. 6-RAV e CAP	Read Assi Value
. 7- CAP	Carica Assi Parametro.
. 8- Assi Rollover	Assi ciclici
. 9- Over travel	Otrecorsa
.10- Assi asserviti in mov.	Assi asserviti solo in movimento

3.1 PID

Proporzionale Integrativo Derivativo

Sono previsti due modi per asservire gli assi; in VELOCITA' e in CORRENTE.

L'asservimento in VELOCITA' viene definito mettendo a zero il bit 0x00000001 di "Kind_axis_0" degli assi in caratterizzazione.

Bisogna rispettare il rapporto tra "Vel_max" e "Volt_vel" perché è in questo modo viene definito alla movimentazione quanti volt abbisognano all'asservimento per la velocità voluta.

Ogni variazione di "Vel_max" deve essere corrisposta da una proporzionale di "Volt_vel".

Il valore di "Dec_max" e "Acc_max" possono essere cambiati senza nessun particolare problema (a parte quelli fisici della macchina).

Siccome "Vel_max" dipende anche da "Step", cambiandolo implicitamente modifica il rapporto tra "Volt_vel" e "Vel_max"; perciò bisogna sempre stabilire in anticipo il valore di "Step".

Il valore di "Volt_vel" dovrebbe essere noto dalle caratteristiche del motore e dal rapporto di trasmissione.

Con assi rotanti può essere dedotta da part-program, che impostando una tensione a loop aperto, ne controlli la velocità effettiva.

Un altro modo per verificare che il valore sia corretto è quello di lanciare un movimento abbastanza lungo da raggiungere la velocità programmata, e controllare l'errore di inseguimento che deve essere:

$$\text{epsi} = \text{velocità al secondo} / K_p.$$

Naturalmente gli altri K del pid devono essere a zero.

L'asservimento in CORRENTE viene definito mettendo a uno il bit 0x00000001 di "Kind_axis_0" degli assi in caratterizzazione.

Bisogna rispettare il rapporto tra "Acc_max" e "Volt_vel" perché è in questo modo viene definito alla movimentazione quanti volt abbisognano all'asservimento per avere l'accelerazione voluta.

Ogni variazione di "Acc_max" deve essere corrisposta da una proporzionale di "Volt_vel".

Il valore di "Dec_max" e "Vel_max" possono essere cambiati senza nessun particolare problema (a parte quelli fisici della macchina).

Siccome "Acc_max" dipende anche da "Step", cambiandolo implicitamente modifica il rapporto tra "Volt_vel" e "Acc_max"; perciò bisogna sempre stabilire in anticipo il valore di "Step".

Il valore di "Volt_vel" dovrebbe essere noto dalle caratteristiche el motore e dall'asservimento.

E' meno agevole rilevare il valore dic "Volt_vel" nei sistema in corrente rispetto a quello in velocita', perche' l'assegnazione i una tensione a loop aperto origenerebbe non una velocita' all'asse, ma una accelerazione che porterebbe l'asse in fuga.

Per motori lineari si puo tentare con l'assegnare una bassa tensione al motore a loop aperto e misurarne la forza con un dinamometro, siccome $Forza = massa * accelerazione$, conoscendo la massa si puo' calcolare l'accelerazione ($accelerazione = Forza / massa$).

Altro metodo e stimarne un valore, tarare alla meglio l'asse è vedere quanti volt si ottengono con una certa accelerazione.

Per vedere quanti volt si hanno si puo' guardare il movimento con la funzione oscilloscopio.

Viene di seguito riportato un esempio di partprogram per rilevare la velocità, con un gradino di tensione:

ESEMPIO

```
;          ASSEGNA UNA TENSIONE ALL'ASSE Z
;          E NE VISUALIZZA LA VELOCITA'
;
;          -TSK/VEL_Z
;          -LET/L1,0
AA         -KYB/DAMMI VOLT Z,L1
;          -CPL/-1:Z
;          -CAP/17:Z,L1
;          -JMP/AA
;          -RET

PartProgram[VEL_Z]
AA         -RAV/4:L1,Z
;          -DIS/1:VELOCITA' Z,L1
;          -JMP/AA
;          -RET
```

Viene di seguito riportato un esempio di partprogram per rilevare la velocità, con un gradino di tensione: e memorizzazione dei dati campionati su file

ESEMPIO

```

; ***** RISPOSTA ASSE XL AD ONDA QUADRA CORTA *****
; ESEGUE UN ONDA QUADRA DI V VOLT DALLA PARTENZA
; PER UN TEMPO, INVERTE LA TENSIONE PER LO STESSO
; TEMPO E LA AZZERA

        -DIS/1:Inizio
        -DIS/2:>
        -AZZ/\OQXL.TXT
        -SCR/\OQXL.TXT,N   Pos_XL   Volt
        -SET/L10,0,L9,0

        -SET/L20,0,L21,100,L22,3
;      L20 = POS. PARTENZA
;      L21 = TEMPO in millisecondi per ONDA QUADRA
;      L22 = VOLT PER DIREZIONE POSITIVA

        -MOV/5000:XL,L20      ;Pos. partenza
        -CAP/28:XL,15        ;Volt Massimi
        -TIM/100
        -RAV/17:L24,XL       ;L24 = V offset
        -RAV/22:L25,XL       ;L25 = V offset old
        -SET/L24,L24+L25
        -DIS/1:VOLT OFFSET ,L24
        -CAP/22:XL,L24       ;Volt Offset
        -TIM/50
        -CPL/-1:XL
        -CAP/51:XL,0         ;buffer off
        -CAP/50:XL,1,XL,17   ;buffer on

        -SET/L5,L22          ;L5 = Volt per andata onda quadra
        -SET/L6,-L5         ;L6 = Volt per ritorno onda quadra
        -CAP/17:XL,L5
        -TMM/L21
        -CAP/17:XL,L6
        -TMM/L21
        -CAP/17:XL,0
        -TIM/50
        -CAP/51:XL,0        ;buffer off
        -CPL/1:XL
        -MOV/5000:XL,L20     ;Ritorno pos. partenza

LOOPS  -GAV/L11,L12,L1,L2
        -JLE/L11,0,FINE
        -JEQ/L12,0,DOPOS
        -SCR/\OQXL.TXT,Sbordamento,L12,L12
        -DIS/2:Overflow,L12
        -JMP/FINE
DOPOS  -ADD/L10,L11
        -ADD/L9,1
        -SCR/\OQXL.TXT,L1,L2,L9
        -JMP/LOOPS
FINE   -DIS/1:Finito,L10
    
```

3.2 ANTICOLLISIONE

Anticollisione tra assi vincolati

Gli assi vengono così raggruppati in modo che possono tra loro lavorare separatamente, gli assi possono incidere però su unico asse fisico, esempio portale con più bracci che lavorano sullo stesso asse portante, in questo caso il sistema è in grado di risolvere il problema dell'anticollisione, gestendo l'arresto momentaneo di un asse in attesa che l'altro si porti in una posizione di fuori ingombro.

In questo caso la programmazione avviene con ciclo separato per ogni braccio, altrimenti sarebbe oneroso pensare alla movimentazione di tutto il sistema, sarà il sistema "AxesBrain" che con la funzionalità di anticollisione controllerà la gestione degli ingombri, unica attenzione è che la programmazione dei cicli non porti dei casi di abbraccio mortale "DEAD LOCK".

PARAMETRI DI ANTICOLLISIONE

Bisogna caricare sull'asse i seguenti campi:

Asse caratterizzato X1

```
Axis_1_name_anticollision="X1"      // A1
Distance_1_anticollision=-30.0     // B1
Kind_axis_1_anticollision=10       // C1
Axis_2_name_anticollision="X2"     // A2
Distance_2_anticollision=-30.0     // B2
Kind_axis_2_anticollision=9        // C2
```

Dove

A1 = nome di un asse in anticollisione con l'asse caratterizzato ("X1")
B1 = valore di anticollisione con questo asse (vedi nota **)
C1 = tipo di anticollisione con questo asse (vedi nota ***)
A2 = nome dell'altro asse in anticollisione con l'asse caratterizzato ("X1")
B2 = valore di anticollisione con quest'altro asse (vedi nota **)
C2 = tipo di anticollisione con quest'altro asse (vedi nota ***)

** Il valore di anticollisione e':
Distanza tra gli zeri degli assi - Ingombro tra gli assi
ES. nel caso di zeri coincidenti, il valore da impostare e pari
al valore negativo dell'ingombro tra gli assi

*** Il tipo di anticollisione e':

Somma tra i seguenti valori

- 1) Codice di approccio che ha i seguenti valori
 - 1 = "X1" concorde con "XT"; "X1" si avvicina a "XT" in positivo
 - 2 = "X1" concorde con "XT"; "X1" si avvicina a "XT" in negativo
 - 3 = "X1" discorde con "XT"; "X1" si avvicina a "XT" in positivo
 - 4 = "X1" discorde con "XT"; "X1" si avvicina a "XT" in negativo
- 2) Accostamento tra gli assi
 - 8 = "X1" si accosta a "XT" sino alla posizione di collisione
 - 0 = "X1" non si accosta a "XT" ma aspetta di avere campo libero
- 3) Inseguimento tra gli assi
 - 16 = "X1" insegue "XT" senza collidere
 - 0 = "X1" non insegue "XT" ma aspetta di avere campo libero
- 4) Calcolo sulla posizione reale o calcolata
 - 32 = il calcolo di anticollisione tra "X1" e "XT" e' fatto sulle posizioni teoriche (SIMULAZIONE)
 - 0 = il calcolo di anticollisione tra "X1" e "XT" e' fatto sulle posizioni reali (LAVORAZIONE)
- 5) Anticollisione avviata da OMO su MARKER o OMO generale
 - 64 = il calcolo di anticollisione tra "X1" e "XT" e' fatto anche se l'asse non e' stato azzerato su MARKER
 - 0 = il calcolo di anticollisione tra "X1" e "XT" e' fatto solamente se l'azzeramento e stato fatto su MARKER

ES. Nel caso si voglia controllare l'anticollisione con assi concordi con avvicinamento negativo (2) , facendo accostare gli assi (8) , ma non facendoli inseguire (0) e su quote reali (0); bisogna impostare $2+8 = 10$

I parametri vanno messi su tutti gli assi interessati all'anticollisione

"XT"

```
Axis_1_name_anticollision=""  
Distance_1_anticollision=0.0  
Kind_axis_1_anticollision=0  
Axis_2_name_anticollision="X1"  
Distance_2_anticollision=-30.0  
Kind_axis_2_anticollision=41 // 32+8+1
```

"X1"

```
Axis_1_name_anticollision="XT"  
Distance_1_anticollision=-30.0
```

```
Kind_axis_1_anticollision=42 // 32+8+2  
Axis_2_name_anticollision="X2"  
Distance_2_anticollision=-30.0  
Kind_axis_2_anticollision=41 // 32+8+1
```

"X2"

```
Axis_1_name_anticollision="X1"  
Distance_1_anticollision=-30.0  
Kind_axis_1_anticollision=42 // 32+8+2  
Axis_2_name_anticollision="X3"  
Distance_2_anticollision=-30.0  
Kind_axis_2_anticollision=41 // 32+8+1
```

3.3 GANTRY

Assi cooperanti

L'asse a portale (Gantry) è una struttura meccanicamente rigida (normalmente è una struttura a portale) e corrisponde pertanto ad un asse unico, ma è trattata dal controllo come se fosse costituita da una coppia di assi (asse master e asse slave, ciascuno con i propri sistemi di conteggio ed il proprio azionamento).

Una delle funzioni del controllo è quella di mantenere la posizione dell'asse "slave" più prossima possibile a quella dell'asse "master".

I movimenti richiesti sugli assi Master-Slave, vengono eseguiti nel modo seguente:

La richiesta di movimento può essere effettuata tramite tutte le istruzioni di movimento.

Nel caso di movimento relativo al solo asse Master, l'asse Slave lo segue o rimane fermo rispettivamente in caso di associazione o disassociazione.

Durante il movimento, l'asse Slave segue il suo Master in tempo reale

Sono gestiti due tipi di GANTRY:

Quello con struttura rigida che non è ingallonabile;
vengono adattati i motori ad avere il medesimo carico.

Quello classico che deve essere guidato dai due motori,
rispettando il più possibile per ogni asse la posizione.

Col tipo rigido l'asse master è comandato alla quota voluta,
l'altro asse si adatta.

Il tipo rigido viene caratterizzato mettendo sull'asse master in

Kind_axis_1=il bit 800000H (SLAVE_CHA_GANTRY)
Axis_name_slave="nome asse slave"
K_follower_slave=100.0 (100 significa gestione mediata , 0 non mediata)

Il tipo classico ha le stesse informazioni, con l'aggiunta in
Kind_axis_1=il bit 1000000H (SLAVE_CHA_HARD)
Con questo tipo bisogna anche assegnare la massima tensione di
scompenso tra i due assi che e' il campo di caratterizzazione
VoltGantry=5.0
Se il valore di VoltGantry e' ≤ 0 lo scompenso tra i due assi
non e considerato:

Inoltre bisogna definire lo scostamento dello zero marker slave
rispetto allo zero marker del master con il seguente campo
di caratterizzazione
DeltaOffset_SlaveGantry=0.640
Questi due campi sono messi sull'asse master

Nota per "DeltaOffset_SlaveGantry"
asse master su zero marker con lettura della quota reale = 0
asse slave con lettura della quota reale = +2.403
Il valore da caratterizzare deve essere -2.403 che e' la
posizione dello zero marker asse slave rispetto al corrispettivo
del master.

Mettendo anche il
Kind_axis_1=il bit 2000000H (SLAVE_CHA_THEOR)
il gantry assegna tutto l'errore di incrocio sull'asse slave; in
tal modo il master segue integralmente il teorico e l'asse slave
accelera o decelera per annullare l'errore tra gli assi.
Usando questo bit, anche per gantry rigido, l'asse slave segue
il teorico e non si adatta.

L'azzeramento degli assi GANTRY e' bene farlo con un part-program, che e' diverso a secondo dei tipi fisici di gantry.

Asse gantry non rigido con assi che non si liberano

Caratterizzazione asse master gantry YM:

Kind_axis_1=0x01800000H (SLAVE_CHA_HARD + SLAVE_CHA_GANTRY)

Axis_name_slave="YS"

K_follower_slave=0.0

VoltGantry=1.0

DeltaOffset_SlaveGantry=-0.640

Come calcolare "DeltaOffset_SlaveGantry="

Lanciare il programma seguente, attendere che l'azzeramento sia fatto e alla richiesta di movimento incrementale di asse master dare i valori che mettono gli assi gantry in posizione di lavoro. Leggere sulla riga 2 il valore da introdurre in caratterizzazione.

```
-TSK/READ_GANTRY ;Lancio del task di lettura assi gantry
-CAP/100:YM,1024 ;Disabilitazione di SLAVE_CHA_HARD
-HOM/YM ;Ricerca micro e marker asse master
-MOV/YM,100 ;Posizionamento asse per azzeramento asse slave
-CAP/0:YM,262144 ;0x40000 revers master slave del gantry
-HOM/YS ;Ricerca micro e marker asse slave
-CAP/0:YM,-262144 ;-(0x40000) ritorno al master slave del gantry
-CAP/100:YM,2048 ;Disabilitazione temporanea del gantry
REP -KYB/MOV INCREM. YM,L1
-INQ/YM
-MOV/500:YM,L1
-ABQ/YM
-JMP/REP
-END
```

PartProgram[READ_GANTRY]

```
LOOP -RAV/42:L1,YM,L2,YS
-DIS/1:1:YM = ,L1
-DIS/1:31:YS = ,L2
-DIS/2:(YM - YS) = ,L1-L2
-TMM/100
-JMP/LOOP
-RET
```

Come eseguire l'azzeramento del sistema di assi gantry
Lanciare il programma seguente

```
-HOM/YM           ;Ricerca micro e marker asse master  
-MOV/YM,100       ;Posizionamento asse per azzeramento asse slave  
-CAP/0:YM,262144  ;0x40000 revers master slave del gantry  
-HOM/YS           ;Ricerca micro e marker asse slave  
-CAP/0:YM,-262144 ;-(0x40000) ritorno al master slave del gantry  
-MOV/YM,200       ;Posizionamento asse per fine azzeramento  
-END
```

Come togliere gli assi gantry dall'ingallonamento
Lanciare il programma seguente e muovere il master
sino ad avere gli assi movibili

```
-CAP/100:YM,4096   ;L'asse YM muove anche senza HOME  
-CAP/100:YS,4096;L'asse YS muove anche senza HOME  
-CAP/100:YM,2048   ;Disabilitazione temporanea del gantry  
REP -KYB/MOV INCREM. YM,L1  
-INQ/YM  
-MOV/500:YM,L1  
-ABQ/YM  
-JMP/REP  
-END
```

Mettendo anche il
Kind_axis_0=il bit 0x00000020 (ASSE CON PARTENZA DOLCE DEL SERVO)
con il GANTRY di tipo non rigido, asservendo i due assi si portano
in posizione piu' dolcemente e il controllo di collision non viene
fatto per tutto il tempo di "Cycles_positioning".
Durante il movimento, se viene superato uno spostamento tra
i due assi del valore di servo error viene segnalato errore.

3.4 Homing

Modalità azzeramento asse

i seguenti bits di "Kind_axis_0" degli assi in caratterizzazione sono riferiti alla funzione homing.

```

;0x00000800 = ESECUZIONE HOMING A BATTUTA "
;0x00001000 = ESECUZIONE HOMING SOLO SUL MARKER
;0x00002000 = ESECUZIONE HOMING SOLO SUL MICRO
;0x00004000 = ASSE CHE SI MUOVE SOLO DOPO HOMING
    
```

Il bit 0x00004000 non permette di muovere l'asse senza azzeramento fatto, a meno che esso non venga mosso in manuale.

Con gli altri bits a zero l'esecuzione della homing e quella standard; ricerca micro di homing, rilascio dello stesso e ricerca del marker.

Con il bit 0x00002000 a uno viene ricercato il micro di homing, rilasciato e azzerato l'asse (non viene ricercato il marker).

Con il bit 0x00001000 a uno viene solo ricercato il marker (non viene ricercato il micro di homing).

Con il bit 0x00000800 a uno viene eseguita la homing a battuta, ossia quando l'asse e' fermo e viene superata la soglia di battuta l'asse viene azzerato. In questo caso la soglia di battuta e' caratterizzata in "Volt_PHASE" e il tempo di permanenza a battuta a "Delay_PHASE". Ricordarsi di togliere l'asse dalla posizione di battuta, perche' la tensione di asservimento rimarrebbe alta.

Altri parametri per la homing sono caratterizzati nei seguenti campi:

Number_OMO_switch_port="Homing_ZA"

Nome dell'input caratterizzato per la funzione homing

FalseTrue_logic_OMO_switch=1

stato dell'input di homing con micro sentito

Space_OMO=-1000.0

spazio massimo e verso di ricerca del micro di omo

Notare che il rilascio micro e la ricerca del marker sono fatte in direzione opposta a quella del micro, quindi se viene fatta un azzeramento solo su marker questo campo deve avere il segno opposto alla direzione voluta (come se facesse una ricerca virtuale del micro).

Vel_OMO=5000.00

velocita' in mm/minuto di ricerca del micro di omo

Vel_MARKER=500.00

velocita' in mm/minuto di rilascio del micro e di ricerca del marker

I bits di caratterizzazione possono essere forzati da part program con l'istruzione -HOM/Type:Asse dove Type ha i seguenti significati:

- 0 = omo secondo caratterizzazione (default)
- 1 = omo senza micro solo con marker
- 2 = omo senza marker solo con micro
- 3 = omo con azzeramento asse incondizionato senza movimento
- 4 = omo a "BATTUTA"
- 8 = test omo su marker per verifica perdita steps encoder
- 16 = spazio omo caratterizzato in direzione positiva per la ricerca micro
- 32 = spazio omo caratterizzato con direzione invertita rispetto alla caratterizzazione per la ricerca micro
- 48 = spazio omo caratterizzato in direzione negativa per la ricerca micro

Quindi -HOM/17:A1 significa esecuzione homing su asse A1 con ricerca solo su marker in direzione negativa. Notare che la direzione e' negativa perche' non viene ricercato il micro, che e' virtualmente ricercato in direzione positiva.

-HOM/18:A1 significa esecuzione homing su asse A1 con ricerca solo su micro in direzione positiva.

Esempio di azzeramento:

```

;
;      1 = mask asse ZL
;      2 = mask asse ZR
;      4 = mask asse XL
;      8 = mask asse XR
;     16 = mask asse Y
;     32 = mask asse SL
;     64 = mask asse SR
;    128 = mask asse LU
;      -DIS/0:>
;      -DIS/1:>
;      -DIS/2:>
;      -DIS/3:>
;      -DIS/4:>
;      -DIS/5:>
;      -DIS/6:>
;      -DIS/7:>
;      -DIS/8:>
;      -EVC/1,2,3,4,5,6,7,8,9
;      -LET/L10,1+2+4+8+16+32+64+128-1
;      -LET/L11,AND(L10,1)
;      -JNE/L11,1,NO_ZL
;      0 = Posizione asse ZL con origine fatta
;      -TSK/OMOZL:0
;      -JMP/DO_ZL
NO_ZL      -EVS/1
DO_ZL      -LET/L11,AND(L10,2)
;      -JNE/L11,2,NO_ZR
;      0 = Posizione asse ZR con origine fatta
;      -TSK/OMOZR:0
;      -JMP/DO_ZR
NO_ZR      -EVS/2
DO_ZR      -EVW/1,2
;      -LET/L11,AND(L10,12)
;      -JNE/L11,12,NO_XLXR
;
;      500 = Posizione asse XR con origine fatta
;     -550 = Posizione asse XR per ricerca MARKER
;     -500 = Posizione asse XL con origine fatta
;      600 = Posizione asse XL per ricerca MARKER
;      -TSK/OMOX:500,-550,-500,600
;      -JMP/DO_XLXR
NO_XLXR    -EVS/9
DO_XLXR    -LET/L11,AND(L10,16)
;      -JNE/L11,16,NO_Y
;      0 = Posizione asse Y con origine fatta
;      250 = Posizione asse Y per ricerca MARKER
;      -TSK/OMOY:0,250
;      -JMP/DO_Y
NO_Y       -EVS/5
DO_Y       -LET/L11,AND(L10,32)
;      -JNE/L11,32,NO_SL
;      0 = Posizione asse SL con origine fatta
;      -TSK/OMOSL:0
;      -JMP/DO_SL
NO_SL      -EVS/6
DO_SL      -LET/L11,AND(L10,64)
;      -JNE/L11,64,NO_SR
;      0 = Posizione asse SR con origine fatta

```

```

- TSK/OMOSR:0
- JMP/DO_SR
NO_SR -EVS/7
DO_SR -LET/L11,AND(L10,128)
      -JNE/L11,128,NO_LU
;      0 = Posizione asse LU con origine fatta
      -TSK/OMOLU:0
      -JMP/DO_LU
NO_LU -EVS/8
DO_LU -EVW/5,6,7,8,9
      -DIS/0: Origine FATTA SU TUTTI GLI ASSI DESIDERATI
      -RET

```

```

PartProgram[OMOZL]
;      Origine ASSE ZL
;
;      L1 = Posizione asse ZL con origine fatta
;
;      -HOM/ZL
;      -MOV/ZL,L1
;      -EVS/1
;      -DIS/1: Origine ZL fatta
;      -RET

```

```

PartProgram[OMOZR]
;      Origine ASSE ZR
;
;      L1 = Posizione asse ZR con origine fatta
;
;      -HOM/ZR
;      -MOV/ZR,L1
;      -EVS/2
;      -DIS/2: Origine ZR fatta
;      -RET

```

```

PartProgram[OMOX]
;      Origine VELOCE ASSI X
;      L1 = Posizione asse XR con Origine fatta
;      L2 = Posizione per inizio Origine asse XR
;           (rispetto al micro di Origine)
;      L3 = Posizione asse XL con Origine fatta
;      L4 = Posizione per inizio Origine asse XL
;           (rispetto al micro di Origine)
;      -EVC/1,2,3,4
;      -TSK/OMOXR:L1,L2 ; -550 = Pos. inizio ricerca marker XR
;      -TSK/OMOXL:L3,L4; ; 600 = Pos. inizio ricerca marker XL
;      -EVW/1,2
;      -EVS/9
;      -DIS/3: Origine FATTA SU ASSI XR e XR
;      -RET

```

```

PartProgram[OMOXL]
;      Origine VELOCE ASSE XL
;
;      L1 = Posizione asse con Origine fatta
;      L2 = Posizione per inizio Origine asse XL
;           (rispetto al micro di Origine)
;
;      -SFP/5000,1000,1000

```

```

-HOM/3:XL
-DIS/2:>
-TMS/XL,-2000:5:160,0,NTCH    ;-2000 = posiz. finale
-DIS/2:NON TOCCATO MICRO XL
-JMP/FINI
NTCH  -EVS/4
      -EVW/3
      -SFP/7000,1000,1000
      -RAV/1:L10,XL
      -MOV/XL,L2+L10
      -SFP/5000,1000,1000
      -HOM/17:XL                ; Origine su Marker dir. positiva
      -MOV/XL,L1                ;-500
      -DIS/4: Origine XL fatta
      -EVS/1
FINI  -
      -RET

```

```

PartProgram[OMOXR]
;      Origine VELOCE ASSE XR
;
;      L1 = Posizione asse con Origine fatta
;      L2 = Posizione per inizio Origine asse XR
;           (rispetto al micro di Origine)
;
;      -SFP/5000,1000,1000
;      -HOM/3:XR
;      -DIS/3:>
;      -TMS/XR,2000:5:159,0,NTCH    ;2000 = posiz. finale
;      -DIS/3:NON TOCCATO MICRO XR
;      -JMP/FINI
NTCH  -EVW/4
      -SFP/7000,1000,1000
      -RAV/1:L10,XR
      -MOV/XR,L2+L10
      -SFP/5000,1000,1000
      -HOM/17:XR                ; Origine su Marker dir. positiva
      -MOV/XR,L1                ;500
      -EVS/3
      -DIS/3: Origine XR fatta
      -EVS/2
FINI  -
      -RET

```

```

PartProgram[OMOY]
;      Origine VELOCE ASSE Y
;
;      L1 = Posizione asse con Origine fatta
;      L2 = Posizione per inizio Origine asse Y
;           (rispetto al micro di Origine)
;
;      -SFP/5000,1000,1000
;      -HOM/3:Y
;      -DIS/1:>
;      -TMS/Y,-2000:5:158,0,NTCH    ;-2000 = posiz. finale
;      -DIS/1:NON TOCCATO MICRO
;      -JMP/FINI
NTCH  -SFP/7000,1000,1000
      -RAV/1:L10,Y

```

```

- MOV/Y, L2+L10
- SFP/5000,1000,1000
- HOM/49:Y ; Origine su Marker dir. negativa
- MOV/Y, L1
- EVS/5
- DIS/5: Origine Y fatta
FINI -
- RET

```

```

PartProgram[OMOSL]
; Origine ASSE SL
;
; L1 = Posizione asse SL con Origine fatta
;
- OMO/SL
- MOV/SL, L1
- EVS/6
- DIS/6: OMO SL fatta
- RET

```

```

PartProgram[OMOSR]
; Origine ASSE SR
;
; L1 = Posizione asse SR con Origine fatta
;
- HOM/SR
- MOV/SR, L1
- EVS/7
- DIS/7: Origine SR fatta
- RET

```

```

PartProgram[OMOLU]
; Origine ASSE LU
;
; L1 = Posizione asse LU con Origine fatta
;
- HOM/LU
- MOV/LU, L1
- EVS/8
- DIS/8: Origine LU fatta
- RET

```

3.5 Dati campionati

Buffer acquisizione dati campionati

Il buffer circolare utilizza un'area interna.

L'attività di bufferizzazione viene attivata con il comando:

CAP/50:X1,14,Y1,17..

dove :

50 significa ATTIVA BUFFER

X1 e' il nome dell'asse

14 e' il tipo di memorizzazione di questo asse

Il tutto può essere ripetuto per un massimo di 8 assi.

Sono previsti i seguenti tipi di memorizzazione:

GAP_QUOTAS	(1)
GAP_POSRIF	(2)
GAP_QUOTAS_ABS	(6)
GAP_POSRIF_ABS	(7)
GAP_EPSI	(14)
GAP_VOLT	(17)
GAP_VOLT_ABS	(18)

L'attività di bufferizzazione viene disattivata con il comando:

CAP/51:X1,0..

dove 51: significa DISATTIVA BUFFER

X1 non e' usato

0 non e' usato

Le posizioni finali del movimento sono in funzione dello stato dell'asse; assolute o incrementali o riferite ad un offset.

Nel caso di movimenti circolari gli assi oltre ai due necessari per eseguire il cerchio, sono mossi in modo interpolato con la circonferenza (elica cilindrica).

Il senso di rotazione e' funzione oltre che da CIR CIL anche dall'ordine degli assi sul movimento.

Vale la regola di considerare il primo asse come ascissa e il secondo come ordinata; guardare gli assi secondo il verso positivo, avendo l'ascissa da sinistra a destra e l'ordinata che si allontana lungo la direzione dello sguardo.

Esempio di utilizzo dati campionati

```

;          ***** BUFFERIZZAZIONE ASSE XL *****
;
;          -DIS/10:Inizio Bufferizzazione XL
;          -DIS/11:>
;          -AZZ/\GET_XL.TXT
;          -SCR/\GET_XL.TXT,N Pos_Reale Pos_Teorica Volt
;
;          -CAP/51:XL,0                ; buffer off
;          -CAP/50:XL,1,XL,2,XL,17 ; buffer on
;
;          -TMM/2000
;
;          -CAP/51:XL,0                ; buffer off
;          -SET/L10,0,L9,0
LOOP -GAV/L11,L12,L1,L3
;          -JLE/L11,0,FINE
;          -JEQ/L12,0,DOPO
;          -SCR/\GET_XL.TXT,Sbordamento,L12,L12
;          -DIS/11:Overflow,L12
;          -JMP/FINE
DOPO -ADD/L10,L11
;          -ADD/L9,1
;          -SCR/\GET_XL.TXT,L1,L3,L9
;          -JMP/LOOP
FINE -DIS/10:Finito,L10

```

3.6 Read Axes Parameters

Legge vari parametri degli assi (da 1 a 8 assi)
Il tipo di parametro dipende dallo switch a1:
-RAV/a1:b1,A1,c1,B1,...

a1 = tipo di parametro da leggere (switch:)

- 0 = INFOrmazioni su asse o sistema
- 1 = Quote assi reali
- 2 = Quote assi teoriche
- 3 = Servizio per marker (dopo OMO)
- 4 = Velocita' asse
- 5 = Quote assi staticizzate (da TCH TPE)
- 6 = Quote assi reali assolute
- 7 = Quote assi teoriche assolute
- 8 = Quote assi staticizzate assolute
- 9 = Offset relativo
- 10 = Over travel positivo
- 11 = Over travel negativo
- 12 = Over travel positivo assoluto
- 13 = Over travel negativo assoluto
- 14 = Errore d'interpolazione
- 15 = KP (proportional gain)
- 16 = VFF (velocity feed forward)
- 17 = Volt a DAC
- 18 = Volt a DAC assoluto
- 19 = KD (derivative gain)
- 20 = KI (integrative gain)
- 21 = Max integrative error
- 22 = Volt offset
- 23 = KD2 (derivative 2 gain)
- 24 = AFF (acceleration feed forward)
- 25 = Collision Error
- 26 = Servo Error
- 27 = Position Error
- 28 = Prescrizione Massima (Volts)
- 29 = Tempo di latenza per STOP COUNTER
- 30 = Tensione massima per STOP COUNTER
- 31 = Velocita' massima per STOP COUNTER
- 32 = KS (smorzatore)
- 33 = Rampe non lineari PENDENZA ACC (variazione acc al sec)
- 34 = Rampe non lineari PENDENZA DEC (variazione dec al sec)

- 35 = Rampe non lineari TEMPO PENDENZA ACC
- 36 = Rampe non lineari TEMPO PENDENZA DEC
- 37 = Valore della correzione ASSE
Quota reale = Quota Encoder + Valore Correzione
- 38 = KT (predictive)
- 39 = Velocita' limite dell'asse o di sistema (SYSTEM)
- 40 = Accelerazione limite dell'asse o di sistema (SYSTEM)
- 41 = Decelerazione limite dell'asse o di sistema (SYSTEM)
- 42 = Quota da marker con correzioni
- 43 = Quota da marker senza correzioni
- 44 = Distanza reale dalla meta
- 45 = Distanza teorica (calcolo) dalla meta
- 46 = Riduzione velocita' per supero errore
- 47 = Step encoder persi in modo cumulativo
- 48 = SAMPLE Tempo di campionatura (millisecondi)
- 49 = SAMPLE Tempo effettivamente usato (millisecondi)
- 50 = Lettura livelli fatti nel BUFFER CIRCOLARE **1
- 51 = Lettura livelli liberi del BUFFER CIRCOLARE **2

- 55 = CFR (Change FeedRate %)
- 56 = CSP (Change SPindle %)
- 57 = RAPPORTO GIRI MANDRINO / MOTORE (Cambio Gamma

Mandrino)

- 58 = GIRI MAX MANDRINO (Cambio Gamma Mandrino)

- 60 = Passo Encoder Caratterizzato
- 61 = Shift Asse (corrente)
- 62 = Velocita' teorica dell'asse
- 63 = Shift per tornio (spostamento da centro R a punta tool)
- 64 = Shift Quota (corrente)

- 65 = KF (Friction)
- 66 = Addolcitore di OVERSHOOT
- 67 = Quote assi teoriche in evoluzione (continuo ...)
- 68 = Quote assi teoriche assolute in evoluzione (continuo ...)

-CAP/98:V,50,B,0
| Offset B
Raggio di sviluppo
con raggio = 0 chiusura prestazione

- 99 = assi usati nel PP (1 = DEFINITI 0 = NON DEFINITI)

- 100 = INFOrmazioni supplementari su asse

- 101 = INFOrmazioni hardware
PRIMA ELECTRINICS = Status_R
NEWTON = ENCODER_NEWTON
SIX = ENCODER_SIX
SERCOS = ENCODER_SERCOS
YASKAWA = ENCODER_YASKAWA
ETEL = ENCODER_ETEL
UNIVERSALE = POSITION_ACTUAL_UNI
SYSTEM = VERSIONE
- 102 = INFOrmazioni hardware
PRIMA ELECTRINICS = Device_R
NEWTON = ENCODER_MARKER
SIX = ENCODER_MARKER
SERCOS = ERROR_SERCOS
YASKAWA = ERROR_YASKAWA
ETEL = ERROR_ETEL
UNIVERSALE = DRIVE_FOLLOWING_ERROR_UNI
Asse STEPPER valore PULSE (generatore di frequenza)
SYSTEM = STATUS_PP
- 103 = INFOrmazioni hardware
PRIMA ELECTRINICS = ErrorCode_R
NEWTON = QUANTI_MARKER
SIX = QUANTI_MARKER
SERCOS = DAC_SERCOS
YASKAWA = DAC_YASKAWA
ETEL = DAC_ETEL
UNIVERSALE = POSITION_SERPOINT
ETHERBOX_FRQ valore FREQUENZA (letta da DAC_OUTPUT)
Asse STEPPER valore FREQUENZA ottenuta con "PULSE"
SYSTEM = NOT USED
- 104 = INFOrmazioni hardware
ETEL = ENCODER_START_ETEL
UNIVERSALE = LATCH_POSITION_UNI
SYSTEM = NOT USED
- 110 = Valore dello scostamento dell'asse fisico di un virtuale
111 = Valore LEN_ARM1_V
112 = Valore LEN_ARM2_V
113 = Valore LEN_ARM3_V

**1 il valore ritornato e' 0 o il numero di livelli memorizzati

**2 il valore ritornato e' 0 o il numero di livelli liberi,
se non e' attiva la gestione buffer il valore e' -1

b1 c1 valori letti dagli assi

A1 B1 nomi degli assi

INFO

valore letto (a bits) :

1H ; ASSE DI SOLA LETTURA
 2H ; ASSE CON LOOP DI POSIZIONE APERTO (DAC)
 4H ; ASSE CON OMO FATTA
 8H ; ASSE IN SERVO ERROR (ASSE FERMO) **E
 10H ; ASSE IN COLLISION (ASSE IN MOVIMENTO) **E
 20H ; ASSE NON IN POSIZIONE **E
 40H ; ASSE IN BLOCCO CONTA **E
 80H ; ASSE INCREMENTALE
 100H ; ASSE CON PID INTEGRATIVO SU MOVIMENTO
 200H ; ASSE CHE NON SI DEVE MUOVERE (UNMOVED)
 400H ; ASSE CON ASSERVIMENTO ATTIVO
 800H ; ASSE IN DRIVER KO **E
 1000H ; ASSE CON CORREZIONE ATTIVA
 2000H ; ASSE CON COLLISION ERROR TESTATO
 4000H ; ASSE CON SERVO ERROR TESTATO
 8000H ; ASSE CON POSITION ERROR TESTATO
 10000H ; ASSE CON STOP COUNTER ERROR TESTATO
 20000H ; ASSE CON DRIVER OK ERROR TESTATO
 40000H ; ASSE GANTRY SCAMBIATO IN SLAVE (MASTER)
 80000H ; ASSE CON MOVIMENTO AGGREGATO
 100000H ; ASSE CON ANTICOLLISION TESTATA
 200000H ; ASSE CON MOVIM. FERMATO DA ANTIC. o CFR/0
 400000H ; ASSE IN ANTICOLLISION
 800000H ; ASSE IN DEADLOCK
 1000000H ; ASSE CON ASSERVIMENTO IN CORRENTE
 2000000H ; ASSE IN DRIVER FAULT
 4000000H ; ASSE IN OVER TRAVEL POSITIVO (SWITCH FINE CORSA +)
 8000000H ; ASSE IN OVER TRAVEL NEGATIVO (SWITCH FINE CORSA -)

Se il nome asse e' SYSTEM

valore letto (a bits) :

1H ; LETTURA ASSI ATTIVA
 2H ; ASSERVIMENTI ATTIVI
 4H ; RUN DI PART PROGRAMMA
 8H ; CYCLE START
 10H ; CYCLE STOP
 20H ; EMERGENZA

40H ; SISTEMA IN RIPARTENZA
 80H ; SISTEMA IN RESET
 100H ; SISTEMA CON ALMENO UN ASSE CON DRIVER KO
 200H ; SISTEMA CON ASSI IN ANTICOLLISIONE O DEADLOCK
 400H ; GRUPPI IN EMERGENZA

 INFO supplementare (100)

valore letto (a bits) :

1H ; ASSE CON KP TIPO SQUARE
 2H ; ASSE CON KD MEDIATA SU 4 CAMPIONATURE
 4H ; ASSE CON LETTURA CORRETTA DA SAMPLE
 8H ; ASSE CON KI TIPO SQUARE
 10H ; ASSE CON ANTICOLLISIONE DISABILITATA
 20H ; ASSE ROLLOVER UNSIGNED CON MOVIMENTO A MINOR SPAZIO
 40H ; ASSE CON ACCELERAZIONE S-CURVE
 80H ; ASSE CON DECELERAZIONE S-CURVE
 100H ; ASSE FUORI SERVIZIO
 200H ; ASSE NORMALMENTE FRENATO
 400H ; GANTRY CHE NON SEGUE (GANTRY RIGIDO)
 800H ; GANTRY DISABILITATO
 1000H ; MOVIMENTO PERMESSO ANCHE SENZA OMO
 2000H ; ASSE MOSSO IN ACCELERAZIONE
 4000H ; ASSE MOSSO IN DECELERAZIONE
 8000H ; ASSE MOSSO A VELOCITA' COSTANTE
 10000H ; ASSE IN PLOTTATURA
 20000H ; ASSE CON MARKER VERO
 40000H ; ASSE CON GESTIONE MOV_SWITCH DISABILITATA
 ;
 100000H ; ASSE "C" ROLLOVER NEAR CONCORDE CIR CIL
 200000H ; ASSE MANDRINO COMANDATO SOLO CON VFF **
 400000H ; ASSE VIRTUALE
 800000H ; ASSE NORMAL. DISABIL. FRENATO FA RIPRESA SERVO

ERROR

1000000H ; ASSE CON FRENO ATTIVO
 2000000H ; ASSE NORMAL. DISABIL. CON RIPRESA CONTINUA DEL

SERVO ERROR

** In questo caso l'errore di inseguimento e messo uguale all'errore
 che l'asse fa in una campionatura (errore della rotazione in UDM asse)

Se il nome asse e' SYSTEM
valore letto (a bits) : MASCHERA DEI GRUPPI DI ASSI IN EMERGENZA

INFO hardware (101)

valore letto : INFOrmazioni hardware (PRIMA ELECTRONICS Status_R)
Asse NEWTON o SIX valore ENCODER letto in impulsi

Se il nome asse e' SYSTEM
valore letto VERSION di AxesBrain

INFO hardware (102)

Asse STEPPER valore PULSE (generatore di frequenza)
Asse NEWTON o SIX ENCODER letto sul MARKER in impulsi

Se il nome asse e' SYSTEM
valore letto CONTATORE stati DISABLE ENABLE che deve
essere 0 (DEBUG)

INFO hardware (103)

Asse STEPPER valore FREQUENZA ottenuta con "PULSE"
Asse NEWTON o SIX QUANTI_MARKER incontrati

Se il nome asse e' SYSTEM
valore letto CONTATORE di HANDLING_MOVE bloccata con
multi CPU o Hyper Treading (con mono CPU deve essere 0)

INFO hardware (104)

Asse ETEL valore ENCODER_START_ETEL "impulsi"

Se il nome asse e' SYSTEM
valore letto CONTATORE di SAMPLE bloccata con
multi CPU o Hyper Treading (con mono CPU deve essere 0)

3 = Servizio per marker

Il valore estratto ha i seguenti significati:

- 1) Dopo una OMO con micro e marker, la distanza del micro dal marker.
- 2) Dopo una OMO senza micro e/o senza marker, la posizione dove l'asse era stato azzerato
- 3) Dopo una OMO/8: o una TCH/8: la posizione del marker; tale valore e' tipicamente zero esatto (a meno che non si sia fatta una OMO su marker precedentemente).
Il non ritrovare zero significa perdita di conteggio.

Note:

RAV/0:G1,X1

Dove G1 ha il bit 800H (2048) settato se il DRIVER e' KO.

Il bit e" settato sempre, anche se il controllo non e' abilitato.

Puo' essere sentita globalmente con la seguente istruzione:

RAV/0:G1,SYSTEM

Dove G1 ha il bit 100H (256) settato se almeno un asse e' in DRIVER KO

In questo caso il bit e' settato solo se il controllo e' abilitato

Ricordarsi che il bit 100H e' attivo con asse non asservito, quindi per "lavorare" bisogna avere il bit 2H attivo e il bit 100H resettato.

3.7 Change Axes Parameters

Cambia vari parametri degli assi (da 1 a 8 assi)

Il tipo di parametro dipende dallo switch a1:

-CAP/a1:A1,b1,B1,c1,...

a1 = tipo di parametro da cambiare (switch:)

- 0 = Informazioni su asse
- 1 = unused
- 2 = unused
- 3 = unused
- 4 = Velocita' asse
- 5 = unused
- 6 = unused
- 7 = unused
- 8 = unused
- 9 = Offset relativo
- 10 = Over travel positivo
- 11 = Over travel negativo
- 12 = Over travel positivo assoluto
- 13 = Over travel negativo assoluto
- 14 = unused
- 15 = KP (proportional gain)
- 16 = VFF (velocity feed forward)
- 17 = Volt a DAC
- 18 = Volt a DAC assoluto
- 19 = KD (derivative gain)
- 20 = KI (integrative gain)
- 21 = Max integrative error
- 22 = Volt offset
- 23 = KD2 (derivative 2 gain)
- 24 = AFF (acceleration feed forward)
- 25 = Collision Error
- 26 = Servo Error
- 27 = Position Error
- 28 = Prescrizione Massima (Volts)
- 29 = Tempo di latenza per STOP COUNTER (Secondi)
- 30 = Tensione massima per STOP COUNTER (Volts)
- 31 = Velocita' massima per STOP COUNTER
- 32 = KS (smorzatore)
- 33 = Rampe non lineari PENDENZA ACC (variazione acc al sec)
- 34 = Rampe non lineari PENDENZA DEC (variazione dec al sec)

35 = Rampe non lineari TEMPO PENDENZA ACC

36 = Rampe non lineari TEMPO PENDENZA DEC

38 = KT (predictive)

39 = Velocita' limite dell'asse o di sistema (SYSTEM)

40 = Accelerazione limite dell'asse o di sistema (SYSTEM)

41 = Decelerazione limite dell'asse o di sistema (SYSTEM)

42 = Cambio quota come con resolver per ENCODER_VIRTUALE

43 = Cambio quota diretta per ENCODER_VIRTUALE

46 = Riduzione velocita' per supero errore

50 = Attiva BUFFER CIRCOLARE

51 = Disattiva o Resetta BUFFER CIRCOLARE

52 = Trigger BUFFER CIRCOLARE

55 = CFR (Change FeedRate %)

56 = CSP (Change SPindle %)

57= RAPPORTO GIRI MANDRINO / MOTORE (Cambio Gamma

Mandrino)

58 = GIRI MAX MANDRINO (Cambio Gamma Mandrino)

61 = Shift Asse (finale)

63 = Shift per tornio (spostamento da centro R a punta tool)

64 = Shift Quota (finale)

65 = KF (Friction)

66 = Addolcitore di OVERSHOOT

98 = Gestione Sviluppo

99 = Definizione assi usati nel Part Program (LIKE_HOLD_TKM)

110 = Valore dello scostamento dell'asse fisico di un virtuale

111 = Valore LEN_ARM1_V

112 = Valore LEN_ARM2_V

113 = Valore LEN_ARM3_V

b1 c1 valori letti dagli assi

A1 B1 nomi degli assi

b1 c1 valori cambiati agli assi

Con 35 36 il TEMPO PENDENZA ha il seguente significato:

> 0 Scurve (tempo aggiuntivo alla rampa lineare)

= 0 Lineare

< 0 Sinusoidale

Se l'accelerazione e' sinusoidale la decelerazione puo' essere lineare o sinusoidale, ma non Scurve; in questo caso viene forzata la decelerazione sinusoidale.

Con 98 i valori passati hanno i seguenti significati:

Primo asse = asse virtuale programmato su figura sviluppata

Suo valore = raggio di sviluppo (con 0 viene annullata)

Secondo asse = asse rotante fisico da movimentare

Suo valore = offset

-CAP/98:V,50,B,0

| Offset B

Raggio di sviluppo

con raggio = 0 chiusura prestazione

Con 99 il valore passato per l'asse ha questi significati:

> 0 l'asse e' definito come asse da usare nel PP;

piu' precisamente con TKM LIKE_HOLD questo asse e' riportato in posizione di fermata a prescindere che sia mosso o no.

<= 0 l'asse viene tolto come asse del PP.

N.B. L'istruzione CAP/99: deve essere data nel Part Program su cui agira' l'istruzione TKM

INFO

CAP/0:ABS_MOV,1 Attiva sul task i movimenti dall'Origine Assoluta

CAP/0:ABS_MOV,-1 Rimette sul task i movimenti dall'Origine Relativa attuale

Con un solo asse (ABS_MOV) avente il parametro >0 attiva sul task il movimento dall'origine assoluta anche in presenza di offset relativi; <=0 disattiva questa prestazione.

CAP/0:X1,4 Forza lo stato di OMO fatta su asse X1 senza azzerare la quota

CAP/0:X1,-4 Toglie lo stato di OMO fatta su asse X1

CAP/0:X1,128 Attiva l'incrementale su asse X1

CAP/0:X1,-128 Toglie l'incrementale da asse X1

CAP/0:X1,256 Attiva l'integrativo con movimento su asse X1

CAP/0:X1,-256 Toglie l'integrativo con movimento su asse X1

CAP/0:X1,512 Forza lo stato di asse bloccato su asse X1

CAP/0:X1,-512 Toglie lo stato di asse bloccato su asse X1

CAP/0:X1,4096 Attiva la correzione su asse X1 (se la tabella e' caricata)

CAP/0:X1,-4096 Toglie la correzione su asse X1

CAP/0:X1,8192 Attiva la gestione su asse X1 di "COLLISION ERROR"

CAP/0:X1,-8192 Toglie la gestione su asse X1 di "COLLISION ERROR"

CAP/0:X1,16384 Attiva la gestione su asse X1 di "SERVO ERROR"
 CAP/0:X1,-16384 Toglie la gestione su asse X1 di "SERVO ERROR"
 CAP/0:X1,32768 Attiva la gestione su asse X1 di "POSITION ERROR"
 CAP/0:X1,-32768 Toglie la gestione su asse X1 di "POSITION ERROR"
 CAP/0:X1,65536 Attiva la gestione su asse X1 di "STOP COUNTER ERROR"
 CAP/0:X1,-65536 Toglie la gestione su asse X1 di "STOP COUNTER ERROR"
 CAP/0:X1,131072 Attiva la gestione su asse X1 di "DRIVER_OK ERROR"
 CAP/0:X1,-131072 Toglie la gestione su asse X1 di "DRIVER_OK ERROR"
 CAP/0:YM,262144 Scambia un master GANTRY caratterizzato con slave
 CAP/0:YM,-262144 Ripristina come da caratterizzazione il GANTRY

INFO supplementare (100)

CAP/100:X1,1 Attiva la gestione del KP in modo SQUARE dell'asse X1
 CAP/100:X1,-1 Mette la gestione del KP in modo LINEARE dell'asse X1
 CAP/100:X1,2 Attiva la gestione "mediata" di KD dell'asse X1 (4 campionature)
 CAP/100:X1,-2 Mette la gestione non "mediata" di KD dell'asse X1 (1 campionatura)
 CAP/100:X1,4 Corregge la quota con il tempo di campionatura dell'asse X1
 CAP/100:X1,-4 Non corregge la quota letta dell'asse X1
 CAP/100:X1,8 Attiva la gestione del KI in modo SQUARE dell'asse X1
 CAP/100:X1,-8 Mette la gestione del KI in modo LINEARE dell'asse X1
 CAP/100:X1,16 Disabilita l'anticollisione dell'asse X1
 CAP/100:X1,-16 Abilita l'anticollisione caratterizzata dell'asse X1
 CAP/100:X1,32 Se l'asse e' ROLLOVER UNSIGNED si muove a minor spazio
 CAP/100:X1,-32 Se l'asse e' ROLLOVER UNSIGNED si muove a spazio comandato
 CAP/100:X1,64 Mette l'accelerazione di tipo S-CURVE
 CAP/100:X1,-64 Toglie l'accelerazione di tipo S-CURVE
 CAP/100:X1,128 Mette la decelerazione di tipo S-CURVE
 CAP/100:X1,-128 Toglie la decelerazione di tipo S-CURVE
 CAP/100:X1,256 Esclude l'asse (ASSE FUORI SERVIZIO)
 CAP/100:X1,-256 Reinclude l'asse (ASSE IN SERVIZIO)
 CAP/100:X1,512 Definisce l'asse come NORMALMENTE FRENATO
 CAP/100:X1,-512 Toglie la funzione NORMALMENTE FRENATO
 CAP/100:YM,1024 Definisce il gantry come gantry rigido
 CAP/100:YM,-1024 Rimette il gantry come "non rigido" se caratterizzato
 CAP/100:YM,2048 Disabilita il sistema gantry (solo su asse master)
 CAP/100:YM,-2048 Riabilita il sistema gantry (solo su asse master)
 CAP/100:YM,4096 Permette il movimento anche senza OMO
 CAP/100:YM,-4096 Riabilita il test su OMO fatta (se e' caratterizzato)
 CAP/100:YM,65536 (0x10000) Definisce l'asse in fase di plottatura
 CAP/100:YM,-65536 (-0x10000) Toglie lo stato di plottatura
 CAP/100:YM,131072 (0x20000) Forza lo stato marker vero (se omo fatta) per debug
 CAP/100:YM,-131072 (-0x20000)Toglie lo stato marker vero (se omo fatta) per debug
 CAP/100:X1,262144 (0x40000) Disabilita la gestione MOV_SWITCH
 CAP/100:X1,-262144 (-0x40000) Riabilita la gestione MOV_SWITCH (se caratterizzata)

CAP/100:X1,0x80000 Altro modo per eseguire HOM/3: (toglie MARKER_VERO)
 CAP/100:X1,-0x80000 Riporta lo zero sul MARKER (mette MARKER_VERO)
 CAP/100:X1,0x100000 Con ROLLOVER_NEAR il movimento e' concorde a CIR CIL
 CAP/100:X1,-0x100000 Tolta la condizione di concordanza a CIR CIL
 CAP/100:X1,0x200000 Con asse mandrino il comando e con solo VFF = 100
 CAP/100:X1,-0x200000 Tolta la condizione di solo VFF all'asse mandrino
 CAP/100:X1,0x800000 Con asse norm. frenato fa ripresa SERVO ERROR (1 VOLTA)
 CAP/100:X1,0x2000000 Con asse norm. frenato continua ripresa di SERVO ERROR
 CAP/100:X1,-0x2000000 Tolta la ripresa continua di SERVO ERROR
 CAP/100:CU,0x8000000 ASSE LINMOV con SERVO attivo in norm. frenato

NOTA se l'asse che viene messo o tolto dalla fase di plottatura e'
 il master di un sistema GANTRY, anche sull'asse slave viene
 eseguita la funzione (CAP/100:YM,0x10000)

Per alcuni valori di caratterizzazione dando come CODICE
 il valore corrispondente in OR con 80H viene anche modificato
 il valore su statica; ossia il valore permane anche dopo RESET

I codici sono:

(10) (138) OVER TRAVEL POSITIVO
 (11) (139) OVER TRAVEL NEGATIVO
 (12) (140) OVER TRAVEL POSITIVO ASSOLUTO
 (13) (141) OVER TRAVEL NEGATIVO ASSOLUTO
 (15) (143) KP
 (19) (147) KD
 (23) (151) KD2
 (20) (148) KI
 (16) (144) VFF
 (24) (152) AFF
 (21) (149) MAX_ERR_I
 (25) (153) Collision Error
 (26) (154) Servo Error
 (27) (155) Position Error
 (28) (156) Prescrizione Massima (Volts)
 (65) (193) KF

(0) (128) INFO per i seguenti sottocodici
 (256) MSK_INTEGRATIVO_MOV
 (512) MSK_DYNAMIC_OFFSET
 (8192) COLLISION ERROR
 (26384) SERVO ERROR
 (32768) POSITION ERROR
 (65536) STOP COUNTER ERROR
 (131072) DRIVER_OK_ERROR

- (100) (228) INFO supplementare per i seguenti sottocodici
- (1) MSK_KP_SQUARE
- (2) MSK_KD_MIDDLE
- (4) MSK_EVALUATE
- (8) MSK_KI_SQUARE
- (64) ACCELERAZIONE S-CURVE
- (128) DECELERAZIONE S-CURVE
- (512) NORMALMENTE FRENATO

CAP/9:X1,50 Mette l'offset relativo su asse X1: viene spostato lo zero rispetto all'offset assoluto di 50 in positivo.
(Se l'asse era a zero dopo questa istruzione diventa -50)
Siccome viene spostato lo zero rispetto all'offset assoluto, e non rispetto allo zero relativo, ogni cambio non tiene conto di quello precedente.
Quindi per tornare a zero assoluto basta dare un cambio con valore 0.

CAP/50:X1,14,X1,17 Abilita il buffer a memorizzare EPSI per X1 e VOLT per X1.
In questo caso il valore che segue l'asse e' il codice RAV corrispondente. Si possono bufferizzare da 1 a 8 assi. Sono validi i seguenti codici:

- 1 = Quote assi reali
- 2 = Quote assi teoriche
- 6 = Quote assi reali assolute
- 7 = Quote assi teoriche assolute
- 14 = Errore d'interpolazione
- 17 = Volt a DAC
- 18 = Volt a DAC assoluto
- 48 = Tempo di campionatura
- 49 = Tempo usato in campionatura
- 90 = Speed hardware (P.E. digitale passiva)
- 91 = Current hardware (P.E. digitale passiva)

CAP/51:X1,0 Disabilita la memorizzazione su buffer
L'asse non e' usato.
Se il valore e' ≥ 0 la memorizzazione del buffer viene fermata (questo avviene naturalmente)

con il buffer pieno)
Se il valore $e' < 0$ l'area del buffer viene
azzerata

CAP/52:X1,1,X1,100 Attiva il trigger su buffer circolare
L'asse X1 fa partire la memorizzazione quando
la quota reale $e' \geq$ a 100.
CAP/52: deve precedere CAP/50:, e viene
annullata da CAP/51:
La condizione di trigger viene fatta su un solo
asse con i seguenti codici:
1 = Quote assi reali
2 = Quote assi teoriche
6 = Quote assi reali assolute
7 = Quote assi teoriche assolute
14 = Errore d'interpolazione
17 = Volt a DAC
18 = Volt a DAC assoluto
90 = Speed hardware (P.E. digitale passiva)
91 = Current hardware (P.E. digitale passiva)
Se il codice e' positivo la condizione $e' \geq$ al
valore; con codice negativo la condizione $e' \leq$ al
valore.

CAP/55:X1,10 Permette il change feedrate di un movimento.
Il valore deve essere tra 0 e 100 (percentuale)
Se quando e' dato il comando, l'asse e' fermo, il
suo valore e' memorizzato e il movimento che seguirà
partirà con il valore di CFR dato con CAP/55:..
A fine movimento il valore di CFR sull'asse e'
riportato a 100. Lo stesso succede dopo un reset.

TENTATIVO IN CORRENTE DI USARE UNA SORTA DI VFF

QR0 = Quota Reale 0
QR1 = Quota Reale 1
QR2 = Quota Reale 2

SPAZ = Velocità Teorica 1 = $QT2 - QT1$

ACC = $SPAZ - (QR2 - QR1) = SPAZ - QR2 + QR1$

ACC = accelerazione necessaria per raggiungere la velocità teorica senza preoccuparsi dello spazio percorso, che verrà controllato dal PID

Perciò verrà usato l'algoritmo seguente da sommare al PID

$$\begin{aligned} & (\text{SPAZ} - \text{QR2} + \text{QR1}) * \text{KC} * \text{VFF} \\ & (\text{SPAZ} - \text{VR2}) * \text{KC} * \text{VFF} \end{aligned}$$

Cercando di correggere anche gli errori di accelerazione si ottiene:

$$\text{ACC} = \text{VR2} - \text{VR1} = (\text{QR2} - \text{QR1}) - (\text{QR1} - \text{QR0}) = \text{QR2} - 2 * \text{QR1} + \text{QR0}$$

perciò

$$\text{ERRACC} = \text{Acc Teorica} - \text{QR2} + 2 * \text{QR1} - \text{QR0}$$

$$\text{ERRACC} = \text{Acc Teorica} - \text{VR2} + \text{VR1}$$

in definitiva

$$(\text{SPAZ} - \text{QR2} + \text{QR1} + \text{AccTeorica} - \text{QR2} + 2 * \text{QR1} - \text{QR0}) * \text{KC} * \text{VFF}$$

Questo algoritmo permette di controllare l'asse solo usando il KP il VFF ed eventualmente KI e AFF

$$\begin{aligned} & (\text{SPAZ} + \text{AccTeorica} - 2 * \text{QR2} + 3 * \text{QR1} - \text{QR0}) * \text{KC} * \text{VFF} \\ & (\text{SPAZ} + \text{AccTeorica} - 2 * (\text{QR2} - \text{QR1}) + (\text{QR1} - \text{QR0})) * \text{KC} * \text{VFF} \\ & (\text{SPAZ} + \text{AccTeorica} - 2 * \text{VR2} + \text{VR1}) * \text{KC} * \text{VFF} \end{aligned}$$

Analizzando meglio il tutto si noterà che:

QR0 = Quota Reale 0

QR1 = Quota Reale 1

QR2 = Quota Reale 2

SPAZ = Velocità Teorica 1 = QT2 - QT1

DVEL = SPAZ - (QR2 - QR1) = QT2 - QT1 - QR2 + QR1 = (QT2-QR2) - (QT1-QR1)

ma QT2-QR2 = E2 e QT1-QR1 = E1

DVEL = E2-E1 = Differenza tra la velocità teorica e quella reale

Analogamente considerando anche l'accelerazione

DACC = (QT2-QT1)-(QT1-QT0)-((QR2-QR1)-(QR1-QR0))

DACC = QT2-QT1-QT1+QT0-QR2+QR1+QR1-QR0

DACC = ((QT2-QR2) - (QT1-QR1)) - ((QT1-QR1) - (QT0-QR0))

DACC = (E2 - E1) - (E1 - E0)

DACC = Differenza tra l'accelerazione teorica e quella reale

Tutto cio' porta a capire meglio il funzionamento del PID

KP trasforma in tensione una parte dell'errore di posizione

KD trasforma in tensione una parte dell'errore di velocita'

KD2 trasforma in tensione una parte dell'errore di accelerazione

INFO

CAP/0:X1,4 Forza lo stato di OMO fatta su asse X1 senza azzerare la quota
CAP/0:X1,-4 Toglie lo stato di OMO fatta su asse X1
CAP/0:X1,128 Attiva l'incrementale su asse X1
CAP/0:X1,-128 Toglie l'incrementale da asse X1
CAP/0:X1,256 Attiva l'integrativo con movimento su asse X1
CAP/0:X1,-256 Toglie l'integrativo con movimento su asse X1
CAP/0:X1,4096 Attiva la correzione su asse X1 (se la tabella e' caricata)
CAP/0:X1,-4096 Toglie la correzione su asse X1
CAP/0:X1,8192 Attiva la gestione su asse X1 di "COLLISION ERROR"
CAP/0:X1,-8192 Toglie la gestione su asse X1 di "COLLISION ERROR"
CAP/0:X1,16384 Attiva la gestione su asse X1 di "SERVO ERROR"
CAP/0:X1,-16384 Toglie la gestione su asse X1 di "SERVO ERROR"
CAP/0:X1,32768 Attiva la gestione su asse X1 di "POSITION ERROR"
CAP/0:X1,-32768 Toglie la gestione su asse X1 di "POSITION ERROR"
CAP/0:X1,65536 Attiva la gestione su asse X1 di "STOP COUNTER ERROR"
CAP/0:X1,-65536 Toglie la gestione su asse X1 di "STOP COUNTER ERROR"
CAP/0:X1,131072 Attiva la gestione su asse X1 di "DRIVER_OK ERROR"
CAP/0:X1,-131072 Toglie la gestione su asse X1 di "DRIVER_OK ERROR"
CAP/0:YM,262144 Scambia un master GANTRY caratterizzato con slave
CAP/0:YM,-262144 Ripristina come da caratterizzazione il GANTRY

INFO supplementare (127)

CAP/127:X1,1 Attiva la gestione del KP in modo SQUARE dell'asse X1
CAP/127:X1,-1 Mette la gestione del KP in modo LINEARE dell'asse X1
CAP/127:X1,2 Attiva la gestione del KD in modo SQUARE dell'asse X1
CAP/127:X1,-2 Mette la gestione del KD in modo LINEARE dell'asse X1
CAP/127:X1,4 Attiva la gestione del KD2 in modo SQUARE dell'asse X1
CAP/127:X1,-4 Mette la gestione del KD2 in modo LINEARE dell'asse X1
CAP/127:X1,8 Attiva la gestione del KI in modo SQUARE dell'asse X1
CAP/127:X1,-8 Mette la gestione del KI in modo LINEARE dell'asse X1

Per alcuni valori di caratterizzazione dando come CODICE
il valore corrispondente in OR con 80H viene anche modificato
il valore su statica; ossia il valore permane anche dopo RESET

I codici sono:

(15) (143) KP

(19) (147) KD

- (23) (151) KD2
- (20) (148) KI
- (16) (144) VFF
- (24) (152) AFF
- (21) (149) MAX_ERR_I
- (25) (153) Collision Error
- (26) (154) Servo Error
- (27) (155) Position Error
- (28) (156) Prescrizione Massima (Volts)

- (0) (128) INFO per i seguenti sottocodici
- (256) MSK_INTEGRATIVO_MOV
- (512) MSK_DYNAMIC_OFFSET
- (8192) COLLISION ERROR
- (26384) SERVO ERROR
- (32768) POSITION ERROR
- (65536) STOP COUNTER ERROR
- (131072) DRIVER_OK_ERROR

- (127) (128) INFO supplementare per i seguenti sottocodici
- (1) MSK_KP_SQUARE
- (2) MSK_KD_SQUARE
- (3) MSK_KD2_SQUARE
- (4) MSK_KI_SQUARE

CAP/9:X1,50 Mette l'offset relativo su asse X1: viene spostato lo zero rispetto all'offset assoluto di 50 in positivo. (Se l'asse era a zero dopo questa istruzione diventa -50) Siccome viene spostato lo zero rispetto all'offset assoluto, e non rispetto allo zero relativo, ogni cambio non tiene conto di quello precedente. Quindi per tornare a zero assoluto basta dare un cambio con valore 0.

CAP/50:X1,14,X1,17 Abilita il buffer a memorizzare EPSI per X1 e VOLT per X1. In questo caso il valore che segue l'asse e' il codice RAV corrispondente. Si possono bufferizzare da 1 a 8 assi. Sono validi i seguenti codici:
1 = Quote assi reali
2 = Quote assi teoriche

6 = Quote assi reali assolute
7 = Quote assi teoriche assolute
14 = Errore d'interpolazione
17 = Volt a DAC
18 = Volt a DAC assoluto

CAP/51:X1,0 Disabilita la memorizzazione su buffer
L'asse e il valore non sono usati.

```
-----  
-DIS/1:>  
-DIS/2:>  
INIZIO      -  
-MOV/Y,0  
-TCH/Y,100:10:L1,Y:TOCCATO  
-DIS/1:NON TOCCATO IN ANDATA  
-TMM/2000  
-JMP/RITORNO  
TOCCATO     -  
-DIS/1:VALORE ,L1  
-TMM/2000  
RITORNO     -  
-TCH/Y,0:1:L1,Y:SENTITO  
-DIS/2:NON SENTITO IN RITORNO  
-TMM/2000  
-JMP/INIZIO  
SENTITO     -  
-DIS/2:VALORE ,L1  
-TMM/2000  
-JMP/INIZIO
```

Change Position Loop

Cambia il loop di posizione dell'asse
Il tipo di "loop" dipende dallo switch a1:
-CPL/a1:A1

a1 = tipo di "loop" (switch:)
-2 = Tolto asservimento + Enable
-1 = Tolto asservimento (DAC)

- 0 = Tolto asservimento (SAC) (Default)
- 1 = Rimesso asservimento + Enable
- 2 = Phase axis

A1 nome dell'asse

Attivando il "loop" -1 (DAC) si puo' comandare l'asse a loop aperto con le istruzioni :
-CAP/4:... , -CAP/17:... e -CAP/18:... .

Attivando il "loop" 0 (SAC) si puo' comandare l'asse a loop aperto con l'istruzione -SAC/..

Esempio di utilizzo di RAV e CAP

```

;      VISUALIZZAZIONE Status_R ZR
VV     -RAV/0:L1,SYSTEM
       -SET/L1,AND(L1,2)
;      -JNE/L1,2,VV
AA     -RAV/101:L1,ZR
       -RAV/102:L2,ZR
       -DIS/1:Status FIIBRA di ZR
       -CAL/,HEX.PP,L1,1,40
       -DIS/2:Device di ZR
       -CAL/,HEX.PP,L2,2,40
;      -RAV/101:L3,Y
;      -RAV/102:L4,Y
;      -DIS/3:Status_R di Y,L3
;      -DIS/4:ErrorCode_R di Y,L4
;      -RAV/101:L5,XR
;      -RAV/102:L6,XR
;      -DIS/5:Status_R di XR,L5
;      -DIS/6:ErrorCode_R di XR,L6
;      -RAV/100:L7,SYSTEM
;      -DIS/7:DDD,L7
;      -TMM/10
;      -JMP/AA

;      VISUALIZZAZIONE IN ESADECIMALE
;
;      L1 = VALORE (0 - 2**32)
;      L2 = RIGA
;      L3 = COLONNA INIZIALE
;
;      -DIS/L2:L3:
;      -LET/L3,L3+7
;      -LET/L1,L1-MOD(L1,1)
;      -JGE/L1,0,RR
;      -LET/L1,L1+0x0100000000
RR     -LET/L4,MOD(L1,16)
;      -JMP/L4
0      -DIS/L2:L3:0
;      -JMP/AA
1      -DIS/L2:L3:1
;      -JMP/AA
2      -DIS/L2:L3:2
;      -JMP/AA
3      -DIS/L2:L3:3
;      -JMP/AA
4      -DIS/L2:L3:4
;      -JMP/AA
5      -DIS/L2:L3:5
;      -JMP/AA
6      -DIS/L2:L3:6
;      -JMP/AA
7      -DIS/L2:L3:7
;      -JMP/AA
8      -DIS/L2:L3:8
;      -JMP/AA
9      -DIS/L2:L3:9
;      -JMP/AA
10     -DIS/L2:L3:A
;      -JMP/AA
11     -DIS/L2:L3:B
;      -JMP/AA
12     -DIS/L2:L3:C

```

```

13      -JMP/AA
        -DIS/L2:L3:D
        -JMP/AA
14      -DIS/L2:L3:E
        -JMP/AA
15      -DIS/L2:L3:F
AA      -LET/L3,L3-1
        -LET/L1,(L1-L4)/16
        -DIS/9:L1,L1
        -JNE/L1,0,RR
        -DIS/10:USCITO
        -RET

;      ***** POSIZIONAMENTI ASSE XL *****
;      MISURAZIONE POSIZIONAMENTI
        -CAP/0:XL,32768      ;Position Error abilit.
        -CAP/27:XL,0.01    ;Position error
        -CAP/16:XL,0      ;Vff
        -CAP/24:XL,6      ;Aff
        -CAP/15:XL,100    ;KP
        -CAP/19:XL,130    ;KD
        -CAP/20:XL,50     ;KI
        -CAP/23:XL,0      ;KD2
;      -CAP/0:XL,256      ;Integrativo con movim.
UUU     -SFP/50000,7500,7500
        -SET/L1,0,L2,700,L3,700,L10,10,L20,0
;      L1 = POSIZIONE DI PARTENZA
;      L2 = POSIZIONE FINALE
;      L3 = INCREMENTO DI POSIZIONE
;      L10 = NUMERO MISURAZIONI AVANTI O INDIETRO
;      L20 = TEMPO SOSTA IN MICROSECONDI
        -SET/L15,ABS((L2-L1)/L3)
        -DIS/1:>
        -DIS/2:>
        -MOV/XL,L1
        -SET/L11,0
        -SET/L14,L11
        -KYB/INIZIO ANDATA,L14
        -SWA/1
RPT1    -SET/L5,L1,L16,0
LOOP1   -MOV/XL,L5
        -DIS/2: POSIZIONE = ,L5
        -TMM/L20
        -ADD/L5,L3,L16,1
        -JLE/L16,L15,LOOP1
        -SET/L11,L11+1
        -JGE/L11,L10,TTT
        -SET/L16,0
LOOP2   -SUB/L5,L3
        -MOV/XL,L5
        -DIS/2: POSIZIONE = ,L5
        -TMM/L20
        -ADD/L16,1
        -JLE/L16,L15,LOOP2
        -SET/L11,L11+1
        -SET/L16,0
        -JLT/L11,L10,RPT1
        -MOV/XL,L1
        -DIS/2: POSIZIONE = ,L1
        -TMM/L20
TTT     -RWA/L1,1
        -DIS/0:TEMPO = ,L1
    
```

3.8 Assi ROLLER

Per definire un asse come ROLLER e' necessario settare Kind_axis_1= il bit 0X00000008 (ROLLER 0 +) e contemporaneamente caricare con il valore del modulo il campo Revolution= (che non deve essere zero).

La visualizzazione dell'asse e' sempre compresa tra 0 e +MODULO; per visualizzare la quota assoluta usare -RAV/42:

Per default i movimenti vanno nella direzione comandata, sino a raggiungere la posizione finale, che e' data in coordinate rollover.

Usando l'istruzione -CAP/100:A1,32 tutti i movimenti che seguono vengono fatti percorrendo lo spazio minore. Non viene percorso un valore > di MODULO anche dando un movimento molto grande.

Questa istruzione rimane attiva sino a che non venga annullata da -CAP/100:A1,-32 o dal rilancio di AxesBrain. Anche il reset non ne cambia lo stato.

Es. asse con modulo 360 visualizzante 60; comandando 3670 (360*10+70) l'asse si porta a 70 percorrendo 10.

Percio' se viene letta la quota con -RAV/42: e poi mosso l'asse a questa quota esso rimane fermo. Questo solo utilizzando la funzione "percorso minore" (-CAP/100:A1,32)

Rammentare che non utilizzando la funzione "percorso minore" (-CAP/100:A1,32) i movimenti percorrono tutto lo spazio comandato.

Es. asse con modulo 360 visualizzante 60; comandando 3670 (360*10+70) l'asse si porta a 70 percorrendo 3610. Esiste anche l'asse ROLLER SEGNATO, che si puo' attivare con Kind_axis_1= il bit 0X00000010 (ROLLER +- MODULO) In questo caso le quote variano tra +- modulo e non funziona -CAP/100:A1,32. Inoltre non essendo veramente utile, potrebbe essere una prestazione da annullare.

3.9 OVER TRAVEL

L'OVER TRAVEL puo' essere definito in caratterizzazione mettendo in Positive_over_travel= e Negative_over_travel= i valori di fine corsa riferiti allo zero assoluto. Quindi se viene cambiato Offset_axis= e' necessario cambiare anche i limiti di over travel perche' viene cambiato l'offset assoluto.

Questi limiti sono attivati dall'azzeramento assi; muovendo in manuale l'asse si ferma sul limite che si vuole superare. Gli altri movimenti non partono se si supera il limite di corsa, ma danno una segnalazione di errore. Se entrambi i limiti sono zero, non viene considerato il limite di OVER TRAVEL.

Movimenti "sino a OVER TRAVEL"

NOTE per i movimenti lanciati da HANDLING_MOV
Mettendo nel campo MODIFYER_MOV della struttura dei movimenti il valore 0x4000 (or con gli altri bit del campo) i movimenti sono eseguiti sino al limite se questi viene superato. Non vengono trattati in questo modo i movimenti in continuo e circolari e HOM.

Utilizzando i campi :

Number_PositiveOverTravel_switch_port=
FalseTrue_logic_PositiveOverTravel_switch=
Number_NegativeOverTravel_switch_port=
FalseTrue_logic_NegativeOverTravel_switch=

vengono anche sentiti i micro di fine corsa. Essi sono gestiti nel seguente modo:

- 1) Caso normale che a prescindere dall'azzeramento assi viene segnalato errore con caduta di asservimento se vogliono superarli (la direzione del movimento viene testata con il corrispondente micro di oltre corsa)
- 2) Caso con Kind_axis_0= che ha il bit 0x00000040 settato; a prescindere dall'azzeramento assi il movimento in manuale quando sente il micro decelera e si ferma senza nessuna segnalazione.

La stessa cosa succede con i movimenti "sino a OVER TRAVEL"
Per tutti gli altri movimenti vale il controllo OVER TRAVEL
con i limiti in caratterizzazione; se viene incontrato il
micro di oltre corsa viene segnalato errore con caduta di
asservimento.

3.10 Assi asserviti solo in movimento

L'asse normalmente DISABILITATO e' un asse che quando e' fermo non e' asservito e se ha il freno e' frenato. Con movimento esso si autoasserve liberando l'eventuale freno sino a movimento concluso.

L'asse puo' essere caratterizzato tale settando in Kind_axis_0= il bit 0x00000080

Anche da part-program si puo' attivare/disattivare sull'asse la funzione di normalmente disabilitato con le seguenti istruzioni:

- 1) CAP/100:A1,512 in questo modo l'asse e' messo normalmente disabilitato, ma dopo di un reset ritorna nello stato di caratterizzazione
- 2) CAP/100:A1,-512 in questo modo l'asse non e' normalmente disabilitato, ma dopo di un reset ritorna nello stato di caratterizzazione
- 3) CAP/258:A1,512 in questo modo l'asse e' messo normalmente disabilitato, ma anche dopo un reset resta in questo stato
- 4) CAP/258:A1,-512 in questo modo l'asse non e' normalmente disabilitato, ma anche dopo un reset resta in questo stato

CAPITOLO 4

4 Configurazione nome assi per interfaccia manuale

Per poter gestire le risorse di sistema nella loro globalità con i cicli di lavorazione è necessario configurare i seguenti parametri.

Nell'ambiente "Sistema" la pagina manuale permette di muovere gli assi in modo manuale.

E' necessario però configurare gli assi sull'interfaccia raggruppandoli.

Nel file di sistema "Homemoveman.ini" nella sessione **[ConfiguraAssi]** è possibile configurare la situazione degli assi desiderata.

[ConfiguraAssi]

NumeroAssi=Numero assi da configurare

NumeroGruppi=Numero gruppi da raggruppare

Asse_01= Nome asse visualizzato nella "Combo Box" !

Nome asse fisico configurato in nel file "sistema.txt" !

Numero Gruppo !

Scritta Bottone +!

Scritta Bottone -!

I asse lineare oppure m asse mandrino

....

Asse_xx= Nome asse visualizzato nella "Combo Box" !

Nome asse fisico configurato in nel file "sistema.txt" !

Numero Gruppo !

Scritta Bottone +!

Scritta Bottone -!

I asse lineare oppure m asse mandrino

Come si è potuto vedere si usa il separatore ! per dividere i campi

Esempio

[ConfiguraAssi]

NumeroAssi=14

NumeroGruppi=4

Asse_01=X1!X1!1!X1+!X1-!l

Asse_02=X2!X2!1!X2+!X2-!l

Asse_03=X3!X3!1!X3+!X3-!l

Asse_04=Y1!Y1!2!Y1+!Y1-!l

Asse_05=Y2!Y2!2!Y2+!Y2-!l

Asse_06=Y3!Y3!2!Y3+!Y3-!l

Asse_07=A!A1!3!A+!A-!l

Asse_08=B!B1!3!B+!B-!l

Asse_09=C!C1!3!C+!C-!l

Asse_10=D!D1!3!D+!D-!l

Asse_11=M1!M1!4!M+!M-!m

Asse_12=M2!M2!4!M+!M-!m

Asse_13=M3!M3!4!M+!M-!m

Asse_14=M4!M4!4!M+!M-!m

CAPITOLO 5

5 Cicli sempre attivi

Per poter gestire le risorse di sistema nella loro globalità con i cicli di lavorazione è necessario configurare i seguenti parametri.

[ParametriAUTOMAZIONE]

StringaLancio=\ AxesBrainSTUDIO \PARTPROG\AUTO.PP

NomePartProgramLancio= AUTO.PP

Note:

Con la caratterizzazione standard qui mostrata, il sistema si avvia automaticamente lanciando il Part program “auto.pp” (oppure con altro nome dato dallo sviluppatore del Sistema)

Essendo questo Part Program editabile come un normale Part program, se si vuole che il Sistema si avvii con un percorso diverso e non accessibile dall’ambiente, occorre definire questo percorso particolare nella “StringaLancio=”

Esempio: **StringaLancio**=\programmi\application\partprog\start.pp

CAPITOLO 6

6 Correzione assi

Per poter gestire le risorse di sistema nella loro globalità con i cicli di lavorazione è necessario configurare i seguenti parametri.

- Lineare correzione lineare ass , di quadratura, di rollio e di beccheggio
- Griglia correzione con griglia di compensazione nel piano

6.1 Lineare

Correzione lineare asse

La correzione asse e' attivata leggendo i valori di correzione sul file specificato nel file “**sistema.txt**”:

sezione: [ParametriGenerali]
voce: FileCorrezione=Correz.txt

Il file deve avere il seguente formato:

```
PHISICAL_AXIS_NAME( CORRECTED="YM" INPUT="YM" )
{
-527.00000 = 0.00000
-514.30000 = -0.00104
.....
.....
31.80000 = -0.00880
44.50000 = -0.01265
}
CORRECTED = e' il nome dell'asse fisico su cui agisce la correzione
INPUT      = e' il nome dell'asse fisico da cui vengono lette le posizioni
            di correzione ( valore a sinistra di = )
```

Se il valore di sinistra di INPUT= sono il nome di due assi (separati da una virgola) si usa la tabella legata al primo o secondo asse a seconda dell'ultimo asse mosso. Naturalmente per avere un funzionamento corretto occorre che due tabelle di correzioni facciano riferimento ad un unico asse corretto e che gli assi input compaiano in entrambe ma con ordine invertito.

Il valore a sinistra del segno = e' la posizione riferita allo zero marker, in coordinate scala, senza offset assoluto. Per ottenere il valore in coordinate scala bisogna sommare alla quota encoder lette l'offset assoluto scritto in caratterizzazione

Il valore a destra del segno = e' l'errore rilevato in quel punto della scala. L'errore e' ottenuto sottraendo alla posizione laser (valore vero) la posizione encoder letta (valore encoder)

```
// THEORETICAL_POSITION : POSITION ENCODER LETTA + OFFSET ABSOLUTE  
CHARATTERIZATION  
// ERROR_POSITION : POSITION LASER (TRUE) - POSITION ENCODER LETTA
```

Per eseguire il laser occorre disabilitare la correzione agli assi interessati.

Questo si puo' ottenere nei seguenti modi:

- 1) Cambiare il nome al file di correzione prima di lanciare la movimentazione; in questo caso nessuna correzione e' attiva.
- 2) Cambiare il nome all'asse sul file di correzione con un nome inesistente prima di lanciare la movimentazione; in questo caso solo gli assi modificati non hanno correzione.

- 3) Disabilitare la correzione con l'istruzione CAP/0:YM,-4096,YS,-4096

In questo caso solo gli assi YM e YS sono senza correzione.

La correzione tolta in questo modo puo' essere riabilitata o dando l'istruzione CAP/0:YM,4096,YS,4096 o con "RESET".

L'errore di posizione del laser e' da trasformare in coordinate uomo (es. da micrometri in mm) con il proprio segno.

Le correzioni sono cumulative sull'asse corretto; questo da la possibilita' di correggere l'asse anche in funzione di errori funzione di altri assi.

6.2 Griglia

Correzione con griglia di compensazione nel piano

Le correzioni nello spazio agiscono su una terna di assi cartesiane
 L'ordine degli assi corretti e' quello messo in CORRECTED
 L'ordine degli assi usati e' quello messo in INPUT

Ogni riga di dati ha il seguente significato:

- valore 1 = posizione asse 1 INPUT
- valore 2 = posizione asse 2 INPUT
- valore 3 = posizione asse 3 INPUT (se previsto)

separatore =

- errore 1 = errore da assegnare all'asse 1 CORRECT
- errore 2 = errore da assegnare all'asse 2 CORRECT
- errore 3 = errore da assegnare all'asse 3 CORRECT (se previsto)

Ci possono essere 2 o 3 assi INPUT e 2 o 3 assi CORRECT
 Il numero di assi CORRECT e' indipendente dal numero di
 assi INPUT

non e' necessario che la tabella sia equispaziata ma i valori
 sulle righe, sulle colonne e sulle altitudini devono essere
 rispettati da tutti i punti.

Il numero di punti della tabella e': Nriga * Ncolonna * Naltitudine

```
MATRIX_AXIS_NAME( INPUT="XL,Y" CORRECTED="XL,Y,ZL" )
{
0.0 , 0.0   = 0.0 , 0.0 , 0.0
10.0 , 0.0  = -0.001 , 0.01 , 0.5
.....

.....
90.0 , 110.0 = -0.05 , 0.04 , -0.5
110.0 , 110.0 = -0.002 , 0.03 , -0.4

}
```

CAPITOLO 7

7 TARATURE ASSI – PID (PROPORZIONALE, INTEGRATIVO, DERIVATIVO)

7.1 GENERALITÀ E PREMESSE

Sono previsti due diversi modi per asservire gli assi che chiameremo:

- asservimento in VELOCITÀ
- asservimento in COPPIA

Di norma viene utilizzato l'asservimento in VELOCITÀ, con il quale si invia al DRIVE del motore un segnale (generalmente compreso tra +/- 10V.) il cui valore rappresenta, per analogia, la velocità a cui deve, istante per istante, girare il motore per realizzare la velocità di spostamento prevista per l'asse in questione nel ciclo di posizionamento.

L'asservimento in VELOCITÀ è previsto con connessione di tipo Analogico e/o digitale (Sercos, ecc.)

L'asservimento in COPPIA (detto anche “in CORRENTE”), utilizzato in casi particolari con azionamenti che lo richiedono (non avendo internamente il calcolo dei coefficienti PID.)

La descrizione del significato dei parametri per l'asservimento in COPPIA è rimandato a uno specifico paragrafo dedicato.

Per selezionare il tipo di pilotaggio occorre predisporre la opportuna maschera esadecimale (mostrata e descritta in seguito) denominata **Kind_Axis_0** ponendo il bit di peso 1 della cifra meno significativa

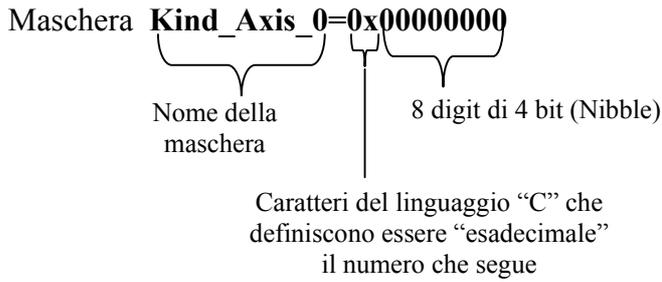
- = **0 (zero)** per l'asservimento in VELOCITÀ (predisposto per default)
- = **1 (uno)** per l'asservimento in COPPIA

NOTE PRELIMINARI

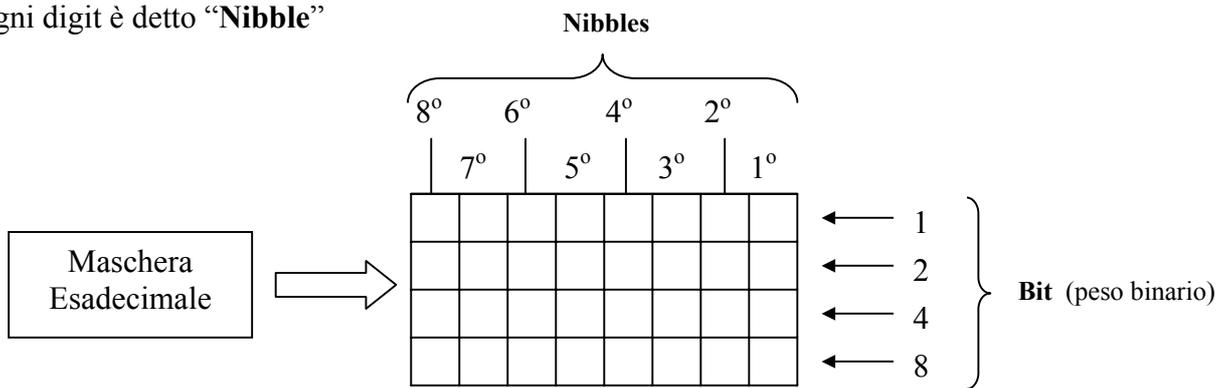
1. Tutti i valori di taratura usano come divisore tra le cifre intere e i decimali il punto **(0.005)** e non la virgola (~~0,005~~)
2. Le unità di misura dimensionali (mm, inches, gradi, rpm, ecc.) definite per ogni Asse nel parametro STEP, sono univoche per tutti i parametri dimensionali di quell'Asse e verranno d'ora in poi chiamate UDM (**Unità di Misura Dimensionale**)nell'intero documento. Pertanto,in applicazioni particolari, è possibile, volendo, assegnare unità di misura diverse ai singoli Assi.
3. Alcune tarature possono essere modificate dinamicamente da Part program tramite le istruzioni **CAP** (**C**hange **A**xis **P**arameter)

7.1.1 CONCETTI PRELIMINARI SULLE MASCHERE

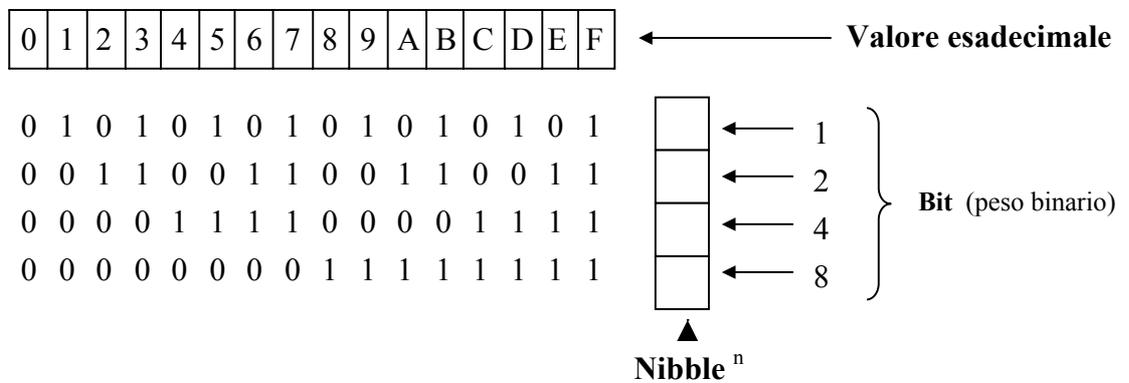
Ogni maschera è formata da un insieme di 8 digit di 4 bit, espressa nella forma seguente:



Ogni digit è detto “**Nibble**”



In ogni Nibble è pertanto possibile scrivere un valore compreso tra 0 ed F, secondo la codifica esadecimale mostrata in figura seguente



Attenzione Per comodità, quando una tabella di tarature prevede il valore da inserire in un Nibble della maschera, si fa riferimento solo al valore decimale corrispondente al bit che si vuole impostare, indipendentemente dal contenuto degli altri bit.

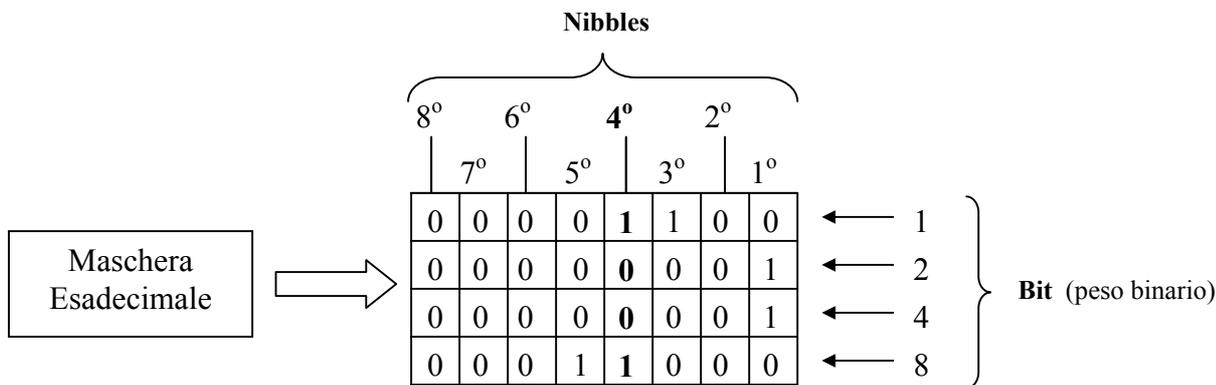
Se, ad esempio, una certa taratura prevedesse di porre a 1 il bit di peso 2 del 4° Nibble della maschera, si direbbe “scrivere 2000 nella maschera.....”, che è il modo per dare la doppia informazione (del bit e del peso decimale)

In realtà si intende che, indipendentemente dal contenuto del Nibble da modificare, deve essere messo a 1 il bit di peso 2 senza alterare il valore degli altri 3 bit del Nibble.

Chi effettua la taratura dovrà perciò ricercare nella tabella precedente il valore esadecimale da scrivere per porre a 1 il bit di peso 2 lasciando inalterato il resto del contenuto del Nibble (effettuando pertanto l'OR dei bit).

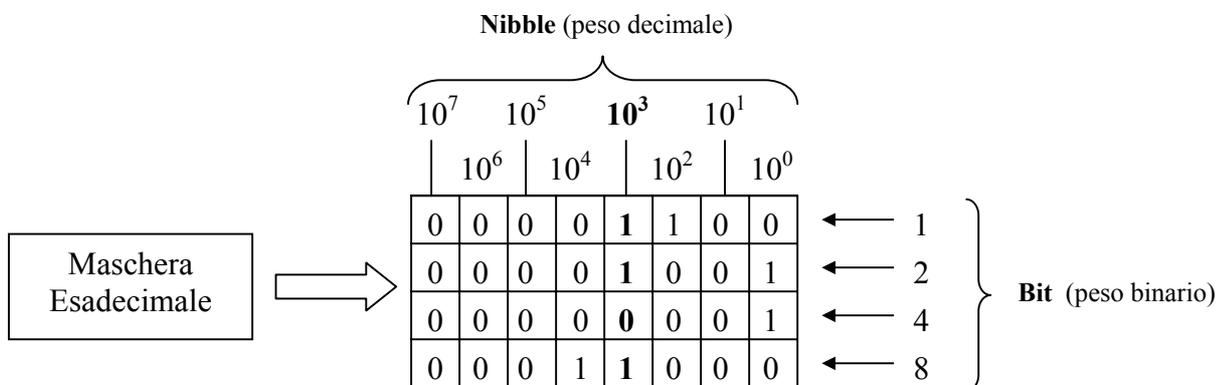
L'esempio seguente chiarisce tutti i concetti su esposti

Contenuto della maschera prima della taratura



Valore esadecimale contenuto 0 0 0 8 9 1 0 6

Conteuto della maschera dopo aver eseguito la taratura espressa come : “scrivere 2000”



Valore esadecimale contenuto 0 0 0 8 **B** 1 0 6

Valore da scrivere ——— ↑

7.2 CARATTERIZZAZIONE ASSI ANALOGICI IN VELOCITA'

7.2.1 TARATURE GENERALI DI SISTEMA RELATIVE A TUTTI GLI ASSI

GeneralStatus=0x00000000 Maschera generale di sistema – Valore di default=**0**
(Utilizzato solo in casi particolari – non modificare)

Time_collision_latency Tempo (in sec.) di mantenimento degli assi sotto
asservimento quando un asse, in Errore, viene fermato. –
Valore di default=**0,2**

7.2.2 CARATTERIZZAZIONE ASSI FISICI - PREDISPOSIZIONI

Per ogni Asse Analogico si hanno a disposizione, e devono essere caratterizzati, tutti i parametri seguenti

[X] Nome o simbolo di individuazione di un Asse
(alfanumerico max 7 caratteri)

i=PHYSICAL_AXIS_NAME(1,"X") Riportare lo stesso nome/simbolo dato all'Asse.
Il numero che precede il nome, nell'esempio =1, è una
indicazione ausiliaria del nome, che può essere utilizzata
o meno, ma che comunque, se esiste, è riconosciuta dalla
programmazione alla pari del nome dell'asse (ad
esempio, la scelta dell'asse su cui operare può essere
definita dal risultato di un calcolo)

b={ Indicazione di Sistema-NON MODIFICARE

Description="Asse X" Nome che compare sulle videate (libero, da definire,
come ad esempio : X, Trasversale, Rotante, ecc.)

Kind_axis="ARM" Indicazione di Sistema-NON MODIFICARE

Kind_axis_0=0x00000000 Maschera delle tarature del **Tipo di Asse**.
Valore di default = **4000**

Valori previsti e caratterizzabili: **NON UTILIZZARE VALORI NON PREVISTI**

0x00000001 - Asservimento Asse in COPPIA (Valore di default = **0** -VELOCITÀ)

0x00000010 - KI con funzionamento non lineare (più grande con errore basso)
Valore di default = **0** (KI con funzionamento lineare)

0x00000020 - Il primo asservimento dell'asse dopo aver dato (o ridato) potenza avviene in modo graduale. Ciò è particolarmente utile per permettere un corretto riassetto meccanico degli Assi Gantry dopo uno spegnimento della macchina.

0x00000040 - L'asse decelera sul micro di fine corsa (se esiste), solo in Manuale. Utile quando si muovono gli Assi prima di aver fatto gli Zeri

0x00000080 - L'abilitazione dell'Asse (Enable) viene automaticamente data e tolta all'inizio e alla fine di un movimento. Utile per quegli Assi che necessitano di bloccaggio a fine movimento.

0x00000800 - Esecuzione OMO a battuta.

0x00001000 - Esecuzione OMO solo sul Marker

0x00002000 - Esecuzione OMO solo sul Micro.

Valore di default = **0** per tutti i 3 campi.
(Esecuzione OMO su Micro + Marker)

0x00004000 - L'Asse si muove solo dopo OMO Valore di default = **4000**

Kind_axis_1=0x00070000 Maschera dei tipi di Asse - Valore di default **1F0000**
Questo valore di default predispone il controllo degli errori di
Collision, Servo, Position, Stop Counter, Drive ok.

Valori previsti e caratterizzabili: NON UTILIZZARE VALORI NON PREVISTI

- 0x00000004 – Asse con abilitazione Servo dopo il 1° movimento - Valore di default = **0**
- 0x00000008 – Asse Rollover (rotativo) con reset a 360° (oppure a un altro valore, ad esempio come N° di stazioni). Se =0 l'asse non si azzerà passando per 360° e continua a contare 361, 362 ecc.
- 0x00000040 - Asse “Traslo” con dinamica per Traslo elevatori e anello di posizione lasco
Valore di default = **0**.
- 0x00000080 – Asse con rampa di accelerazione S-CURVE se il bit di peso **80** è uguale a 1.
Valore di default = **0** (l'accelerazione predisposta è pertanto sinusoidale)
- 0x00000100 – Asse con rampa di decelerazione S-CURVE se il bit di peso **100** è uguale a 1.
Valore di default = **0** (la decelerazione predisposta è pertanto sinusoidale)
- 0x00000200 – Asse con sistema di fasatura (Motori lineari) – Valore di default = **0**
- 0x00010000 – Attivazione dell'Errore di COLLISION. Se messo = 0 la collision non viene controllata – Valore di default = **1**
- 0x00020000 – Attivazione del SERVO ERROR AD ASSE FERMO. Se messo = 0 il servo error non viene controllato – Valore di default = **1**
- 0x00040000 – Attivazione del POSITION ERROR (ASSE IN MOVIMENTO). Se messo = 0 il posizionamento non viene controllato – Valore di default = **1**
- 0x00080000 – Attivazione della protezione di STOP COUNTER ERROR. Se messo = 0 la protezione non agisce – Valore di default = **1**
- 0x00100000 – Abilitazione al segnale di DRIVE OK. Se messo = 0 non viene controllato il segnale di Driver Ok – Valore di default = **1**
- 0x00800000 – ASSE MASTER DI UN SISTEMA DI ASSI GANTRY
Se = 0 non è un Asse Gantry. Se =1 è un Asse Master di un sistema Gantry, dove, come descritto nel paragrafo “NOTE SULLA GESTIONE DEGLI ASSI GANTRY”, verranno date tutte le informazioni relative alla caratterizzazione ed al funzionamento GANTRY
- Valore di default = **0**
- 0x01000000 – ASSE MASTER CON SLAVE CHE DEVE SEGUIRE (se no va in Errore).
Se = 0 l'Asse Slave segue come riesce (ossia con i suoi parametri di taratura, ma non va in Errore. Se = 1 L'Asse Slave deve seguire strettamente il Master.
In caso non riesca, va in errore
- Valore di default = **0**
- 0x02000000 – ASSE MASTER CON SLAVE CHE SEGUE IL (Calcolo) TEORICO
Se = 0 lo Slave segue l'Encoder del Master - Valore di default = **0**

0x08000000 – Abilitazione PID INTEGRATIVO ANCHE IN MOVIMENTO

Se = 0 solo ad asse fermo – Valore di default = 0

0x10000000 – SOLO PER ASSE STEPPER - BOOST ATTIVO IN ACCELERAZIONE

Valore di default = 0

Nota – La taratura degli Assi Stepper è trattata in apposito capitolo

Kind_board_TransducerPosition=40

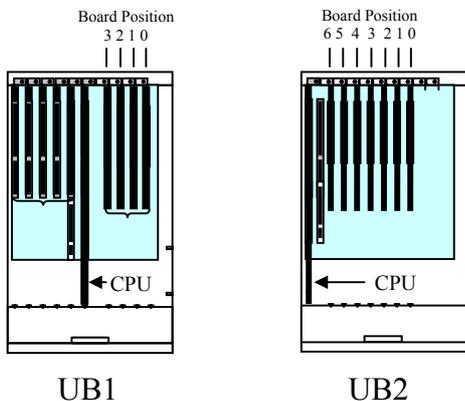
N° di riconoscimento del tipo di schede Assi utilizzate sulle quali risiede il canale Encoder.

Il valore di default = 40 corrisponde alle schede PCI 2000 - NON MODIFICARE

NOTA – Le 3 caratterizzazioni seguenti sono riferite solamente alla scheda PCI 2000.

Board_number_TransducerPosition=0

Posizione fisica del Back-plane in cui si trova la scheda su cui risiede il canale Encoder di questo Asse (che si sta caratterizzando) Il valore, compreso tra 0 ed N (numero massimo di schede inseribili nell'Unità Base scelta) è predisposto in fase di collaudo. NON MODIFICARE



NOTA- Nelle Unità Base MICROSYS 4120 la numerazione da 0 ad N parte dalla scheda più lontana dalla CPU, come mostrato in figura a lato.

Address_port_TransducerPosition=0x00000000

NON UTILIZZATO

Physical_number_TransducerPosition=1÷6

Numero fisico del canale Encoder della scheda Assi associato all'asse che si sta caratterizzando. Di norma si utilizza il canale 1 per il primo Asse, il 2 per il secondo, ecc. ma ciò non è strettamente obbligatorio, per cui è necessario effettuare questa caratterizzazione.

NOTA - Per default il N. del trasduttore Encoder viene associato al corrispondente N. dell'Asse in fase di collaudo dell'Unità di Governo

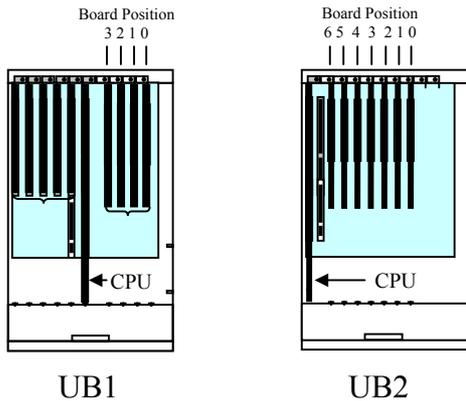
Kind_board_DAC=40

Tipo di scheda Convertitore DAC usata

Valore predefinito = 40 per le schede PCI 2000 -
NON MODIFICARE

NOTA – Le 3 caratterizzazioni seguenti sono riferite solamente alla scheda cod. PCI 2000.

Board_number_DAC=0



Posizione fisica del Back-plane in cui si trova la scheda su cui risiede il Convertitore DAC di questo Asse (che si sta caratterizzando) Il valore, compreso tra 0 ed N (numero massimo di schede inseribili nell'Unità Base scelta) è predisposto in fase di collaudo. NON MODIFICARE

NOTA- Nelle Unità Base MICROSYS 4120 la numerazione da 0 ad N parte dalla scheda più lontana dalla CPU, come mostrato nella figura a lato.

Address_port_DAC=0x00000000

NON UTILIZZATO

Physical_number_DAC=1÷6

Numero fisico del Convertitore DAC della scheda Assi associato all'asse che si sta caratterizzando. Di norma si utilizza il canale 1 per il primo Asse, il 2 per il secondo, ecc. ma ciò non è strettamente obbligatorio, per cui è necessario effettuare questa caratterizzazione.

Per default il N. del Convertitore DAC viene associato al corrispondente N. dell'Asse in fase di collaudo dell'Unità di Governo

NOTA – Se di un Asse (DAC + ENCODER) si usa solo il canale di misura, come ad esempio per collegare un Volantino, il **Physical_number_DAC** deve essere posto = 0

NOTA - di norma il valore assegnato a questa caratterizzazione è lo stesso assegnato alla **Physical_number_TransducerPosition=1÷6**

Number_SERVO_switch_port="EnableX"	Nome logico dell'output del segnale di DRIVE ENABLE di quest'asse (in taratura)
FalseTrue_logic_SERVO_switch=1	Livello logico del segnale di ENABLE (1 oppure zero - se diverso da 1 oppure da zero il sistema non gestisce l'Enable) Valore di default = 1
Number_DRVOK_switch_port="DriverOKX"	Nome logico dell'Input del segnale di DRIVER OK di quest'asse
FalseTrue_logic_DRVOK_switch=1	Livello logico del segnale di Driver OK (1 oppure zero - se diverso da 1 oppure da zero il sistema non gestisce il Driver Ok) Valore di default = 1
Level_TCH=0	<p>Ogni Asse ha un ingresso LATCH connesso con il corrispondente canale Encoder. La stretta connessione dell'HW permette di ottenere strobature sincrone con l'acquisizione della relativa posizione dell'Encoder. L'ingresso Latch viene abilitato ponendo Level_TCH=1 e disabilitato ponendolo = 0.</p> <p>Ogni Asse può strobare valori in modo indipendente dagli altri.</p> <p>NOTA - Volendo strobare contemporaneamente la posizione di più Assi, (ad esempio con il segnale di un Tastatore) si deve portare il segnale elettrico in parallelo su tutti gli Input Latch associati agli assi corrispondenti, dando lo stesso valore di caratterizzazione ai Level_TCH degli assi interessati (il valore da assegnare deve essere scelto tra 2 e 64)</p> <p>Valore di default = 0</p>
Revolution=360.0	<p>Reset della misura di un asse rotativo a 360°.</p> <p>Quando l'asse raggiunge i 360°, ricomincia a contare da 0 (zero). Per abilitare questo funzionamento occorre porre la maschera Kind_Axis_1 = 0x00000008. Se il bit di peso 8 è messo = 0, la conta non si resetta a 360° ma continua 361, 362 ecc.</p>

Nota – Se si deve contare una divisione diversa da 360°, come ad esempio le posizioni di un cambio utensile formato da 105 stazioni, si tarerà la maschera = 8 ma nel valore di Revolution si scriverà 105 (ciò significa che dopo la stazione 104 si ripresenterà la Zero)

7.2.3 VALORI E COEFFICIENTI PER ASSERVIMENTO DINAMICO ASSI

- KP=20** **Coefficiente di Proporzionalità** tra il valore dell'Errore ed il valore della Velocità. (Errore = Valore della posizione calcolata – (meno) Valore della posizione reale dell'Asse)
NOTA – *Una verifica significativa della buona taratura di un Asse è la seguente:*
Con tutti gli altri coefficienti messi = zero (KI, KD, KD2, MOLT_VFF, MOLT_AFF), muovendo l'Asse a velocità costante, l'errore di inseguimento deve essere esattamente = VEL (mm/sec.) / KP
Valore di default = **20**
- KI=0** **Coefficiente di Integrazione**, che tende a ridurre le derive di posizione dell'Asse rispetto alla posizione calcolata.
Ad esempio, ad Asse fermo, recupera l'offset dell'Azionamento (che nel tempo può variare per cause diverse) È buona norma tararlo dopo aver azzerato l'offset dell'asse tramite il parametro **Volt_offset** e aver minimizzato l'offset sull'Azionamento.
Valore di default = **0**
- VoltMaxIntegratedError=1** **Livello di errore massimo accumulato** fino al quale è abilitato l'intervento del coefficiente KI. Per valori superiori si ha comunque l'intervento di KI limitato al valore impostato. Il valore è espresso in Volt. Valore di default = **1**
- KD=0** **Coefficiente Derivativo** - Serve a ridurre la variazione dell'errore di inseguimento.
In regolazione di VELOCITÀ questo parametro ha poca influenza. Valore di default = **0**
- KD2=0** NON UTILIZZATO IN REGOLAZ. DI VELOCITÀ
Valore di default = **0**

Molt_VFF=0

Percentuale del valore del segnale di velocità che viene trasmessa all'Azionamento dell'Asse. Il valore di taratura sarà prossimo a 100, e sarà raggiunto mettendo a punto tutto il servosistema.

NOTA – Il valore di default del VFF è posto = 0 per permettere di rilevare l'andamento dell'errore di posizione che, con VFF = 0, fa muovere l'Asse.

Molt_AFF=0

DI NORMA NON UTILIZZATO IN REGOL: DI VELOCITÀ Valore di default = 0

KS=0.0005

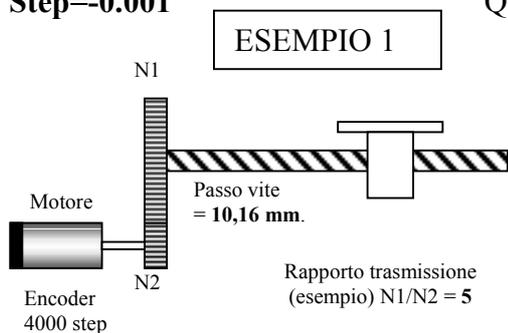
Coefficiente di Smorzamento ad asse fermo.

Il valore è espresso in UDM (nell'unità di misura dimensionale scelta con STEP per questo Asse) ed effettua un arrotondamento sul valore del segnale di errore, (arrotondamento sui valori calcolati, evitando che si generi un errore cumulativo dovuto al calcolo), e tende a smorzare eventuali oscillazioni sulla quota finale.

Questo smorzamento è particolarmente utile per la stabilizzazione di Assi leggeri e/o senza attrito, come assi a rulli, idrosostentati o pneumosostentati. Taratura tipica = ½ del valore della risoluzione (Valore UDM di 1 step Encoder)

Valore di default = **0.0005**

Step=-0.001



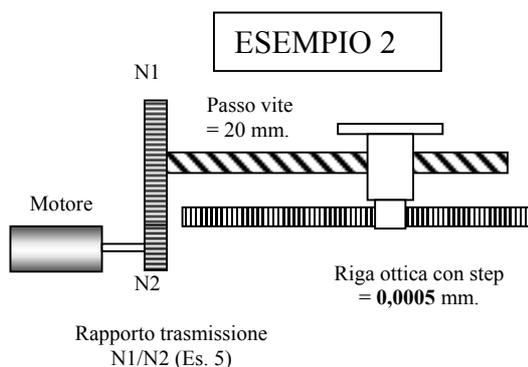
Questo parametro serve a calcolare lo

spostamento effettivo di un asse (in UDM) per ogni step dell'Encoder. Lo spostamento può essere espresso direttamente col valore oppure con la formula di calcolo. Il segno - (meno) permette di invertire il verso di conta se risulta contrario al verso del movimento. La formula da inserire è:

(Passo vite / Rapporto di trasm.) / N° step Encoder

Nell'Esempio 1 il risultato del calcolo è il seguente: $(10.16/5)/4000 = \mathbf{0.000508}$ mm.

ed è evidente che è meglio introdurre la formula che non introdurre direttamente il valore, che non deve essere arrotondato.



Nell'Esempio 2, indipendentemente dalla realizzazione meccanica, uguale alla prima, il valore dello Step è = **0.0005** (quello della riga).

In questo caso sarà più conveniente introdurre direttamente il valore del passo della riga.

Il valore di default = **-0.001** è stato scelto perché:

- Spesso le righe utilizzate hanno passo 0.001mm.
- Si rammenta che si usa il punto e non la virgola per separare i decimali
- Si rammenta che il valore può essere segnato (la mancanza di segno è intesa = segno +)

Volt_max =10

Questo parametro, tarabile da 0 a 10, definisce il valore massimo a cui limitare l'ampiezza del segnale di riferimento dato all'Azionamento del motore.

NOTA - Durante la messa a punto di un Asse è bene comunque limitare la velocità massima raggiungibile ponendo un limite inferiore, riportandolo a 10 per la messa a punto finale.

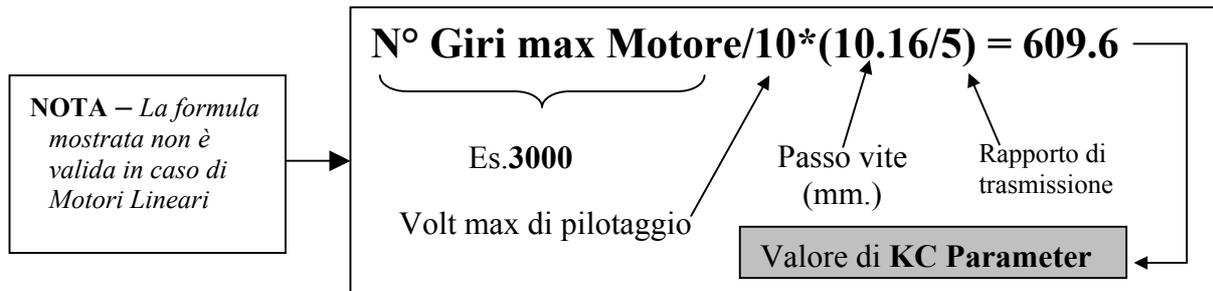
Valore di default=**10**

Volt_offset=0.0

Taratura del valore dell'offset per tenere fermo l'asse in posizione, compensando l'offset dell'azionamento. Il valore può essere positivo o negativo. In questo caso andrà fatto precedere dal segno - (meno).

Valore di default = **0.0**

KC_Parameter=3000/10*(20/1) **Coefficiente di Conversione**, che definisce la velocità dell'Asse (catena cinematica, elettrica e di misura comprese), applicando all'azionamento un riferimento di velocità pari a **1Volt**.
Calcolare come da esempio 1 precedente (relativo a STEP):



Come Valore di default si è predisposta la formula **3000/10*(20/1)** perché è un valore normale con i seguenti parametri:

- Motore da **3000** rpm con livello analogico a **10V**olt
- Passo vite = **20**mm.
- Rapporto di trasmissione = **1** (motore in asse sulla vite)

Con il riferimento di velocità al valore massimo (10V.), risulta una velocità di rapido max = 60.000mm/min.

Il valore, calcolato nell'esempio, può essere inserito come valore di KC ma si consiglia di utilizzare la formula predisposta come default perché adattando alcuni valori durante la caratterizzazione, non si è costretti a rifare tutte le volte il calcolo ed inserire il nuovo risultato

Vel_max=45000

Velocità massima, (espressa in UDM / min.) da definire per l'asse in taratura.

NOTA - La Velocità massima non deve essere superiore al valore di KC Parameter x 10 (10V = valore massimo di riferimento applicabile all'Azionamento)

Si noti che se viene scritto un valore superiore a quello massimo, si accumula errore di inseguimento perché il calcolo sopravanza la posizione istantanea raggiungibile dall'asse.

Acc_max=1000.0
Dec_max=1000.0

I valori di accelerazione e decelerazione sono espressi in UDM /sec² e sono definiti dal costruttore della macchina.

Valore di default = **1000.0 (UDM /sec²)**

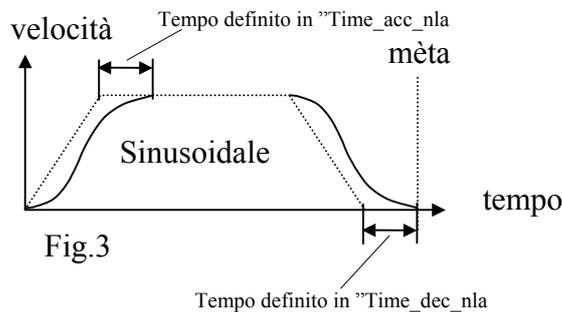
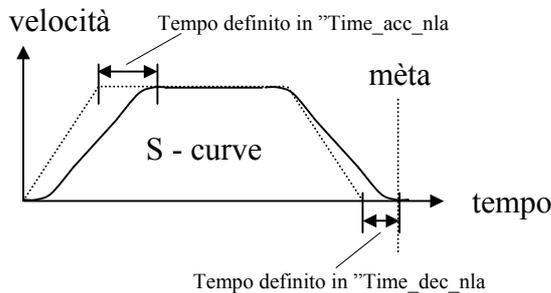
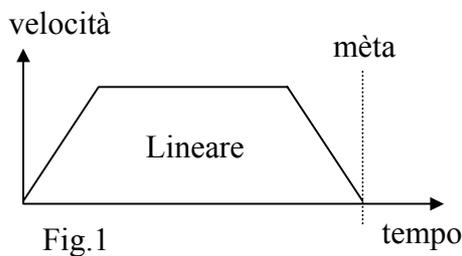
NOTA – *Poter definire valori diversi per Accelerazione e Decelerazione permette di poter abbassare leggermente la decelerazione quando, operando con valori particolarmente alti, si fosse in presenza di overshoot sulla quota finale.*

Diversamente si penalizzerebbe il tempo ciclo perché si abbasserebbe anche l'accelerazione

Time_acc_nla=-1
Time_dec_nla=-1

Queste tarature definiscono il **tipo di rampe di Accelerazione e di Decelerazione**, che possono essere di tre tipi

- **Lineare** – per rampe veloci
- **S-curve o trapezoidale** – per rampe più dolci di quelle lineari, ottenute inserendo un tempo aggiuntivo (ai tempi di accelerazione e/o di decelerazione ottenuti con rampe lineari dovute alla caratterizzazione di **Acc_max** e **Dec_max**)
- **Sinusoidali** – per rampe meno veloci ma la macchina ha movimenti più dolci.



Valori di taratura di Time_acc_nla / _dec_nla

Per Rampa Lineare assegnare 0 (Fig.1)

Per Rampe S-CURVE (Fig.2)

Acceleraz. (Kind_Axis_1 con bit **80 =1**)

Deceleraz. (Kind_Axis_1 con bit **100 =1**)

assegnare-1 per la rampa naturale

(Accelerazione max = alla programmata)

assegnare tempo in sec. (0.02 = 20msec.)

(Accelerazione max adattata dal calcolo)

Per Rampe Sinusoidali (Fig.3)

Acceleraz. (Kind_Axis_1 con bit **80 =0**)

Deceleraz. (Kind_Axis_1 con bit **100 =0**)

assegnare-1 per la rampa naturale

(Accelerazione max = alla programmata)

assegnare tempo in sec. (0.02 = 20msec.)

(Accelerazione max adattata dal calcolo)

NOTA - Per ottimizzare il ciclo è possibile utilizzare tipi diversi di accelerazione e decelerazione per lo stesso Asse.

7.2.4 CONTROLLO ED ERRORI ASSI

ServoError_standby=2.0 Taratura del Servo error ad asse fermo.(UDM)

Serve a proteggere l' asse da eventuali derive o forzature meccaniche che tendano a spostarlo dalla posizione sotto controllo in posizione.

Il valore, espresso in UDM (Unità di Misura definita con STEP)., è modificabile da programma con l'istruzione **CAP/26:** Valore di default = **2.0**

ServoError_motion=15.0 Taratura del Servo error ad asse in movimento

(differenza istantanea tra la quota calcolata e quella reale).

Il valore, da definire in fase di messa a punto dell'Asse, è modificabile anche da Part program con l'istruzione **CAP/25:**

Di norma il valore è dell'ordine di qualche mm.

Viene però predisposto un valore di default =**15.0 UDM**.per permettere di muovere gli assi senza andare in servo error anche se non sono ancora tarati bene, proteggendoli comunque da possibili fughe per errori di cablaggio o altro.

Position_error=0.5

Definisce la **tolleranza di posizionamento** ed è

espresso in UDM (grandezze di programmazione= mm, gradi, ecc).Il valore, strettamente legato alle caratteristiche della macchina, di norma viene scelto non troppo stretto, perché la bontà del posizionamento non dipende da questo valore bensì da quella del servosistema e della meccanica.

E' comunque modificabile da Part program con l'istruzione **CAP/27:**

Qualora fosse richiesta una tolleranza molto stretta, ad esempio su un interasse di fori o altro, la tolleranza è modificabile da programma.

Valore di default = **0.5 UDM**.

Cycles_positioning=2.0

Tempo max (in sec.) di posizionamento entro la tolleranza, a partire dall'arrivo in posizione del calcolo. Scaduto il tempo (time-out), se l'asse non entra in tolleranza, nasce una segnalazione di errore.
Valore di default = **2.0**

Cycles_wait=0.02

Tempo (in sec.) durante il quale l'Asse deve rimanere nella tolleranza di posizionamento. Serve a non scambiare l'attraversamento della posizione finale con l'effettivo posizionamento in tolleranza. Va minimizzato perché è un tempo che si somma sempre a quello di posizionamento.
Valore di default = **0.02**

- VelStopCounter=1.0** Velocità reale dell'Asse (espressa in UDM/min.) sotto alla quale il sistema verifica la coerenza tra riferimento dato agli Azionamenti (espresso dal valore VoltStopCounter) e la velocità a cui l'Asse sta muovendo,. Se viene rilevata incoerenza (asse lento e comando alto), parte il tempo "Delay stop counter" al termine del quale nasce il relativo errore.
E' modificabile da Part program con l'istruzione **CAP/31:** Valore di default = **1.0**
- VoltStopCounter=0.5** Livello del Riferimento di Velocità, espresso in Volt, superato il quale il controllo "StopCounter". viene effettuato. Permette cioè di intervenire prima di aumentare eccessivamente la tensione al motore, con l'asse apparentemente impedito a muovere. E' modificabile da Part program con l'istruzione **CAP/30:** Valore di default = **0.5**
- DelayStopCounter=0.2** Finestra di tempo, espresso in sec. entro il quale l'asse deve uscire dalle condizioni di Stop Counter definite dai 2 valori **VelStopCounter** e **VoltStopCounter** su descritti
Permette di controllare, ad esempio, che non sia interrotto il canale di conta Encoder, oppure se l'asse, arrivato in prossimità della meta, non riesce a raggiungerla per duri meccanici, impedimenti o guasti. Questo controllo è abilitato dal valore della maschera **Kind_Axis_1. = 0x00080000**
Trascorso il tempo, viene dato il comando di STOP COUNTER ERROR
Il tempo è modificabile da Part program con l'istruzione **CAP/29:** Valore di default=**0.2**

TimeOutDriverOk=1.0

Tempo, in sec. entro il quale deve arrivare il segnale di DRIVER OK se in uno dei controlli ciclici effettuati automaticamente dal sistema, il segnale di Driver Ok viene trovato non attivo.

Il Timer è abilitato dal valore della maschera

Kind_Axis_1=0x00100000 Valore di default = **1.0**

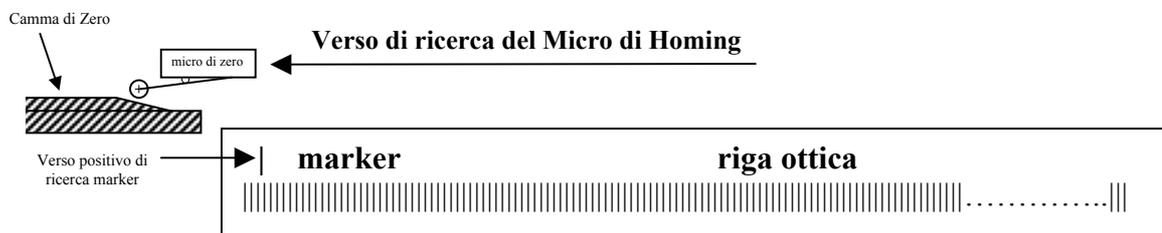
NOTA – Alcuni DRIVE non restituiscono il segnale di DRIVE OK. In questo caso il valore diella maschera sarà = 0

7.2.5 AZZERAMENTO ASSI, FINE CORSA E BACKLASH

Number_OMO_switch_port="HomeX" **Nome logico dell'Input** del segnale di Homing di quest'asse

FalseTrue_logic_OMO_switch=1 **Livello logico del segnale di Homing**
 (1 oppure zero. Se diverso da 1 oppure da zero il sistema non gestisce l'input)
 Valore di default = 1

Space_OMO=-10000 **Verso di ricerca del Micro di Homing**
 Viene caratterizzata anche la distanza massima percorribile dall'asse entro la quale il Micro deve sicuramente essere incontrato. Serve come protezione in caso di sregolazioni meccaniche o altro.
 Il valore della distanza è in UDM e il verso della ricerca è definito dal segno +/-
 Valore di default = -10000



Space_OMO_Micro_OFF = Spazio massimo che l'asse può compiere nel sentire il rilascio del Micro di zero .

Space_OMO_Marker = Spazio massimo che l'asse può compiere nel cercare la tacca del Marker .

Vel_OMO=1000 **Velocità di ricerca del Micro di Homing**
 Espressa in UDM/min.
 Valore di default =1000

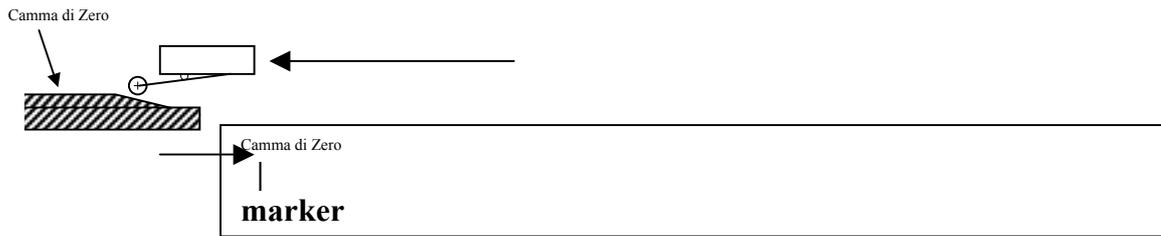
Vel_MARKER=500 **Velocità di ricerca del Marker (Index)**
 Espressa in UDM/min.
 Valore di default =500

Offset_axis=-0.0	Rappresenta la distanza fisica, espressa in UDM., tra la posizione dove si vuole porre lo "Zero asse" e il Marker di zero dell'Encoder ricercato sul micro di zero. Il valore può essere positivo (senza segno) o negativo (preceduto dal segno-) a seconda di dove si trova il micro di zero. Valore di default = -0.0
Positive_over_travel=0.0	Quota di fine corsa positiva rispetto allo Zero Asse (espressa in UDM.). Valore di default = 0.0
Negative_over_travel=0.0	Quota di fine corsa negativa rispetto allo Zero Asse (espressa in UDM.). Valore di default = 0.0 Se entrambi i valori di Positive e Negative over travel sono = 0 il controllo è disabilitato (gli Assi non hanno fine corsa)
Number_PositiveOverTravel_switch_port=""	Nome logico dell'Input del segnale del Micro di Fine Corsa positiva dell'asse da inserire tra le virgolette. (A volte, per poter muovere la macchina in Manuale prima di fare le origini, vengono predisposti i Micro di fine corsa a salvaguardia di errori di manovra).
FalseTrue_logic_PositiveOverTravel_switch=2	Livello logico del segnale di Fine Corsa Positiva. (1 oppure zero. Se diverso da 1 oppure da zero il sistema non gestisce l'input) Valore di default = 2 (perché di norma non esistono questi Micro)
Number_NegativeOverTravel_switch_port=""	Nome logico dell'Input del segnale del Micro di Fine Corsa negativa dell'asse da inserire tra le virgolette.
FalseTrue_logic_NegativeOverTravel_switch=2	Livello logico del segnale di Fine Corsa Negativa. Valore di default = 2 (perché di norma non esistono questi Micro)

Backlash=-0

Valore di compensazione del gioco meccanico, rilevato sperimentalmente ed espresso in UDM.

Il valore può essere positivo oppure negativo (preceduto dal segno meno). Il segno dovrà essere negativo se la direzione dell'asse nella fase di ricerca del marker di zero è positiva e viceversa (come mostrato in figura seguente) Valore di default = **-0**



7.2.6 GESTIONE DELLA FRENATURA ASSI

Number_BRAKE_switch_port=""

Nome logico dell'output del segnale di BRAKE di comando del freno di questo asse.

FalseTrue_logic_BRAKE_switch=2

Livello logico del segnale di BRAKE (1 oppure zero - se diverso da 1 oppure da zero il sistema non gestisce il freno)
Valore di default = **2** (perché di norma gli assi non sono dotati di freno)

Time_BRAKE_Lock=1

Tempo in sec. impiegato dal freno per agire. Per tutto questo tempo l'asse rimane sotto controllo. (caso tipico un asse verticale che deve essere tenuto in posizione se no può cadere) Questo tempo è conteggiato solo se il freno (BRAKE) è caratterizzato.
Valore di default = **1(sec.)**

Time_BRAKE_Enable=1

Tempo in sec., di anticipo della abilitazione (Enable) dell'asse prima di dare il comando di sbloccaggio del freno. (garanzia che l'asse sia sotto controllo prima di sfrenarlo) Questo tempo è conteggiato solo se il freno (BRAKE) è caratterizzato. Valore di default = **1(sec.)**

Time_BRAKE_Unlock=1

Tempo in sec. impiegato dal freno per sbloccarsi. Per tutto questo tempo l'asse rimane sotto controllo in posizione. Trascorso questo tempo il controllo è abilitato a muovere l'asse.
Questo tempo è conteggiato solo se il freno (BRAKE) è caratterizzato. Valore di default = **1(sec.)**

7.2.7 ANTICOLLISIONE

La taratura dei valori di anticollisione necessita di alcune spiegazioni ed esempi iniziali.

Ogni Asse che può meccanicamente collidere con altri deve essere protetto in entrambe le direzioni (destra e sinistra oppure sopra e sotto ecc.)

Chiameremo genericamente **Asse X** quello in taratura, **Axis 1** e **Axis 2** quelli con cui l'**Asse X** non deve collidere.

Per tarare l'anticollisione dell'Asse X, occorre pertanto dare al Sistema le seguenti informazioni:

- **Axis_1_name_anticollision=""** nome dell' Asse 1 scritto tra le virgolette
- **Distance_1_anticollision=0.0** distanza di sicurezza di anticollisione verso l'Asse 1
- **Kind_axis_1_anticollision=0** tipo (modalità) di anticollisione che si vuole attuare
- **Axis_2_name_anticollision=""** nome dell' Asse 2 scritto tra le virgolette
- **Distance_2_anticollision=0.0** distanza di sicurezza di anticollisione verso l'Asse 1
- **Kind_axis_2_anticollision=0** tipo (modalità) di anticollisione che si vuole attuare

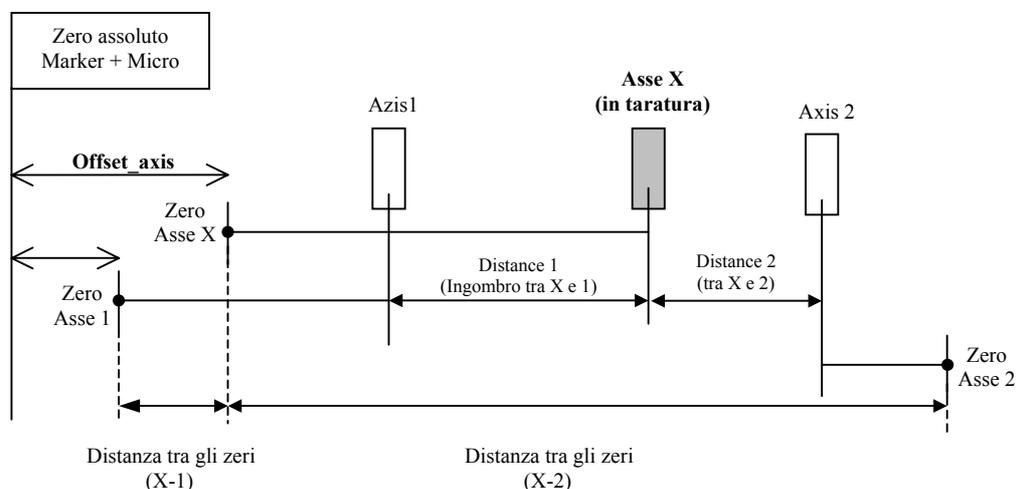
Dovranno essere compilati gli stessi campi anche per gli Assi 1 e 2, che andranno a loro volta difesi da collisione con l'Asse X e con eventuali altri assi (con i quali l'Asse X non può collidere perché si trova tra l'1 ed il 2).

Il valore di anticollisione, espressa in UDM sarà uguale a:

$$\text{Distance} = \text{Distanza tra gli zeri (dell'Asse X e dell'altro Asse)} - (\text{meno}) \text{ Ingombro tra i due Assi}$$

Nota – Se gli Assi hanno la stessa origine, il valore di Distance è uguale al valore dell'ingombro con segno – (meno)

La figura seguente mostra le distanze da considerare



Quando si tarerà l'Asse 1 dovrà essere impostato lo stesso valore di Distance tra l'attuale Asse 1 e l'Asse X, e così pure per l'Asse 2.

Tipo di anticollisione

Kind_axis_1_anticollision=0 tipo (modalità) di anticollisione che si vuole attuare

Il valore da impostare è dato dalla somma decimale dei valori assegnati alle grandezze **A, B, C, D, E** come da schema seguente:

A Codice di approccio tra gli Assi. Può assumere i seguenti valori:

Valori decimali della grandezza
A

1 AsseX concorde con Asse2 (Asse X si avvicina all'Asse 2 muovendo in positivo)



2 AsseX concorde con Asse1 (Asse X si avvicina all'Asse 1 muovendo in negativo)



3 AsseX discorde da Asse1 (Asse X si avvicina all'Asse 1 muovendo in positivo)



4 AsseX discorde da Asse1 (Asse X si avvicina all'Asse 1 muovendo in negativo)



Valori decimali delle grandezze
B, C, D, E



- B Modalità di accostamento tra gli Assi. Può assumere i seguenti valori:**
8 AsseX si accosta all'Asse 1 sino alla posizione di collisione (**Distance**)
0 AsseX non si accosta ma aspetta di avere l'intero campo libero
- C Modalità di inseguimento tra gli Assi. Può assumere i seguenti valori:**
16 AsseX insegue l'Asse 1 senza collidere (se 1 si allontana, X muove inseguendo)
0 AsseX non insegue l'Asse 1 ma aspetta di avere il campo libero
- D Modalità di calcolo di "Distance". Può assumere i seguenti valori:**
0 l'attuazione dell'anticollisione si basa sulle posizioni reali (Movimentazione)
- E Calcolo dell'anticollisione dopo OMO. Può assumere i seguenti valori:**
64 Il calcolo della "**Distance**" viene fatto dopo qualsiasi modalità di azzeramento assi **OMO**, anche se gli assi non sono stati azzerati sul Marker dell'Encoder
0 Il calcolo della "**Distance**" viene fatto solo dopo la modalità di azzeramento assi sul Marker dell'Encoder

Esempio di caratterizzazione:

Nel caso si voglia controllare l'anticollisione con:

A	Assi concordi con avvicinamento negativo	=	2
B	facendo accostare gli Assi	=	8
C	facendoli inseguire	=	0
D	calcolando "Distance" su quote reali	=	0
E	solo dopo aver fatto gli zeri solo sul Micro	=	64

Valore decimale da assegnare a **Kind_Axis_collision= 74**

7.2.8 ASSI SLAVE – GANTRY

ASSE SLAVE - Maschera Kind_Axis_1=0x00800000 (bit a Zero)

Axis_name_slave="" Nome dell'Asse con funzionamento "Slave" rispetto a questo che si sta caratterizzando

K_follower_slave=0.0 Coefficiente percentuale di inseguimento dell' Asse Slave.

GANTRY - Maschera Kind_Axis_1=0x00800000 (bit a Uno)

Axis_name_slave="" Nome dell'Asse con funzionamento "Slave" rispetto a questo che si sta caratterizzando

K_follower_slave=0.0 In funzionamento Gantry assume il significato di coefficiente di ripartizione degli errori di posizione istantanei tra i due assi Master e Slave. Se = 0 non c'è ripartizione. Se = 100, gli errori vengono ripartiti equamente tra gli assi. (Se ad esempio uno degli assi viene un po' rallentato da un duro meccanico o altro, mentre lo si accelera per riallinearsi, si rallenta un po' l'altro fino a minimizzazione degli errori). Ciò garantisce un miglior mantenimento istantaneo dell'allineamento degli assi.

VoltGantry=0.0 Differenza in Volt dei segnali di riferimento di Velocità ai 2 Drives di un asse GANTRY non rigido, dichiarato in Kind_Axis_1 = 0x01800000 (Allineamento gestito da due motori sincronizzati e non da un sistema meccanicamente rigido) Questo valore garantisce che, per mantenere gli assi allineati, non si superi un certo sforzo. La differenza tra i 2 segnali di riferimento non può mai superare il valore impostato. Valore di default = **0.0**

DeltaOffset_SlaveGantry=0.0

Distanza tra lo zero (marker) del Master rispetto allo zero (marker) dello Slave di un sistema GANTRY non rigido . La distanza, espressa in UDM, serve al sistema per tener conto della distanza tra i rispettivi Zeri per tenere gli assi allineati.

Valore di default = **0.0**

7.2.9 TARATURA MOTORI LINEARI

Premesse

- 1) – Per definire un Asse come “Lineare”, occorre settare la maschera:
Kind_axis_1= al valore **0x00000200** – Asse con sistema di fasatura (Motori lineari) con le regole spiegate nel paragrafo **CONCETTI PRELIMINARI SULLE MASCHERE**
- 2) - Quando si utilizzano motori lineari non esiste ovviamente alcun rapporto di trasmissione tra motore ed Asse.
Inoltre la riga ottica di rilevamento della posizione dell’Asse è fisicamente la stessa utilizzata dall’Azionamento per la gestione delle fasi del motore.
Pertanto la taratura del **KC** non potrà più essere definita dalla formula **KC_Parameter=3000/10*(20/1)** valida per motori rotanti, ma dalla nuova **KC_Parameter=velocità asse⁽¹⁾/volt**
Se ad esempio l’Asse deve muovere a 45 mt/min. con 10Volt, il valore di taratura sarà **KC_Parameter= 4500 (45.000mm./10)**
⁽¹⁾ La velocità è sempre espressa in UDM/min. Nella descrizione si è usata l’espressione “45mt./min perché è la più usuale, ma quella corretta sarebbe stata 45000UDM/min.

Fasatura Motore

A differenza degli Assi a motore rotante, gli assi con Motore lineare necessitano di norma di una “Ricerca della fase” per individuare la posizione relativa tra avvolgimenti e magneti. Questa operazione deve avvenire prima di qualsiasi movimento, compresa la ricerca del micro di Zero perché, non conoscendo la fase su detta, non è possibile far muovere il motore.

La ricerca della fase è un compito dell’Azionamento, per il quale sono stati predisposti i seguenti comandi in seguito descritti.

Number_PHASE_switch_port="" Nome dell’Output utilizzato per avviare il processo di “ricerca della fase” (per gli azionamenti che lo richiedono)

FalseTrue_logic_PHASE_switch=2 Definizione dello stato logico dell’Output di cui sopra (1 oppure 0) Valore di default = **2** perché non definito a priori.

Delay_PHASE=15 Tempo massimo di attesa che termini il processo di ricerca della fase. Valpre di default = **15** (sec.)

Esistono i due casi seguenti:

Azionamenti che restituiscono il segnale di DRIVER OK al termine del processo di ricerca fase. Se il processo termina prima e l’Azionamento segnala “fase eseguita OK” (tramite l’input di DRIVER OK), il sistema prende immediatamente sotto controllo l’Asse senza attendere lo scadere del tempo. Se invece il segnale di DRIVER OK non arriva entro il tempo, viene segnalato errore.

Azionamenti che non restituiscono il segnale di DRIVER OK. In questo caso il sistema attende comunque lo scadere del tempo e poi prende sotto controllo l’Asse. Se la fase non era stata ricercata entro il tempo, l’asse si muoverà male e nascerà un errore di movimentazione (collision, ecc.)

Volt_PHASE=0 **Attenzione** – Questa taratura serve solo per gli Azionamenti che richiedono di definire il valore della tensione da dare agli avvolgimenti del motore durante la ricerca della fase. Di norma gli azionamenti non richiedono questo valore, che deve in questi casi essere tassativamente = zero. Valore di default = **0**

7.3 TARATURA IN CORRENTE

7.3.1 PREDISPOSIZIONI

Selezione della modalità in Coppia, settando la Maschera al valore 0x00000001

Kind_axis_0=0x00000001 Maschera delle tarature del **Tipo di Asse** con asservimento in Coppia

7.3.2 COEFFICIENTI E SIGNIFICATI PER LA TARATURA IN CORRENTE

KP=20

Coefficiente di Proporzionalità tra il valore dell'Errore ed il valore della Corrente di comando.

Valore di default = **20**

KI=0

Coefficiente di Integrazione, che tende a ridurre le derive di posizione dell'Asse rispetto alla posizione calcolata.

Ad esempio, ad Asse fermo, recupera l'offset dell'Azionamento (che nel tempo può variare per cause diverse) È buona norma tararlo dopo aver azzerato l'offset dell'asse tramite il parametro **Volt_offset** e aver minimizzato l'offset sull'Azionamento.

Valore di default = **0**

VoltMaxIntegratedError=1

Livello di errore massimo accumulato fino al quale è abilitato l'intervento del coefficiente KI. Per valori superiori si ha comunque l'intervento di KI limitato al valore impostato. Il valore è espresso in Volt. Valore di default = **1**

KD=0

Coefficiente Derivativo – E' indispensabile per tenere sotto controllo l'Asse unitamente al KP. Tipicamente ha valore circa doppio di quello del KP.

Si noti che più l'azionamento è veloce nella risposta, più il valore di KD si avvicina a quello di KP

Valore di default = **0** (perché di norma si usa la taratura in Velocità).

Ricordarsi di porlo al valore =40 prima di iniziare le tarature in Coppia

KD2=0

Minimizza l'errore dovuto alla Accelerazione, tenendola quanto più possibile uguale a quella programmata.

Non usato in taratura di Velocità. Valore di default = **0**

Molt_VFF=0

Percentuale del valore del segnale di Velocità che viene trasmessa all'Azionamento dell'Asse. **Il valore di taratura sarà prossimo a zero** perché in questo caso non si controlla la Velocità. Valore di default = **0**

NON MODIFICARE

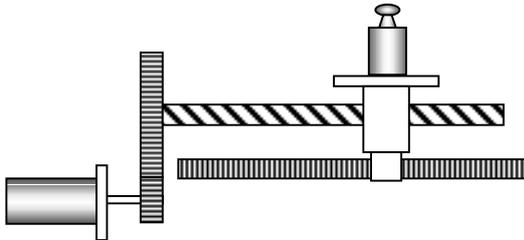
Molt_AFF=0

Percentuale del valore di Accelerazione che serve a minimizzare l'errore di inseguimento in fase di Accelerazione e Decelerazione (tanto come agisce normalmente il segnale di Velocity Feed Forward nelle tarature di Velocità)

Di norma non utilizzato in regolazione di VELOCITÀ.

Valore di default = **0**

KC_Parameter=3000/10*(20/1)



Coefficiente di Conversione, che definisce l'accelerazione dell'Asse (catena cinematica, elettrica e di misura comprese), espressa in UDM/sec², ottenuta applicando all'azionamento un riferimento di Corrente equivalente a un valore di tensione all'uscita del DAC pari a **1Volt**.

ATTENZIONE – Il valore di default **“3000/10*(20/1)”** vale per taratura dell'Asse in Velocità che di norma è la più utilizzata. Deve però essere cancellata la formula ed inserito il valore per taratura in Coppia

Un buon metodo per definire un valore approssimativo del KC da inserire è il seguente:

Il pilotaggio in Corrente in sintesi si differenzia da quello in Velocità per l'uso e interpretazione dei segnali di comando, essendo l'uscita del DAC (Volt) proporzionale alla Corrente data al Motore e non alla sua Velocità

- 1) Calcolare la **Forza** in NEWTON che IL Motore riesce ad applicare alla Massa da spostare, sostituendo i dati reali nella formula

$$\text{Forza (Newton)} = \frac{\text{Coppia Motore} * 2\pi * (N1/N2) * 1000}{\text{Passo Vite (in mm.)}}$$

- 2) Calcolare l'**Accelerazione**, sostituendo i dati reali nella formula

$$\text{Accelerazione (mm/sec.}^2\text{)} = \frac{\text{Forza (Newton)} * 10.000}{\text{Massa}}$$

- 3) Calcolare infine il **KC grossolano**, sostituendo i dati reali nella formula

$$\text{KC grossolano} = \frac{\text{Accelerazione} * \text{Guadagno Drive}}{\text{Corrente Nominale Motore}}$$

NOTE

- Per **KC grossolano** si intende un valore di KC di partenza, dell'ordine di grandezza di quello definitivo.
- Per **Guadagno Drive** si intende il rapporto (per il Drive dell'Asse in taratura) tra il valore della corrente data al Motore e il relativo comando (in tensione) in uscita dal DAC. ($I_{\text{Motore}} / V_{\text{DAC}}$). Questo valore si ottiene dalla taratura del Drive
- La **Corrente Nominale Motore** (I_n) è ricavabile direttamente dalla targhetta del Motore stesso

7.4 TARATURA STEP-MOTORS

Di seguito vengono elencati i parametri da modificare e che cambiano significato quando un asse viene tarato per pilotare Step Motors.

Kind_board_TransducerPosition=40

N° di riconoscimento del tipo di schede Assi utilizzate sulle quali risiede il canale Step Motor.. Il valore di default = **40** corrisponde alle schede PCI 2000.

Esistono i casi seguenti: Se il sistema prevede l'utilizzazione di Step Motor dotati di Encoder per la lettura degli steps effettuati (funzionamento ad anello chiuso) lasciare il valore **40** e scegliere il canale Encoder nella taratura del parametro:

Physical_number_TransducerPosition=1÷6

in quanto si utilizza uno dei canali Encoder degli Assi Analogici. Il relativo DAC può perciò essere usato solo come uscita analogica per un Asse Mandrino ad anello aperto o altro.

Se il sistema prevede l'utilizzazione di Step Motor senza Encoder (ad anello aperto), modificare il valore in:

1002 se il canale Step Motor è sulla Scheda 6 Assi Analogici PCI 2000

1003 se è SULLA Scheda 24+24 I/O

Board_number_TransducerPosition=0

Posizione fisica del Back-plane in cui si trova la scheda su cui risiede il canale di pilotaggio Step Motor di questo Asse (che si sta caratterizzando) Il valore, compreso tra 0 ed N (numero massimo di schede inseribili nell'Unità Base scelta) è predisposto in fase di collaudo. NON MODIFICARE

NOTA- Nelle Unità Base MICROSYS SERIE 4000 la numerazione da 0 ad N parte dalla scheda più lontana dalla CPU

Kind_board_DAC=40

Tipo di scheda con Uscita PULSE usata.

Inserire il valore:

1002 se il canale Step Motor è sulla Scheda 6
Assi Analogici PCI 2000

1003 se è sulla Scheda 24+24 I/O

Board_number_DAC=0

Posizione fisica del Back-plane in cui si trova la scheda su cui risiede l'Uscita PULSE di questo Asse (che si sta caratterizzando) Il valore, compreso tra 0 ed N (numero massimo di schede inseribili nell'Unità Base scelta) è predisposto in fase di collaudo. NON MODIFICARE

NOTA- *Nelle Unità Base MICROSYS della serie 4000 la numerazione da 0 ad N parte dalla scheda più lontana dalla CPU, come mostrato nella figura a lato.*

Address_port_DAC=0x00000000

NON UTILIZZATO

Physical_number_DAC=1÷3

Numero fisico dell'Uscita PULSE della scheda su cui risiede il pilotaggio dell'asse che si sta caratterizzando.

ATTENZIONE! Se l'Asse Step Motor è **senza Encoder**, il numero (1, 2, 3) scelto dovrà essere lo stesso indicato nel campo:

Physical_number_TransducerPosition=1÷3

Dovranno essere uguali tra loro anche i numeri indicati nei campi:

Board_number_TransducerPosition

Board_number_DAC

Di conseguenza dovranno essere tra loro uguali anche i numeri (**1002** oppure **1003**) indicati nei campi:

Kind_board_TransducerPosition

Kind_board_DAC

7.4.1 TARATURE PER ASSERVIMENTO DINAMICO ASSI STEPPER

- KP=20** **Coefficiente di Proporzionalità** tra il valore dell'Errore ed il valore della Velocità. (Errore = Valore della posizione calcolata – (meno) Valore della posizione reale dell'Asse)
Per Assi con Motori Step, KP agisce solo ad **Asse fermo**.
Valore di default = **20**
- KD=0** **Frequenza massima della fascia bassa (espressa in Hz)**
(e di Asservimento ad Asse Fermo).
Definisce la frequenza di pilotaggio del Motore Step sopra alla quale il sistema deve accelerare costantemente fino alla fascia alta (la cui frequenza minima è espressa dal coefficiente **KD2**). La zona di frequenze intermedia tra la fascia bassa e la fascia alta è quella in cui il Motore ha un buco di coppia.. (Valori da ricavare dai Data - Sheet del motore)
Se tarato =**0** il motore si muove senza controllo di fascia.. Valore di default = **0**
- KD2=0** **Frequenza minima della fascia alta (Vedi KD)**
Valore di default = **0**
- KS=0.0005** **Coefficiente di Smorzamento ad asse fermo.**
Il valore è espresso in UDM (nell'unità di misura dimensionale scelta con STEP per questo Asse) ed evita che si oscilli sulla quota finale per il mancato raggiungimento di un numero intero di passi.
La taratura tipica è un valore uguale allo spostamento fisico dell'Asse corrispondente a 1(uno) passo del motore.
Nota – Nei casi di motorizzazione ad anello chiuso (con Encoder), il valore della risoluzione espresso da **KS** può essere diverso dalla risoluzione dell'Encoder. Valore di default = **0.0005, da modificare come su detto, prima di iniziare le tarature dinamiche.**
- KM=0.01** **Valore (in UDM) dello spostamento dovuto a 1 passo del Motore.**
La taratura tipica è un valore uguale allo spostamento fisico dell'Asse.
Nota – Di norma **KM** è uguale a **KS**
Valore di default = **0.01, da modificare come su detto, prima di iniziare le tarature dinamiche.**

Volt_max=10	<p>Questo parametro, per i Motori Step, definisce il valore della frequenza massima, in Hz, della fascia superiore alla quale il motore è in grado di funzionare. (il valore minimo della fascia è espresso da KD2) Valore di default=10 da modificare secondo data sheet del motore.</p>
Volt_vel=10	<p>NON MODIFICARE IL VALORE. Se la direzione del Motore non è concorde a quella dell'Encoder, invertire il segno</p>
Step=-0.001	<p>Con Assi Stepper dotati di Encoder, questo parametro serve a calcolare lo spostamento effettivo di un asse (in UDM) per ogni step dell'Encoder. Lo spostamento può essere espresso direttamente col valore oppure con la formula di calcolo, come per gli Assi analogici. Con Assi senza Encoder, riportare il valore assegnato a KM. Il segno - (meno) permette di invertire il verso di conta se risulta contrario al verso del movimento. Valore di default = -0.001</p>
Vel_max=45000	<p>Velocità massima, (espressa in UDM / min.) da definire per l'asse in taratura. NOTA - La Velocità massima, rappresentata dalla frequenza massima a cui il motore può funzionare è definita dalla formula $\text{Vel_max} = \text{KM} * \text{Volt_max} * 60.$ Si noti che se viene scritto un valore superiore a quello massimo calcolato, il sistema procede comunque alla velocità massima consentita. In realtà pertanto questa taratura è necessaria se si vuole limitare la velocità (frequenza) massima di funzionamento. Valore di default = 45000 da modificare in base al valore tipico del Motore</p>
Number_MARKER_switch_port=""	<p>Nome logico dell'Input del segnale usato come Marker per i motori senza Encoder. (Input di norma realizzato tramite feritoia e fotocellula)</p>
FalseTrue_logic_MARKER_switch=2	<p>Livello logico del segnale del Marker. (1 oppure zero. Se diverso da 1 oppure da zero il sistema non gestisce l'input) Valore di default = 2</p>

Number_DIRECTION_switch_port=""

Nome logico dell'output del segnale di **DIRECTION** di quest'asse

FalseTrue_logic_DIRECTION_switch=1

NON MODIFICARE IL VALORE
Livello logico del segnale **DIRECTION**
(Se = **1** l'asse muove in un verso, mentre
se = **0** muove in verso opposto).
Valore di default = **1**

Number_STEP_switch_port=""

Nome logico dell'output del segnale **STEP** di quest'asse.

FalseTrue_logic_STEP_switch=0

NON MODIFICARE IL VALORE
Livello logico del segnale **STEP**.
Valore di default = **0**

Number_BOOST_switch_port=""

Nome logico dell'output del segnale di **BOOST** di quest'asse. Questo segnale si attiva automaticamente con due diverse modalità a seconda della taratura della maschera **Kind_axis_1=**

- Se la maschera ha il bit **Kind_axis_1=0x10000000** (settato=**1**) l'output commuta solo nella fase di accelerazione.
- Se **Kind_axis_1=0x00000000** (posto =**0**) l'output commuta all'inizio del movimento e rimane attivo per tutta la durata dello spostamento.

Questi segnali vengono utilizzati da alcuni tipi di azionamenti che alzano automaticamente la corrente in accelerazione oppure per tutto il movimento e la abbassano a motore fermo.

FalseTrue_logic_BOOST_switch=2

Livello logico del segnale di **BOOST**
(1 oppure zero. Se diverso da 1 oppure da zero il sistema non gestisce l'output)
Valore di default = **2**

7.5 TRACCIA SU COME PROCEDERE NELLA TARATURA ASSI

Per prima cosa bisogna controllare che l'asse abbia le quote con la direzione e l'unità di misura (UDM) voluti. Se il risultato non è corretto cambiare il valore di **Step=** , agendo sul segno per la direzione e sul valore per l'unità di misura. Questo controllo non dipende dal tipo di asservimento usato e si può fare con il loop di posizione aperto (asse non asservito).

Prima di chiudere il loop di posizione verificare che **Volt_max** non abbia un valore alto, allo scopo di evitare disastrose fughe dell'asse. I valori cautelativi dipendono dall'asse, ma di norma si usano valori dell'ordine di 1 volt.

Prima di chiudere il loop di posizione verificare che i parametri relativi al PID (**KP KI KD KD2 AFF VFF**) siano tutti a zero ad esclusione del **KP** per gli asservimenti in velocità e dei **KP KD** per gli asservimenti in coppia. Per l'asservimento in velocità si può iniziare con un **KP** di 20; per quelli in coppia con un **KP** di 10 e **KD** di 20.

Chiudendo il loop di posizione (dopo aver controllato che le quote siano corrette), si possono avere i seguenti casi:

1) L'asse rimane asservito. Ciò significa che **KC_Parameter** ha il segno giusto e il suo valore non è molto scorretto.

2) L'asse scappa e va in errore di asservimento. Ciò può significare che **KC_Parameter** ha il segno sbagliato o che il suo valore è molto scorretto. Cercare di correggere **KC_Parameter** abbassando eventualmente i parametri del PID, sino a che l'asse rimanga asservito.

notare che l'asse potrebbe sembrare asservito se l'errore di posizione rimane a zero; perciò è bene "perturbare l'asse" cercando di smuoverlo manualmente, affinché agisca la retroazione del PID. Un ausilio si può avere da "Taratura Assi" nell'ambiente "Sistema", dove è possibile vedere il comando dell'asse in taratura, che deve rimanere ad un valore stabile o oscillare reagendo alle perturbazioni date all'asse.

Quando l'asse rimane asservito, bisogna alzare **Volt_max** per poter dare dei comandi all'asservimento sufficienti a farlo muovere , anche se ancora limitati (es. 5 volt)

A questo punto le procedure divergono a seconda degli asservimenti.

7.5.1 Asservimento in velocità

Per questo tipo di asservimento bisogna assicurarsi di aver tarato bene **KC_Parameter**, che definisce il valore di trasformazione tra i volt del DAC e la corrispondente velocità ottenuta dall'asse. Questo parametro può essere ottenuto da valori noti o dedotto. Se l'asse in questione è un asse rotante un metodo è quello di far eseguire un Part-program, che utilizzando l'asse senza controllo (open loop) assegna una tensione all'asservimento e calcola la velocità. Il Part-program è del tipo mostrato di seguito:

```

;          ASSEGNA UNA TENSIONE ALL'ASSE A1
;          E NE VISUALIZZA LA VELOCITA'
;
;          -TSK/VEL_A1
;          -CPL/-1:A1
;
;          SDO/nome freno,1      ;Istruzione da usarsi con assi frenati
;          -LET/L1,0
AA        -KYB/VOLT A1,L1
;          -CAP/17:A1,L1
;          -JMP/AA
;          -END

PartProgram[VEL_A1]
AA        -RAV/1:L1,A1
;          -TMM/100
;          -RAV/1:L2,A1
;          -LET/L3,ABS(L2-L1)*600
;          -DIS/1:VELOCITA' A1,L3
;          -JMP/AA
;          -RET

```

Assegnando un volt ,la velocità definisce il valore di **KC_Parameter** (a parte il segno che dipende dalla direzione di rotazione del motore rispetto alla direzione di conta dell'encoder) L'uso di questo Part-program è condizionato dal tipo di asse, che deve essere rotante o comunque con un lungo spostamento. Inoltre l'asse non deve essere frenato, altrimenti all'apertura del loop di posizione, il freno entra in funzione, e se non viene sfrenato (con **SDO/nome freno,1**) si rischia di bruciare il motore. Naturalmente il Part-program va adattato con il nome dell'asse da tarare.

Un altro metodo consiste nel far eseguire un Part-program che fa muovere l'asse ad una certa velocità, e con il solo parametro KP utilizzato (vedi PID) viene controllato l'errore di inseguimento che deve essere:

$$\text{Errore} = (\text{Velocità asse} / 60) / \text{KP}$$

```

;          CONTROLLO ERRORE DI INSEGUIMENTO DELL'ASSE A1
;
;          -TSK/ERR_A1
;          -SFP/3000,500,500      ;Feed , Acc , Dec
AA        -MOV/A1,500      ;Muove asse A1 a 500
;          -MOV/A1,0        ;Muove asse A1 a 0
;          -JMP/AA
;          -END

PartProgram[ERR_A1]
AA        -RAV/14:L1,A1
;          -DIS/1:ERRORE A1,L1
;          -JMP/AA
;          -RET
    
```

Naturalmente il Part-program va adattato con il nome dell'asse da tarare e la velocità da assegnare all'asse, velocità che deve essere raggiunta e mantenuta perché il test funziona per assi in velocità costante.

Se l'errore si discosta molto da quello calcolato cambiare il **KC_Parameter** o agire sul guadagno dell'asservimento qualora sia possibile. la retroazione del PID.

Un ausilio si può avere da "Taratura Assi" nell'ambiente "Sistema", dove è possibile vedere il comando e l'errore di inseguimento dell'asse che si sta muovendo.

Nell'esempio precedente ,considerando una velocità di 3000 UDM/min. e un KP di 10 il valore da ottenersi è: $(3000/60)/10 = 5$

7.5.2 Asservimento in coppia

Anche per questo tipo di asservimento bisogna assicurarsi di aver tarato bene **KC_Parameter**, che definisce il valore di trasformazione tra i volt del DAC e la corrispondente accelerazione ottenuta dall'asse. Questo parametro può essere ottenuto da valori noti o dedotto. Inoltre in questi asservimenti è necessario che siano presenti i parametri **KP** e **KD**, altrimenti l'asse non si può asservire (oscilla con il solo **KP**). Per dedurre il valore (in modulo) di **KC_Parameter** si può procedere nei seguenti modi:

Questo metodo si può usare con assi lineari. E' necessario far eseguire un Part-program che assegna una tensione in Volt al convertitore e misurare con un dinamometro la forza in Kg che sviluppa il motore sull'asse. Se si conosce la massa in Kg dell'asse il valore da assegnare sarà **KC_Parameter** = (10000 * Forza / Massa) / Volt

Il Part-program da usare è :

```

;           ASSEGNA UNA TENSIONE ALL'ASSE A1
;
;           -CPL/-1:A1
;           SDO/nome freno,1           ;Istruzione da usarsi con assi frenati
;           -LET/L1,0
AA          -KYB/VOLT A1,L1
;           -CAP/17:A1,L1
;           -JMP/AA
;           -END

```

Naturalmente il Part-program va adattato con il nome dell'asse da tarare e devono essere impostati valori bassi di tensione (+0.1) in crescendo sino a che l'asse si muova perché altrimenti l'asse acquisterebbe un alta velocità anche con valori piccoli di tensione (viene originata un'accelerazione e non una velocità).

Un altro metodo consiste nel far eseguire un Part-program che fa muovere l'asse con una certa accelerazione.

Con l'ausilio di "Taratura Assi" nell'ambiente "Sistema", è possibile vedere il comando dato con l'accelerazione programmata.

KC_Parameter = (Accelerazione) / Volt comando

```

;           CONTROLLO ERRORE DI INSEGUIMENTO DELL'ASSE A1
;
;           -SFP/3000,500,500           ;Feed , Acc , Dec
AA          -MOV/A1,500           ;Muove asse A1 a 500
;           -MOV/A1,0               ;Muove asse A1 a 0
;           -JMP/AA
;           -END

```

7.5.3 Considerazioni finali

Quando il **KC_Parameter** è aggiustato bisogna ritoccare anche gli altri valori del PID ricordandosi che valori alti rendono l'asse instabile, ma con valori bassi gli errori di posizione sono più alti. Perciò bisogna cercare valori abbastanza alti, ma non tanto da rendere instabile l'asse.

Con gli asservimenti in velocità è indispensabile **KP**, importante **VFF** e migliorativo **KI**

Con gli asservimenti in coppia sono indispensabili **KP** e **KD**, importanti **KI** e **AFF**. Il rapporto **KD / KP** può variare da 1 a 5; di solito è intorno a 2. Un metodo per cercare i valori da impostare è quello di far eseguire il Part-program seguente, variando con "editor" nell'ambiente "automazione" i valori da testare. La bontà del movimento e il tempo impiegato danno un'idea del corretto impiego dei parametri

```

; ***** POSIZIONAMENTI ASSE A1 *****
;
;           -CAP/0:A1,32768      ;Position Error abilitata
;           -CAP/27:A1,0.008     ;Valore Position error
;
;           -CAP/15:A1,35        ;KP
;           -CAP/20:A1,200       ;KI
;           -CAP/19:A1,80        ;KD
;           -CAP/23:A1,0         ;KD2
;           -CAP/16:A1,0         ;VFF
;           -CAP/24:A1,100       ;AFF
;
;           -SFP/120000,25000,15000
;
;           -SET/L1,110,L2,2.5,L3,100
;           L1 = POSIZIONE DI PARTENZA
;           L2 = INCREMENTO DI POSIZIONE
;           L3 = NUMERO MISURE
;
;           -MOV/A1,L1
;           -KYB/INIZIO ANDATA
;           -SET/L10,0
;           -SWA/1
LOOP      -MOV/A1,L1
;           -ADD/L1,L2
;           -ADD/L10,1
;           -JLT/L10,L3,LOOP
;           -RWA/L1,1
;           -DIS/1:TEMPO = ,L1
;           -DIS/2:TEMPO PER POSIZIONE = ,L1/L10
;           -END
    
```


CAPITOLO 8

8 Visione

Il pacchetto VisAlgo lavora tramite richieste di funzioni d'analisi dell'immagine che effettuano operazioni di:

- Filtro e manipolazione dell'immagine
- Analisi luce e estrazioni automatiche di soglie
- Blobs su rettangoli d'interesse
- Estrazioni di contorni su intersezione di figure su rettangoli d'interesse
- Estrazioni di bordi
- Misurazione di contorni
- Estrazioni di enti sui contorni
- Misurazione sugli enti

Per attivare le funzionalità di visione con telecamere reali è necessario configurare il file "Vision.ini", con i seguenti parametri.

[ParametriVisione]

NumeroTelecamere=0

Progressiva=0

SchedaDoppia=0

APPENDICE A

A. Elenco Errori

Errore	1	Funzionalità ancora attiva
Errore	2	Nome asse/mandrino/IO non corretto
Errore	3	OVERFLOW Tabella nomi assi
Errore	4	OVERFLOW Tabella assi
Errore	5	Tipo asse incongruente
Errore	6	MANCA ASSE FISICO PER SISTEMA VIRTUALE
Errore	7	ASSE HELP NON USABILE (CYL_TRASLANTE)
Errore	8	OVERFLOW Tabella JOB di movimentazione
Errore	9	JOB di movimentazione NON DEFINITO
Errore	10	OVERFLOW AREA movimentazione
Errore	11	PARAMETRI Start continuo ERRATI
Errore	16	PARAMETRI Movimentazione ERRATI
Errore	17	TROPPE TPE CONTEMPORANEE
Errore	18	ASSE GIA' USATO IN ALTRI JOBS
Errore	19	ASSE FUORI CAMPO DI LAVORO
Errore	20	RAGGIO FINALE <> RAGGIO INIZIALE
Errore	21	JOB NON DEFINIBILE
Errore	22	TROPPI ASSI (MAX 8)
Errore	23	NON CI SONO PIU' TIMER
Errore	24	ASSI IN MOVIMENTO
Errore	25	STATO INCORRETTO
Errore	26	ERRORE DI SISTEMA NELL'USO DEL JOB
Errore	27	ASSE FRENATO
Errore	28	CLOSE O OPEN POSITION LOOP INUTILE
Errore	29	MANCA AREA SLAVE
Errore	30	CAMPIONATURA NON ATTIVA
Errore	31	TROPPI PARAMETRI
Errore	32	MANCA ASSERVIMENTO ATTIVO
Errore	33	TROPPE PIASTRE I/O CONFIGURATE
Errore	34	PIASTRA I/O SBAGLIATA
Errore	35	BIT NON SU PIASTRA
Errore	36	OVERFLOW DEL BIT LOGICO
Errore	37	BIT I/O SBAGLIATO
Errore	38	TROPPE PIASTRE ANALOGICHE CONFIGURATE
Errore	39	PIASTRA ANALOGICA SBAGLIATA
Errore	40	CANALE ANALOGICO NON SU PIASTRA

Errore	41	OVERFLOW DEL CANALE ANALOGICO	(LOGICO)
Errore	42	CANALE ANALOGICO SBAGLIATO	
Errore	43	SCELTA SCONOSCIUTA	
Errore	44	TEMPO SCADUTO (Time OUT)	
Errore	45	INTERPOLAZIONE SPLINE ERRATA	
Errore	47	AZZERAMENTO ASSE NON FATTO (MANCA HOMING)	
Errore	48	OVERFLOW TABLE SPINDLE	
Errore	49	MANDRINO IN ATTIVITA' SAMPLE	
Errore	50	MANDRINO IN ROTAZIONE	
Errore	51	ASSERVIMENTO DISABILITATO	
Errore	52	QUOTA CORREZIONE ERRATA (UGUALE)	
Errore	53	OVERFLOW AREA CORREZIONE	
Errore	54	BUFFER CIRCOLARE NON ATTIVO	
Errore	55	OVERFLOW COUNTER NELL'ASSE STEPPER	
Errore	56	COUNTER NON DEFINITO NELL'ASSE STEPPER	
Errore	58	MOVIMENTO IN DEADLOCK	
Errore	59	TASK IN ABORT (TKM)	
Errore	60	OVERFLOW TKM	
Errore	61	ASSE SU MICRO DI OLTRE CORSA POSITIVO	
Errore	62	ASSE SU MICRO DI OLTRE CORSA NEGATIVO	
Errore	63	FASE NON FINITA	
Errore	64	TASK CON TKM ANCORA ATTIVA	
Errore	65	GRM SU GRUPPO ANCORA ATTIVO	
Errore	66	TABELLA ELECTRONIC CAM NON DEFINITA	
Errore	67	OVERFLOW TABLE ELECTRONIC CAM	
Errore	68	DIREZIONE SBAGLIATA IN ELECTRONIC CAM	
Errore	69	HEC BUFFER PIENO	
Errore	70	HEC VELOCITA' MASTER ECCESSIVA	
Errore	71	TABELLA HEC NON CORRETTA	
Errore	72	FUORI SERVIZIO	
Errore	73	SPD BLOCCATO DA GRM/1:..	
Errore	74	TABELLA HEC IN SAMPLE (NON CANCELLABILE)	
Errore	75	Movimento con ABC (DEFAULT LOCK)	
Errore	1001	HANDLE INCORRETTO	
Errore	1002	DRIVER OCCUPATO	
Errore	1003	INIZIALIZZAZIONE ERRATO	
Errore	1004	ANOMALIA IN FASE DI AZZERAMENTO	
Errore	1006	TROPPI HANDLER DI MOVIMENTAZIONE ATTIVI	
Errore Driver Fault	0x5	Configuration error - Decision Table	
	0x7	Calibration error	
	0x10E	Fiber not connected to digital driver	
	0x111	Digital driver ass'h't hardware enable	

0x113 Fiber rx Frame error
0x114 Fiber rx Parity error
0x116 Zero command refused for movement
0x117 Lost zero axis

APPENDICE B

B. Esempio “Sistema.txt”

[ParametriGenerali]

Campionatura=1000
NumeriDecimali=3
NumeriCifre=10
TolleranzaCerchio=1000
KN=-1.0
NUMBER_SAMPLE_SWITCH_PORT=""
FALSETRUE_LOGIC_SAMPLE_SWITCH=1
SegnalePotenza=""
RitardoPotenza=5.0
NumeroLocali=32
TickMicro=1
GeneralStatus=0
TickCounter=1
FALSETRUE_LOGIC_SegnalePotenza=1
NumeriDecimaliDRT=3
NumeriCifreDRT=7
NumeriDecimaliFILE=4
NumeriCifreFILE=10
NumeroByteReadFieldBus=8
NumeroByteWriteFieldBus=8
Time_collision_latency=0.2
FileCorrezione=Correz.txt
IO_PLC_management=0
NumeroIstruzioniMax=128
PartProgramG89=G89.PP
VolantinoECS=1
GraficaCNC=1
ModuloFieldBusHilscher=1
ModuloFieldBusSofting=0
FlagFastWaitIO=0
GlobaleNumeroAllarmi=9999
TempoLatenzaErrori=1
FlagHoldG8183=1
ECSVolantinoSuRapido=0

WaitSEC=0.03
TempoScansione=0.010
SeGlobaleErroreNoDISM1=1
GlobaleBloccoMovimentazione=0

[IO]

```
i=IO_NAME(1 , "ISA_OUT1" ) // A1 di 1
b={
Description=""
Index_system=0
FalseTrue_logic=0
Kind_sensor="DIGITAL_OUTPUT"
Kind_board=9
Rack_number=0
Address_port=0
Physical_number=1
e=}
```

```
i=IO_NAME( 401 , "EnZ" )
b={
Description="Enable Z"
Index_system=0
FalseTrue_logic=0
Kind_sensor="DIGITAL_OUTPUT"
Kind_board=20
Rack_number=0
Address_port=0
Physical_number=1
e=}
```

```
i=IO_NAME( 101 , "CAN_O1" )
b={
Description=""
Index_system=0
FalseTrue_logic=0
Kind_sensor="DIGITAL_OUTPUT"
Kind_board=5
Rack_number=0
Address_port=0
Physical_number=1
e=}
```

```
i=IO_NAME( 1 , "ISA_INP1" )    // A2 di 1
b={
Description=""
Index_system=1
FalseTrue_logic=0
Kind_sensor="DIGITAL_INPUT"
Kind_board=9
Rack_number=0
Address_port=3
Physical_number=1
e=}
```

```
i=IO_NAME( 401 , "HomZ011" )
b={
Description="Home Z St01 Un011"
Index_system=1
FalseTrue_logic=0
Kind_sensor="DIGITAL_INPUT"
Kind_board=20
Rack_number=0
Address_port=0
Physical_number=1
e=}
```

```
i=IO_NAME( 402 , "DrOKZ011" )
b={
Description="Driver OK Z St01 Un011"
Index_system=1
FalseTrue_logic=0
Kind_sensor="DIGITAL_INPUT"
Kind_board=20
Rack_number=0
Address_port=0
Physical_number=9
e=}
```

```
[Z]
i=PHYSICAL_AXIS_NAME( 1,"Z" )
b={
Description="A9 Asse Z Fori_8.2_7 St4 Un041"
Kind_axis="ARM"
Kind_axis_0=0x00004000
Kind_axis_1=0x08070000
Kind_board_TransducerPosition=40
Board_number_TransducerPosition=1
Address_port_TransducerPosition=0x00000000
Physical_number_TransducerPosition=1
Kind_board_DAC=40
Board_number_DAC=1
Address_port_DAC=0x00000000
Physical_number_DAC=1
Level_TCH=0
NotifyEnabled=0
KP=40.000
KI=1000.000
KD=0.000
KD2=0.000
VoltMaxIntegratedError=2.500
K_DAC_1Volt=819.2 // 3276.7
Molt_VFF=100.000
Molt_AFF=0.000
KS=0
Volt_max=10
Volt_vel=-9
Volt_offset=0.0
Step=-1/16384*2.5
Backslash=0.0
Revolution=0.0
Vel_max=7500
Acc_max=1000.0
Dec_max=1000.0
ServoError_standby=2.0
ServoError_motion=5.0
Position_error=0.05
Cycles_positioning=2
Cycles_wait=0.01
Offset_axis=18
Positive_over_travel=0
Negative_over_travel=0
DelayStopCounter=0.2
```

```
VoltStopCounter=0.5
VelStopCounter=1.0
TimeOutDriverOk=1.0
Time_acc_nla=-1
Time_dec_nla=-1
Axis_1_name_anticollision=""
Distance_1_anticollision=0.0
Kind_axis_1_anticollision=0
Axis_2_name_anticollision=""
Distance_2_anticollision=0.0
Kind_axis_2_anticollision=0
Axis_name_slave=""
K_follower_slave=0.0
VoltGantry=0.0
DeltaOffset_SlaveGantry=0.0
Number_SERVO_switch_port="AbilZ041"
FalseTrue_logic_SERVO_switch=1
Number_DRVOK_switch_port="DrOKZ041"
FalseTrue_logic_DRVOK_switch=2
Number_PHASE_switch_port=""
FalseTrue_logic_PHASE_switch=0
Delay_PHASE=0
Volt_PHASE=0
Number_OMO_switch_port="HomZ041"
FalseTrue_logic_OMO_switch=1
Space_OMO=500.0
Vel_OMO=1000.0
Vel_MARKER=500.0
Number_MARKER_switch_port=""
FalseTrue_logic_MARKER_switch=0
Number_DIRECTION_switch_port=""
FalseTrue_logic_DIRECTION_switch=0
Number_BRAKE_switch_port="A210"
FalseTrue_logic_BRAKE_switch=0
Time_BRAKE_Enable=0.1
Time_BRAKE_Lock=0.1
Time_BRAKE_Unlock=0.1
Number_STEP_switch_port=""
FalseTrue_logic_STEP_switch=0
e=}
```