Volumen

3

AxesBrain

Configuración y caracterización

Configuración y caracterización

AxesBrain

Configuración y caracterización

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

Informaciones legales

Las informaciones con este documento, inclusos URL y las otras referencias sobre el situado Internet, pueden cambiar sin ninguna advertencia.
A menos que específica anotación, las referencias a compañías, organizaciones, producidas, personas y acontecimientos son ficticios y no asociáis con reales compañías, organizaciones, producidas, personas y acontecimientos.
EL AB&T Tecnologías Informaticheä puede registrar, despedir, solicitar el copyright o marcas y reivindicar la propiedad intelectual a todos los argumentos te tratada en este documento.
Sin limitar los derechos bajo copyright, ninguna parte de este documento puede ser reproducida, o modificada o transmitida bajo cada forma o medio, electrónico, meccanico,per fotocopiar, para registrar, u otro, sin la expresa autorización dell Informaticheä.
Salvo que por acuerdos escritos con el AB&T Tecnologías Informaticheä el suministro de este documento no autoriza a nadie a la grabación, a dar licencias, a solicitar el copyright o marcas y a reivindicar la propiedad intelectual a los argumentos tratada en este documento.
AxesBrainä, VisAlgoä, ScadaMERCURIOä son marcas registradas
©AxesBrain, ©VisAlgo tutelado por copyright
ActiveX, DirectX, JScript, Microsoft, Microsoft Press, MS-DOS, Visual Basic, Visual C++, Win32, Win32s, Windows, WDM, Windows NT, Windows 2000 y Windows Yo son producidos u o marcas registradas por el Microsoft Corporation en los Estados Unidos e/o en los otros Países.
Los nombres de compañías y productos emplazados en este documento pueden ser marcas registradas por sus propietarios.

ÍNDICE DE LOS ARGUMENTOS

1 INTRODUCCIÓN....1-1

1.1 COMO UTILIZAR ESTE MANUAL....1-1
1.2 SISTEMA TXT....1-2

2 CONFIGURACIÓN....2-1

2.1 PARAMETROS GENERALES....2-1
2.1.1 Parámetros General Tablero sinóptico8....2-7
2.2 I/O SEÑALA, DIGITAL Y ANALÓGICOS ,....2-12
2.2.1 I/O digitales señalan....2-13
2.2.2 Señales Analógicos I/O....2-19
2.2.3 Parámetros de los I/O digitales y analógicos - Tableros sinópticos....2-21
2.3 ASES Y MANDRILES....2-30
2.3.1 Ases....2-30
2.3.2 Mandriles....2-76
2.3.3 Ases virtuales....2-84
2.4 PARAMETROS ESPECIFICOS DE LOS ENTORNOS....2-94
2.5 PREDEFINIZIONI ISO....2-95

3 CARACTERIZACIÓN....3-112

3.1 PID....3-113
3.2 ANTICOLLISIONE....3-116
3.3 GANTRY....3-119
3.4 HOMING....3-123
3.5 DETERMINADOS CAMPEONATOS....3-130
3.6 READ AXES PARAMETERS....3-132
3.7 CHANGE AXES PARAMETERS....3-137
3.8 ASES ROLLOWER....3-150
3.9 OVER TRAVEL....3-151
3.10 ASES SOLO SOMETIDOS EN MOVIMIENTO....3-152

4 CONFIGURACIÓN NOMBRE ASES POR INTERFAZ MANUAL....4-153

5 CICLOS SIEMPRE ACTIVOS....5-1

6 CORRECCIÓN ASES....6-1

6.1 LINEAL....6-2
6.2 PARRILLA....6-4

7 CALIBRADOS ASESPID, PROPORCIONAL, COMPLEMENTARIO, DERIVATIVO....7-5

7.1 DATOS PERSONALES ES PREMISAS....7-5
7.2 CARACTERIZACIÓN ASES ANALÓGICOS EN VELOCIDAD....7-7

7.2.1 CALIBRADOS GENERALES DE SISTEMA RELATIVOS ATUTTI ÉL ASES 7-7
7.2.2 CARACTERIZACIÓN ASES FÍSICOS - PREDISPOSICIONES 7-7
7.2.3 VALORES ES COEFICIENTES POR SUMISIÓN DINÁMICOS ASES 7-14
7.2.4 CONTROL Y ERRORES ASES 7-20
7.2.5 AZZERAMENTO ASSI, FINE CORSA E BACKLASH 7-24
7.2.6 GESTIÓN DEL FRENATURA ASES 7-27
7.2.7 ANTICOLLISIONE 7-28
7.2.8 ASES ESLAVAS-GANTRY 7-31
7.2.9 CALIBRADO MOTORES LINEALES 7-33
7.3 CALIBRADO EN CORRIENTE 7-35
7.3.1 PREDISPOSICIONES 7-35
7.3.2 COEFICIENTES ES SENTIDOS POR EL CALIBRADO EN COR 7-35
7.4 CALIBRADO STEP-MOTORS 7-38
7.4.1 CALIBRADOS POR SUMISIÓN DINÁMICOS ASES STEPPER 7-41
7.5 TRAZA SOBRE COMO PROCEDER EN EL CALIBRADO ASES 7-2
7.5.1 Sumisión en velocidad 7-3
7.5.2 Sumisión en pareja 7-5
7.5.3 Consideraciones finales 7-6

VISION 7-7

EJEMPLO “SISTEMA.TXT” 3

CAPÍTULO 1

1 INTRODUCCIÓN

1.1 Como utilizar este manual

Este manual describe cómo utilizar las funciones dedicadas a la caracterización del sistema AxesBrain.
Su objetivo es proveerles especialmente a los programadores una rápida ayuda y soporte por función que raramente son utilizadas y proveer ofrecer una rápida guía en mérito.
El manual tiene, sin embargo, breves textos descriptivos fácilmente comprensibles.
Os familiarizáis primera con los símbolos representados en consecuencia de para interpretar de ello el sentido en las ocasiones de su empleo en los capítulos siguientes.

1.2 Sistema TXT

El producto "AxesBrain" se configura por el expediente ***"sistema.txt"*** *que es colocado en el sottodirectory "DAT" del directorio del entorno "AxesBrainStudio"*

Para utilizar los servicios puestos a disposición de AxesBrain se tienen que configurar los parámetros de los ases de los I/O digitales y analógicos caracterizando los valores de los ases de los mandriles y los I/O.

Y útil primera configurar el sistema en términos de ases y mandriles, de I/O sea dedaleras que analógicos, luego caracterizarlos según las específicas solicitudes.

El sistema provee un "***sistema.ini"*** con la configuración dotada, que deberá ser caracterizada según las exigencias específicas (nombres de las señales) nombre de los ases, parámetros velocidad etcétera.).

El presente manual quiere proveer las nociones esenciales por la caracterización del producto **AxesBrain.**

En particular ello es tan organizado:

El **Capítulo 2 Configuración** describe los parámetros que configurar para poder administrar los recursos del sistema.

El **Capítulo 3 Caracterización** describe los parámetros que configurar para caracterizar el sistema.

El **Capítulo 4 Manual** provee los rudimentos esenciales sobre como configurar los ases y los relativos grupos presentes en la aplicación deseada.

El **Capítulo 5 Ciclos siempre activos** provee las informaciones necesarias a la configuración de los cordones de lanzamiento por los parámetros PLC y AUTOMACIÓN.

El **Capítulo 6 Corrección ases** provee las informaciones necesarias a la configuración de los parámetros de corrección Lineal y Parrilla.

El **Capítulo 7 Calibrados ases** ilustra los conceptos base inherente al calibrado de los ases y mandriles.

El **Capítulo 8 Visión** ilustra los conceptos base inherente el paquete VisAlgo con función de análisis de imagen.

L ' **Apéndice A.** es una lista sottoforma de lista de todas las alarmas que pueden ser engendrado por el sistema en fase de programación y caracterización.

L ' **Apéndice B** describe un ejemplo de expediente *sistema.txt*

CAPÍTULO 2

2 Configuración

En el Expediente "Sistema.txt" están presentes todos los parámetros de caracterizar subdividido en:

Parámetros Generales-Informaciones que conciernen toda la aplicación, cuales a esempio:Campionatura, n. de decimales, etc
Parámetros de los I/O digitales y analógicos-Informaciones sobre el tipo de I/O, nombres asignados a las señales, n. del canal físico utilizado etcétera
Parámetros relativos a los ases-Informaciones relativas a la individuación del tipo de ficha y eje, analógico, sercos, etcétera, valores de calibrado PID etc
Parámetros relativos a los ases virtuales-Informaciones relativas a la caracterización de los ases que necesitan una transformación de las coordenadas en el plan, Ases SCARA, Cilíndricos etcétera,

Parámetros Mandriles-Informaciones relativas al canal de pilotaje utilizado, ej. sobre ficha ases analógicos, calle KAN, etcétera, N. de vueltas, aceleración, etcétera
Parámetros específicos de los Entornos-Informaciones sobre las condiciones de lanzamiento del part program de automación
Definición de los "procesos ISO" - Predefinizioni ISO

2.1 Parámetros generales

Los parámetros generales son los primeros contenidos en el expediente "Sistema.TXT" y conciernen de los registros interiores utilizados por "AxesBrain."
Son indicados bajo la voz: **[ParametriGenerali]**
En seguida hay un ejemplo para poder definir de ello el sentido:

EJEMPLO

```
[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

Veámoslos en detalle:

Campionatura = 500 valor del muestreo ases en microsegundos
NumeriDecimali = 3 valor de estándard de decimales usado por el sistema
NumeriCifre = 10 valor de estándard de cifras usado por el sistema
TolleranzaCerchio = 1 el valor indica el máximo error entre rayo inicial del círculo y aquel final.
KN = valor que de estándard es –1 en este caso los valores interiores del sistema él autotarano, se determina de otro modo según el factor indicado.

NUMBER_SAMPLE_SWITCH_PORT " si indicara el nombre de la señal de salida es activado, ves **FALSETRUE_LOGIC_SAMPLE_SWITCH** si 1 o 0, cuando el sistema es activo, (potencia activa con capacitación de los ases.
FALSETRUE_LOGIC_SAMPLE_SWITCH = 1 activo a uno, 0 activo a cero

SegnalePotenza = " si indicara como referencia a una señal de intput digitale,la potencia de la máquina, capacitación de los ases, será dada por el sistema después de un tiempo indicado en segundo por el valor siguiente
RitardoPotenza = 5.0 valor de retraso como señal automática de potencia

NumeroLocali = 32 número de VARIABLES LOCALES
juntado a cada part program de automación (- DIM para cambiar de modo local este parámetro,

TickMicro = Número de Tick por el microcampionatura
, sólo el PID),normalmente 1, si el sistema necesita un muestreo más espeso aumentar de algunas unidades.

GeneralStatus = Máscara general de sistema a bits
;0x00000001 = Coche activación de USB MAVEL
;0x00000002 = Desactivación del WATCH DOG con USB MAVEL
;0x00000004 = Gestión de DEADLOCK sin ERROR, silente,
;0x00000008 = Gestión CFR a nivel (modo nuevo)
;0x00000010 = no viene resettato COMPENSACIÓN RELATIVA
TickCounter = Número de tick del muestreo la gestión
de los motor STEP usando las señales I/O

NumeriDecimaliDRT = Número decimal en la visualización de los datos activada por la instrucción DRT

NumeriCifreDRT = Número cifras en la visualización de los datos activada por la instrucción DRT

NumeriDecimaliFILE = Número decimal en la visualización de los datos activada por la instrucción FWR

NumeriCifreFILE = Número decimal en la visualización de los datos activada por la instrucción FWR

NumeroByteReadFieldBus = Número de byte leído en la gestión del BÚS de campo

NumeroByteWriteFieldBus = Número de byte escrito en la gestión del BÚS de campo

FileCorrezione =Nombre expediente de corrección ases sea lineal que de cuadratura

Time_collision_latency = Tiempo en segundo de retraso por antes de inhabilitar los accionamientos, el mando de disabilitazione pone la referencia de los accionamientos a zero,effettua el time_collision_latency, saca las capacitaciones.

IO_PLC_management = Flag por la capacitación de la gestión de la adquisición de todos los YO Digitales y Analógicos con la gestión de las tres señales por la Página PLC señala.

NumeroIstruzioniMax = Para evitar que una secuencia de instrucciones lógicas y de control condicionen el tiempo dedicado por la UCP, viene indicado el número de instrucciones en el que si el task no viene suspendido, el sistema fuerza una pausa de **TempoScansione** milisegundos por el part program en examen.
Estándard = 64

TempoScansione = Tiempo él suspensión corchete automático por ejecución continua del número de instrucciones indicada **NumeroIstruzioniMax,** sin suspensión.
Estándard = 0.050

PartProgramG89 = Nombre del part program que es ejecutado con las modalidades de los ciclos fijos.
Estándard = G89.pp

VolantinoECS = Activa la octavilla de cónsul ECS

GraficaCNC = Activa la señal de los entes ejecutados

ModuloFieldBusHilscher = Activa la ficha Hilscher por la gestión del FieldBus

ModuloFieldBusTCP-IO = Activa la gestión del FieldBus TCP-IO

ModuloFieldBusSofting = Activa la ficha Softing por la gestión del FieldBus Can

FlagFastWaitIO = Valor que activa la modalidad fast por las instrucciones "WDI"

activas. Estándard = 0
attive. Default = 0

GlobaleNumeroAllarmi = El número indicado si mucho de cero indica el Global en ella cuál el sistema carga el número de alarmas presentes.

TempoLatenzaErrori = Tiempo en segundo que tiene que transcurrir entre dos mensajes con el mismo número de error engendrado por "-SEC/SET_MESSAGE, #", para actualizarlo sobre la zona de vídeo dedicada.

Estádard = 1

FlagHoldG8183 = Flag que define**, si** a 1 que a los ciclos fijos de G81,G82 y G83 pueden irle en HOLD

Estádard = 0

ECSVolantinoSuRapido = Habilita la gestión de la manopla de R como octavilla

WaitSEC = Tiempo él suspensión corchete automático que es introducido a cada instrucción de "SEC" y "DIS/-1:"

Estándard = 0.03

SeGlobaleErroreNoDISM1 = Para evitar el sovrascrittura en la parte de señal con la instrucción-DIS/-1: …. si son presentes señales, el **GlobaleNumeroAllarmi** averiguando, el sistema no manda el mensaje.

Estándard = 1

GlobaleBloccoMovimentazione = El número indicado si mucho de cero la indica Global en el que el sistema averigua la inhibición de los movimientos manuales, no programados, en el caso caso que el valor sea diferente de cero 0.

Estándard = 0

Time_collision_latency = Tiempo en segundo de permanencia sumisión con referencia "eje firme" por cada SERVO OFF

Time_default_lock_wait = Tiempo en segundo de permanencia sumisión a fin animo con los ases “DEFAULT LOCK”

NoDisplayMachineGroupError = Inhabilita si está a 1 la visualización de la señal anomalía ases o mandriles juntados a los Grupos Maquina. Estándard =0, Visualiza la anomalía,

ReturnLocalFromEndTask= Habilita la vuelta de los valores de las Locales al part program chiamante al final de la ejecución del TASK activada con la instrucción: TSK/lll,xxxxx:L1,Lo, Estándard 1 = Vuelta de los valores en las Locales,

Time_NoCheckOverRunMicro = Tiempo en segundo antes de señalar la anomalía cuando se activan los ases, Enable a los accionamientos, para los ases en más allá de carrera. Y tan posible desplazar el eje de la posición de más allá de carrera, con el hacia en la dirección opuesta y en el tiempo indicado en este parámetro.

NumberRetryFieldbus= Número de tentativas de reembargo al periférico de FieldBus antes de interrumpir la comunicación.

Newto_IOErrCnt= **Globale** sobre que es incrementado los errores de comunicación con la ficha NEWTO o 0

Newto_WTDogEnable= **Globale** que si pone a 1 se para el attulizzazione de las Salidas Digitales

GlobaleLingua = Número del Global sobre que es cargado el número de la lengua activa y en particular:

1=Italiano
2=Inglese
3=Tedesco
4=Francese
5=Spagnolo

Si el número es 0 no es provisto tal servicio, Estándard 0

2.1.1 Parámetros General Tablero sinóptico8

Voces de caracterizar	Explicación
[ParametriGenerali]	Nombre del tipo de Caracterización
Campionatura=500	Expresada en microsegundos
NumeriDecimali=3	
NumeriCifre=10	
TolleranzaCerchio= 0,01	Tolleranzatra típico entre rayo iniz.e final
KN= -1.0	NO MODIFIQUES
NUMBER_SAMPLE_SWITCH_PORT=""	Indicar el NOMBRE de la salida que se utiliza en la aplicación para permitir la capacitación de la potencia - Señala que el gobierno es activo a régimen.
FALSETRUE_LOGIC_SAMPLE_SWITCH=1	Estado lógico de la señal NUMBER_SAMPLE_SWITCH_PORT
SegnalePotenza=""	Input de Servo ON que señala "Cabina eléctrica bajo potencia"
FALSETRUE_LOGIC_SegnalePotenza=1	Estado lógico de la Señal Potencia
RitardoPotenza=2.0	Retraso (sec.) antes de someter los Ases
NumeroLocali=32	N. de Variables Locales asignado por estándard a cada P.Program
TickMicro=1	NO MODIFIQUES
TickCounter=1	NO MODIFIQUES
GeneralStatus =0	Máscara de definición de los estados generales del Sistema con los siguientes sentidos: GeneralStatus =0 Modo normal GeneralStatus =4 Dead-lock en anticollision_ no señala errorGeneralStatus =16 (=0x10 en esadecimale) Las compensaciones relativas, dadas por los Part Program no son anulados en sigue a Reposición.

NumeriDecimaliDRT=**3**	N. de decimales (3) visualizado en los valores de las cuotas y los Globales sobre la pantalla DRT (Display Real Time) de la página Automación del entorno "Automación" Valor de estándard = **3** NORMALMENTE NO MODIFICAR
NumeriCifreDRT=**7**	N. de cifras máximas de la pantalla DRT. Valor de estándard = **7** NO MODIFIQUES
NumeriDecimaliFILE=**4**	N. de decimales (4) del tamaño de los números escrito en los files por instrucciones Valor de estándard = **4** NORMALMENTE NO MODIFICAR
NumeriCifreFILE=**10**	N. de cifras del tamaño de los números escrito en los files por instrucciones.Valor de estándard = **10** NORMALMENTE NO MODIFICAR
NumeroByteReadFieldBus=32	Valor típico del CAN
NumeroByteWriteFieldBus=32	Valor típico del CAN
Time_collision_latency=0.2	Tiempo de ulterior finca en sumisión de los ases en caso de emergencia
FileCorrezione=Correz.txt	Nombre del Expediente en que son registradas las correcciones de precisión de los ases
IO_PLC_management=0	NO MODIFIQUES
NumeroIstruzioniMax=64	N. máximo de instrucciones ejecutado en secuencia dentro de cada task antes de suspender el Task por 1 ciclo- NORMALMENTE no MODIFICAR
PartProgramG89=G89.PP	Nombre del P.Program ISO lanzado por el ciclo fijo G89, en el ejemplo el P.Program se llama G89.PP,
VolantinoECS=**0**	SE=0 la Octavilla ECS no se enlaza por el Teclado Se=1 la Octavilla es conectada pero también usa el código de la tecla F11 Valor de estándard = **0**
GraficaCNC=1	Se=1 la gráfica en ambienteCNC es habilitada Se=0 es inhabilitado. Valor de estándard =1 NORMALMENTE NO MODIFICAR

ModuloFieldBusHilscher=**1**	Se=1 es habilitado la ficha Field-bús Hilscher Se=0 es inhabilitado Valor de default=1
ModuloFieldBusTCP-IO =**0**	Se=1 es habilitado la ficha Field-bús me Modulas Se=0 es inhabilitado Valor de default=0
ModuloFieldBusSofting=**0**	Se=1 es habilitado la ficha Field-bús Softing Se=0 es inhabilitado Valor de default=0
FlagFastWaitIO=1	**Si =1 habilita la gestión veloz de la instrucción FWI**Se=0 el FWI como el WDI obra Valor de estándard =1 **Nota-cautela en el empleo de FWI**
GlobaleNumeroAllarmi=0	Número del Global en que el sistema registra de modo puntual cuantas alarmas están presentes Valor de default=0
TempoLatenzaErrori=1	Le expresado en secondi.per dar tiempo de visualizar errores diferentes. Valor de estándard =1(sec).
FlagHoldG8183=**0**	**Se=1 permite de habilitar la ida en HOLD en los ciclos fijos G81-82-83.**Valor de default=0
ECSVolantinoSuRapido=0	**Se=1 el encoderino de teclado ECS, aquel de Exprés, es habilitado.**Valor de default=0
WaitSEC=**0.03**	**Tiempo de retraso automático en las transmisiones de la instrucción SEC.**Valor de default=0.03, segundo,
TempoScansione=**0.050**	Tiempo de barrido de los I/O, en sec. Valor de default=0.050 NO MODIFIQUES
SeGlobaleErroreNoDISM1=**1**	Se=1 protege la visualización de los errores en la tira DIS/-1:……. de superposiciones - Valpore de default=1
GlobaleBloccoMovimentazione=**0**	Número de un global que, si vale cero no inhibe el movimentazione de los ases. Si invece=1 la inhibe Valor de default=0, ninguno global,

Lingua=**ITA**	**Y la lengua de todos los mensajes y las inscripciones del sistema. Default=ITA** ITA = italiano ENG=Inglese etc SPA =Spagnolo
AbilitaMinimizzazione=**1**	**Se=1 la minimización a icono es habilitada.**Se=0 no sipuò minimizar el cuadro Valor de default=1
NoDisplayMachineGroupErro =**0**	Inhabilita si está a 1 la visualización de la señal anomalía ases o mandriles juntados a los Grupos Maquina. Estándard =0, Visualiza la anomalía,
Time_wait_disabile_AxesDefaultLock= **0**	Tiempo(sec, de permanencia de la capacitación para los ases que DISABILTATI CUANDO SIDO "NORMALMENTE es predispuesto PARAS." , Ves calibrado **Kind_Axis_0=0x00000080,** Valor de default=0
ReturnLocalFromEndTask = 1	Habilita la vuelta de los valores de las Locales al part program chiamante al final de la ejecución del TASK activada con la instrucción: TSK/lll,xxxxx:L1,Lo, Estándard 1 = Vuelta de los valores en las Locales,
Time_NoCheckOverRunMicro = 0	Tiempo en segundo antes de señalar la anomalía cuando se activan los ases, Enable a los accionamientos, para los ases en más allá de carrera. Y tan posible desplazar el eje de la posición de más allá de carrera, con el hacia en la dirección opuesta y en el tiempo indicado en este parámetro. Valor de default=0

GlobaleLingua = 0	Número del Global sobre que es cargado el número de la lengua activa y en particular: 1=Italiano 2=Inglese 3=Tedesco 4=Francese 5=Spagnolo Si el número es 0 no es provisto tal servicio, Estándard 0

2.2 I/O señala, digital y analógicos ,

Señalas input y salidas digitales y analógicas

En seguida hay un ejemplo para poder definir de ello el sentido:

EJEMPLO

```
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=}
```

2.2.1 I/O digitales señalan

Parte por la definición del nombre, número y descripción del I/O

IO_NAME(indice NOMBRE, "nombre simbólico"")
NOMBRE tiene que ser comprendido entre 1 y NUM_MAX_DI_BIT (INPUT)
NOMBRE tiene que ser comprendido entre 1 y NUM_MAX_DO_BIT (OUTPUT)

Data_type ="Word" ; "Bit" "Byte" "Word" "Dword" **Default Bit**
Description ="LMInput" **Default “”**
Estension ="1" **Default Non Attiva**

Parte por l ' activación automática del cambio de estado puede ser

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 **es** puesto a cero lógico sobre el mando de REPOSICIÓN y las instrucciones –RST/0 -RST/2
0 no efectúa a ninguna operación

NoThreeState= 1 1 no es puesto en modalidad threestate con la instrucción-RST/4

Parte por la descripción física del I/O

Index_system = índice de sistema para usos futuros
FalseTrue_logic = 1 por bit en lógica negada normalmente y' 0
Kind_sensor ="DIGITAL_OUTPUT" o "DIGITAL_INPUT"

Kind_board= tipo de digital I/O

0; VIRTUAL, PAQUETE K, 1 - 1024 BITS INPUT/OUTPUT
1; LÁMINA I/O IP-UNIDIG-Y-48
2; LÁMINA ASES TIP102 I/O
3; USB MAVEL I/O
4; LÁMINA ASES TPMC118 LOS
5; LÁMINA I/O FIELD AUTOBÚS
6; LÁMINA I/O ETN PRIMERO ELECTRONICS
7; LÁMINA ASES YIDDISHES MAVEL I/O
8: FICHA I/O pci 8255
9: FICHA I/O isa 8255
10: LÁMINA ASES PRIMEROS ELECTRONICS
11 ;PIGGY BOARD I/O ETN su SERVO PRIMA ELECTRONICS
12 : LÁMINA ASES PRIMA ELECTRONICS ANALOGIC PASSIVE
13 : LÁMINA ASES PRIMA ELECTRONICS DIGITAL PASSIVE
14 : LÁMINA MLTA DIGITAL I/O
15 : LÁMINA ASES MLTA per DIGITAL Input
17 : LÁMINA I/O NEWTON (48 INPUT 24 OUTPUT)
18 : LÁMINA ASES NEWTON por DIGITAL Input
20 : LÁMINA ASES PCI-2000
21 : LÁMINA I/O PCI-2001
22 :SERCOS I/O
100 :VIRTUAL_MIX ,PAQUETE K, 1 - 2048 BITS INPUT/OUTPUT

Board_number= número lámina Rack_number

0 ; número usado (0)
1 ; número BOARD IP (0 - N)
2 ; número BOARD IP (0 - N)
3 ; número cestello (0 - 7)
4 ; número usado (0)
5 ; número usado (0)
6 ; número usado (0)
7 ; número BOARD YIDDISH (0 - N)
8 ; número BOARD I/O pci 8255
9 ; número BOARD I/O isa 8255
10 ; número BOARD PIASTRA ASSI PRIMA ELECTRONICS (0 - N)
11 ; número usado (0)
12 ; número BOARD PIASTRA ASSI PRIMA ELECTRONICS (0 - N)
13 ; número BOARD PIASTRA ASSI PRIMA ELECTRONICS (0 - N)
14 ; número BOARD MLTA (0 - N)
15 ; número BOARD ASSI MLTA (0 - N)
17 ; número BOARD NEWTON (0 - N)
18 ; número BOARD ASSI NEWTON (0 - N)

20 ; número BOARD ASSI PCI-2000 (0 - N)
21 ; número BOARD Digital I/O PCI-2001 (0 - N)
22 : número BOARD SERCOS (0 - N)
100 ; número usado (0)

Address_port= offset
0 ;offset usado (0)
1 ;offset Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
2 ;offset Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
3 ; número dispositivo USB MAVEL (0 - 7)
4 ; número usado (0)
5 ; número dispositivo (0 - 767)
6 ; número dispositivo (0 - 255)
7 ; número dispositivo (0 - 15) (0 = locale)
8 ; número dispositivo (0 - 5)
9 ; número dispositivo (0 - 5)
10 ; número asse PRIMA ELECTRONICS (1 - 12)
;unused per bits 65 - 96 perche' informazioni di sistema
11 ; número dispositivo (0 - 255)
12 ; número asse PRIMA ELECTRONICS (1 - 12)
13 ; número asse PRIMA ELECTRONICS (1 - 12)
14 ; número dispositivo (0)
15 ; número asse MLTA (1 - 6)
17 ; número dispositivo (0)
18 ; número asse NEWTON (1 - 4)
20 ;unused
21 ;unused
22 ; número asse SERCOS (1 - 8)
100 :Num WORD di 16 bit (0 - 256)

Phisical_Number = posición física del I/O sobre lámina

0 ;1 - 1024
---
1 ;1 - 48
---
2 ;1 - 16 INPUT (1-8 eje 1 1-4 OUTPUT (1-2 eje 1
9-16 eje 2) 3-4 eje 2)
---
3 ;1 - 64
---
4 ;1 - 6 solo INPUT
---
5 ;1 - 12288

```
  ---------------------------------------------------------
6  ;1 - 1024
  ---------------------------------------------------------
7  ;1 - 32 INPUT    1-32 OUTPUT (   1-8  eje 1
                                     9-16 eje 2
                                    17-24 eje 3
                                    25-32 eje 4  )
  ---------------------------------------------------------
8  ;1 - 48
  ---------------------------------------------------------
9  ;1 - 48
  ---------------------------------------------------------
10 ;1 - 32  Device_R   input: (  9) Abilitazione eje
                               (12) Micro di zero (HOM)
                               (13) Fine-corsa
                               (15) Fault asse
   ;33- 64  Status_R           (33) Axis enabled
                               (34) Axis servo error
                               (35) Axis emergency
                               (36) Axis zeroing
                               (37) Axis zeroed
                               (38) Axis stopping
                               (40) Axis rinting
                               (41) Axis moving
                               (47) Axis driver OK
                               (48) Axis enable OUT
11 ;1 - 1024
  ---------------------------------------------------------
12 ;1 - 8                      (1) Driver OK     input
                               (3) ERRF           "
                               (4) OM             "
                               (5) FCA            "
                               (6) FCI            "
                               (7) FAULT          "
                               (8) ZEROF          "
     1 - 1                     (1) Servo Enable   output
  ---------------------------------------------------------
13 ;1 - 32                    (12) Micro di Hom    input
   ;1 - 32                     (1) Servo Enable   output
  ---------------------------------------------------------
14 ;1 - 16  ( 1-8 INPUT     9-16 OUTPUT )
  ---------------------------------------------------------
15 ;1 - 8   ( 1-8 INPUT )
  ---------------------------------------------------------
```

17 ;1 - 48 (1-48 INPUT) (1-24 OUTPUT)

---

18 ;1 - 3 (1-3 INPUT) (1 OUTPUT)

---

20 ;1 - 16 OUTPUT

1 Enable eje 1
2 Enable eje 3
3 Enable eje 5
4 Boost 1
5 Enable eje 2
6 Enable eje 4
7 Enable eje 6
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 eje 1
2 Switch OMO eje 2
3 Switch OMO eje 3
4 Switch OMO eje 4
5 Switch OMO eje 5
6 Switch OMO eje 6
7 Unused
8 Unused
9 Driver OK eje 1
10 Driver OK eje 2
11 Driver OK eje 3
12 Driver OK eje 4
13 Driver OK eje 5
14 Driver OK eje 6
15 Unused
16 Unused

---

21 ;1 - 40 OUTPUT

1 Enable eje 1
2 Enable eje 3
3 Enable eje 5
4 Boost 1
5 Enable eje 2

		6 Enable eje 4
		7 Enable eje 6
		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
		17 - 40 Output user
1-40	INPUT	1 Switch OMO eje 1
		2 Switch OMO eje 2
		3 Switch OMO eje 3
		4 Switch OMO eje 4
		5 Switch OMO eje 5
		6 Switch OMO eje 6
		7 Unused
		8 Unused
		9 Driver OK eje 1
		10 Driver OK eje 2
		11 Driver OK eje 3
		12 Driver OK eje 4
		13 Driver OK eje 5
		14 Driver OK eje 6
		15 Unused
		16 Unused
		17 - 40 Input user

---

22 ;1 - 32 OUTPUT

1 - 32 INPUT

---

100 ;1 - 2048

2.2.2 Señales Analógicos I/O

IO_NAME(NOMBRE, "nombre simbólico")
indice NOMBRE tiene que ser comprendido entre 1 y NUM_MAX_AI_BIT (INPUT)
indice NOMBRE tiene que ser comprendido entre 1 y NUM_MAX_AO_BIT (OUTPUT)
Kind_sensor="ANALOG_OUTPUT" o "ANALOG_INPUT"
Kind_board= tipo de analog

2 ; LÁMINA ASES TIP102 ANALOG I/O
3 ; LÁMINA IP-FastDAC ANALOG OUTPUT
4 ; LÁMINA ASES TPMC118 ANALOG OUTPUT
5 ; LÁMINA ANALOG FIELD BUS
6 ; LÁMINA ANALOG ETN PRIMA ELECTRONICS
7 ; LÁMINA ASES YIDDISH MAVEL
11 ;PIGGY BOARD ANALOG ETN sobre SERVO PRIMA ELECTRONICS
12 : LÁMINA ASES PRIMA ELECTRONICS ANALOGIC PASSIVE
13 : LÁMINA ASES PRIMA ELECTRONICS DIGITAL PASSIVE
15 : LÁMINA ASES MLTA
16 :PIGGY BOARD MLTA DAC ADC
17 : LÁMINA ASES NEWTON
19 : LÁMINA NEWTON ADC
20 : LÁMINA ASES PCI-2000
22 :SERCOS ANALOG

Board_number= numero piastra Rack_number

2 ;numero BOARD IP (0 - N)
3 ;numero BOARD IP (0 - N)
4 ;numero aparato (0)
5 ;numero aparato (0)
6 ;numero aparato (0)
7 ;numero BOARD YIDDISH (0 - N)
11 ;numero aparato (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 ASSI PCI-2000 (0 - N)
22 ;numero BOARD SERCOS (0 - N)

Address_port= offset

2 ;offset Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
3 ;offset Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000)
4 ; número usado (0)
5 ; número aparato (0 - 767)
6 ; número aparato (0 - 255)
7 ; número aparato (0 - 15) (0 = locale)
11 ; número aparato (0 - 255)
12 ; número eje PRIMA ELECTRONICS ANALOG PASSIVE (1 - 12)
13 ; número eje PRIMA ELECTRONICS DIGITAL PASSIVE (1 - 12)
15 ;unused
16 ;offset PIGGY BOARD MLTA (0 - 3)
17 ;unused
19 ;unused
20 ;unused
22 ;unused

Phisical_Number= posición física del canal analógico I/O sobre lámina

2 ;1 - 2 INPUT 1 - 2 OUTPUT
3 ;1 - 8
4 ;1 - 6 solo OUTPUT
5 ;1 - 12288/16
6 ;1 - 10
7 ;1 - 4
11 ;1 por aparato
12 ;unused
13 ;1 -1
(1) = SERVO en Corriente
(-1) = SERVO en Tensión
15 ; número eje MLTA (1 - 6)
16 ; número canale MLTA DAC ADC (1 - 2)
17 ; número eje NEWTON (1 - 4)
19 ; número canal NEWTON ADC (1 - 4)
20 ; número eje PCI-2000 DAC = (1 - 6+3) ADC = (1 - 3)

2.2.3 Parámetros de los I/O digitales y analógicos - Tableros sinópticos

Los I/O digitales y analógicos de la aplicación que está realizándose encuentran sobre más fichas HW, a menudo de tipo diferente.
En el Expediente sistema.txt, la caracterización de todos los I/O inicia después de la indicación **[IO]** escrita una sola vez en cabeza a todo los I/O
Si se trata de I/O virtuales, surgida de flags por situaciones lógicas internas, no unidos a señales externas, pero en todo caso administráis como si fueran I/O, caracterizar el **Kind_board=0**
Si en cambio se trata de I/O reales, el **Kind_board** = valor indicado para cada tipo de ficha, ej. 20,
Son caracterizados todo en esta fase que, como ejemplo, reconduce algunos casos con las indicaciones adecuadas a describir la metodología más que la simple regulación.
Input y Salida se encuentran sobre las fichas:

- PCI 2001 6 Ases Analógicos + 3 Mandrilas (20)
- PCI 2001 24+24 I/O (21)
- 48 I/O TTL (8) PCI (9) ISA
- FICHA FIELD-BUS (5)
- SERCOS (22)

2.2.3.1 Salidas Digitales - Ejemplo de caracterización de la ficha PCI 2000

[IO] ← Escrito una sola vez antes de la caracterización de todos los I/O, Digitales y Analógicos,

Voces de caracterizar	*Informaciones por la definición de un Salida digital*
i=IO_NAME(1 , "EnableX1")	Número lógico =1, se asigna de costumbre 1 a la primera Salida, 2 al segundo etcétera,-Ves nota 1 Nombre lógico = **EnableX1**
b={	
Description="Abilitazione Asse X1"	Descripción en claro, de insertar, como se quiere comparezca en la pantalla de sistema de los I/O
Index_system=**0**	Reservado a AWL - no MODIFICAR
FalseTrue_logic=**0**	Si = 0 los niveles lógicos y físicos coinciden Si = 1 es invertido
Kind_sensor="DIGITAL_OUTPUT"	DIGITAL_OUTPUT define que es una Salida digital - no MODIFICAR
Kind_board=**20**	**20** es el código de reconocimiento obligatorio de todos los I/O de la ficha PCI 2000
Rack_number=**0**	Indicar la posición de la ficha en el Back-plane, la ficha "0" es la más cercana a la UCP, "1" la segunda del mismo tipo etcétera,
Address_port=**0**	En este caso queda =0. , porque la ficha Kind_board=20 no tiene I/O dedicado a priori a cada eje, Si en cambio por ejemplo, se utilizan fichas con Kind_board=22, para ases SERCOS con 8 secciones, será escrito un número por 1 a 8, según la sección utilizada por los I/O de cada individual Eje.
Physical_number=**1**	**1** es el número con el que el Sistema reconoce esta salida sobre la ficha, Kind_board=20. Como enseñado en el Tablero de las Correspondencias, si se cambia el número se cambia físicamente la salida utilizada.
e=}	

NOTAS-
1, el Número Lógico tiene que ser unívoco en el ámbito del mismo tipo de I/O. podrán existir pues el input Digital =1, lo analógico =1, La salida Digital =1, lo analógico =1 etcétera. No es aceptada la situación de dos o más I/O del mismo tipo con el mismo número.
2, las otras Salidas de la ficha, previstos por los mandos Enable de los otros 5Assi y de las señales de Enable, Dir, Boost dedi cati al pilotaje de los Step-motors se caracterizan con el mismo criterio de aquél enseñado en el ejemplo.
3, las Salidas no utilizadas por el objetivo original, si no se tienen por ejemplo Step-Motors, pueden ser utilizados como normales Salidas, asignando oportunamente el número lógico y el nombre lógico, primera raya.
4, La caracterización de una Salida digital de la ficha 24+24I/O PCI 2001 se realiza con las mismases modalidades, asignando pero Kind_board=21 en vez de 20 y el Phisical_number indicados en el tablero de las correspondencias.
Mismo principio por las Salidas administradas por Field-autobús u otro.

2.2.3.2 Salidas Analógicas Ejemplo de caracterización de la ficha PCI 2000

NOTA - Usted Caracterización de las Salidas Analógicas es igual a la de aquellos Digitales, en cuánto sirve a indicar al Sistema de gestión cuál calle se quiere utilizar entre aquellas predispuestas, y cuál nombre se quiere dar a la señal.

En el ejemplo siguiente en efecto se caracteriza la salida Analógica, Kind_sensor = "**ANALOG_OUTPUT",**

f) denominado, **2,"DACM2",**
qué la descripción aclara como **("DAC Mandr. 2",**
domiciliado sobre la 2a Ficha a partir de la UCP (**Rack_number=1)**
ella que ficha es tipo 6Assi analógica + 3Mandrini, PCI 2000 (**Kind_board=20)**
y, no teniendo secciones múltiples, (Address_port=0,

- es reconocida por el Sistema con el, **Physical_number=8-Ves Tablero de las Correspondencias,**

```
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 Digitales - Ejemplo de caracterización de la ficha PCI 2000

Voces de caracterizar	*Informaciones por la definición de un Input digital*
i=IO_NAME(1 , "AUX_ON")	Número lógico =1, se asigna de costumbre 1 a la primera Salida, 2 al segundo etcétera, Nombre lógico = **AUX_ON**
b={	
Description=" Auxiliary ON "	Descripción en claro, de insertar, como se quiere comparezca en la pantalla de sistema de los I/O
Index_system=1	Reservado a AWL - no MODIFICAR
FalseTrue_logic=0	Si = 0 los niveles lógicos y físicos coinciden Si = 1 es invertido
Kind_sensor=" DIGITAL_INPUT "	DIGITAL_INPUT define que es un Input digital - no MODIFICAR
Kind_board=21	**21** es el código de reconocimiento obligatorio de todos los I/O de la ficha PCI 2001
Rack_number=0	Indicar la posición de la ficha en el Back-plane, la ficha "0" es la más cercana a la UCP, "1" la segunda del mismo tipo etcétera,
Address_port=0	En este caso queda =0. , La indicación sirve a localizar la eventual sección de ficha,
Physical_number=17	**17** es el número con el que el Sistema reconoce esta salida sobre la ficha, Kind_board=20. Como enseñado en el Tablero de las Correspondencias, si se cambia el número se cambia físicamente la salida utilizada..
e=}	

NOTA-1, Con el mismo criterio se caracterizan los otros Input de la ficha.

Otro ejemplo de Caracterización del input de Homing sobre la ficha ases 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 Analógicos - Ejemplo de caracterización de la ficha PCI 2000

NOTA - Usted Caracterización de los Input Analógicos es igual a la de aquellos Digitales, en cuánto sirve a indicar al Sistema de gestión cuál calle se quiere utilizar entre aquellas previstas, y cuál nombre se quiere dar a la señal. Obviamente no se caracterizan las voces Index_system = y FalseTrue_logic = en cuanto tratasi de señales analógicas

```
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 CARACTERIZACIÓN DE I/O DIGITALES CONSECUTIVOS A. GRUPOS

Creación de la estructura

Para minimizar el número de tableros cuando se tienen bastantes Input o Salidas consecutivas, virtuales o reales, como por ejemplo típicamente organizados por la gestión Field Autobús, es posible disfrutar del automatismo aquí descrito, que permite de definir un entero grupo homogéneo de Input o bien Salida con un solo tablero.
En el expediente de calibrado típico de un Input, pero la misma lógica se aplica al expediente de una Salida, se introducen las dos siguientes voces que caracterizar, que definen la estructura de Input, o bien de Salida, múltiple:

Data_type = "Word"
Extension = "n"

Dónde "**Word"**, valor selecto aquí como ejemplo, puede asumir los siguientes nombres y valores:

"Bit" = 1 input individuales, Caso sin sentido. Por un sólo Input se rellena la estructura.)
"Nibble" = 4 input consecutivos
"Byte" = 8 input consecutivos
"Word" =16 input consecutivos
"Dword" =32 input consecutivos

Y dónde "**n"** representa la extensión del nombre fecha automáticamente del sistema.
Eso es debido al hecho que con esta estructura, la voz i=IO_NAME (…,…) ya no es el nombre de una individual señal pero el Apellido de la familia de "señales." Los individuales Nombres son asignados automáticamente y en secuencia a las botaduras Input de la familia, a partir del número indicado en Extension con "**n"**

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

Voces de caracterizar	*Informaciones por la definición de un Salida digital*
i=IO_NAME(501 , "I_011")	Número lógico =501 Nombre lógico = "I_011"
b={	
Description="LM Input"	Descripción en claro, de insertar, como se quiere comparezca en la pantalla de sistema de los I/O Sarà iguales por todos los Input
Data_type="Dword"	VOZ DE AGGIUNGERE A. LA ESTRUCTURA BASE, con tablero por 32 Input "Dword"
Extension=1	Número de que el Extension parte a contar, en el ejemplo hasta 32,
Index_system=0	Reservado a AWL - no MODIFICAR
FalseTrue_logic=0	Si = 0 los niveles lógicos y físicos coinciden Si = 1 es invertido
Kind_sensor="DIGITAL_INPUT"	DIGITAL_INPUT define que es un Input digital - no MODIFICAR
Kind_board=**5**	**5** es el código de reconocimiento obligatorio de todos los I/O de la ficha Field Bus
Rack_number=0	Indicar la posición de la ficha en el Back-plane, la ficha "0" es la más cercana a la UCP, "1" la segunda del mismo tipo etcétera,
Address_port=0	En este caso queda =0. , La indicación sirve a localizar la eventual sección de ficha,
Physical_number=7	**7** es el número con el que el Sistema reconoce lo primero Input de la familia de señales consecutivas sobre la interfaz KAN (Kind_board=5). Como enseñado en el Tablero de las Correspondencias, si se cambia el número se cambia físicamente el input utilizado.
e=}	

2.2.3.6 LANZAMIENTO DE TASK ES P.PROGRAM SOBRE EL CAMBIO DE ESTADO DE I/O

Creación de la estructura

Para lanzar P.Program sobre la conmutación de una señal de Input, o bien de una Salida, es posible insertar las voces, aquí en seguida descritas, dentro de la estructura típica de caracterización de los Input, o bien de las Salidas.

Las voces pueden ser insertáis en un cualquiera posición, pero se sugiere de ponerle en el fondo, enseguida antes del cierre e=}

Nota - si estas voces son insertadas en una estructura múltiple, es suficiente la conmutación de uno cualquiera de las señales de la estructura para activar el lanzamiento del P.Program.

Definiciones de la estructura de lanzamiento P.Program sobre el estado de un I/O

a	TaskChangeFront=1	Se=1 siente la frente de conmutación de la señal Al cambio de nivel de 0 a 1 lanza el Part Program "InOne" indicado en raya "y" que reside en el Expediente "LM_011.pp" indicado en raya "d."Al cambio de nivel de 1 a 0 lanza el Part Program "InZero" indicado en raya "g" que reside en el Expediente "LM_011.pp" indicado en raya "f." Se=0 siente el nivel de la señal. Si nivelo 1 lanza el Part Program "InOne" indicado en raya "y" que reside en el Expediente "LM_011.pp" indicado en raya "d."Si nivelo 0 lanza el Part Program "InZero" indicado en raya "g" que reside en el Expediente "LM_011.pp" indicado en raya "f." Nota-los nombres de los P.Program y los Files son dados como ejemplo.
c=b	TaskModeRun=**1**	Si =1 los P.Program de las rayas "y" o bien "g", en acuerdo con el calibrado de raya "a", sólo son lanzados si soy no activas. Si =0 los P.Program son reenvidados en todo casoSi =2 señala error si un task activo es reenvidado Valor de default=1
d=c	TaskOnePP="LM_011.pp"	N ombre del Expediente de P.Program tipo. pp
e=d	TaskOneName="InOne"	Nombre de P.Program en el Expediente indicado en "d"

f=e	TaskZeroPP="LM_011.pp"	Nombre del Expediente de P.Program tipo. pp
g=f	TaskZeroName="InZero"	Nombre de P.Program en el Expediente indicado en "g"
h=g	ZeroOnReset=0	Se=0 El mando de Reposición determinada en Manual e/o de P.Program no actúa sobre esta Salida. , Nota - El mando no actúa si el calibrado es relativa a un Input, Se=1 El mando de Reposición es habilitado a actuar llevando en todo caso la salida a cero
i	NoThreeState=**0**	Si =0 este Input/Output va en la condición lógica de ThreeState. Si = 1 este Input/Output no va en la condición lógica de ThreeState.Valor de default=0

2.3 Ases y mandriles

Parámetros inherentes a los ases y mandriles

2.3.1 Ases

```
[YS]
PHYSICAL_AXIS_NAME( 1, "YS" )
                ;Numero e Nome dell'asse
b={


Kind_axis="ARM"
      ; Define un eje

Kind_axis_0=0x00005001
    ; Máscara 0 a bits del eje
        ;0x00000001 = SUMISIÓN EN CORRIENTE, 0 = EN VELOCIDAD',
        ;0x00000002 = GESTIÓN "SQUARE" de KP
        ;0x00000004 = GESTIÓN "SQUARE" de KD
        ;0x00000008 = GESTIÓN "SQUARE" de KD2
        ;0x00000010 = GESTIÓN "SQUARE" de KI
        ;0x00000020 = EJE CON SALIDA DULCE DEL SIRVO
        ;0x00000040 = EJE QUE DECELERA SOBRE MICRO DE FINA CARRERA
        ;0x00000080 = EJE NORMALMENTE INHABILITADO, CUANDO Y' FIRME,
        ;0x00000800 = EJECUCIÓN OMO "A. GOLPE"
        ;0x00001000 = EJECUCIÓN OMO SOLISTA SOBRE EL MARKER
        ;0x00002000 = EJECUCIÓN OMO SOLISTA SOBRE LO MICRO
        ;0x00004000 = EJE QUE SÓLO se MUEVE DESPUÉS DE OMO
        ;0x00010000 = SISTEMA VIRTUAL SCARA
        ;0x00020000=SISTEMA VIRTUAL SCARA PANTÓGRAFO COMP. POSIT.
        ;0x00040000 = SISTEMA VIRTUAL SCARA PANTÓGRAFO COMP. NEGAT.
        ;0x00080000 = SISTEMA VIRTUAL CILÍNDRICO TIPO LANCHA
        ;0x00100000 = SISTEMA VIRTUAL CILÍNDRICO TIPO TRASLANTE
        ;0x00200000 = SISTEMA VIRTUAL LASERVAL
```

Kind_axis_1=0x00070200

; Máscara 1 a bits del eje
;0x00000002 = EJE TIPO POTENZIOMETRICO
;0x00000004 = EJE CON CAPACITACIÓN SIRVO DESPUÉS DE MOV.
;0x00000008 = EJE ROLLOVER CON VALORES entre 0 y +MODULO
;0x00000010 = EJE ROLLOVER CON VALORES entre - MÓDULO y +MODULO
;0x00000020 = EJE POR PLOTTER
;0x00000040 = EJE "TRASLADO"
;0x00000080 = TRAMOS CON ACELERACIÓN S_CURVE
;0x00000100 = TRAMOS CON DESACELERACIÓN S-CURVA
;0x00000200 = EJE CON PUESTA EN FASE (MAVEL)
;0x00000400 = EJE FUERA DE SERVICIO
;0x00010000 = COLLISION ERROR
;0x00020000 = SERVO ERROR (EJE FIRME)
;0x00040000 = POSITION ERROR (EJE EN MOVIMIENTO)
;0x00080000 = ALTO COUNTER ERROR
;0x00100000 = DRIVER VALE ERROR
;0x00800000 = EJE MÁSTER CON ESLAVOS CARAT. GANTRY
;0x01000000 = EJE MÁSTER CON ESLAVAS QUE TIENE QUE SEGUIR
;0x02000000 = EJE MÁSTER CON ESLAVAS QUE EL TEÓRICO SIGUE
;0x04000000 = EJE MÁSTER CON ESLAVOS INTERPOLANTE
;0x08000000 = PID COMPLEMENTARIO TAMBIÉN CON MOVIMIENTO
;0x10000000 = BOOST ACTIVADO EN ACC. , SÓLO CON EJE STEPPER,

Kind_board_TransducerPosition=1

; Tipo de lámina transductor
; 1 = EJE TIP102 (ENCODER+DAC)
; 2 = EJE TPMC118 (ENCODER+DAC)
; 10 = EJE USB MAVEL (ENCODER+DAC)
; 11 = EJE YIDDISHES MAVEL (ENCODER+DAC)
; 20 = EJE MLTA (ENCODER+DAC)
; 30 = EJE NEWTON (ENCODER+DAC)
; 40 = EJE PCI2000 N (ENCODER+DAC)
; 100 = EJE PRIMEROS ELECTRONICS (POSICIÓN)
; 101 = EJE PRIMEROS ELECTRONICS, ANALÓGICA VOZ PASIVO,
; 102 = EJE PRIMEROS ELECTRONICS, DIGITAL VOZ PASIVO,
; 103 = EJE ENCODER CON RSM304 P.E. (ETN)
; 200 = EJE SERCOS, POSICIÓN ')
; 201 = EJE SERCOS (VELOCIDAD)
; 202 = EJE SERCOS, CORRIENTE,
; 210 = EJE YASKAWA (POSICIÓN)
; 220 = EJE ETEL (POSICIÓN)
;1000 = EJE POR STEPPER A. BITS
:1002 = EJE POR STEPPER A. CONTADOR PCI 2000

:1003 = EJE POR STEPPER A. CONTADOR PCI 2001
:1100 = EJE POR STEPPER RSM304 ETN SIRVO

Board_number_TransducerPosition=1

; Número de lámina por transductor
; 1 0 - n numero lámina TIP102
; 2 0 - n numero lámina TPMC118
; 10 0 - 7 numero cesta USB MAVEL
; 11 0 - n numero lámina YIDDISH
; 20 0 - n numero lámina MLTA
; 30 0 - n numero lámina NEWTON
; 40 0 - n numero lámina PCI 2000
; 100 0 - n numero lámina PRIMEROS ELECTRONICS
; 101 0 - n numero lámina PRIMEROS ELECTRONICS
; 102 0 - n numero lámina PRIMEROS ELECTRONICS
; 103 unused EJE ENCODER CON RSM304 P.E. (ETN)
; 200 0 - n numero lámina SERCOS
; 201 0 - n numero lámina SERCOS
; 202 0 - n numero lámina SERCOS
; 210 0 - n numero lámina YASKAWA, POSICIÓN,
; 220 0 - n numero lámina ETEL, POSICIÓN,
;1000 unused STEPPER A. BITS
;1002 0 - n numero lámina PCI 2000
;1003 0 - n numero lámina PCI 2001
:1100 unused EJE POR STEPPER RSM304 ETN SIRVO

Address_port_TransducerPosition=0x00004000

; Dirección eje transductor
; 1 compensaciones sobre Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000,
; 2 unused
; 10 unused
; 11 0 - 15 numero lámina, 0 = local,
; 20 unused
; 30 unused
; 40 unused
; 100 unused
; 101 unused
; 102 unused
: 103 6 - n dirijo DRIVER CONTROL modulo, M-1)*2
4 módulos consecutivos con M>=4
; 200 unused
; 201 unused
; 202 unused
; 210 unused

; 220 unused
;1000 unused
;1001 unused
;1002 unused
;1003 unused
:1100 6 - n dirijo DRIVER CONTROL modulo, M-1)*2
4 módulos consecutivos con M>=4

Physical_number_Transducerposition=2
; Número físico eje transductor
; 1 1 - 2 numero ases lámina TIP102
; 2 1 - 6 numero ases lámina TPMC118
; 10 0 - 7 numero eje USB MAVEL
; 11 1 - n numero eje lámina YIDDISH, n = 4 local 2 remoto,
; 20 1 - n numero eje lámina MLTA, n = 2 o 4 o 6,
; 30 1 - n numero eje lámina NEWTON, n = 4,
; 40 1 - n numero eje lámina PCI2000(n = 6,
; 100 1 - 16 numero ases lámina PRIMEROS ELECTRONICS
; 101 1 - 16 numero ases lámina PRIMEROS ELECTRONICS
; 102 1 - 16 numero ases lámina PRIMEROS ELECTRONICS
: 103 1 - 3 Numero eje
; 200 1 - 8 numero ases lámina SERCOS
; 201 1 - 8 numero ases lámina SERCOS
; 202 1 - 8 numero ases lámina SERCOS
; 210 1 - 15 número ases lámina YASKAWA (POSICIÓN)
; 220 1 - 4 numero ases lámina ETEL, POSICIÓN,
;1000 unused STEPPER A. BITS
;1002 1 - 3 STEPPER A. CONTADOR PCI2000, pulse 1 - 3,
;1003 1 - 3 STEPPER A. CONTADOR PCI2001, pulse 1 - 3,
:1100 1 - 3 Numero motor

Kind_board_DAC=1
; Tipo de lámina eje convertidor
; 1 = EJE TIP102 (ENCODER+DAC)
; 2 = EJE TPMC118 (ENCODER+DAC)
; 10 = EJE USB MAVEL (ENCODER+DAC)
; 11 = EJE YIDDISHES MAVEL (ENCODER+DAC)
; 20 = EJE MLTA (ENCODER+DAC)
; 30 = EJE NEWTON (ENCODER+DAC)
; 40 = EJE PCI2000 (ENCODER+DAC)
; 100 = EJE PRIMEROS ELECTRONICS (POSICIÓN)
; 101 = EJE PRIMEROS ELECTRONICS, ANALÓGICA VOZ PASIVO,
; 102 = EJE PRIMEROS ELECTRONICS, DIGITAL VOZ PASIVO,
; 103 = tipo ilegal
; 200 = EJE SERCOS (POSICIÓN)

; 201 = EJE SERCOS, VELOCIDAD',
; 202 = EJE SERCOS, CORRIENTE,
; 210 = EJE YASKAWA (POSICIÓN)
; 220 = EJE ETEL (POSICIÓN)
;1000 = EJE POR STEPPER A. BITS
;1002 = EJE POR STEPPER A. CONTADOR PCI200
;1003 = EJE POR STEPPER A. CONTADOR PCI2001
:1100 = EJE POR STEPPER RSM304 ETN SIRVO

Board_number_DAC=1

```
                ; Número de lámina eje convertidor
  ; 1 0 - n      numero lámina TIP102
  ; 2 0 - n      numero lámina TPMC118
  ; 10 0 - 7     numero cesta USB MAVEL
  ; 11 0 - n     numero lámina YIDDISH
  ; 20 0 - n     numero lámina MLTA
  ; 30 0 - n     numero lámina NEWTON
  ; 40 0 - n     numero lámina PCI2000
  ; 100 0 - n    numero lámina PRIMEROS ELECTRONICS
  ; 101 0 - n    numero lámina PRIMEROS ELECTRONICS
  ; 102 0 - n    numero lámina PRIMEROS ELECTRONICS
  ; 103 unused  Tipo ilegal
  ; 200 0 - n    numero lámina SERCOS
  ; 201 0 - n    numero lámina SERCOS
  ; 202 0 - n    numero lámina SERCOS
  ; 210 0 - n    numero lámina YASKAWA, POSICIÓN,
  ; 220 0 - n    numero lámina ETEL, POSICIÓN
  ;1000 unused          STEPPER A. BITS
  ;1002 0 - n    numero lámina PCI2000
  ;1003 0 - n    numero lámina PCI2001
  :1100 unused          EJE POR STEPPER RSM304 ETN SIRVO
```

Address_port_DAC=0x00004000

```
                ; Dirección eje convertidor
  ; 1 compensaciones sobre Carrier ("A"=0x1000 "B"=0x2000 "C"=0x3000 "D"=0x4000,
  ; 2 unused
  ; 10 0 - 7     numero aparato
  ; 11 0 - 15    numero lámina, 0 = local,
  ; 20 unused
  ; 30 unused
  ; 40 unused
  ; 100 unused
  ; 101 unused
  ; 102 unused
  ; 103 unused  Tipo ilegal
  ; 200 unused
  ; 201 unused
  ; 202 unused
  ; 210 unused
```

; 220 unused
;1000 unused
;1001 unused
;1002 unused
;1003 unused
:1100 6 - n dirijo DRIVER CONTROL modulo, M-1)*2
4 módulos consecutivos con M>=4

Physical_number_DAC=2

; Número físico eje convertidor
; 1 1 - 2 numero ases lámina TIP102
; 2 1 - 6 numero ases lámina TPMC118
; 10 0 - n numero ases cesta USB MAVEL
; 11 1 - n numero ases lámina YIDDISH, n = 4 local 2 remoto,
; 20 1 - n numero ases lámina MLTA, n = 2 o 4 o 6,
; 30 1 - n numero ases lámina NEWTON, n = 4,
; 40 1 - n numero ases lámina PCI2000, n = 6,
; 100 1 - 16 numero ases lámina PRIMEROS ELECTRONICS
; 101 1 - 16 numero ases lámina PRIMEROS ELECTRONICS
; 102 1 - 16 numero ases lámina PRIMEROS ELECTRONICS
; 103 unused Tipo ilegal
; 200 1 - 8 numero ases lámina SERCOS
; 201 1 - 8 numero ases lámina SERCOS
; 202 1 - 8 numero ases lámina SERCOS
; 210 1 - 15 numero ases lámina YASKAWA, POSICIÓN,
; 220 1 - 4 numero ases lámina ETEL, POSICIÓN,
;1000 unused STEPPER A. BITS
;1002 1 - 3 STEPPER A. CONTADOR PCI2000, pulse 1 - 3,
;1003 1 - 3 STEPPER A. CONTADOR PCI2001, pulse 1 - 3,
:1100 1 - 3 Numero motor

Level_TCH=1

Cada eje puede' staticizzare de modo independiente de los otros.
Parallelizzando con el hardware el input de touch de los ases intencionales,
el staticizzazione y' contemporánea, pero tiene que ser definida al
software poniendo LEVEL_TCH_ASSE con un número igual por cada uno
de estos ases que tienen la señal de touch en paralelo.
El valor de LEVEL_TCH_ASSE y' incluido tro 0 a 32;
0 significa eje sin touch.

NotifyEnabled=0

/ / Tiene que ser = 0

KP=3000.0

renta proporcional (error de posición)

KI=16.0

renta complementaria (sumatoria de los errores de posición)

KD=20000.0

renta derivativa (error de velocidad')
Para ases stepper define la frecuencia de sumisión (Hz)

KD2=10000.0
renta acelerador (error de aceleración)
Para ases stepper define la frecuencia de scavalcamento resonancia, Hz,

VoltMaxIntegratedError=2.0
máxima tensión, Voltio, conseguido por el complementario KI

Molt_AFF=100.0
Acceleration Feed Forward
Porcentaje de accelerazione/decelerazione usado en el PID
Y' típico por driver en corriente, análogo al VFF, pero puede usarse
también por driver en velocidad'

K_DAC_1Volt=3276.7
valor digital del DAC para conseguir 1 Voltios
poniendo 0 el valor y realzado automáticamente del código eje

Molt_VFF=0.0
Velocity Feed Forward
de usarse típicamente por driver en velocidad'
Y' típico por driver en velocidad', análogo a AFF,; por driver en
corriente puede servir a reducir el efecto del
Fuerza Electro Motriz inducido, a pacto que a la tensión de AUTOBÚS del
driver sea suficiente,

KS=0.0
Valor de truncamiento por el PID. Generalmente es puesto a 0.0005

KT=0.0
Tiempo de muestreo en porcentaje correspondiente al retraso entre
lectura posición y correspondiente mando activo

KM=0.001
Para ases stepper mixtos ("Kind_board_TransducerPosition =... y
"Kind_board_DAC=1000", este campo tiene que representar el valor de
un step del motor, porque' el campo Step = y' usado por el encoder
Con ases stepper que tienen "Kind_board_TransducerPosition=1000" y
"Kind_board_DAC=1000" es usado el solo valor de "Step =.. y
luego "KM =.. no y' usado

KN=0
Para ases stepper a contador difinisce la frecuencia de base
del contador en Hz / / más USADO

Volt_max=9.0
máxima tensión le fecha al DAC (Voltio)

Volt_vel=-8.0
tensión (Voltio) necesaria para moverse a Vel_max por driver en velocidad'
tensión (Voltio) necesaria para acelerarle a Acc_max por driver en corriente

KC_Parameter=0
Este parámetro permite de definir la relación de conversión por
la sumisión, en alternativa a Volt_vel Vel_max o a Volt_vel Acc_max.
Si y' cero son utilizados de otro modo los dos campos susodichos viene
usado éste.
El sentido físico del campo y' la velocidad' (u/min) por 1 voltios en el
caso de sumisiones en velocidad'; la aceleración (u/sec2) por 1 voltios
por las sumisiones en corriente.

Volt_offset=+0.0
compensación de tensión necesaria para parar el eje (Voltio)

Step=+0.001
multiplicador empujado -> coordinado cálculo

Backslash=0.0
Valor del juego mecánico de recobrar

Revolution=0.0
MÓDULO de rotación para ases ROLLOVER

Vel_max=50000.0
Velocidad' máxima del eje
por sumisiones en velocidad' y' la velocidad' conseguidas con "Volt_vel"

Acc_max=20000.0
Aceleración máxima del eje
por sumisiones en corriente y' la aceleración conseguida con "Volt_vel"

Dec_max=15000.0
Desaceleración máxima del eje

ServoError_standby=2.0
Error de posición con eje firme
administrado si Kind_axis_1 tiene el settato bit 0x00020000 = SERVO ERROR

ServoError_motion=10.0
Error de posición con eje en movimiento
administrado si Kind_axis_1 tiene el settato bit 0x00010000 = COLLISION ERROR

Position_error=0.03
Error de posición del eje a fin animo
administrado si Kind_axis_1 tiene el settato bit 0x00040000 = POSITION ERROR

Cycles_positioning=1.0
Tiempo máximo de posicionamiento en segundo, para entrarle en "Position_error",

Cycles_wait=0.002
Tiempo mínimo de permanencia en segundo (eje en "Position_error")

Offset_axis=-534.4
Compensación eje (cero intencional con respecto de cero MARKER)

Positive_over_travel=572.0
Campo positivo de trabajo con respecto del cero offsettato
es controlado después de HOM

Negative_over_travel=-2.0
Voy tirando negativo de trabajo con respecto del cero offsettato
es controlado después de HOM

DelayStopCounter=0.2
Ventana en segundo por control ALTO COUNTER

VoltStopCounter=0.5
Tensión en Voltio; con valores => es activado el control ALTO COUNTER

VelStopCounter=1.0
Velocidad' en mm/min; con valores <= es activado el control ALTO COUNTER
administrado si Kind_axis_1 tiene el settato bit 0x00080000 = ALTO COUNTER ERROR

TimeOutDriverOk=1.0
Time Out en segundos DRIVER VALE sin señalar error
administrado si Kind_axis_1 tiene el settato bit 0x00100000 = DRIVER VALE ERROR

Time_acc_nla=0.0
tiempo de variación aceleración con respecto de tramo lineal caracterizado
> 0 = JERK
= 0 = TRAMO LINEAL
<0 = TRAMO SINUSOIDAL, acc mediano = acc max * 2 / PE,

Time_dec_nla=0.0
tiempo de variación desaceleración con respecto de tramo lineal caracterizado
> 0 = JERK
= 0 = TRAMO LINEAL
<0 = TRAMO SINUSOIDAL, dec mediano = dec max * 2 / PE,

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 = ''
nombre del eje eslavas

K_follower_slave=0.0
coeficiente porcentual de persecución de las eslavas
con ases GANTRY porcentual de corrección cruzada

VoltGantry=5.0
máximo desequilibrio de tensión por GANTRY rígidos

DeltaOffset_SlaveGantry=0.640
valor de desplazamiento de las cero eslavas con respecto del máster (GANTRY)

Number_SERVO_switch_port = "
nombre de la salida de SIRVO
con nombre nulo (") en los siguientes casos es usado la salida de estándard
Kind_board_DAC=1 (TIP102) 1; sirvo Amplifier Enable
Kind_board_DAC=10 (USB MAVEL) 1; Sirvo Amplifier Enable
Kind_board_DAC=11, YIDDISHES MAVEL, 1; Sirvo Amplifier Enable
Kind_board_DAC=102, PRIMEROS ELECTRONICS DEDALERA PASIVO, 1 ;
Flag_W
en el caso YIDDISH son usados 8 bits por eje, en orden de eje

FalseTrue_logic_SERVO_switch=1
estado salida con SERVO ON
= 0 salida puesta a cero
= 1 salida puesta a uno
<> 0 o 1 salidas administradas

Number_DRVOK_switch_port = "INP13"
con nombre nulo (") en los siguientes casos es usado el input de estándard
Kind_board_TransducerPosition=1 (TIP102) 16; General Input Axis
Kind_board_TransducerPosition=10 (USB MAVEL) 1 ; Driver VALE
Kind_board_TransducerPosition=11, YIDDISHES MAVEL, 1 ; Driver VALE
Kind_board_TransducerPosition=102 (P. E. DIGITAL PASIVO) 2; Device_R + 1
si el eje tiene que ser fasato
Kind_board_TransducerPosition=102 (P. E. DIGITAL PASIVO) 1; Device_R
si el eje y' sin puesta en fase
en el caso YIDDISH son usados 8 bits por eje, en orden de eje

FalseTrue_logic_DRVOK_switch=1
estado input con DRIVER VALE
= 0 input a cero con DRIVER VALE
= 1 input a uno con DRIVER VALE
<> 0 o 1 input administrados

Number_PHASE_switch_port = "OUT13"
con nombre nulo (") en los siguientes casos es usado la salida de estándard
Kind_board_DAC=1 (TIP102) 2; Phase axis (LED)
Kind_board_DAC=10 (USB MAVEL) 2; Phase axis
Kind_board_DAC=11, YIDDISHES MAVEL, 2; Phase axis
en el caso YIDDISH son usados 8 bits por eje, en orden de eje

FalseTrue_logic_PHASE_switch=1
estado salida con PHASE
= 0 salida puesta a cero
= 1 salida puesta a uno
<> 0 o 1 salidas administradas

Delay_PHASE=2.0
tiempo de puesta en fase en segundo, subida tensión en meta' tiempo,
para ases sin fase tiempo de OMO a golpe

Volt_PHASE=3.0
tensión de puesta en fase
para ases sin fase tensión de OMO a golpe

Number_OMO_switch_port = "
con nombre nulo (") en los siguientes casos es usado el input de estándard
Kind_board_TransducerPosition=1 (TIP102) 4 ; Reference Switch Input
Kind_board_TransducerPosition=10 (USB MAVEL) 2 ; Reference Switch Input
Kind_board_TransducerPosition=11, YIDDISHES MAVEL, 2 ; Reference Switch Input
Kind_board_TransducerPosition=102 (P. E. DIGITAL PASIVO) 8; Device_R + 1
en el caso YIDDISH son usados 8 bits por eje, en orden de eje

FalseTrue_logic_OMO_switch=1
estado input de OMO
= 0 input a cero con micros OMO pisados
= 1 input a uno con micros OMO pisados
<> 0 o 1 input administrados

Space_OMO=-1000.0
dimensión y hacia de búsqueda micro de omo

Vel_OMO=5000.00
velocidad' en mm/minuto de búsqueda micro

Vel_MARKER=5000.00
velocidad' en mm/minuto de búsqueda marker

Number_BRAKE_switch_port = ''
nombre de la salida que manda el freno eje

FalseTrue_logic_BRAKE_switch=0
estado salida por eje frenado

Time_BRAKE_Lock=0
tiempo en segundo de frenatura eje, sirve y freno activos,

Time_BRAKE_Unlock=0
tiempo en segundo de sfrenatura eje, sirve y freno activos,

Time_BRAKE_Enable=0
tiempo en segundo de sumisión habilitada antes de iniciar a desenfrenar

Number_PositiveOverTravel_switch_port = ''
nombre del input usado para sentir OVER TRAVEL POSITIVOS

FalseTrue_logic_PositiveOverTravel_switch=0
estado input con eje en OVER TRAVEL POSITIVOS

Number_NegativeOverTravel_switch_port = ''
nombre del input usado para sentir OVER TRAVEL RECHAZOS

FalseTrue_logic_NegativeOverTravel_switch=0
estado input con eje en OVER TRAVEL RECHAZOS

Number_MOV_READY_switch_port = ''

nombre de la salida que señala que el movimiento y' listo

FalseTrue_logic_MOV_READY_switch=1

estado salida por movimiento listo

Number_MOV_EXEC_switch_port = ''

nombre del input usado por outorizzare el movimiento

FalseTrue_logic_MOV_EXEC_switch=1

estado input por movimiento lícito

Number_MARKER_switch_port = "
nombre del input usado para leer la muesca de marker (STEPPER)

FalseTrue_logic_MARKER_switch=0
estado input con muesca de marker encontrado, STEPPER,
Number_DIRECTION_switch_port = "
nombre de la salida que indica dirección (STEPPER)

FalseTrue_logic_DIRECTION_switch=0
estado a punto de salida tener dirección positiva, STEPPER,

Number_STEP_switch_port = "
nombre de la salida que habilita la ejecución de 1 step (STEPPER)

FalseTrue_logic_STEP_switch=0
estado a punto de salida tener 1 step, STEPPER,

Number_BOOST_switch_port = "
nombre de la salida que habilita el aumento de corriente (STEPPER)

FalseTrue_logic_BOOST_switch=0
Con bit 0x10000000 de Kind_axis_1 = 0 (STEPPER)
estado a punto de salida tener el aumento de corriente
con eje en movimiento
Con bit 0x10000000 de Kind_axis_1 = 1 (STEPPER)
estado a punto de salida tener el aumento de corriente con
el eje en aceleración

MachineGroup=Abbinamento del eje a Grupo Maquina

TaskErrorPP=File "Task" ejecutado automáticamente cuando él
averigua una anomalía sobre el eje

TaskErrorName=Nome del Task dentro del expediente precisado por "**TaskErrorPP"**

TaskErrorModeRun=0; Modalidad de activación del TASK:
0=sempre (Estándard)
1=solo si es no activo ya

StepsEncoder = Número impulsos del Encoder, Sólo si se desea la recuperación de los
pérdidas de cuenta,

MaxStepsLost = Máximo número de impulsos Encoder perdido en la vuelta, no acumulativa,
antes de señalar error

MarkerSteps = Número impulsos del ancho del marker, Estándard 2 si la activa verificación de la pérdida pasa encoder,

Space_OMO_Micro_OFF = Espacio máximo que el eje puede cumplir en sentir el liberación de lo Micro de cero.

Space_OMO_Marker = Espacio máximo que el eje puede cumplir en buscar la muesca del Marker.

e=}

; Término parámetros eje

Con sumisión en velocidad' el KC es calculado, transforma los valores interiores en voltio digit, usando Volt_vel y Vel_max.
Con sumisiones en corriente KC es calculado Volt_vel usando y Acc_max.
Para no cambiar los valores de KP KI KD KD2 KS hace falta cambiar los valores usados manteniendo inalterado la relación entre ellos. y Acc_max.
Para no cambiar los valores de KP KI KD KD2 KS hace falta cambiar los valores usados manteniendo inalterado la relación entre ellos.

2.3.1.1 Parámetros relativos a los ases Tablero sinóptico

La caracterización de cada Eje inicia con el nombre sintético del eje encerrado entre 2 corchetes, en nuestro caso = [X1])

Notas Importantes
Los carácteres escritos en este campo pueden ser mayúsculos o minúsculos, pero en los dos casos son reconocidos como ases diferentes. , por ejemplo los Ases **X1 y x1 son dos ases diferentes,**
El nombre del eje puede ser a lo sumo de **7 carácteres**
El PHYSICAL_AXIS_NAME tiene que ser el mismo indicado al principio entre corchetes.
h) Por la elección de los valores de insertar en caracterización, se vea el capítulo "PID", en el que son tratadas todas las varias posibilidades de calibrado para Ases Analógicos, Digitales, con motor lineales, con motor Stepper etcétera
i) En la estructura siguiente son indicados los valores de estándard.

[X1]	
i=PHYSICAL_AXIS_NAME(1,"X1")	Número y nombre
b={	
Description="Asse X1"	
Kind_axis="ARM"	
Kind_axis_0=**0x00000000**	
Kind_axis_1=**0x001F0000**	
Kind_board_TransducerPosition=**40**	N° de reconocimiento del tipo de ficha
Board_number_TransducerPosition=**0**	Posición física del Back-plane en que se encuentra la ficha sobre que reside el canal Encoder de este Eje
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 Parámetros relativos a los ases virtuales

Los ases virtuales sirven a transformar coordenadas cartesianas, como expresadas por el programador, en movimientos de los ases físicos, no cartesianos, tales de realizar los desplazamientos a lo largo de y sobre los puntos del espacio cartesiano.
Pues la programación de un Robot SCARA o Cilíndrico es efectuada como si fuera un Robot Cartesiano y el Control provee en estos casos a correlacionar automáticamente las características físicas del ROBOT a las cuotas programadas.
Por la estructura siguiente, se comunican al Sistema las características físicas, Largo de los individuales bracci, etc, para darle las informaciones para ejecutar correctamente las transformaciones.
También por programación de Ases Virtuales se pueden efectuar interpolaciones Lineales y Circulares y Spline.
El tablero siguiente sirve a agrupar los ases reales que realizan el brazo del Robot, en las tipologías:

j) Brazo SCARA
k) Brazo SCARA Pantógrafo
l) Brazo Cilíndrica Lancha
m) Brazo Cilíndrico Traslante

TABLERO DE DEFINICIÓN DE LOS ASES VIRTUALES

[XV1]	Nombre del primer eje Virtual (XV1)
i=VIRTUAL_AXES_NAME(10,"XV1","YV1")	Número y nombres de los dos ases virtuales
b={	
Kind_axis="ARM"	NO MODIFIQUES
Kind_axis_0=**0x00000000**	0x000**1**0000 Sistema SCARA (*) 0x000**2**0000 SCARA Pantógrafo, compensado en positivo 0x000**4**0000 SCARA Pantógrafo, compensado en negativo, 0x000**8**0000 Cilíndrica Lancha 0x00**1**00000 Cilindrica Traslante
Kind_axis_1=**0x00000000**	NO MODIFIQUES
Axis_name_ascisse="XV1"	Nombre del eje Virtual selecto como **Abscisas**
Axis_name_ordinate="YV1"	Nombre del eje Virtual selecto como **Ordenado**
Axis_name_1="A1"	1° Eje físico Es: **A1]** (*)
Axis_name_2="B1"	2° Eje físico Es: **[B1]** (*)
Axis_name_3="C1"	3° Eje físico Es: **[C1]** (*)
Axis_name_4="X1"	4° Eje físico Es: **[X1]** (*)
Lenght_arm_1=200.0	Primera parte largo Brazo (**)
Lenght_arm_2=200.0	Segunda parte largo Brazo (**)

Abs_offset_A=100.0	Valor de la abscisa de la compensación Absoluta (***)
Vel_max_axis_A=20000.0	Velocidad max del eje virtual a lo largo de las Abscisas
Acc_max_axis_A=200.0	Aceleración max del Assse virtual a lo largo de las Abscisas
Dec_max_axis_A=200.0	Desaceleración max del Assse virtual a lo largo de las Abscisas
Abs_offset_O=100.0	Valor dell de la compensación Absoluta (***)
Vel_max_axis_O=20000.0	Velocidad max del eje virtual a lo largo de las Ordenadas
Acc_max_axis_O=1000.0	Aceleración max del Assse virtual a lo largo de las Ordenadas
Dec_max_axis_O=1000.0	Desaceleración max del Assse virtual a lo largo de las Ordenadas
e=}	

(*) A1, B1, C1, X1 son ejemplos de nombres determinados a los Ases físicos. Cada uno de estos ases físicos tiene que ser tarado en su Estructura Ases que por ejemplo inicia con [A1] etcétera
Los ases rotatorios, por ejemplo por el SCARA = A1, B1, C1, tiene que ser caracterizados en grados en la voz **Step=-0.001,** El valor de estándard = –0,001 es típico de ases lineales. Para los ases rotatorios hará falta insertar un valor tal que, teniendo en cuenta etc las relaciones de transmisión, haga corresponder 1 vuelta de brazo con 360°. El hacia de cuenta del Encoder del motor tiene que ser positivo cuando el brazo gira a izquierdas en sentido, como de normas de la trigonometría.)

(* *) Los ases lineales deben Robot tienen que tener todas las mismas unidades de medida, sea por cuánto concierne los Largo, las Compensaciones, las Velocidades, las Aceleraciones.

(* * *) Valor de la abscisa y la ordenada de la compensación Absoluta (Unidad de medida de los Ases Lineales)
El origen de los ases Virtuales que realizan el Robot, puede ser shiftato con respecto del origen por las dos voces **Abs_offset_A = y Abs_offset_O =**

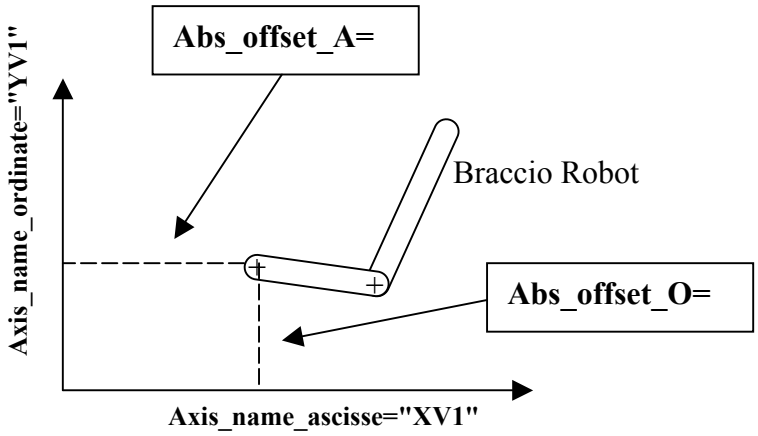
2.3.1.3

2.3.1.4

2.3.1.5

2.3.1.6

2.3.1.7



2.3.1.8 NOTAS VARIAS SOBRE LOS SISTEMAS VIRTUALES

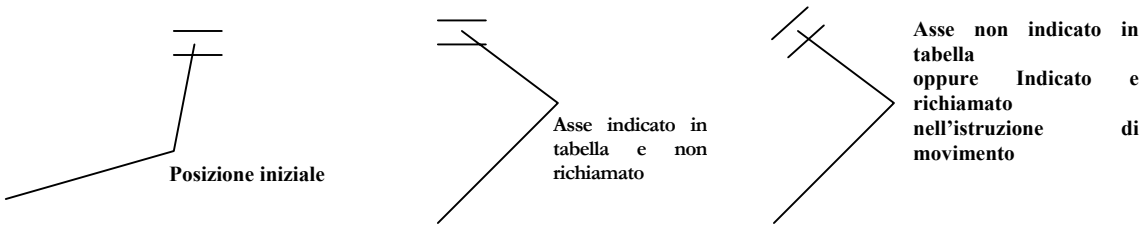
Máxima velocidad: corresponde a la máxima velocidad querida sobre el eje virtual; si esta velocidad fuera imposible que efectuar, hace falta tener presente que los dos ases reales juntados al eje virtual tienen cada uno una máxima velocidad, por estándard es utilizada el principio posible.

Máxima aceleración y máxima desaceleración: corresponden a la máxima aceleración y desaceleración querida sobre el eje virtual; también por este fechas si los valores indicados fueran superiores a lo posible, por estándard son utilizados los máximos posible.

Tercer Eje - En el caso esté presente el tercer eje, en el ejemplo C1 del SCARA, el eje pinza tiene que ser concéntrico con el eje de rotación del tercer eje, en este caso el largo del segundo eje está fijo.

Si en cambio el eje pinza el largo del segundo eje no fuera concéntrico, con respecto del eje pinza, sería variable, siendo dependiente de la posición angular de la pinza. Por consiguiente, ya que se impuesta un solo largo, el sistema cartesiano conseguido sólo será válido en las condiciones de concentricità.

Nombre del tercer eje: indica el eventual eje rotatorio de la muñeca, sitio a la extremidad del segundo eje;, en el ejemplo C1.
La indicación o menos que este eje en tablero, influyen sólo en la geometría del sistema a los efectos de la compensación automática de dicho rotatorio y precisamente:
La compensación automática consiste en mantener automáticamente la pinza de la muñeca paralela a su posición de salida. Eso permite de equiparar el desplazamiento de la pinza al de un sistema cartesiano.



3°Asse indicado en la estructura Virtual: en caso de requerido movimiento ases virtual tal eje rotatorio, si no solicitara en el movimiento, viene automáticamente aglomerado y la pinza apuesta sobre tal eje queda por lo tanto siempre paralela a él mismo. Si solicitara en el movimiento, el eje girará, de modo coordinado con el movimiento de los otros ases, hasta la posición angular programada.

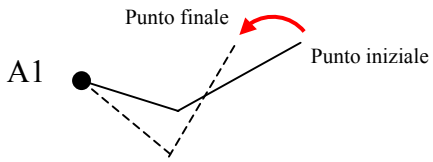
Vínculos y criterios de programación para ases virtuales

- Con un sistema de Ases Virtuales, en la misma instrucción de movimiento, los primeros dos Ases, A1 y B1, tienen que siempre ser ambos virtuales o bien físicos, pero nunca uno virtual y el otro físico.
  Ejemplo de instrucción errada: **MOV/XV,50,B1,20,** XV es virtual mientras B1 es físico,

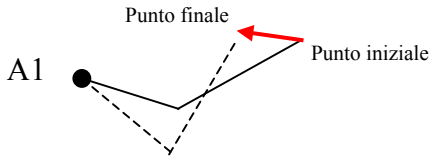
  Y viceversa posible solicitar juntamente ases virtuales al 3° e/o 4° eje real del sistema.
  Es: **MOV/XV,50,YV,100,C1,25**

- El 4° eje (traslante), es utilizado en las interpolaciones virtuales pero tiene que ser programado con valores explícitos, indicados en la instrucción de movimiento,
-
- Programando la instrucción de MOV sin expresar la velocidad, el movimiento de los ases virtuales para alcanzar la cuota final no sigue un trayecto rectilíneo pero aquella natural.
  Expresando en cambio la velocidad, el movimiento sigue un trayecto lineal sobre el plan Virtual

  MOV sin velocidad explícito Es: **MOV/XV1,80,YV1,40 (**coordenadas del punto final)



MOV con velocidad explícito Es: **MOV/1000:XV1,80,YV1,40 (**coordenadas del punto final)

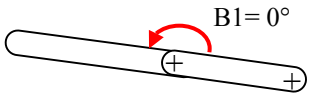


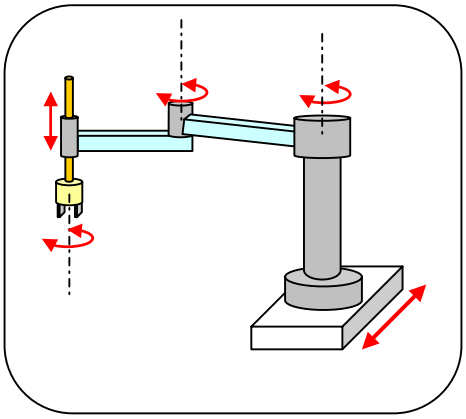
2.3.1.9 El sistema SCARA

El sistema SCARA es definido caracterizando **Kind_axis_0=0x00010000**

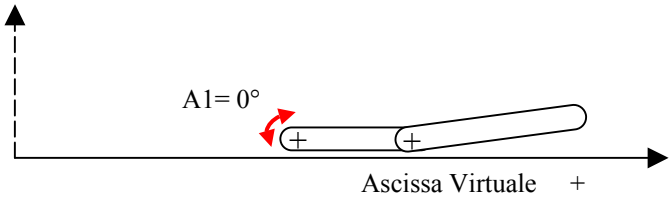
Vínculos de Calibrado

n) Con el eje rotatorio B1 a Cero grados (0°) los dos bracci tienen que ser alineados y extendidos





Con el eje rotatorio A1 a Cero grados, 0°, se define la dirección y el hacia de la Abscisa Virtual

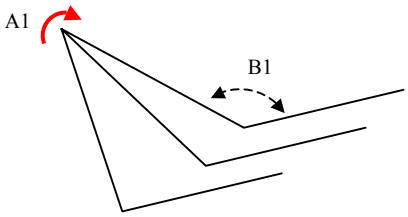


El calibrado **Axis_name_1** = "A1" define el eje físico A1, 1° rotatorio-obligatorio por SCARA,
El calibrado **Axis_name_2** = "B1" define el eje físico B1, 2° rotatorio-obligatorio por SCARA,
El calibrado **Axis_name_3** = "C1" define el eje físico rotatorio de la muñeca, si existe.
El calibrado **Axis_name_4** = "X1" define el eje traslante, si existe

Nota - Si es usado el eje traslante X1, tiene que ser paralelo y concorde a la Abscisa Virtual.
El valor **Lenght_arm_1=200.0** es el largo del brazo entre el centro de rotación de A1 y B1
El valor **Lenght_arm_2=200.0** es el largo del antebrazo entre el centro de rotación de B1 y el eje de la mano, o bien, si existe, el centro de rotación de C1,

2.3.1.10 SCARA PANTÓGRAFO

Constructivamente el SCARA pantógrafo, al mover eje A1, 1° rotatorio, mantiene mecánicamente el antebrazo siempre paralelo a él mismo. En realidad el asseB1, 2° rotatorio, rueda, arrastrada mecánicamente, en sentido opuesto del mismo rincón mandado por A1.
De este modo el movimiento es aquel esquemáticamente representado en figura



Notas sobre la gestión SCARA pantógrafo
Toda la estructura de la gestión de los ases virtuales es aquel del SCARA.
El único campo diferente es aquel relativo al **Kind_axis_0** = que tiene que ser caracterizado como sigue:

Kind_axis_0=0x00020000 SCARA Pantógrafo, compensado en positivo,
Kind_axis_0=0x00040000 SCARA Pantógrafo, compensado en negativo,

El hacia de la compensación debe ser entendida en el modo siguiente:

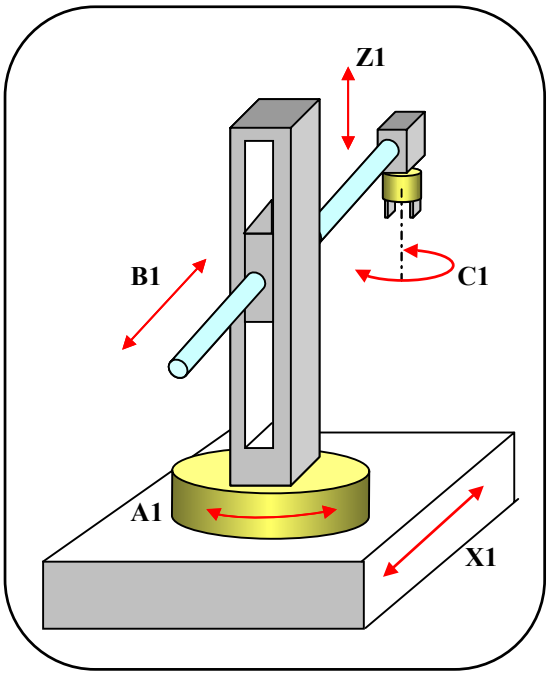
Con compensación positiva se quiere compensar el efecto pantógrafo sumando B1 el rincón efectuado por el eje A1.

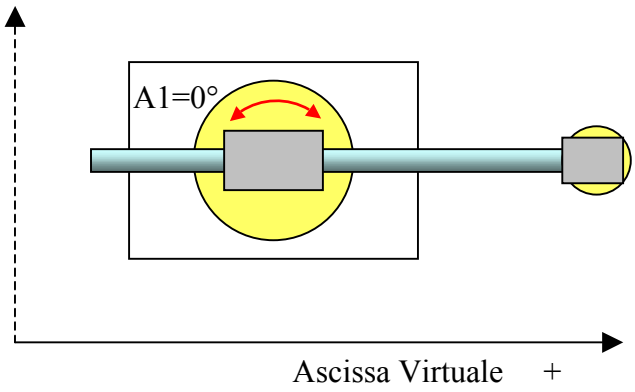
Con compensación negativa se quiere compensar el efecto pantógrafo sustrayendo al eje B1 el rincón efectuado por el eje A1.

2.3.1.11 Sistema CILÍNDRICA LANCHA

El sistema CILÍNDRICA LANCHA es definida **Kind_axis_0=0x00080000** caracterizando

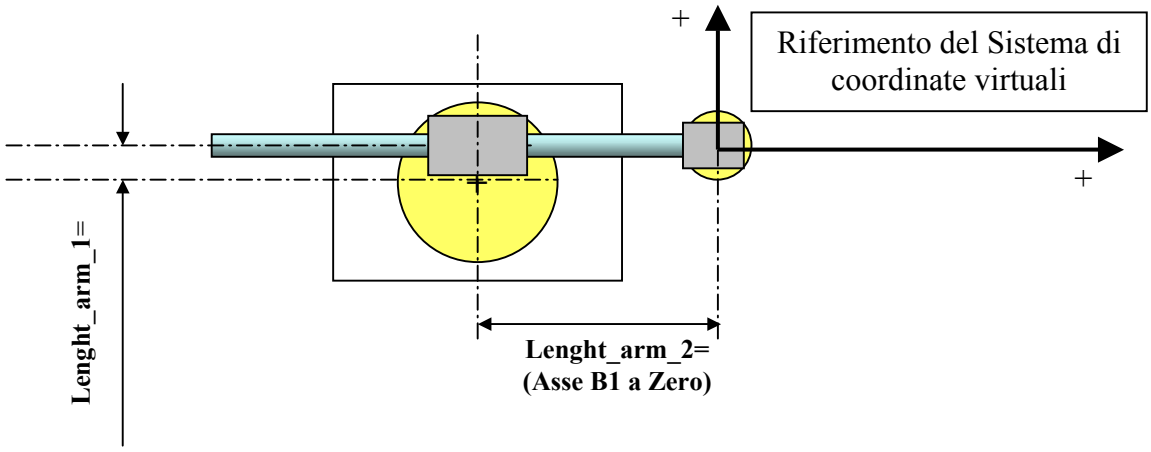
Con el eje rotatorio A1 a Cero grados, 0°, se define la dirección y el hacia de la Abscisa Virtual





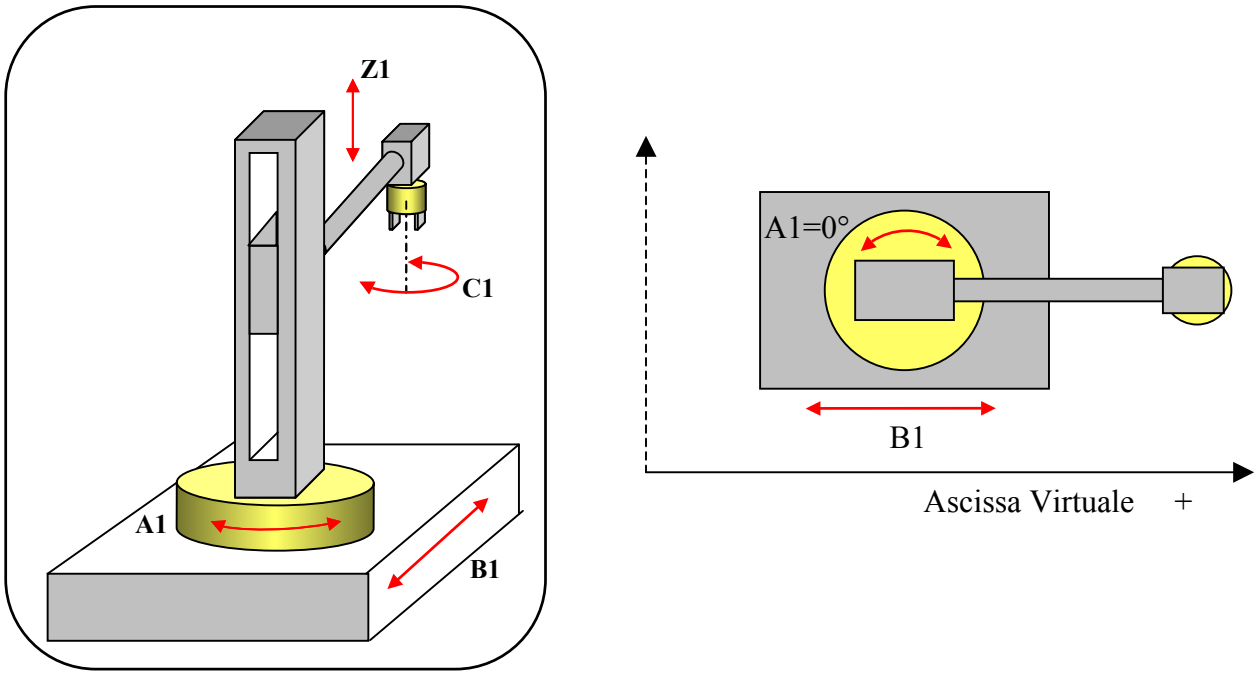
El calibrado **Axis_name_1** = "A1" define el eje físico A1, 1° rotatorio-obligatorio por CILÍNDRICO,
El calibrado **Axis_name_2** = "B1" define el eje físico B1, Lancha-obligatorio por CILÍNDRICO,
El calibrado **Axis_name_3** = "C1" define el eje físico rotatorio de la muñeca, si existe.
El calibrado **Axis_name_4** = "X1" define el eje traslante, si existe

Nota - Si es usado el eje del carro traslante X1, tiene que ser paralelo y concorde a la Abscisa Virtual.
El valor **Lenght_arm_1=0** es el valor del **disassamento** entre el centro de rotación de A1 y el eje lancha.
El valor **Lenght_arm_2=200.0** es la distancia entre él Cero del eje lancha y el centro de rotación del asseA1

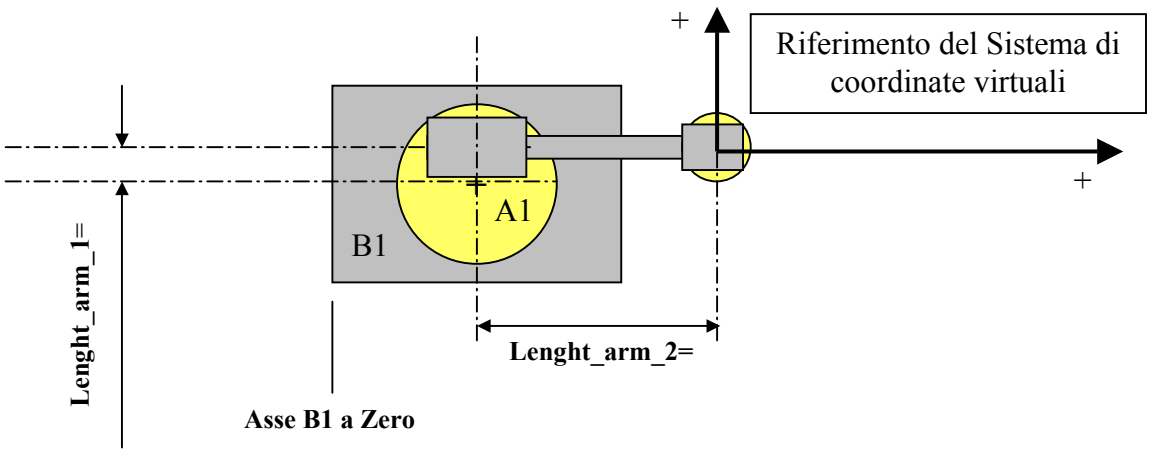


2.3.1.12 Sistema CILÍNDRICO TRASLANTE

El sistema CILÍNDRICO TRASLANTE es definido **Kind_axis_0=0x00100000** caracterizando
Con el eje rotatorio A1 a Cero grados, 0°, se define la dirección y el hacia de la Abscisa Virtual



El calibrado **Axis_name_1** = "A1" define el eje físico A1, 1° rotatorio-obligatorio por CILÍNDRICO,
El calibrado **Axis_name_2** = "B1" define el eje fisicoB1, Carro-obligatorio por CILÍNDRICO,
El calibrado **Axis_name_3** = "C1" define el eje físico rotatorio de la muñeca, si existe.
El calibrado **Axis_name_4** = "X1" EN ESTE CASO no EXISTE
Nota - El eje del carro traslante es B1 y tiene que ser paralelo y concorde a la Abscisa Virtual.
El valor **Lenght_arm_1=0** es el valor del **disassamento** entre el centro de rotación de A1 y el eje lancha.
El valor **Lenght_arm_2=200** es la distancia entre el centro del eje de rotación de la muñeca y el centro de rotación del eje A1


+
Riferimento del Sistema di coordinate virtuali
+
A1
B1
Lenght_arm_1=
Lenght_arm_2=
Asse B1 a Zero

2.3.1.13 Notas sobre los ases stepper

Los ases stepper es definido en caracterización utilizando
los siguientes campos de caracterización eje

Kind_axis_1=0x10070000
; Máscara 1 a bits del eje
;0x10000000 = BOOST ACTIVADO EN ACC. , SÓLO CON EJE STEPPER,

Kind_board_TransducerPosition =
; Tipo de lámina transductor
;1000 = EJE POR STEPPER A. BITS
;1002 = EJE POR STEPPER A. CONTADOR PCI2000
;1003 = EJE POR STEPPER A. CONTADOR PCI2001
;1100 = EJE POR STEPPER RSM304 ETN SERVO P.E.
Board_number_TransducerPosition=0
; Número de lámina por transductor
;1000 unused STEPPER A. BITS
;1002 0 - n numero lámina PCI2000
;1003 0 - n numero lámina PCI2001
;1100 unused EJE POR STEPPER RSM304 ETN SERVO P.E.
Address_port_TransducerPosition=0x0
; Dirección eje transductor
;1000 unused
;1001 unused
;1002 unused
;1003 unused
:1100 6 - n dirijo DRIVER CONTROL modulo, M-1)*2
4 módulos consecutivos con M>=4
Physical_number_Transducerposition =
; Número físico eje transductor
;1000 unused STEPPER A. BITS
;1002 1 - 3 EJE POR STEPPER A. CONTADOR PCI2000(pulse 1 - 3,
;1003 1 - 3 EJE POR STEPPER A. CONTADOR PCI2001, pulse 1 - 3,
:1100 1 - 3 Numero motor

Kind_board_DAC =
; Tipo de lámina eje convertidor
;1000 = EJE POR STEPPER A. BITS
;1002 = EJE POR STEPPER A. CONTADOR PCI2000
;1003 = EJE POR STEPPER A. CONTADOR PCI2001
;1100 = EJE POR STEPPER RSM304 ETN SIRVO

Board_number_DAC=0
; Número de lámina eje convertidor
;1000 unused STEPPER A. BITS
;1002 0 - n numero lámina PCI2000
;1003 0 - n numero lámina PCI2001
;1100 unused EJE POR STEPPER RSM304 ETN SIRVO

Address_port_DAC=0
; Dirección eje convertidor
;1000 unused STEPPER A. BITS
;1002 unused EJE POR STEPPER A. CONTADOR PCI2000
;1003 unused EJE POR STEPPER A. CONTADOR PCI2001
;1100 6 - n dirijo DRIVER CONTROL modulo, M-1)*2
4 módulos consecutivos con M>=4

Physical_number_DAC=2
; Número físico eje convertidor
;1000 unused STEPPER A. BITS
;1002 1 - 3 STEPPER A. CONTADOR PCI2000, pulse 1 - 3,
;1003 1 - 3 STEPPER A. CONTADOR PCI2001, pulse 1 - 3,
;1100 1 - 3 Numero motor

KP=10.0
renta proporcional (error de posición usado a eje firme)

KD=100
Para ases stepper define la frecuencia de sumisión (Hz)

KD2=400
Para ases stepper define la frecuencia de scavalcamento resonancia, Hz,

KS=0.01
Valor de truncamiento por el PID. Tiene que ser igual o mayor del paso del motor a stepper

KM=0.01
Para ases stepper mixtos ("Kind_board_TransducerPosition =... y "Kind_board_DAC=1000", este campo tiene que representar el valor de un step del motor, porque' el campo Step = y' usado por el encoder Con ases stepper que tienen "Kind_board_TransducerPosition=1000" y "Kind_board_DAC=1000" es usado el solo valor de "Step =.. y luego "KM =.. no y' usado. , Cuando y' ocasión viene en todo caso siempre automáticamente hecho absoluto,

KN=0
Para ases stepper a contador define la frecuencia de base del contador en Hz / / ACTUALMENTE no SIRVE RESERVADO A OTROS EMPLEOS FUTUROS

Volt_max=12000
máxima "frecuencia" en Hz. Con los ases stepper no es controlado la tensión pero la frecuencia máxima que puede soportar el motor

Volt_vel = 10
"frecuencia" necesaria para moverse a Vel_max
Usado por los plottature del mando (frecuencia) que y' visualizado en volts. Si el valor y' la frecuencia necesaria para ir a vel_max el valor visualizado y' la frecuencia. Si el valor y' diferente todo los frecuencias son visualizadas en proporción. Eso' permite de programar valores que puedan ser plottati.

KC_Parameter = 0
Este parámetro permite de definir la relación de conversión por la sumisión, en alternativa a Volt_vel Vel_max.
Si y' cero son utilizados de otro modo los dos campos susodichos viene usado éste.
El sentido físico del campo y' la velocidad' (u/min) por 1 Hces o por valores proporcionales a la frecuencia. Si por ejemplo se quieren tener 10 con una velocidad' 600 el valor sara' 600/10. Este campo sirve para poner el valor intencional por el "plottatura"; la señal puede servir a invertir la dirección del eje.

Step = +0.001
multiplicador empujado -> coordinado cálculo, En stepper es forzado automáticamente en absoluto,

StepsEncoder = Número impulsos del Encoder, Sólo si se desea la recuperación de los pérdidas de cuenta,

MaxStepsLost = Máximo número de impulsos Encoder perdido en la vuelta, no acumulativa,
antes de señalar error

MarkerSteps = Número impulsos del ancho del marker, Estándard 2 si la activa verificación de la pérdida pasa encoder,

Vel_max=3000
Velocidad' máxima del eje
La velocidad' tiene que ser <= a KM * max_Hz * 60

Number_MARKER_switch_port = "
nombre del input usado para leer la muesca de marker

FalseTrue_logic_MARKER_switch=0
estado input con muesca de marker encontrado
Number_DIRECTION_switch_port = "
nombre de la salida que indica dirección
FalseTrue_logic_DIRECTION_switch=0
estado a punto de salida tener dirección positiva
Number_STEP_switch_port = "
nombre de la salida que habilita la ejecución de 1 step
En el caso de STEPPER PCI2000 no es usado, pero la corresponsal salida PULSE
En el caso de STEPPER RSM304 no es usado, pero el corresponsal bit de salida.
FalseTrue_logic_STEP_switch=0
estado a punto de salida tener 1 step
Number_BOOST_switch_port = "
nombre de la salida que habilita la reducción de corriente
FalseTrue_logic_BOOST_switch=0
estado a punto de salida tener la reducción de corriente

Por la justa señal de las "CUOTAS" actuar cambiando la señal a "Step" =

Para cambiar la dirección al motor existen dos modos; cuando ella dirección y' fecha de una salida digital se puede invertir el estado de la salida "FalseTrue_logic_DIRECTION_switch = ". O bien cambiar el le señalo a "Volt_vel" = o a "KC_Parameter." Este y' el sólo modo posible cuando la dirección no y' dada por una salida digital, cod 1100 con EJE POR STEPPER RSM304 ETN SERVO P.E.

ALGORITMOS STEPPER "BITS"

TC = tiempo muestreo en milisegundos

TickCounter = ciclos de counter, 1 - n,

Frecuencia en Hz = 1000 * TickCounter /, TC * 2 * Pulse,

KM = incremento por paso del motor STEPPER

Vel Reg. en SAMPLE = KM * Frecuencia * TC / 1000

Vel Reg. en SAMPLE = KM * TickCounter /, 2 * Pulse,

KN = Vel Max en SAMPLE = KM * TickCounter /, 2 * 1,

Vel Reg. en SAMPLE = KN / Pulse

Pre.Max =, KM * TC * Freq.Max, / 1000

Pulse =, 1000 * TickCounter /, Frecuencia * TC * 2,

ALGORITMOS STEPPER "PCI2000" Y "RSM304"

La gestión de la frecuencia y par; Tclk y' siempre a 1 con RSM304

Período en microsegundos = Tclk * 2 *, Pulse + 1,

Frecuencia en Hz = 1000000 /, Tclk * 2 *, Pulse + 1,)

TC = tiempo muestreo en milisegundos

KM = incremento por paso del motor STEPPER

KN = Vel Max en SAMPLE =, 1000 * KM * TC, /, Tclk * 2 *, 0 + 1,)

Vel Reg. en SAMPLE =, 1000 * KM * TC, /, Tclk * 2 *, Pulse + 1,)

Vel Reg. en SAMPLE = KN /, Pulse + 1,

Pre.Max =, KM * TC * Freq.Max, / 1000

Pulse =, 1000000 /, Frecuencia * Tclk * 2,) - 1

2.3.1.14 NOTAS SOBRE LOS ASES SERCOS

Los ases SERCOS en velocidad' es caracterizado poniendo el valor 200 en los campos "Kind_board_TransducerPosition" = y "Kind_board_DAC" =
En la gestión de estos ases el valor de tensión de dar al DAC no y' real, por tanto' y' buena norma considerar de trabajar con 10 Voltios hipotéticos, de modo que los gráficos por la puesta a punto da indicaciones en el range que normalmente y' usado, 10 Voltios.
Eso' comporta la carga de los siguientes campos en el modo qué sara' descrito:

El campo "K_DAC_1Volt" = tiene que ser cargado con el valor "SALIDA" que SERCOS utiliza para mover el eje a un décimo de la velocidad' máxima.

"Volt_max=11" será cargado con 11 Voltios. El valor no debe borde la velocidad' máxima que se entiende conseguida con 10 voltios.

En "KC_Parameter" = será puesta la velocidad' conseguida con 1 voltios, que y' la velocidad' máxima dividida 10.

En "Step" = la relación entre los unidos' SERCOS y aquellas usáis en AxesBrain. De costumbre el valor y' 0.0001; o sea si él caracteriza el eje en mm el step y' un décimo de micrómetro.

"Vel_max" = y' el valor de velocidad' máxima, que y' hipotéticamente conseguida con 10 Voltios.

E.G.
Imaginamos de tarar un eje con velocidad' máxima de 3000 mm/min, con unido' SERCOS paras a 0.0001 mm y SALIDA SERCOS de 30000000 step sercos/min (3000*10000), los campos serán cargados con los siguientes valores:

```
[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=}

2.3.1.15 NOTAS SOBRE LOS ASES ROLLOVER

Para definir un eje como ROLLOVER y' necesario settare
Kind_axis_1 = el bit 0X00000008, ROLLOVER 0 +)
y al mismo tiempo cargar con el valor del módulo
el campo Revolution =, que no debe ser cero.
La visualización del eje y' siempre incluida entre
0 y +MODULO; para visualizar la cuota absoluta usar
- RAV/42:
Por estándard los movimientos van en la dirección mandado,
hasta a alcanzar la posición final, que y' fecha en
coordinados rollover.
Usando la instrucción - CAP/100:A1,32 todos los movimientos
qué siguen son hechos recorriendo lo espacio menor.
No es recorrido un valor> de MÓDULO también dando
um movimiento muy grande.
Esta instrucción queda activa hasta tiene que no venga
anulada de - CAP/100:A1,-32 o del relance de
AxesBrain. También la reposición no cambia el estado.
Ej. eje con módulo 360 visualizzante 60; mandando
3670 (360*10+70) el eje se lleva a 70 recorriendo 10.

Por tanto' si es leída la cuota con - RAV/42: y luego
movido el eje a esta cuota ello queda firme. Éste
sólo utilizando la función "recorrida menor" (- CAP/100:A1,32,

Recordar que no utilizando la función "recorrida
menor" (- CAP/100:A1,32, los movimientos recorren todo
el espacio mandado.

Ej. eje con módulo 360 visualizzante 60; mandando
3670 (360*10+70) el eje se lleva a 70 recorriendo 3610.

También existe el eje ROLLOVER SEÑALADO, que se puede'
activar con Kind_axis_1 = el bit 0X00000010, ROLLOVER + - MÓDULO,
En este caso las cuotas varían entre + - módulo
y no funciona - CAP/100:A1,32. Además no siendo
verdaderamente útil, podría ser una prestación
de anular.

veramente utile, potrebbe essere una prestazione
da annullare.

2.3.1.16 NOTAS SOBRE LOS GANTRY

Son administrados dos tipos de GANTRY:

Aquel con estructura rígida que no y' ingallonabile;
son conformados los motores a tener la misma carga.

Aquel clásico que tiene que ser conducido por los dos motores,
respetando el más posible por cada eje la posición.

Con el tipo rígido el eje máster y mandado a la cuota intencional,
el otro eje se acostumbra.
Es caracterizado poniendo sobre el eje máster en
Kind_axis_1=il bit 800000H (SLAVE_CHA_GANTRY)
Axis_name_slave = "nombre eje eslavas"
K_follower_slave=100.0, 100 significa gestión indirecta, 0 no mediado,

El tipo clásico tiene las mismases informaciones, con la añadidura en
Kind_axis_1=il bit 1000000H (SLAVE_CHA_HARD)
Con este tipo también hace falta asignar la máxima tensión de
descompensación entre los dos ases que y' el campo de caracterización
VoltGantry=5.0
Si el valor de VoltGantry y' <=0 la descompensación entre los dos ases
no y considerado:

Además hace falta definir la desviación de los cero marker eslavos
con respecto del cero marker del máster con el siguiente campo
de caracterización
DeltaOffset_SlaveGantry=0.640
Estos dos campos son puestos sobre el eje máster

Nota por "DeltaOffset_SlaveGantry"
eje máster sobre cero marker con lectura de la cuota real = 0
eje eslavas con lectura de la cuota real = +2.403
El valor de caracterizar tiene que ser -2.403 que y' ella
posición del cero marker eje eslavas con respecto de lo correspondiente
del máster.
También poniendo el
Kind_axis_1=il bit 2000000H (SLAVE_CHA_THEOR)
el gantry asigna todo el error de cruce sobre el eje eslavas; en
tal modo el máster sigue integralmente el teórico y el eje eslavas
acelera o decelera para anular el error entre los ases.
Usando este bit, también por gantry rígido, el eje eslavas sigue
el teórico y no se acostumbra..

La puesta a cero de los ases GANTRY y' bien hacerlo con un part-program,
qué y' diferente a según de los tipos físicos de gantry.

Eje gantry no rígido con ases que no se libran
Caracterización eje máster 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

Como calcular "DeltaOffset_SlaveGantry" =
Lanzar el programa siguiente, esperar que la puesta a cero
sea hecho y a la solicitud de movimiento incremental de eje
máster dar los valores que ponen los ases gantry en posición
de trabajo. Ligeras sobre la raya 2 el valor de introducir en
caracterización.

```
- TSK/READ_GANTRY   ; Lanzamiento del task de lectura ases gantry
- CAP/100:YM,1024        ; Disabilitazione de SLAVE_CHA_HARD
- OMO/YM           ; Búsqueda micro y marker eje máster
- MOV/YM,100     ; Posicionamiento eje por puesta a cero eje eslavas
- CAP/0:YM,262144        ;0x40000 revers máster eslavas del gantry
- OMO/YS           ; Búsqueda micro y marker eje eslavas
- CAP/0:YM,-262144       ; - (0x40000) vuelta al máster eslavas del gantry
- CAP/100:YM,2048        ; Disabilitazione temporal 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
```

Como ejecutar la puesta a cero del sistema de ases gantry
Lanzar el programa siguiente

- OMO/YM ; Búsqueda micro y marker eje máster
- MOV/YM,100 ; Posicionamiento eje por puesta a cero eje eslavas
- CAP/0:YM,262144 ;0x40000 revers máster eslavas del gantry
- OMO/YS ; Búsqueda micro y marker eje eslavas
- CAP/0:YM,-262144 ; - (0x40000) vuelta al máster eslavas del gantry
- MOV/YM,200 ; Posicionamiento eje por fina puesta a cero
- END

Como sacar los ases gantry del ingallonamento
Lanzar el programa siguiente y mover el máster
hasta a tener los ases móviles

- CAP/100:YM,4096 ; El eje YM también mueve sin HOME
- CAP/100:YS,4096 ; El eje YS también mueve sin HOME
- CAP/100:YM,2048 ; Disabilitazione temporal del gantry

REP - KYB/MOV INCREM. YM,L1
- INQ/YM
- MOV/500:YM,L1
- ABQ/YM
- JMP/REP
- END

También poniendo el
Kind_axis_0=il bit 0x00000020, EJE CON SALIDA DULCE DEL SIRVO,
con el GANTRY de tipo no rígido, sometiendo los dos ases se llevan
en posición más dulcemente y el control de collision no viene
hecho por todo el tiempo de "Cycles_positioning."
Durante el movimiento, si es superado un sposizionamento entre
los dos ases del valor de sirvo error es señalado error.

2.3.1.17 NOTAS SOBRE LA GESTIÓN DE OVER TRAVEL

EL OVER TRAVEL puede ser definido en caracterización
poniéndole en Positive_over_travel = y Negative_over_travel =
los valores de fina carrera referidos al cero absoluto. Luego
si Offset_axis es cambiado = y' necesario también cambiar
los límites de over travel porque' es cambiado la compensación
absoluto.
Estos límites son activados por la puesta a cero ases; moviendo
en manual el eje se para sobre el límite que se quiere superar.
Los demás movimientos no parten si se supera el límite de
corrida, pero da una señal de error.
Si ambos los límites son cero, no es considerado el
límite de OVER TRAVEL.
Animas "hasta" a OVER TRAVEL
NOTAS por los movimientos lanzados por HANDLING_MOV
Poniendo en el campo MODIFYER_MOV de la estructura de los movimientos
el valor 0x4000 (or con los otros bites del campo) los movimientos
son ejecutados hasta al límite si éste es superado. No
son tratados de este modo los movimientos en continuo y
circulares y HOM.

Utilizando los campos:
Number_PositiveOverTravel_switch_port =
FalseTrue_logic_PositiveOverTravel_switch =
Number_NegativeOverTravel_switch_port =
FalseTrue_logic_NegativeOverTravel_switch =
también son sentidos los micros de fina carrera. Ellos son
administrados en el siguiente modo:
1, caso normal que para prescindir de la puesta a cero ases
es señalado error con caída de sumisión
si quieren superarlos, la dirección del movimiento
es probada con el corresponsal micro de más allá de carrera,
2, caso con Kind_axis_0 = que tiene el bit 0x00000040 settato;
a prescindir de la puesta a cero ases el movimiento en manual
cuando siente lo micro decelera y se para sin nadie
señal.
La misma cosa sucede con los movimientos "hasta" a OVER TRAVEL
Por todos los demás movimientos vale el control OVER TRAVEL
con los límites en caracterización; si es encontrado el
micro de más allá de carrera es señalado error con caída de
sumisión.

2.3.1.18 NOTAS SOBRE LA GESTIÓN DE ASES NORMALMENTE FRENADA

El eje normalmente inhabilitado y' un eje que cuando y' paro no
y' sometido y si tiene el freno y' frenado. Con movimiento ello
él autoasserve liberando el eventual freno hasta a movimiento
concluido.
El eje puede ser caracterizado tal settando en
Kind_axis_0 = el bit 0x00000080

También de part-program se puede' attivare/disattivare sobre el eje
la función de normalmente inhabilitado con las siguientes instrucciones:

1, CAP/100:A1,512 de este modo el eje y' enviado
normalmente inhabilitado, pero después de una reposición
vuelve en el estado de caracterización
2, CAP/100:A1,-512 de este modo el eje no y'
normalmente inhabilitado, pero después de una reposición
vuelve en el estado de caracterización
3, CAP/258:A1,512 de este modo el eje y' enviado
normalmente inhabilitado, pero también después de una reposición
queda en este estado
4, CAP/258:A1,-512 de este modo el eje no y'
normalmente inhabilitado, pero también después de una reposición
queda en este estado

2.3.1.19 NOTAS SOBRE EL MOVIMENTAZIONE CON GESTIÓN A. BIT

Por la gestión con bit de los movimientos hace falta poner sobre SISTEMA.TXT
los siguientes campos en las informaciones ASES.

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 = define como la salida usada
indicador de eje en movimiento o listo a mover.
Number_MOV_EXEC_switch_port = define el input que permite
el movimentazione verdadero de aquel eje. El movimiento queda en
wait hasta a que el input sea como settato
FalseTrue_logic_MOV_EXEC_switch.
El movimiento ejecutado acaba, señalando los estados de eje para,
pero no notifica su fin si el bit input no viene resettato.

RESUMIENDO:
Number_MOV_READY_switch_port = permite de saber
cuando un eje y' en movimiento, pero no actúa sobre su ejecución.
Number_MOV_EXEC_switch_port = permite de administrar el desarrollarse del
movimentazione; el movimiento tiene que encontrar todos los ases del
movimiento, con bit input definido, settati para partir, y a movimiento
terminado todos tienen que ser resettati para notificar de ello el fin.
Cuando el movimiento y' en continuo la gestión de
Number_MOV_READY_switch_port = y' inalterada.
Number_MOV_EXEC_switch_port = en cambio no permite los movimientos del
continuo sólo la primera vez que es lanzado un movimiento con
al menos un eje con el input definido. Luego a bit settato, todo los
movimientos son lanzados sin ningún control. Sólo la instrucción
de cierre continúo, HLC, espera que todos los bites, de todo él
ases que han hecho parte de aquellos continuo, sean antes resettati
de notificar continuo terminado.

La gestión MOV_EXEC puede ser inhabilitada con la partidaria
instrucción modal:
CAP/100:X1,0x40000
Usted puede rehabilitar con la función inversa:
CAP/100:X1,-0x40000

2.3.2 Mandriles

i=IO_NAME(1, "MAND1",
b =
Kind_sensor = "ANALOG_OUTPUT"
Kind_board=4
Rack_number=0
Address_port=0
Physical_number=6
y =

[M1]
; Nombre del mandril
SPINDLE_NAME(1, "M1",
; Número y Nombre del mandril

Kind_spindle = "SPINDLE"
; Define un mandril

Kind_spindle_0=0x00000000
; Tipo de mandril

Number_DAC_switch_port = "
1,
nombre de la salida analógica usado para mandar el mandril
2,
Si el nombre no existe ("o 0, es usado como mandril
l' eje definido con "Axis_SERVO_with_move" =

Number_ACD_switch_port = "
nombre del input analógico usado para leer la velocidad' del mandril
(unused)

NotifyEnabled=0
(unused)

Axis_SERVO_with_move = "
1,
Nombre eje que permite la rotación si y' sometido
Si el eje no y' definido la rotación y' siempre permitida
"Number_DAC_switch_port" = con el valor de una salida analógica
2,
nombre del eje usado como mandril
"Number_DAC_switch_port" = con el valor nulo ("o 0,

K_DAC_1Volt=3276.7
valor digital del DAC para conseguir 1 Voltios
poniendo 0 el valor y realzado automáticamente del código mandril

Volt_max=10.0
máxima tensión le fecha al DAC (Voltio)

Volt_turn=10.0
tensión (Voltio) necesaria para girarle a "Turn_max" = vueltas

KC_parameter=0
Este parámetro permite de definir la relación de conversión por
el mandril, en alternativa a Volt_turn Turn_max.
Si y' cero son utilizados de otro modo los dos campos susodichos viene
usado éste.
El sentido físico del campo son las vueltas (giri/min) por 1 voltios.

Volt_offset=+0.0
tensión de shift (Voltio) necesaria para hacer girar el mandril
a Turn_min; bajo el que el mandril y' detención

Turn_max=110000.0
giri/min conseguido con "Volt_turn" =

Acc_max=300.0
Acc. en giri/sec2

Dec_max=300.0
Dec. en giri/sec2

Turn_min=0.0
Giras min del mandril; debajo de estas vueltas es puesto 0

Delay_spindle=1
Tiempo de espera en segundo por vel. régimen alcanzado o por parada

Number_ENABLE_switch_port = ''
nombre de la salida por la capacitación del mandril

FalseTrue_logic_ENABLE_switch=1
estado salida por la capacitación del mandril

Number_DRVOK_switch_port = ''
nombre del input por driver mandril VALE

FalseTrue_logic_DRVOK_switch=0
estado input por driver mandril VALE

Number_REGIM_switch_port = ''
nombre del input por mandril a régimen (velocidad' alcanzada)

FalseTrue_logic_REGIM_switch=0
estado input con mandril a régimen

Number_STOP_switch_port = ''
nombre del input que señala el mandril firme

FalseTrue_logic_STOP_switch=1
estado input con el mandril firme

Number_LOCK_switch_port = ''
nombre de la salida que frena el mandril (MLTA)

FalseTrue_logic_LOCK_switch=1
estado a punto de salida frenar el mandril, MLTA,

Los Volts de darle al DAC del mandril son calculados considerando la rotación como función lineal del mando.

**VOLTIO = Volt_offset +, VELOCIDAD' INTENCIONAL - Turn_min, *
, Volt_turn - Volt_offset /, Turn_max - Turn_min,**

Si la VELOCIDAD' INTENCIONAL y' menor de Turn_min entonces VOLT = 0

Con el eje usado como mandril hace falta poner los parámetros eje y mandril de modo que sean correlacionados:

K_DAC_1Volt = y K_DAC_1Volt = tienen que ser iguales
Volt_vel = y Volt_turn = tienen que ser iguales
Vel_max = y Turn_max = tienen que estar en la misma relación de los unida' de medida representada.
es, con eje en grados, para tener las vueltas hace falta dividir la velocidad' eje con 360; por lo tanto 360000 gradi/min diventono 1000 giri/min
Acc_max = el mismo razonamiento de Vel_max
Dec_max = el mismo razonamiento de Vel_max
Todos los input/output de la estructura mandril no soy usados
Volt_offset = y Turn_min = son forzados a cero

2.3.2.1 Parámetros Mandriles Tablero sinóptico

La caracterización de cada Mandril inicia con el nombre sintético del mandril mismo encerrado entre 2 corchetes, en nuestro caso = [M2])

Notas Importantes
Los carácteres escritos en este campo pueden ser mayúsculos o minúsculos, pero en los dos casos son reconocidos como mandriles diferentes. , por ejemplo los Mandriles **M2 y m2 son dos mandriles diferentes,**
El nombre del mandril puede ser a lo sumo de **7 carácteres**
El SPINDLE_NAME tiene que ser el mismo indicado al principio entre corchetes
Si el Mandril le es a LOOP ABIERTO, (sin Encoder, es suficiente caracterizar toda la estructura siguiente, dedicada a los Mandriles.

Si en cambio se trata de un Mandril a LOOP CERRADO, (con Encoder, por la gestión del que se utiliza el canal DAC y Encoder de un Eje, hará falta tarar limitadamente toda la estructura Eje y luego la estructura Mandril siguiente a los las voces indicadas (*), dejando los valores de estándard en los rimaneti vocea de tablero.

> Un mandril administrado como un Eje deberá por tanto ser tarado como Eje y también como Mandril.

Ejemplo de caracterización del Mandril M2

[M2]	(*)
i=SPINDLE_NAME(2,"M2")	(*)
b={	NO MODIFIQUES
Description="Mandrino M2"	(*)
Kind_spindle="SPINDLE"	NO MODIFIQUES
Number_DAC_switch_port="DACM2"	A segunda que el mandril sea a loop abierto, sin Encoder, o bien a loop cerrado, con Encoder, actuar como sigue: Loop abierto-dar el nombre de la salida analógica selecta en el calibrado de las Salidas analógicas Loop cerrado-indicar" ", sin nombre porque se usa un canal de un eje con Encoder, definido en **Axis_SERVO_with_move =** (*)
Number_ACD_switch_port=""	NO UTILIZADO
Axis_SERVO_with_move=""	A segunda que el mandril sea a loop abierto, sin Encoder, o bien a loop cerrado, con Encoder, actuar como sigue: Loop abierto-Si el eje indicado con el nombre

	en este calibrado es bajo no controlo, el Mandril no gira. , si no indicara un nombre de eje, el mandril no es coondizionato, Loop cerrado-indicar en" "el nombre del eje usado por el mandril (*) (*)
K_DAC_1Volt=0	Insertar el peso atado al n. de bit del convertidor DAC utilizado, tan artificioso:**2(n.bit-1)/10.**Ejemplo: con DAC a 12 bites, comprensivo de señal, el valor será: 2(12-1)/10 = 2(11)/10 =2048/10 =204,8 Valor de estándard = **0,** El sistema es predispuesto por las fichas 6 Ases +3 Mandriles PCI 2000 (*)
KC_parameter=300	Por Mandriles a loop abiertos mandado de modo Analógico, indicar el N. de vueltas del Motor mandril por 1Volt de salida del DAC. , por mandriles con mando digital ves los calibrados específicos.) Por Mandriles a loop cerrado ves NOTAS a pie de página (*)
Volt_max=**10**	Msassima tensión en salida por el DACValor de estándard = **10** (*)
Volt_turn=**0**	NO MODIFIQUES
Volt_offset=-0	Tensión de compensación por tener paro el mandrilValor de estándard = **0**
Turn_min=**0**	De costumbre = 0Por mandriles especiales que no pueden bajar bajo un N. de giri/min. predefinido, indicar este límite en N.di gira (es.2000) Valor de estándard = **0**
Turn_max=3000	N. de giri/minuto max al que puede girar el mandril. Valor de estándard = **3000** (*)
Acc_max=50	Aceleración max expreso en N. de vueltas / sec2 Valor de estándard = **50** (*)
Dec_max=50	Desaceleración max expreso en N. de vueltas / sec2 Valor de estándard = **50** (*)
Delay_spindle=0	Tiempo suplementario de espera después de que el Mandril ha llegado al régimen programado.

	Valor bdi estándard = **0 (**Calibrado algo usado)
Number_ENABLE_switch_port=""	Nombre de la salida digital utilizado para dar el Enable a la UNIDAD DE DRIVER
FalseTrue_logic_ENABLE_switch=1	Nivel lógico de la señal Enable Valor de estándard = **1**
Number_DRVOK_switch_port=""	Nombre del input digital utilizado por la señal de Unidad de Driver VALE
FalseTrue_logic_DRVOK_switch=2	Nivel lógico de la señal Unidad de driver VALEValor de estándard = **2,** ningún control,
Number_REGIM_switch_port=””	Nombre del input digital que señala Mandril a régimen. (Control algo usado)
FalseTrue_logic_REGIM_switch=2	Nivel lógico de la señal REGIMValor de estándard = **2,** ningún control,
Number_STOP_switch_port=""	Nombre del input digital que señala Mandril firme. (Control algo usado)
FalseTrue_logic_STOP_switch=2	Nivel lógico de la señal ALTOValor de estándard = **2,** ningún control,
Number_LOCK_switch_port=""	Nombre de la salida digital usado para parar el Mandril. (Mando algo usado)
FalseTrue_logic_LOCK_switch=2	Nivel lógico de la señal LOCKValor de estándard = **2,** ningún control,
e=}	NO MODIFIQUES

CONOCIDOS SOBRE LA ADMINISTRE DE LOS MANDRILES (*)
Mandriles a loop cerrado, utilizan el DAC y el Encoder dí un Eje.

En el calibrado de la estructura Eje relativo a este Mandril, (por ejemplo llamada C1, siendo la estructura relativa a un Mandril e impar a un eje, si tarara dí norma

Kind_axis_0=0x00000008
Revolution=360, En este novio el eje vendrá gesto con reposición de la cuota a cada vuelta,
Step = Estimas deforestas que 1 vuelta dí mandril valgo 360°, En acuerdo al Encoder usado,

Da notaré que un mandril a loop cerrado, Usted impar tarado come sobre, si comportara come un eje, cuya cuota impar si resetta a cada vuelta. Pues, en este caso, programando la cuota Cero, el mandril la alcanzará volviendo atrás dí todas las vueltas hechas a la ida.

En este caso si utilizan todo él calibrados dí un Eje teñendo contó siguientes uniones:

K_DAC_1Volt de la estructura Mandril tiene que ser igual a aquél estimas escrito en **K_DAC_1Volt** del calibrado Eje (C1)

KC_parameter =, de la estructura Mandril, tiene que ser igual a 1/3600 del valor de **Vel_max** definido por la estructura Eje, C1,

Turn_max =, de la estructura Mandril, tiene que ser igual a 1/360 del valor de **Vel_max =,** de la estructura Eje C1,

Acc_max =, de la estructura Mandril, tiene que ser igual a 1/360 del valor de **Acc_max =,** de la estructura Eje C1,

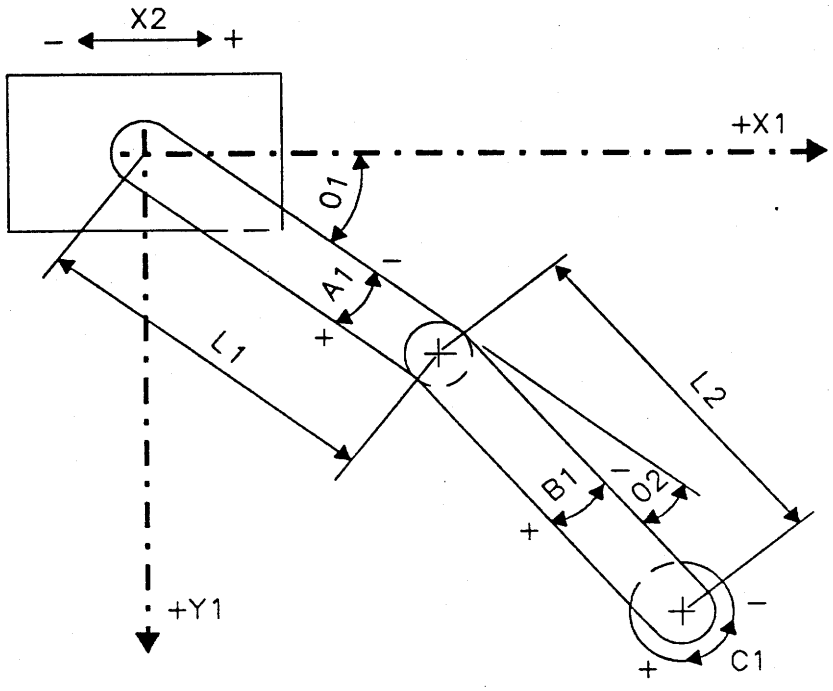
Dec_max =, de la estructura Mandril, tiene que ser igual a 1/360 del valor de **Dec_max =,** de la estructura Eje C1,

2.3.3 Ases virtuales

Los ases virtuales son los que necesitan una transformación en el plan de las coordenadas.

Los ases virtuales, como dice la palabra misma, no son ases físicos pero ases que por comodidad usuario son caracterizados sobre sistemas a geometría SCARA o CILÍNDRICA de modo que además de efectuar movimientos sobre los ases físicos se pueda efectuar también movimientos según un sistema cartesiano.
Con tal sistema, además se pueden efectuar interpolaciones lineales y circulares.

2.3.3.1 Sistema SCARA



Número de ases SCARA: indica el número de ases que tomar en consideración; mínimo 2, los primeros 2,; tres, 1,2,3 o bien 4 (todo)

En el ejemplo indicado, son tomados en consideración todo y 4, las 2 articulaciones A1 y B1,il 3° eje rotatorio C1 puesto a la extremidad de la segunda articulación y al 4° eje traslante X2,

Nombre del primer eje: indica el primero eje real del sistema, 1° articulación,; en el ejemplo el eje A1

Largo del primer eje: indica el largo del primer eje (distancia entre lo primero y según articulación); en el ejemplo L1

Nombre del segundo eje: indica el segundo eje real del sistema, 2° articulación,; en el ejemplo B1

Largo del segundo eje: indica el largo del segundo eje, distancia entre articulación central y eje pinza,; en el ejemplo L2.
En el caso esté presente el tercer eje, en el ejemplo C1, el eje pinza tiene que ser concéntrico con el eje de rotación del tercer eje, en este caso el largo del segundo eje está fijo. Si en cambio hay excentricidad, el largo del segundo eje, con respecto del eje pinza, es variable, siendo dependiente de la posición angular de la pinza. Por consiguiente, ya que se impuesta un solo largo, el sistema cartesiano conseguido sólo será válido en aquellas condiciones.
Con base en estas consideraciones, en caso de excentricidad, hace falta cuanto tener presente sigue:

o) no es posible conseguir trayectos rectilíneos y circulares y el error de trayecto será tanto mayor como mayor es la excentricidad.

Nombre del tercer eje: indica el eventual eje rotatorio sitio a la extremidad del segundo eje; en el ejemplo C1.
La indicación o menos que este eje en tablero, influyen sólo en la geometría del sistema a los efectos de la compensación automática de dicho rotatorio y precisamente:

p) Eje no indicado en tablero: en caso de requerido movimiento ases, reales o virtuales, tal eje rotatorio es ignorado, naturalmente si no solicitara en el movimiento,

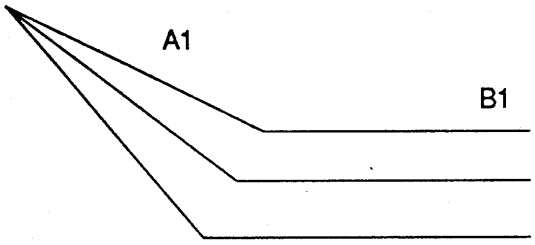
q) Eje indicado en tablero: en caso de requerido movimiento ases, reales o virtuales, tal eje rotatorio, naturalmente si no solicitara en el movimiento, viene automáticamente aglomerado; la pinza apuesta sobre tal eje queda por lo tanto siempre paralela a él mismo.
Y alguno que si este eje es solicitado en el movimiento, la indicación o menos que el eje en el tablero no tiene ningún sentido más siendo el movimiento requerido determinante.

Nombre del cuarto eje: indica el eventual eje traslante, traslado de todo Yo SCARA,; en el ejemplo X2

Compensación mecánica: este fecho sirve para localizar el tipo de robot según la convención siguiente:

0 = robot de tipo SCARA tradicional
± 1 = robot de tipo SCARA pero con sistema a pantógrafo

dónde sólo moviendo uno eje, en el ejemplo de abajo A1, el segundo eje, en el ejemplo B1, se mantiene paralelo a él mismo.



La señal algebraica debe ser entendida en el modo siguiente:
Con la señal positiva se entiende la compensación automática sobre el eje B1 sumando el rincón efectuado por el eje A1.

Con la señal negativa se entiende la compensación automática sobre el eje B1 sustrayendo el rincón efectuado por el eje A1.

Compensación absoluta: este fecho es utilizado cada vez se desea conseguir un cero eje práctico diferente de aquel teórico.

Máxima velocidad: corresponde a la máxima velocidad querida sobre el eje virtual; si esta velocidad fuera imposible que efectuar, hace falta tener presente que los dos ases reales juntados al eje virtual tienen cada uno una máxima velocidad, por estándard es utilizada el principio posible.

Máxima aceleración y máxima desaceleración: corresponden a la máxima aceleración y desaceleración querida sobre el eje virtual; también por este fechas si los valores indicados fueran superiores a lo posible, por estándard son utilizados los máximos posible.

Notas:

a, Ya que todos los cálculos para los ases virtuales son ejecutados según la geometría clásica obrando sobre senos y cosenos de rincones expresos en grados, hace falta tener presente cuanto sigue:

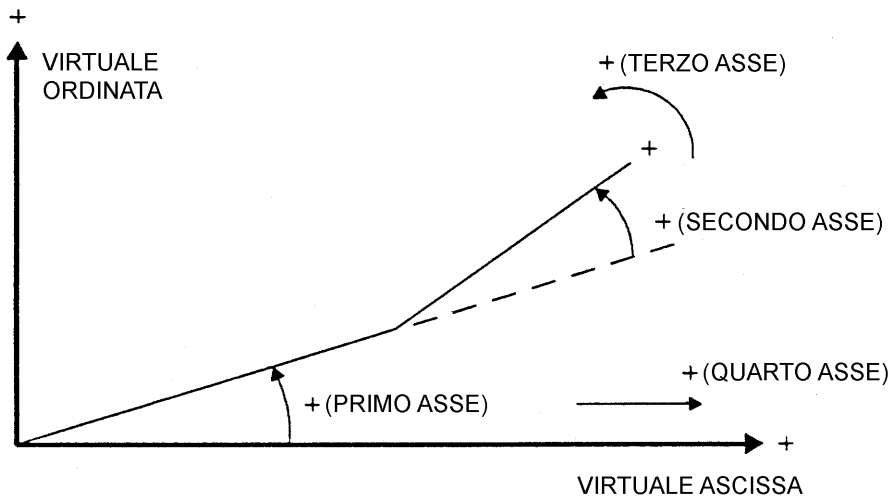
1 - La unidad de medida de los primeros 3 ases reales tiene que ser el grado.

2 - La unidad de medida del cuarto eje, puede ser el mm o pulgares a segunda que a los largo primero y según eje sean dadas respectivamente en mm o inc.

3 - La compensación absoluta del primer eje real tiene que ser dado, si está presente el 4° eje traslante, de modo que el eje virtual abscisa sea paralelo al cuarto eje SCARA.

4 - La compensación absoluta del segundo eje real tiene que ser dado de modo que los primeros dos ases resulten alineados, brazo todo tenso.

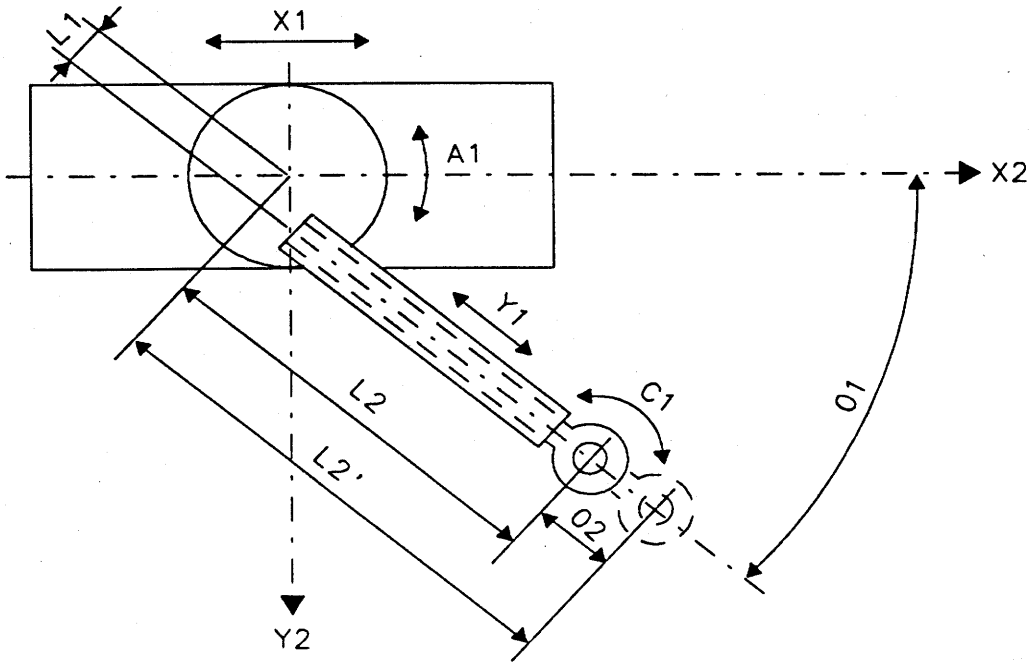
5 - Las direcciones de movimiento, positivo, rechazo, de los ases del sistema tienen que ser todo concordes: por los primeros 3 ases rotatoria dirección positiva significa ir de abscisa positiva a ordenada positiva; el eventual cuarto eje tiene que tener la misma dirección de la abscisa. La figura siguiente provee un ejemplo de dichas direcciones.



6 - Con base en las consideraciones vistas en los puntos 1-5 precedentes, se tendrá que después de un movimiento a cero de todo y a 4 los ases reales SCARA, el eje virtual abscisa se encontrará a una coordenada equivalente a la suma de los largo de lo primero y según eje; el eje virtual ordenado a coordenada cero.

b, Con un sistema SCARA, no es posible solicitar por una misma instrucción de movimiento, ases virtuales juntamente a los primeros dos ases reales del sistema, uno de los dos o ambos.
Y viceversa posible solicitar juntamente ases virtuales al 3° e/o 4° eje real del sistema.

2.3.3.2 Sistema CILINDRICO



Número de ases del sistema CILÍNDRICO: indica el número de ases que tomar en consideración; 1°+2°,1°+2°+ 3°,1°+2°+3°+ 4°. Se aconseja siempre indicar 4 y si algún eje es no presiente (2°) 3°, 4°, no lo indiques en las rayas adecuadas.
En el ejemplo indicado, son tomados en consideración todo y 4 los ases.

Nombre del primer eje: indica el primero eje real del sistema; en el ejemplo A1. El primer eje siempre es el eje rotatorio general y tiene que siempre estar presente

Largo del primer eje: indica la distancia, también puede ser 0, entre el eje de rotación del primer eje y el eje pinza; tal distancia debe ser dada perpendicularmente al eje cannotto, si cannotto faltante, al brazo lleva pinza,; en el ejemplo L1.

Nombre del segundo eje: indica el segundo eje real del sistema; en el ejemplo Y1.
Tal eje siempre es el eje cannotto y podría ser no presiente también; en tal caso tiene que pero necesariamente estar presente el cuarto eje

Largo del segundo eje: indica la distancia entre el eje de rotación del primer eje y el eje pinza; tal distancia debe ser dada paralelamente al eje cannotto, si cannotto faltante, al brazo lleva pinza,; en el ejemplo, L2 si el segundo eje, cannotto, no tiene compensación absoluta; 12, si este eje tiene un compensaciones suyo, 02.
En el caso esté presente el tercer eje, en el ejemplo C1, el eje pinza tiene que ser concéntrico con el eje de rotación del tercer eje, en este caso el largo del segundo eje está fijo. Si en cambio hay excentricidad, el largo del segundo eje, con respecto del eje pinza, es variable, siendo independiente de la posición angular de la pinza. Por consiguiente, ya que se impuesta un solo largo, el sistema cartesiano conseguido sólo será válido en aquellas condiciones.
Con base en estas consideraciones, en caso de excentricidad, hace falta cuanto tener presente sigue:

- no es posible conseguir trayectos rectilíneos y circulares y el error de trayecto será tanto mayor como mayor es la excentricidad.

Nombre del tercer eje: indica el eventual eje rotatorio sitio a la extremidad del cannotto, o brazo lleva pinza si cannotto faltante,; en el ejemplo C1.
La indicación o menos que este eje en tablero, influyen sólo en la geometría del sistema a los efectos de la compensación automática de dicho rotatorio y precisamente:

- Eje no indicado en tablero: en caso de requerido movimiento ases, reales o virtuales, tal eje rotatorio es ignorado, naturalmente si no solicitara en el movimiento,
- Eje indicado en tablero: en caso de requerido movimiento ases, reales o virtuales, tal eje rotatorio, naturalmente si no solicitara en el movimiento, viene automáticamente aglomerado; la pinza apuesta sobre tal eje queda por lo tanto siempre paralela a él mismo.

Y alguno que si este eje es solicitado en el movimiento, la indicación o menos que el eje en el tablero no tiene ningún sentido más siendo el movimiento requerido determinante.

Nombre del cuarto eje: indica el eventual eje traslante, traslado de todo lo CILÍNDRICO,; en el ejemplo X1. Tal eje podría ser no presiente también conque esté presente el segundo eje;

Compensación mecánica: tal dato siempre va dato con valor 0.

Compensación absoluta: este fecho es utilizado cada vez que cada se desea conseguir un cero eje práctico diferente de aquel teórico.

Máxima velocidad: corresponde a la máxima velocidad querida sobre el eje virtual; si esta velocidad fuera imposible que efectuar, (cada eje real miembro al sistema tiene un mismo principio velocidad, por estándard es utilizada el principio posible.
Máxima aceleración y desaceleración: corresponden a la máxima aceleración y desaceleración querida sobre el eje virtual; también por este fechas si los valores indicados fueran superiores a lo posible, por estándard son utilizados los máximos posible.

Notas:

a, Ya que todos los cálculos para los ases virtuales son ejecutados según la geometría clásica, obrando sobre senos y cosenos de rincones expresos en grados, hace falta tener presente cuanto sigue:

1 - La unidad de medida de los ases rotatorios, primero y tercer eje, reales tiene que ser el grado.

2 - La unidad de medida del cuarto eje real puede ser el mm o el inc. a segunda que el largo según eje sea dada respectivamente en mm o inc.

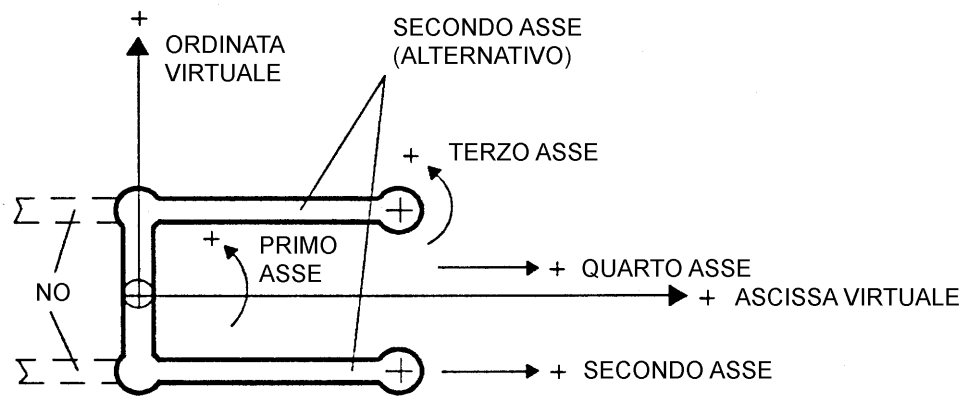
3 - La compensación absoluta del primer eje real tiene que ser dado, si está presente el 4 eje traslante, de modo que el eje virtual abscisa sea paralelo al cuarto eje CILÍNDRICO.

4 - ll según eje, cannotto, puede o menos tener compensación; tal compensación implica claramente un diferente largo del segundo eje.

5 - Las direcciones de movimiento, positivo, rechazos, de los ases del sistema tienen que ser todo concordes: para los ases rotatorios, primero y tercer eje, dirección positiva significa ir de abscisa positiva a ordenada positiva; los ases lineales, según y cuarto, tiene que tener la misma dirección de las abscisas.

6 - ll según eje, cannotto, no puede encontrarse en abscisa negativa en caso contrario girar el primer eje de modo que se encuentra en abscisa positiva.

La figura siguiente provee como un ejemplo de tienen que ser puestos los ases sobre de un sistema cilíndrico



7 - Con base en las consideraciones vistas en los puntos 1-6 precedentes, se tendrá que después de un movimiento a cero de todo y a 4 los ases reales CILÍNDRICOS, el eje virtual abscisa se encontrará a una coordenada igual al largo del segundo brazo y el eje virtual ordenados a una coordenada igual al largo del primer brazo. , Por este último eje la coordenada será positiva o rechazo,

b, Con un sistema CILÍNDRICO, no es posible solicitar por una misma instrucción de movimiento, ases virtuales juntamente a los dos ases reales, uno de los dos o ambos, constituyentes el sistema cilíndrico y precisamente: 1° eje del sistema, Rotatorio general, 2° eje del sistema, cannotto o 4° eje del sistema (traslante) en el caso de ausencia del eje cannotto
Y en cambio posible solicitar juntamente ases virtuales al 3° e/o 4° eje, si está presente el 2° eje,
Recapitulación de los ases reales no abbinabili a los virtuales en una misma instrucción de movimiento:

- Sistema cilíndrico compuesto por el 1° y 2° eje, rotatorio general + cannotto, no es admitido virtuales +1° e/o 2° eje real

- Sistema cilíndrico compuesto por el 1° y 4° eje, rotatorio general + traslante, no es admitido virtuales + 1° e/o 4° eje real

- Sistema cilíndrico traje de 1°, 2° y 4° eje

No son admitidos virtuales + 1° e/o 2° eje, solo o con el 4°.

[X3]
i=**VIRTUAL_AXES_NAME**("X3" , "Y3")
b={
Kind_Axis = "ARM"
Kind_Axis_0=0x00010000 / / SCARA ESTÁNDAR
Kind_Axis_1=0x00000000
Axis_name_ascisse = "X3"
Axis_name_ordinate = "Y3"
Axis_name_1 = "A1" / / EJE ANGULAR BRAZO
Axis_name_2 = "B1" / / EJE ANGULAR ANTEBRAZO
Axis_name_3 = "/ / EJE ROTATORIO (MUÑECA)
Axis_name_4 = "/ / EJE TRASLANTE SUPLEMENTARIO
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 / / CILÍNDRICO CON LANCHA
Kind_Axis_1=0x00000000
Axis_name_ascisse = "X2"
Axis_name_ordinate = "Y2"
Axis_name_1 = "C1" / / EJE ANGULAR
Axis_name_2 = "D1" / / EJE LINEAL (LANCHA)
Axis_name_3 = "/ / EJE ROTATORIO (MUÑECA)
Axis_name_4 = "/ / EJE TRASLANTE SUPLEMENTARIO
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 Parámetros específicos de los Entornos

Informaciones sobre las condiciones de lanzamiento del part program

[ParametriAUTOMAZIONE]
StringaLancio =
NomePartProgramLancio=via.pp

Notas:
Con la caracterización estándar aquí enseñada, el sistema se encamina automáticamente el Part lanzando program "via.pp", o bien con otro nombre dado por el desarrollador del Sistema,
Siendo como este Part Program editabile un normal Part program, si se quiere que el Sistema se encamina con un recorrido diferente y no accesible del entorno, hace falta definir este recorrido particular en el "StringaLancio ="

Ejemplo: **StringaLancio=\programmi\applicazione\partprog\start.pp**

2.5 Predefinizioni ISO

En el expediente "SISTEMA.TXT" puede ser insertados mandos de abbinamento de los recursos a los "procesos" ISO y además algunos predefinizioni.

Los procesos de elaboración ISO, G-COLA, son máximo 32, son localizados por un ID (identificativo numérico) y de un Nombre (identificativo alfanumérico)

Los parámetros de los procesos ISO es definido como lo primero por "Base" y los otros "Prueba" con el número de proceso-1 hasta a lo sumo de procesos menos uno:

ejemplo 5 procesos tendremos

Base_xxxx

Testa1_xxxxx

Testa2_xxxxx

Testa3_xxxxx

Testa4_xxxxx

Estas definiciones conciernen los valores de estándard, que son activadas al principio de cada part

Las sesiones son:

a - Sesión identificativo nombre del proceso

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

b- Sesión identificativo ID del proceso

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

Ejemplo

[AbbinamentoNumeroTesta]
Base_NumeroTesta=11
Testa1_NumeroTesta12
Testa2_NumeroTesta=21

c- Sesión abbinamento ases maquina a ases ISO

Los 9 ases canónicos ISO son X,Y,Z,A,B,C,U,V,W con esta sesión son juntados a los ases físicos o virtuales del sistema.
El valor –999 es insertado al sitio del nombre eje físico o virtual, cuando no existe el abbinamento.

[AbbinamentoAssiTeste]
Base_X = [nombre eje físico o virtual]
Base_Y = [nombre eje físico o virtual]
Base_Z = [nombre eje físico o virtual]
Base_A = [nombre eje físico o virtual]
Base_B = [nombre eje físico o virtual]
Base_C = [nombre eje físico o virtual]
Base_U = [nombre eje físico o virtual]
Base_V = [nombre eje físico o virtual]
Base_W = [nombre eje físico o virtual]
Testa1_X = [nombre eje físico o virtual]
Testa1_Y = [nombre eje físico o virtual]
Testa1_Z = [nombre eje físico o virtual]
Testa1_A = [nombre eje físico o virtual]
Testa1_B = [nombre eje físico o virtual]
Testa1_C = [nombre eje físico o virtual]
Testa1_U = [nombre eje físico o virtual]
Testa1_V = [nombre eje físico o virtual]
Testa1_W = [nombre eje físico o virtual]
…
…
Testa31_X = [nombre eje físico o virtual]
Testa31_Y = [nombre eje físico o virtual]
Testa31_Z = [nombre eje físico o virtual]
Testa31_A = [nombre eje físico o virtual]
Testa31_B = [nombre eje físico o virtual]
Testa31_C = [nombre eje físico o virtual]
Testa31_U = [nombre eje físico o virtual]
Testa31_V = [nombre eje físico o virtual]
Testa31_W = [nombre eje físico o virtual]

Ejemplo

```
[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 - Sesión abbinamento mandril S ISO a mandril físico

El mandril S ISO con esta sesión es juntado a un mandril físico del sistema.
El valor –999 es insertado al sitio nombre mandril, cuando no existe el abbinamento.

[AbbinamentoMandrinoTeste]
Base_Mandrino = [nombre mandril]
Testa1_Mandrino = [nombre mandril]
..
Testa31_Mandrino = [nombre mandril]

Ejemplo

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

y - Sesión abbinamento directorio de trabajo part program ISO al proceso

Cada proceso ISO trabaja sobre un directorio separado, de modo que obrar sobre un propio archivo de programas ISO, xxxxx.prg.

[AbbinamentoDirectoryPP]
Base_Directory_PP = [path directorio \]
Testa1_Directory_PP = [path directorio \]
…
Testa31_Directory_PP = [path directorio \]

Ejemplo

[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 \

Cada proceso ISO trabaja sobre un tablero propio de **orígenes,** por esta sesión es posible precisar el entero recorrido y el nombre del expediente continente el tablero.

[AbbinamentoORIGINI]
Base_FILE_ORIGINE=[percorso y nombre expediente]
Testa1_FILE_ORIGINE=[percorso y nombre expediente]
…
Testa31_FILE_ORIGINE=[percorso y nombre expediente]

Ejemplo

[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 - Sesión abbinamento expediente tablero TOOL al proceso ISO

Cada proceso ISO trabaja sobre un tablero propio de **Tool,** por esta sesión es posible precisar el entero recorrido y el nombre del expediente continente el tablero.

[AbbinamentoTOOL]
Base_FILE_ TOOL = [recorrido y nombre expediente]
Testa1_FILE_ TOOL = [recorrido y nombre expediente]
…
Testa31_FILE_ TOOL = [recorrido y nombre expediente]

Ejemplo

[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 - Sesión abbinamento expediente tablero parámetros P al proceso ISO

Cada proceso ISO trabaja sobre un tablero propio de **parámetros P,** por esta sesión es posible precisar el entero recorrido y el nombre del expediente continente el tablero.

[AbbinamentoPARAMETRI]
Base_FILE_ PARÁMETROS = [recorrido y nombre expediente]
Testa1_FILE_ PARÁMETROS = [recorrido y nombre expediente]
…
Testa31_FILE_ PARÁMETROS = [recorrido y nombre expediente]

Ejemplo

[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 - Sesión abbinamento expediente tablero Funciona M y G al proceso ISO

Cada proceso ISO trabaja sobre un tablero propio de **Funciones M y G,** por esta sesión es posible precisar el entero recorrido y el nombre del expediente continente el tablero.

[AbbinamentoFUNZIONI_MG]
Base_FILE_ FUNZIONI_MG = [recorrido y nombre expediente]
Testa1_FILE_ FUNZIONI_MG = [recorrido y nombre expediente]
…
Testa31_FILE_ FUNZIONI_MG = [recorrido y nombre expediente]

Ejemplo

[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 - Sesión abbinamento expediente tablero Correctores Útiles al proceso ISO

Cada proceso ISO trabaja sobre un tablero propio de **Correctores Útiles,** por esta sesión es posible precisar el entero recorrido y el nombre del expediente continente el tablero.

[AbbinamentoCORREZ]
Base_FILE CORRECCIONES = [recorrido y nombre expediente]
Testa1_FILE_ CORRECCIONES = [recorrido y nombre expediente]
…
Testa31_FILE_ CORRECCIONES = [recorrido y nombre expediente]

Ejemplo

[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 - Sesión abbinamento gestión de la pareja por un global G cargado por la lógica de máquina por el proceso ISO

Si la lógica de máquina adquiere el valor de pareja y lo provee como valor porcentual en una variable global G, este última puede venir indicada al proceso ISO juntado.
El valor –999 es insertado al sitio del número del Global, cuando no existe el abbinamento.

[AbbinamentoCoppia]
Base_ Globale_Coppia = [número Global]
Testa1_ Globale_Coppia = [número Global]
…
Testa31_ Globale_Coppia = [número Global]

Ejemplo

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

n - Sesión abbinamento capacitación y disabilitazione del proceso ISO por el estado contenido un global G precisa

Por el estado 1 = habilitado y 0=disabilitato contenido en un G precisa por cada proceso ISO, es posible administrar la función de activación y desactivación del proceso ISO.
El valor –999 es insertado al sitio del número del Global, cuando no existe el abbinamento.

[AbbinamentoG_Disabilitazione]
Base_ Globale_Disabilitazione = [número Global]
Testa1_ Globale_Disabilitazione = [número Global]
…
Testa31_ Globale_Disabilitazione = [número Global]

Ejemplo

[AbbinamentoG_Disabilitazione]
Base_Globale_Disabilitazione=-999
Testa1_Globale_Disabilitazione=-999
Testa2_Globale_Disabilitazione=-999

o - Sesión abbinamento estándard arregla medida en mm o pulgares

El proceso ISO a un propio sistema de medida métrica o a pulgares, si no precisara por esta voz es métrico, en caso de que lo se quiera en pulgares hace falta definir al valor numérico 1 el parámetro del proceso ISO deseado.

El valor –999 indica sistema métrico, como por lo demás también 0.

[AbbinamentoInch]
Base_INCH=[flag 1 por sistema en pulgares]
Testa1_INCH=[flag 1 por sistema en pulgares]
..
Testa31_INCH=[flag 1 por sistema en pulgares]

Ejemplo

[AbbinamentoInch]
Base_INCH=-999

p - Sesión abbinamento definición en los G02 y G03 del modo de interpretar los valores del centro del rayo (EL-J-K) en absoluto con respecto de las orinas trozo activo u en incremental con respecto de la posición corriente.

En cada proceso ISO es posible definir los valores del centro del rayo les contenidos en los G02 y G03 en absoluto con respecto del sistema de referencia en curso odin incremental con respecto de la posición corriente. Si no declarara de modo explícito en esta sesión el sistema los interpreta de modo absoluto.
El valor –999 indica sistema absoluto, como por lo demás también 0.

[AbbinaIncrementaleG0302]
Base_ INCREMENTALEG0302=[flag 1 por sistema en incremental]
Testa1_ INCREMENTALEG0302=[flag 1 por sistema en incremental]
..
Testa31_ INCREMENTALEG0302=[flag 1 por sistema en incremental]

Ejemplo

[AbbinaIncrementaleG0302]
Base_INCREMENTALEG0302=-999
Testa1_INCREMENTALEG0302=-999
Testa2_INCREMENTALEG0302=-999

q - Sesión abbinamento modalidad de ejecución G01

En cada proceso ISO es posible definir si la ejecución de los bloques G01-G02-G03 están en modalidad continua (defaul) o bien con detención de los ases a cada bloque.
El valor –999 indica modalidad continua, como por lo demás también 0.

[AbbinaNonContinuo]
Base_ NONCONTINUO = [flag 1 por modalidad con detención ases]
Testa1_ NONCONTINUO = [flag 1 por modalidad con detención ases]
..
Testa31_ NONCONTINUO = [flag 1 por modalidad con detención ases]

Ejemplo

[AbbinaNonContinuo]
Base_NONCONTINUO=-999
Testa1_NONCONTINUO=-999
Testa2_NONCONTINUO=-999

r - Sesión abbinamento salida digital por M08 refrigerante médico en jefe

En cada proceso ISO es posible definir la salida digital por a función de refrigerador primario M08

[AbbinamentoRefrigerantePrincipale]
Base_ RefrigerantePrimario = [nombre salida digital]
Testa1_ RefrigerantePrimario = [nombre salida digital]
..
Testa31_ RefrigerantePrimario = [nombre salida digital]

Ejemplo

[AbbinamentoRefrigerantePrincipale]
Base_RefrigerantePrimario=EV_RP011
Testa1_RefrigerantePrimario EV_RP012
Testa2_RefrigerantePrimario EV_RP021

s - Sesión abbinamento salida digital por M07 refrigerante secundario

En cada proceso ISO es posible definir la salida digital por a función de refrigerador secundario M07

[AbbinamentoRefrigeranteSecondario]
Base_ RefrigeranteSecondario = [nombre salida digital]
Testa1_ RefrigeranteSecondario = [nombre salida digital]
..
Testa31_ RefrigeranteSecondario = [nombre salida digital]

Ejemplo

[AbbinamentoRefrigeranteSecondario]
Base_RefrigeranteSecondario=EV_RS011
Testa1_RefrigeranteSecondario = EV_RS012
Testa2_RefrigeranteSecondario=EV_RS021

Función spefiche definidos automáticamente, por la sincronización con la lógica de máquina.

Son previstos de los automatismos por la interconexión con la LÓGICA de MÁQUINA, por 16 o 32 bites consecutivos por el cambio de estados y solicitudes.

Y en particular:

Lógica Máquina CNC-ISO

Bit 0 Lógica VALE = 1	Bit 0 Part Program Start = 1
Bit 1 Rich. Alto	Bit 1 Part Program Alto = 1
Bit 2 Rich. PassoPasso	Bit 2 Part Program en PassoPasso = 1
Bit 3 Rich. Kill Part program	Bit 3 Part Program Killed
Bit 4 Alarma = 1	Bit 4 Part Program Alarma adquiridos y Part Program en Alto
Bit 5 Warning = 1	Bit 5 Part Program Warning adquiridos
Bit 6 Info = 1	Bit 6 Part Program Info adquiridos
Bit 7 Lock = 1	Bit 6 Part Program Lock adquiridos
Bit 8 Next Step	Bit 8
Bit 9 Manual	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

Son escritos por el proceso ISO

[AbbinamentoOutputDigitaleControllo]

Base_OutputDigitaleControllo=[nome input u salida digital bit 0]
Testa1_OutputDigitaleControllo=[nome input u salida digital bit 0]
..
Testa31_OutputDigitaleControllo=[nome input u salida digital bit 0]

Son leídos por el proceso ISO
[AbbinamentoInputDigitaleControllo]
Base_InputDigitaleControllo=[nome input u salida digital bit 0]
Testa1_InputDigitaleControllo=[nome input u salida digital bit 0]
..
Testa31_InputDigitaleControllo=[nome input u salida digital bit 0]

Número de bit 16 o 32
[AbbinamentoNumeroBitControllo]
Base_NumeroBitControllo=[NumeroBit 16 0 32]

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

Funciones futuras definidas automáticamente, pero no interpretadas por el sistema.

[AbbinamentoOutputDigitaleCicloFisso]
Base_OutputDigitaleCicloFisso=[nome input u salida digital]
Testa1_OutputDigitaleCicloFisso=[nome input u salida digital]
..
Testa31_OutputDigitaleCicloFisso=[nome input u salida digital]

[AbbinamentoInputDigitaleCicloFisso]
Base_InputDigitaleCicloFisso=[nome input u salida digital]
Testa1_InputDigitaleCicloFisso=[nome input u salida digital]
..
Testa31_InputDigitaleCicloFisso=[nome input u salida digital]

[AbbinamentoOutputAnalogicoCicloFisso]
Base_OutputAnalogicoCicloFisso=[nome salida analógica]
Testa1_InputDigitaleCicloFisso=[nome input u salida digital]
..
Testa31_InputDigitaleCicloFisso=[nome input u salida digital]

[AbbinamentoInputAnalogicoCicloFisso]
Base_InputAnalogicoCicloFisso=[nome input analógico]
Testa1_InputAnalogicoCicloFisso=[nome input analógico]
..
Testa31_InputAnalogicoCicloFisso=[nome input analógico]

[AbbinamentoChiaveInputControllo]
Base_ChiaveInputControllo=[nome input u salida digital]
Testa1_ChiaveInputControllo=[nome input u salida digital]
..
Testa31_ChiaveInputControllo=[nome input u salida digital]

Parámetros globales a todos los procesos ISO

[ParametriCNC]

Rapido=Valore predefinido de exprés si él desira un valor menor de aquél programado en caracterización

Feed=Feeed de estándard programado al principio del programa ISO, será reemplazado por el primer F encontrado en el part program

Speed = Speed de estándard programado al principio del programa ISO, será reemplazado por el primer S encontrado en el part program

X_offsetTavola=offset eje X MESA, será reemplazada antes por el O expediente en el part program
Y_offsetTavola = compensación eje Y MESA, será reemplazado antes por el O expediente en el part program
Z_offsetTavola = compensación eje Z MESA, será reemplazado antes por el O expediente en el part program

Multiprocesso=Se cero es activado multi la función prueba mono proceso, Estándard 1,

GlobaleStatoMacchina=Globale por el intercambio con la Lógica de máquina
BLOCCA_STATI -1 si leyera por el sistema a –1 para los estados maquina
NO_STATI 0
AUTOMÁTICO 1
MANUAL 3

GlobaleComandiMacchina=Globale por el intercambio con la Lógica de máquina
BLOCCA_COMANDI -1 si leyera por el sistema a –1 para los mandos maquina
NO_CICLO 0
VIA_CICLO 3
BREAK 10
HOLD 11

GlobaleServizi=Globale por el intercambio con la Lógica de máquina
BLOCCA_SERVIZI -1 si leyera por el sistema a –1 para la activación de los servicios
NO_SERVIZI_ATTIVI 0
MDI 1
COMM 2
FUNZIONI_MANUALI 3
IMPOSTA_PEZZO 4
MODALITA_CICLO 5

GlobaleProtezioni=NON utilizado por el sistema, futuros desarrollos

GlobaleStatoPLC=Indicando un número de global, este última es utilizada por el sistema para escribir sobre la interfaz la inscripción indicada en los dos parámetros siguientes 1 o 0.
ScrittaAttivaStatoPLC=STRINGA escrito si **GlobaleStatoPLC=1**
ScrittaNonAttivaStatoPLC = CORDÓN inscripción si **GlobaleStatoPLC=0**
TipoIOFeedHold=0 o 1 = input o 2 = salida hace las funciones del **GlobaleStatoPLC**
NomeIOFeedHold = Nombre YO

GlobaleComandoPLC = Indicando un número de global, este última es utilizada por el sistema para escribir sobre la interfaz la inscripción indicada en los dos parámetros siguientes 1 o 0.
ScrittaAttivaComandoPLC=STRINGA escrito si **GlobaleComandoPLC = 1**
ScrittaNonAttivaComandoPLC = CORDÓN inscripción si **GlobaleComandoPLC = 0**
TipoIOHold=0 o 1 = input o 2 = salida hace las funciones del el **GlobaleComandoPLC**
NomeIOHold=Nome YO

Break = Puesto al valor de 1 reemplaza el interruptor de "**Selección Proceso" (F10)** con el interruptor de "**BREAK",** sobre la interfaz CNC.

GlobaleUnitaSelezionata = Indicando un número de global, este última es cargada por el sistema con el ID del proceso selecto sobre la interfaz.

GlobaleUnitaSelezionabileDaPLC = Indicando un número de global, este última es utilizada **cuando varia,** por lo tanto sólo al cambio, para seleccionar el proceso ISO sobre la interfaz con el ID del proceso indicado.

¡Atención!!! Sólo el cambio de valor del **GlobaleUnitaSelezionabileDaPLC,** efectúa el cambio de selección del proceso ISO.

Ejemplo

[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

CAPÍTULO 3

3 Caracterización

Para poder administrar los recursos de sistema en su totalidad con los ciclos de elaboración es necesario configurar los siguientes parámetros.

. 1-PID	Proporcional Complementario Derivativo
. 2-Anticollisione	Anticollisione entre ases vinculados
. 3-Gantry	Ases cooperanti
. 4-Homing	Modalidad puesta a cero eje
. 5-Determinados campeonatos	Buffer adquisición fecha campeonatos
. 6-RAV y CAP	Read Ases Value
. 7 - CAP	Cargados Ases Parámetro.
. 8 - Ases Rollover	Ases cíclicos
. 9 - Over travel	Otrecorsa
.10 - Ases sometidos en mov.	Ases sólo sometidos en movimiento

3.1 PID

Proporcional Complementario Derivativo

Son previstos dos modos para someter los ases; en VELOCIDAD' y en CORRIENTE.

La sumisión en VELOCIDAD' es definida poniendo a zeroil bit 0x00000001 de "Kind_axis_0" de los ases en caracterización.

Hace falta respetar la relación entre "Vel_max" y "Volt_vel" porque' y' de este modo es definido al movimentazione cuantos voltios precisan a la sumisión por la velocidad' intencional.

Cada variación de "Vel_max" tiene que ser correspondida por un proporcional de "Volt_vel."

El valor de "Dec_max" y "Acc_max" pueden ser cambiados sin ningún particular problema, aparte aquellos físicos de la máquina.

Puesto que "Vel_max" también depende de "Step", cambiándolo implícitamente modificación la relación entre "Volt_vel" y "Vel_max"; por tanto' siempre hace falta establecer en antelación el valor de "Step."

El valor de "Volt_vel" debería ser conocido de las características del motor y de la relación de transmisión.

Con ases rotatorios puede ser deducida por part-program, que programando una tensión a loop abierto, controlas de ello la velocidad efectiva.

Otro modo para averiguar que el valor sea corregido y' aquel de lanzar un movimiento bastante largo que alcanzar la velocidad' programada, y controlar el error de persecución que tiene que ser:

epsi = velocidad' al segundo / Kp.

Naturalmente los otros K del pid tienen que ser a cero.

La sumisión en CORRIENTE es definido poniendo a uno el bit 0x00000001 de "Kind_axis_0" de los ases en caracterización.

Hace falta respetar la relación entre "Acc_max" y "Volt_vel" porque' y' de este modo es definido al movimentazione cuantos voltios precisan a la sumisión para tener la aceleración intencional.

Cada variación de "Acc_max" tiene que ser correspondida por un proporcional de "Volt_vel."

El valor de "Dec_max" y "Vel_max" pueden ser cambiados sin ningún particular problema, aparte aquellos físicos de la máquina.

Puesto que "Acc_max" también depende de "Step", cambiándolo implícitamente modificación la relación entre "Volt_vel" y "Acc_max"; por tanto' siempre necesita tabilire en antelación el valor de "Step."

El valor de "Volt_vel" debería ser conocido de los característicos el motor y de la sumisión.

Y' menos fácil notar el valor dic "Volt_vel" en los arregla en corriente con respecto de aquel en velocidad', porque' la asignación los una tensión a loop abierto origenerebbe no una velocidad' al eje pero a una aceleración que llevaría el eje en fuga.

Por motor lineales se puede intentar con el asignar una baja tensión al motor a loop abierto y medir de ello la fuerza con un dinamometro como Fuerza = masa * aceleración, conociendo la masa se puede calcular la aceleración, aceleración = Fuerza / masa.

Otro método y estimar de ello un valor, tarar al mejor el eje es ver cuantos voltios se consiguen con cierta aceleración.

Para ver cuántos voltios se la tienen puede mirar el movimiento con la función osciloscopio.

Viene en seguida indicado un ejemplo de partprogram para notar la velocidad, con un peldaño de tensión:

EJEMPLO

```
;               ASIGNA UNA TENSIÓN A. EL EJE Z
;               Y VISUALIZA DE ELLO LA VELOCIDAD'
;
                - TSK/VEL_Z
                - LET/L1,0
AA              - KYB/DAMMI VOLTIO 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 en seguida indicado un ejemplo de partprogram para notar la velocidad, con un peldaño de tensione:e memorización de los determinados campeonatos sobre expediente

EJEMPLO

```
;       * * * * * * CONTESTADA EJE XL A OLA CUADRA CORTA * * * * * *
;       EJECUTA UN UNDULA CUADRANTE DE V VOLTIO DE LA SALIDA
;       POR UN TIEMPO, INVIERTE LA TENSIÓN POR EL MISMO
;       TIEMPO ES LA BORRA
                - DIS/1:Inizio
                - DIS/2:>
                - AZZ/\OQXL.TXT
                - SCR/\OQXL.TXT,N Pos_XL Voltio
                - SET/L10,0,L9,0
                - SET/L20,0,L21,100,L22,3
;               L20 = POS. SALIDA
;               L21 = TIEMPO en milisegundos por OLA CUADRANTE
;               L22 = VOLTIO POR DIRECCIÓN POSITIVA
                - MOV/5000:XL,L20       ; Pos. salida
                - CAP/28:XL,15          ; Voltios Máximos
                - TIM/100
                - RAV/17:L24,XL         ; L24 = V compensación
                - RAV/22:L25,XL         ; L25 = V compensación old
                - SET/L24,L24+L25
                - DIS/1:VOLT COMPENSACIÓN, L24
                - CAP/22:XL,L24         ; Voltio Compensación
                - TIM/50
                - CPL/-1:XL
                - CAP/51:XL,0                   ; buffer off
                - CAP/50:XL,1,XL,17     ; buffer on
                - SET/L5,L22              ; L5 = Voltio por ida ola cuadrante
                - SET/L6,-L5              ; L6 = Voltio por vuelta ola cuadrante
                - 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       ; Vuelta pos. salida
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
FIN             - DIS/1:Finito,L10
```

3.2 ANTICOLLISIONE

Anticollisione entre ases vinculados

Los ases vienen tan agrupados de modo que pueden entre trabajarles separadamente, los ases pueden incidir pero en único eje físico, ejemplo portal con más bracci que trabajan sobre el mismo eje portante, en este caso el sistema está capaz de solucionar el problema del anticollisione, administrando la detención momentánea de un eje en espera que lo otro se lleva por fuera en una posición estorbo.

En este caso la programación ocurre con ciclo separado por cada brazo, de otro modo sería oneroso pensar en el movimentazione de todo el sistema, será el sistema "AxesBrain" que con la función de anticollisione controlará la gestión de los estorbos, única atención es que la programación de los ciclos no entregada casos de abrazo mortal "DEAD LOCK."

PARÁMETROS DE ANTICOLLISIONE

Hace falta cargar sobre el eje los siguientes campos:

Eje caracterizado X1

```
Axis_1_name_anticollision = "XT"        / / 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
```

Dónde
A1 = nombre de un eje en anticollisione con el eje caracterizado ("X1",
B1 = valor de anticollisione con este eje, ve nota * *)
C1 = tipo de anticollisione con este eje, ve nota * * *)
A2 = nombre del otro eje en anticollisione con el eje caracterizado ("X1",
B2 = valor de anticollisione con este otro eje, ve nota * *)
C2 = tipo de anticollisione con este otro eje, ve nota * * *)

* * El valor de anticollisione y':
Distancia entre los ceros de los ases - Estorbo entre los ases
EJ. en el caso de ceros coincidentes, el valor de programar e igual
al valor negativo del estorbo entre los ases

* * * El tipo de anticollisione y':
Suma entre los siguientes valores

1, código de aproche que tiene los siguientes valores
- 1 = "X1" concorde con "XT"; "X1" se acerca a "XT" en positivo
- 2 = "X1" concorde con "XT"; "X1" se acerca a "XT" en negativo
- 3 = "X1" discorde con "XT"; "X1" se acerca a "XT" en positivo
- 4 = "X1" discorde con "XT"; "X1" se acerca a "XT" en negativo

2, acercamiento entre los ases
- 8 = "X1" se acerca a "XT" hasta a la posición de colisión
- 0 = "X1" no se acerca a "XT" pero espera de tener campo libre

3, persecución entre los ases
- 16 = "X1" sigue "XT" sin chocar
- 0 = "X1" no sigue "XT" pero espera de tener campo libre

4, cálculo sobre la posición real o artificiosa
- 32 = el cálculo de anticollisione entre "X1" y "XT" y' hecho sobre las posiciones teóricas (SIMULACIÓN)
- 0 = el cálculo de anticollisione entre "X1" y "XT" y' hecho sobre las posiciones reales (ELABORACIÓN)

5, Anticollisione encaminado por OMO sobre MARKER u OMO general
- 64 = el cálculo de anticollisione entre "X1" y "XT" y' hecho aunque el eje no y' es borrado sobre MARKER
- 0 = el cálculo de anticollisione entre "X1" y "XT" y' hecho sólo si la puesta a cero y estado hecho sobre MARKER

EJ. En el caso se quiera controlar el anticollisione con ases concordes con acercamiento negativo (2), haciendo acercar los ases (8), pero no haciéndolos seguir, 0, y sobre cuotas reales (0); necesita programar 2+8 = 10

Los parámetros deben ser puestos sobre todos los ases interesados al 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

Ases cooperanti

El eje a portal, Gantry)è una estructura mecánicamente rígida, es normalmente una estructura a portal, y corresponde por tanto a un eje único, pero es tratada por el control como si fuera constituida por una pareja de ases, eje máster y eje eslavas, cada uno con los mismos sistemas de cuenta y el propio accionamiento.
Una de las funciones del control es mantener la posición del eje "eslavas" más próximo posible a la del eje "máster."

Los movimientos requeridos sobre los ases máster-eslavas, son ejecutadas en el modo siguiente:

La solicitud de movimiento puede ser efectuada por todas las instrucciones de movimiento.

En el caso de movimiento relativo al solo eje Máster, el eje eslavas lo sigue o queda respectivamente firme en caso de asociación o disassociazione.

Durante el movimiento, el eje eslavas sigue su Máster a tiempo real

Son administrados dos tipos de GANTRY:

Aquel con estructura rígida que no y' ingallonabile;
son conformados los motores a tener la misma carga.

Aquel clásico que tiene que ser conducido por los dos motores,
respetando el más posible por cada eje la posición.

Con el tipo rígido el eje máster es mandado a la cuota intencional,
el otro eje se acostumbra.

El tipo rígido es caracterizado poniendo sobre el eje máster en

Kind_axis_1=il bit 800000H (SLAVE_CHA_GANTRY)
Axis_name_slave = "nombre eje eslavas"
K_follower_slave=100.0, 100 significa gestión indirecta, 0 no mediado,

El tipo clásico tiene las mismases informaciones, con la añadidura en
Kind_axis_1=il bit 1000000H (SLAVE_CHA_HARD)
Con este tipo también hace falta asignar la máxima tensión de descompensación entre los dos ases que y' el campo de caracterización
VoltGantry=5.0
Si el valor de VoltGantry y' <=0 la descompensación entre los dos ases no y considerado:

Además hace falta definir la desviación de los cero marker eslavos con respecto del cero marker del máster con el siguiente campo de caracterización
DeltaOffset_SlaveGantry=0.640
Estos dos campos son puestos sobre el eje máster

Nota por "DeltaOffset_SlaveGantry"
eje máster sobre cero marker con lectura de la cuota real = 0
eje eslavas con lectura de la cuota real = +2.403
El valor de caracterizar tiene que ser -2.403 que y' ella posición del cero marker eje eslavas con respecto de lo correspondiente del máster.

También poniendo el
Kind_axis_1=il bit 2000000H (SLAVE_CHA_THEOR)
el gantry asigna todo el error de cruce sobre el eje eslavas; en tal modo el máster sigue integralmente el teórico y el eje eslavas acelera o decelera para anular el error entre los ases.
Usando este bit, también por gantry rígido, el eje eslavas sigue el teórico y no se acostumbra.

La puesta a cero de los ases GANTRY y' bien hacerlo con un part-program, qué y' diferente a según de los tipos físicos de gantry.

Eje gantry no rígido con ases que no se libran
Caracterización eje máster 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

Como calcular "DeltaOffset_SlaveGantry" =
Lanzar el programa siguiente, esperar que la puesta a cero sea hecho y a la solicitud de movimiento incremental de eje máster dar los valores que ponen los ases gantry en posición de trabajo. Ligeras sobre la raya 2 el valor de introducir en caracterización.

```
- TSK/READ_GANTRY   ; Lanzamiento del task de lectura ases gantry
- CAP/100:YM,1024        ; Disabilitazione de SLAVE_CHA_HARD
- HOM/YM          ; Búsqueda micro y marker eje máster
- MOV/YM,100      ; Posicionamiento eje por puesta a cero eje eslavas
- CAP/0:YM,262144        ;0x40000 revers máster eslavas del gantry
- HOM/YS          ; Búsqueda micro y marker eje eslavas
- CAP/0:YM,-262144       ; - (0x40000) vuelta al máster eslavas del gantry
- CAP/100:YM,2048        ; Disabilitazione temporal 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
```

Como ejecutar la puesta a cero del sistema de ases gantry
Lanzar el programa siguiente

- HOM/YM ; Búsqueda micro y marker eje máster
- MOV/YM,100 ; Posicionamiento eje por puesta a cero eje eslavas
- CAP/0:YM,262144 ;0x40000 revers máster eslavas del gantry
- HOM/YS ; Búsqueda micro y marker eje eslavas
- CAP/0:YM,-262144 ; - (0x40000) vuelta al máster eslavas del gantry
- MOV/YM,200 ; Posicionamiento eje por fina puesta a cero
- END

Como sacar los ases gantry del ingallonamento
Lanzar el programa siguiente y mover el máster
hasta a tener los ases móviles

- CAP/100:YM,4096 ; El eje YM también mueve sin HOME
- CAP/100:YS,4096 ; El eje YS también mueve sin HOME
- CAP/100:YM,2048 ; Disabilitazione temporal del gantry

REP - KYB/MOV INCREM. YM,L1
- INQ/YM
- MOV/500:YM,L1
- ABQ/YM
- JMP/REP
- END

También poniendo el
Kind_axis_0=il bit 0x00000020, EJE CON SALIDA DULCE DEL SIRVO,
con el GANTRY de tipo no rígido, sometiendo los dos ases se llevan
en posición más dulcemente y el control de collision no viene
hecho por todo el tiempo de "Cycles_positioning."
Durante el movimiento, si es superado un sposizionamento entre
los dos ases del valor de sirvo error es señalado error.

3.4 Homing

Modalidad puesta a cero eje

los siguientes bits de "Kind_axis_0" de los ases en caracterización
son referidos a la función homing.

;0x00000800 = EJECUCIÓN HOMING A. GOLPE"
;0x00001000 = EJECUCIÓN HOMING SOLISTA SOBRE EL MARKER
;0x00002000 = EJECUCIÓN HOMING SOLISTA SOBRE LO MICRO
;0x00004000 = EJE QUE SÓLO se MUEVE DESPUÉS DE HOMING

El bit 0x00004000 no permite de mover el eje sin puesta a cero
hecho, a menos que ello no sea movido en manual.
Con los otros bits a cero la ejecución del homing y aquella estándar;
búsqueda micro de homing, liberación del mismo y búsqueda del marker.
Con el bit 0x00002000 a uno es investigado lo micro de homing,
concedido y borrado el eje, no es investigado el marker.
Con el bit 0x00001000 a uno viene solista fugitivo el marker
, no es investigado lo micro de homing.
Con el bit 0x00000800 a uno es ejecutado el homing a golpe,
o sea cuando el eje y' detención y es superada el umbral de golpe
el eje es borrado. En este caso el umbral de golpe y'
caracterizada en "Volt_PHASE" y el tiempo de permanencia a golpe
a "Delay_PHASE." Acordarse de sacar el eje de la posición de
golpe, porque' la tensión de sumisión quedaría alta.
Otros parámetros por el homing son caracterizados en los siguientes campos:

Number_OMO_switch_port = "Homing_ZA"
Nombre del input caracterizado por la función homing

FalseTrue_logic_OMO_switch=1
estado input de homing con micro oído

Space_OMO=-1000.0
espacio máximo y hacia de búsqueda de lo micro de omo
Notar que la liberación micro y la búsqueda del marker son hechas
en dirección opuesta a aquel de lo micro, por lo tanto si es hecha
una puesta a cero sola sobre marker este campo tiene que tener la señal
opuesto a la dirección intencional, como si hiciera una búsqueda virtual
de lo micro.

Vel_OMO=5000.00
velocidad' en mm/minuto de búsqueda de lo micro de omo

Vel_MARKER=500.00

velocidad' en mm/minuto de liberación de lo micro y búsqueda del marker

Los bits de caracterización pueden ser forzados por part program
con la instrucción - HOM/Type:Asse dónde Type tiene los siguientes sentidos:

0 = omo según caracterización (estándard)
1 = omo sin micro sólo con marker
2 = omo sin marker solo con micro
3 = omo con puesta a cero eje incondicional sin movimiento
4 = omo a "GOLPE"
8 = prueba omo sobre marker por verificación pérdida steps encoder
16 = espacio omo caracterizado en dirección positiva
por la búsqueda micro
32 = espacio omo caracterizado con dirección invertida
con respecto de la caracterización por la búsqueda micro
48 = espacio omo caracterizado en dirección negativa
por la búsqueda micro

Luego - HOM/17:A1 significa ejecución homing sobre eje A1 con búsqueda
sólo sobre marker en dirección negativa. Notar que
la dirección y' rechazo porque' no es investigado
lo micro, que y' virtualmente rebuscado en dirección
positiva.

- HOM/18:A1 significa ejecución homing sobre eje A1 con búsqueda
sólo sobre micro en dirección positiva.

Ejemplo de puesta a cero:

```
;
;               1 = mask eje ZL
;               2 = mask eje ZR
;               4 = mask eje XL
;               8 = mask eje XR
;               16 = mask eje Y
;               32 = mask eje SL
;               64 = mask eje SR
;               128 = mask eje 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 = posición eje ZL con origen hecho
                - TSK/OMOZL:0
                - JMP/DO_ZL
NO_ZL           - EVS/1
DO_ZL           - LET/L11,AND(L10,2,
                - JNE/L11,2,NO_ZR
;               0 = posición eje ZR con origen hecho
                - 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 = posición eje XR con origen hecho
;               -550 = posición eje XR por búsqueda MARKER
;               -500 = posición eje XL con origen hecho
;                600 = posición eje XL por búsqueda 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 = posición eje Y con origen hecho
;                250 = posición eje Y por búsqueda 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 = posición eje SL con origen hecho
                - TSK/OMOSL:0
                - JMP/DO_SL
NO_SL           - EVS/6
DO_SL           - LET/L11,AND(L10,64,
                - JNE/L11,64,NO_SR
;               0 = posición eje SR con origen hecho

                - TSK/OMOSR:0
                - JMP/DO_SR
NO_SR           - EVS/7
DO_SR           - LET/L11,AND(L10,128,
                - JNE/L11,128,NO_LU
;               0 = posición eje LU con origen hecho
                - TSK/OMOLU:0
                - JMP/DO_LU
NO_LU           - EVS/8
DO_LU           - EVW/5,6,7,8,9
                - DIS/0: Origen HECHO SOBRE TODOS LOS ASES DESEADOS
                - RET


PartProgram[OMOZL]
;               Origen EJE ZL
;
;               L1 = Posición eje ZL con origen hecho
;
                - HOM/ZL
                - MOV/ZL,L1
                - EVS/1
                - DIS/1: Origen ZL hecho
                - RET


PartProgram[OMOZR]
;               Origen EJE ZR
;
;               L1 = Posición eje ZR con origen hecho
;
                - HOM/ZR
                - MOV/ZR,L1
                - EVS/2
                - DIS/2:Origine ZR hecho
                - RET


PartProgram[OMOX]
```

```
;               Origen VELOCES ASES XES
;               L1 = Posición eje XR con Origen hecho
;               L2 = Posición por principio Origen eje XR
;                       , con respecto de lo micro de Origen,
;               L3 = Posición eje XL con Origen hecho
;               L4 = Posición por principio Origen eje XL
;                       , con respecto de lo micro de Origen,
                - EVC/1,2,3,4
                - TSK/OMOXR:L1,L2; -550 = Pos. inicio búsqueda marker XR
                - TSK/OMOXL:L3,L4;; 600 = Pos. inicio búsqueda marker XL
                - EVW/1,2
                - EVS/9
                - DIS/3: Origen HECHO SOBRE ASES XR y XR
                - RET


PartProgram[OMOXL]
;               Origen VELOZ EJE XL
;
;               L1 = Posición eje con Origen hecho
;               L2 = Posición por principio Origen eje XL
;                       , con respecto de lo micro de Origen,
;
                - SFP/5000,1000,1000
                - HOM/3:XL
                - DIS/2:>
                - TMS/XL,-2000:5:160,0,NTCH     ; -2000 = posiz. final
                - DIS/2:NON TOCADO 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             ; Origen sobre Marker dir. positiva
                - MOV/XL,L1             ; -500
                - DIS/4: Origen XL hecho
                - EVS/1
FINOS           -
                - RET



PartProgram[OMOXR]
;               Origen VELOZ EJE XR
;
;               L1 = Posición eje con Origen hecho
;               L2 = Posición por principio Origen eje XR
;                       , con respecto de lo micro de Origen,
```

```
;
                - SFP/5000,1000,1000
                - HOM/3:XR
                - DIS/3:>
                - TMS/XR,2000:5:159,0,NTCH      ;2000 = posiz. final
                - DIS/3:NON TOCADO 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             ; Origen sobre Marker dir. positiva
                - MOV/XR,L1             ;500
                - EVS/3
                - DIS/3: Origen XR hecho
                - EVS/2
FINOS           -
                - RET


PartProgram[OMOY]
;               Origen VELOZ EJE Y
;
;               L1 = Posición eje con Origen hecho
;               L2 = Posición por principio Origen eje Y
;                       , con respecto de lo micro de Origen,
;
                - SFP/5000,1000,1000
                - HOM/3:Y
                - DIS/1:>
                - TMS/Y,-2000:5:158,0,NTCH      ; -2000 = posiz. final
                - DIS/1:NON TOCADO MICRO
                - JMP/FINI
NTCH            - SFP/7000,1000,1000
                - RAV/1:L10,Y
                - MOV/Y,L2+L10
                - SFP/5000,1000,1000
                - HOM/49:Y              ; Origen sobre Marker dir. rechazo
                - MOV/Y,L1
                - EVS/5
                - DIS/5: Origen Y hecho
FINOS           -
                - RET




PartProgram[OMOSL]
```

```
;               Origen EJE SL
;
;               L1 = Posición eje SL con Origen hecho
;
                - OMO/SL
                - MOV/SL,L1
                - EVS/6
                - DIS/6:OMO SL hecho
                - RET


PartProgram[OMOSR]
;               Origen EJE SR
;
;               L1 = Posición eje SR con Origen hecho
;
                - HOM/SR
                - MOV/SR,L1
                - EVS/7
                - DIS/7: Origen SR hecho
                - RET


PartProgram[OMOLU]
;               Origen EJE LU
;
;               L1 = Posición eje LU con Origen hecho
;
                - HOM/LU
                - MOV/LU,L1
                - EVS/8
                - DIS/8: Origen LU hecho
                - RET
```

3.5 Determinados campeonatos

Buffer adquisición fecha campeonatos

El buffer circular utiliza un área interior.

La actividad' de bufferizzazione es activada con el mando:
CAP/50:X1,14,Y1,17..
dónde:
50 significa ACTIVO BUFFER
X1 y' el nombre del eje
14 y' el tipo de memorización de este eje
Lo todo puede ser repetido por un máximo de 8 ases.
Son previstos los siguientes tipos de memorización:

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

La actividad' de bufferizzazione es desactivada con el mando:
CAP/51:X1,0...
dónde 51: significa DESACTIVA BUFFER
X1 no y' usado
0 no y' usado

Las posiciones finales del movimiento están en función del estado eje; absolutas o incrementales o referidas a un compensación.
En el caso de movimientos circulares los ases además de los dos necesarios para ejecutar el círculo, son movidos de modo interpolado con ella circunferencia (hélice cilíndrica).
El senzo de rotación y' función más allá de que de CIR CIL también del orden de los ases sobre el movimiento.
Vale la regla de considerar el primer eje como abscisa y el según como ordenada; quardare los ases según el verso positivo, teniendo a la derecha la abscisa de izquierda y la ordenada que se aleja a lo largo de la dirección de la mirada.

Ejemplo de empleo fecha campeonatos

```
;       * * * * * * * * * * * BUFFERIZZAZIONE EJE XL * * * * * * * * * * *
;
                - DIS/10:Inizio Bufferizzazione XL
                - DIS/11:>
                - AZZ/\GET_XL.TXT
                - SCR/\GET_XL.TXT,N Pos_Reale Pos_Teorica Voltio
;
                - 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
DESPUÉS         - ADD/L10,L11
                - ADD/L9,1
                - SCR/\GET_XL.TXT,L1,L3,L9
                - JMP/LOOP
FIN     - DIS/10:Finito,L10
```

3.6 Read Axes Parameters

Lee varios parámetros de los ases, de 1 a 8 ases,
El tipo de parámetro depende del switch a1:
- RAV/a1:b1,A1,c1,B1,...

a1 = tipo de parámetro de leer, switch:)
0 = informaciones sobre eje o sistema
1 = cuotas ases reales
2 = cuotas ases teóricas
3 = servicio por marker, después de OMO,
4 = velocidad' eje
5 = cuotas ases staticizzate, de TCH TPE,
6 = cuotas ases reales absolutos
7 = cuotas ases teóricas absolutas
8 = cuotas ases staticizzate absolutos
9 = compensación relativa
10 = Over travel positivo
11 = Over travel negativo
12 = Over travel positivo absoluto
13 = Over travel negativo absoluto
14 = error de interpolación
15 = KP , proportional gain,
16 = VFF , velocity feed forward,
17 = voltio a DAC
18 = voltio a DAC absoluto
19 = KD , derivativos gain,
20 = KI , complementarios gain,
21 = Max complementarios errores
22 = voltio compensación
23 = KD2 , derivativos 2 gain,
24 = AFF , acceleration feed forward,
25 = Collision Error
26 = sirvo Error
27 = Position Error
28 = prescripción Máxima (Volts)
29 = tiempo de latencia por ALTO COUNTER
30 = tensión máxima por ALTO COUNTER
31 = velocidad' máxima por ALTO COUNTER
32 = KS (sordina)
33 = tramos no lineal INCLINACIÓN ACC, variación acc al sec,
34 = tramos no lineal INCLINACIÓN DEC, variación dec al sec,

35 = tramos no lineal TIEMPO INCLINACIÓN ACC
36 = tramos no lineal TIEMPO INCLINACIÓN DEC
37 = valor de la corrección EJE
Cotiza real = Cuota Encoder + Valor Corrección
38 = KT (predictive)
39 = velocidad' límite del eje
40 = aceleración límite del eje
41 = desaceleración límite del eje
42 = cuota de marker con correcciones
43 = cuota de marker sin correcciones
44 = distancia real de la meta
45 = distancia teórico cálculo de la meta
46 = reducción velocidad' por supero error
47 = pasos perdidos por el último Homing si activara el control sobre el encoder
50 = lectura nivela hechos en el BUFFER CIRCULAR * *1
51 = lectura nivela libre del BUFFER CIRCULAR * *2

55 = CFR, Change FeedRate%,
56 = CSP, Change SPindle%,
60 = paso Encoder Caracterizado

100 = informaciones suplementarias sobre eje

101 = informaciones hardware
PRIMEROS ELECTRINICS = Status_R
NEUTONIO = ENCODER_NEWTON
PCI2000 = ENCODER_PCI2000
SERCOS = ENCODER_SERCOS
YASKAWA = ENCODER_YASKAWA
ETEL = ENCODER_ETEL
102 = informaciones hardware
PRIMEROS ELECTRINICS = Device_R
NEUTONIO = ENCODER_MARKER
PCI2000 = ENCODER_MARKER
SERCOS = ERROR_SERCOS
YASKAWA = ERROR_YASKAWA
ETEL = ERROR_ETEL
Eje STEPPER valor PULSE, generador de frecuencia,
103 = informaciones hardware
PRIMEROS ELECTRINICS = ErrorCode_R
NEUTONIO = QUANTI_MARKER
PCI2000 = QUANTI_MARKER
SERCOS = DAC_SERCOS

YASKAWA = DAC_YASKAWA
ETEL = DAC_ETEL
Eje STEPPER valor FRECUENCIA conseguida con "PULSE"

* *1 el valor vuelto y' 0 o el número de niveles memorizados
* *2 el valor vuelto y' 0 o el número de niveles libres,
si no y' activa la gestión buffer el valor y' -1

b1 c1.... valores leídos por los ases
A1 B1.... nombres de los ases

INFO
valor leído, a bits,:
1H; EJE DE SOLA LECTURA
2H; EJE CON LOOP DE POSICIÓN ABIERTO (DAC)
4H; EJE CON OMO RALEA
8H; EJE EN SERVO ERROR (EJE FIRME) * * Y
10H; EJE EN COLLISION (EJE EN MOVIMIENTO) * * Y
20H; EJE NO EN POSICIÓN * * Y
40H; EJE EN BLOQUE CUENTA * * Y
80H; EJE INCREMENTAL
100H; EJE CON PID COMPLEMENTARIO SOBRE ANIMO
200H; EJE QUE no se TIENE QUE MOVER, UNMOVED,
400H; EJE CON SUMISIÓN ACTIVA
800H; EJE EN DRIVER KO * * Y
1000H; EJE CON CORRECCIÓN ACTIVA
2000H; EJE CON COLLISION ERROR PROBADO
4000H; EJE CON SIRVO ERROR PROBADO
8000H; EJE CON POSITION ERROR PROBADO
10000H; EJE CON ALTO COUNTER ERROR PROBADO
20000H; EJE CON DRIVER VALE ERROR PROBADO
40000H; EJE GANTRY INTERCAMBIADO EN ESLAVAS (MÁSTER)
80000H; EJE CON MOVIMIENTO AGREGADO
100000H; EJE CON ANTICOLLISION PERIÓDICO
200000H; EJE CON MOVIM. PARADO POR ANTIC. O CFR/0
400000H; EJE EN ANTICOLLISION
800000H; EJE EN DEADLOCK
1000000H; EJE CON SUMISIÓN EN CORRIENTE
2000000H; EJE EN DRIVER FAULT
4000000H; EJE EN OVER TRAVEL POSITIVO, SWITCH FIN CARRERA +)
8000000H; EJE EN OVER TRAVEL NEGATIVO, SWITCH FIN CARRERA -)

Si el nombre eje y' SYSTEM
valor leído, a bits,:
1H; LECTURA ASES ACTIVA
2H; SUMISIONES ACTIVAS
4H; RUN DE PART PROGRAMA
8H; CYCLE START
10H; CYCLE ALTO
20H; EMERGENCIA
40H; SISTEMA EN RIPARTENZA
80H; SISTEMA EN REPOSICIÓN
100H; SISTEMA CON AL MENOS UN EJE CON DRIVER KO
200H; SISTEMA CON ASES EN ANTICOLLISIONE O DEADLOCK

INFO suplementario (100)
valor leído, a bits,:
1H; EJE CON KP TIPO SQUARE
2H; EJE CON KD TIPO SQUARE
4H; EJE CON KD2 TIPO SQUARE
8H; EJE CON KI TIPO SQUARE
10H; EJE CON ANTICOLLISIONE INHABILITADA
20H; EJE ROLLOVER UNSIGNED CON ANIMO A. MENOR ESPACIO
40H; EJE CON ACELERACIÓN S-CURVA
80H; EJE CON DESACELERACIÓN S-CURVA
100H; EJE FUERA DE SERVICIO
200H; EJE NORMALMENTE INHABILITADO
400H; GANTRY QUE NO SIGUE (GANTRY RÍGIDO)
800H; GANTRY INHABILITADO
1000H; MOVIMIENTO PERMITIDO TAMBIÉN SIN OMO
2000H; EJE MOVIDO EN ACELERACIÓN
4000H; EJE MOVIDO EN DESACELERACIÓN
8000H; EJE MOVIDO A. VELOCIDAD' CONSTANTE
10000H; EJE EN PLOTTATURA
20000H; EJE CON MARKER AUTÉNTICO
40000H; EJE CON GESTIÓN MOV_SWITCH INHABILITADA

INFO hardware (101)
valor leído: Como de lámina inteligente

3 = servicio por marker
El valor extraído tiene los siguientes sentidos:

1, después de un OMO con micro y marker, la distancia de lo micro del marker.
2, después de un OMO sin micros e/o sin marker, la posición dónde el eje fue borrado
3, después de un OMO/8: o un TCH/8: la posición del marker; tal valor y' típicamente cero exacto, a menos que no él sea hecha anteriormente un OMO sobre marker. El no hallar cero significa pérdida de cuenta.

Notas:

RAV/0:G1,X1

Dónde G1 tiene el bit 800H, 2048, settato si el DRIVER y' KO.
El bit y" settato siempre, aunque el control no y' habilitado.

Puede ser sentida globalmente con la siguiente instrucción:

RAV/0:G1,SYSTEM

Dónde G1 tiene el bit 100H, 256, settato si al menos un eje y' en DRIVER KO
En este caso el bit y' settato solo si el control y' habilitado

Acordarse que el bit 100H y' activo con eje sometido, por lo tanto para "trabajar" hace falta tener el bit 2H activo y el bit 100H resettato.

3.7 Change Axes Parameters

Cambia varios parámetros de los ases, de 1 a 8 ases,
El tipo de parámetro depende del switch a1:
- CAP/a1:A1,b1,B1,c1,...
a1 = tipo de parámetro de cambiar, switch:)
0 = informaciones sobre eje
1 = unused
2 = unused
3 = unused
4 = velocidad' eje
5 = unused
6 = unused
7 = unused
8 = unused
9 = compensación relativa
10 = Over travel positivo
11 = Over travel negativo
12 = Over travel positivo absoluto
13 = Over travel negativo absoluto
14 = unused
15 = KP , proportional gain,
16 = VFF , velocity feed forward,
17 = voltio a DAC
18 = voltio a DAC absoluto
19 = KD , derivativos gain,
20 = KI , complementarios gain,
21 = Max complementarios errores
22 = voltio compensación
23 = KD2 , derivativos 2 gain,
24 = AFF , acceleration feed forward,
25 = Collision Error
26 = sirvo Error
27 = Position Error
28 = prescripción Máxima (Volts)
29 = tiempo de latencia por ALTO COUNTER, Segundo,
30 = tensión máxima por ALTO COUNTER (Volts)
31 = velocidad' máxima por ALTO COUNTER
32 = KS (sordina)
33 = tramos no lineal INCLINACIÓN ACC, variación acc al sec,
34 = tramos no lineal INCLINACIÓN DEC, variación dec al sec,

35 = tramos no lineal TIEMPO INCLINACIÓN ACC
36 = tramos no lineal TIEMPO INCLINACIÓN DEC

38 = KT (predictive)
39 = velocidad' límite del eje
40 = aceleración límite del eje
41 = desaceleración límite del eje
43 = cuota de marker sin correcciones
46 = reducción velocidad' por supero error

50 = activo BUFFER CIRCULAR
51 = desactiva o Resetta BUFFER CIRCULAR
52 = Trigger BUFFER CIRCULAR

55 = CFR, Change FeedRate%,
56 = CSP, Change SPindle%,

b1 c1.... valores leídos por los ases
A1 B1.... nombres de los ases
b1 c1.... valores cambiados a los ases

Con 35 36 el TIEMPO INCLINACIÓN tiene el siguiente sentido:
> 0 Scurve (tiempo adicional al tramo lineal)
= 0 lineal
<0 sinusoidal
Si la aceleración y' sinusoidal la desaceleración puede'
ser lineal o sinusoidal pero no Scurve; en éste
caso es forzado la desaceleración sinusoidal.

INFO
CAP/0:X1,4 Fuerza el estado de OMO hecho sobre eje X1 sin borrar la cuota
CAP/0:X1,-4 Sacado el estado de OMO hecho sobre eje X1
CAP/0:X1,128 Activo lo incremental sobre eje X1
CAP/0:X1,-128 Sacado lo incremental de eje X1
CAP/0:X1,256 Activo lo complementario con movimiento sobre eje X1
CAP/0:X1,-256 Sacado lo complementario con movimiento sobre eje X1
CAP/0:X1,512 Fuerza el estado de eje atascado sobre eje X1
CAP/0:X1,-512 Sacado el estado de eje atascado sobre eje X1
CAP/0:X1,4096 Activo la corrección sobre eje X1, si el tablero y' cargada,
CAP/0:X1,-4096 Sacado la corrección sobre eje X1
CAP/0:X1,8192 Activo la gestión sobre eje X1 de "COLLISION ERROR"
CAP/0:X1,-8192 Sacado la gestión sobre eje X1 de "COLLISION ERROR"
CAP/0:X1,16384 Activo la gestión sobre eje X1 de "SERVO ERROR"
CAP/0:X1,-16384 Sacado la gestión sobre eje X1 de "SERVO ERROR"

CAP/0:X1,32768 Activo la gestión sobre eje X1 de "POSITION ERROR"
CAP/0:X1,-32768 Sacado la gestión sobre eje X1 de "POSITION ERROR"
CAP/0:X1,65536 Activo la gestión sobre eje X1 de "ALTO COUNTER ERROR"
CAP/0:X1,-65536 Sacado la gestión sobre eje X1 de "ALTO COUNTER ERROR"
CAP/0:X1,131072 Activo la gestión sobre eje X1 de "DRIVER_OK ERROR"
CAP/0:X1,-131072 Sacado la gestión sobre eje X1 de "DRIVER_OK ERROR"
CAP/0:YM,262144 Intercambiado un máster GANTRY caracterizado con eslavas
CAP/0:YM,-262144 Restablecido como de caracterización el GANTRY

INFO suplementario (100)
CAP/100:X1,1 Activo la gestión del KP de modo SQUARE del eje X1
CAP/100:X1,-1 Puesto la gestión del KP de modo LINEAL del eje X1
CAP/100:X1,2 Activo la gestión del KD de modo SQUARE del eje X1
CAP/100:X1,-2 Puesto la gestión del KD de modo LINEAL del eje X1
CAP/100:X1,4 Activo la gestión del KD2 de modo SQUARE del eje X1
CAP/100:X1,-4 Puesto la gestión del KD2 de modo LINEAL del eje X1
CAP/100:X1,8 Activo la gestión del KI de modo SQUARE del eje X1
CAP/100:X1,-8 Puesto la gestión del KI de modo LINEAL del eje X1
CAP/100:X1,16 Inhabilitado el anticollisione del eje X1
CAP/100:X1,-16 Habilitado el anticollisione caracterizado eje X1
CAP/100:X1,32 Si el eje y' ROLLOVER UNSIGNED se mueve a menor espacio
CAP/100:X1,-32 Si el eje y' ROLLOVER UNSIGNED se mueve a espacio mandado
CAP/100:X1,64 Puesto la aceleración de tipo S-CURVA
CAP/100:X1,-64 Sacado la aceleración de tipo S-CURVA
CAP/100:X1,128 Puesto la desaceleración de tipo S-CURVA
CAP/100:X1,-128 Sacado la desaceleración de tipo S-CURVA
CAP/100:X1,256 Excluido el eje, EJE FUERA DE SERVICIO,
CAP/100:X1,-256 Reinclude el eje (EJE EN SERVICIO)
CAP/100:X1,512 Definido el eje como NORMALMENTE INHABILITADO
CAP/100:X1,-512 Sacado la función NORMALMENTE INHABILITADA
CAP/100:YM,1024 Definido el gantry como gantry rígido
CAP/100:YM,-1024 Repuesto el gantry como "no rígido" si caracterizara
CAP/100:YM,2048 Inhabilitado el sistema gantry, sólo sobre eje máster,
CAP/100:YM,-2048 Rehabilitado el sistema gantry, sólo sobre eje máster,
CAP/100:YM,4096 Permitido el movimiento también sin OMO
CAP/100:YM,-4096 Rehabilitado la prueba sobre OMO hecho, si y' caracterizado,
CAP/100:YM,65536 Definido el eje en pase de plottatura
CAP/100:YM,-65536 Sacado el estado de plottatura
CAP/100:YM,131072 Fuerza el estado marker verdadero, si omo hecho, por debug
CAP/100:YM,-131072 Sacado el estado marker verdadero, si omo hecho, por debug
CAP/100:X1,262444 Inhabilitado la gestión MOV_SWITCH
CAP/100:X1,-262444 Rehabilitado la gestión MOV_SWITCH, si caracterizara,
CAP/100:X1,0x80000 Otro modo para ejecutar HOM/3: , MARKER_VERO saca,
CAP/100:X1,-0x80000 Reconducido él cero sobre el MARKER, MARKER_VERO pone,

Por algunos valores de caracterización dando como CÓDIGO
el valor correspondiente en OR con 80H también es modificado
el valor sobre estática; o sea el valor también queda después de REPOSICIÓN
Los códigos son:
(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, Sirvo Error
(27) (155, Position Error
(28) (156, Prescripción Máxima (Volts)

(0) (128, INFO para los siguientes sottocodici
(256) MSK_INTEGRATIVO_MOV
(512) MSK_DYNAMIC_OFFSET
(8192) COLLISION ERROR
(26384) SIRVO ERROR
(32768) POSITION ERROR
(65536) ALTO COUNTER ERROR
(131072) DRIVER_OK_ERROR

(100) (228, INFO suplementario para los siguientes sottocodici
(1) MSK_KP_SQUARE
(2) MSK_KD_SQUARE
(4) MSK_KD2_SQUARE
(8) MSK_KI_SQUARE
(64) ACELERACIÓN S-CURVA
(128) DESACELERACIÓN S-CURVA
(512) NORMALMENTE INHABILITADO

CAP/9:X1,50 Puesto la compensación relativa sobre eje X1: es desplazado
él cero con respecto de la compensación absoluta de 50 en positivo.
, Si el eje fuera a cero después de esta instrucción se convierte en -50,
Puesto que lo es desplazado cero con respecto de la compensación absoluta,
y no con respecto del cero relativo, cada cambio no tiene
cuenta de aquel precedente.
Luego para volver a cero absoluto basta con dar un cambio

con valor 0.

CAP/50:X1,14,X1,17 Habilita el buffer a memorizar EPSI por X1 y VOLTIO por X1.
En este caso el valor que sigue el eje y' el código RAV correspondiente. Se pueden tamponar de 1 a 8 ases. Son válidos los siguientes códigos:
1 = cuotas ases reales
2 = cuotas ases teóricas
6 = cuotas ases reales absolutos
7 = cuotas ases teóricas absolutas
14 = error de interpolación
17 = voltio a DAC
18 = voltio a DAC absoluto
48 = tiempo de muestreo
49 = tiempo usado en muestreo
90 = Speed hardware, P.E. digital pasiva,
91 = Current hardware, P.E. digital pasiva,

CAP/51:X1,0 Inhabilita la memorización sobre buffer
El eje no y' usado.
Si el valor y' >= 0 la memorización del buffer es parado, éste ocurre naturalmente con el buffer lleno,
Si el valor y' <0 el área del buffer viene borrada

CAP/52:X1,1,X1,100 Activa el trigger sobre buffer circular
El eje X1 hace partir la memorización cuando la cuota real y' >= a 100.
CAP/52: tiene que preceder CAP/50:, y viene anulada por CAP/51:
La condición de trigger es hecha sobre un solista eje con los siguientes códigos:
1 = cuotas ases reales
2 = cuotas ases teóricas
6 = cuotas ases reales absolutos
7 = cuotas ases teóricas absolutas
14 = error de interpolación
17 = voltio a DAC
18 = voltio a DAC absoluto

90 = Speed hardware, P.E. digital pasiva,
91 = Current hardware, P.E. digital pasiva,
Si el código y' positivo la condición y' >= al valor; con código negativo la condición y' <= al valor.

TENTATIVA EN CORRIENTE DE USAR UN TIPO DE VFF

QR0 = Cuota Real 0
QR1 = Cuota Real 1
QR2 = Cuota Real 2

SPAZ = Velocidad' Teórica 1 = QT2 - QT1

ACC = SPAZ -, QR2 - QR1, = SPAZ - QR2 + QR1

ACC = aceleración necesaria para alcanzar la velocidad' teórica
sensa preocuparse espacio recorrido, que será controlado del PID

Por tanto' será usado el algoritmo siguiente por sumarle al PID
, SPAZ - QR2 + QR1, * KC * VFF
, SPAZ - VR2, * KC * VFF

Tratando de también corregir los errores de aceleración se consigue:
ACC = VR2-VR1 =, QR2 - QR1, -, QR1 - QR0, = QR2 - 2*QR1 + QR0
por tanto'
ERRACC = Acc Teórica - QR2 + 2*QR1 - QR0
ERRACC = Acc Teórica - VR2 + VR1
en fin
, SPAZ - QR2 + QR1 + AccTeorica - QR2 + 2*QR1 - QR0, * KC * VFF

Este algoritmo permite de controlar el eje solo usando el KP
el VFF y eventualmente KI y AFF

, SPAZ + AccTeorica - 2*QR2 + 3*QR1 - QR0, * KC * VFF
, SPAZ + AccTeorica - 2*(QR2 - QR1, +, QR1 - QR0,) * KC * VFF
, SPAZ + AccTeorica - 2*VR2 + VR1, * KC * VFF

Analizando mejor el todo se notará' que:

QR0 = Cuota Real 0
QR1 = Cuota Real 1
QR2 = Cuota Real 2

SPAZ = Velocidad' Teórica 1 = QT2 - QT1

DVEL = SPAZ -, QR2 - QR1, = QT2 - QT1 - QR2 + QR1 = (QT2-QR2) - (QT1-QR1)
pero QT2-QR2 = E2 y QT1-QR1 = E1
DVEL = E2-E1 = Diferencia entre la velocidad' teórica y aquella real

Análogamente también considerando la aceleración

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 = Diferencia entre la aceleración teórica y aquella real

Todo eso' lleva a entender mejor el funcionamiento del PID
KP transforma en tensión una parte del error de posición
KD transforma en tensión una parte del error de velocidad'
KD2 transforma en tensión una parte del error de aceleración

INFO
CAP/0:X1,4 Fuerza el estado de OMO hecho sobre eje X1 sin borrar la cuota
CAP/0:X1,-4 Sacado el estado de OMO hecho sobre eje X1
CAP/0:X1,128 Activo lo incremental sobre eje X1
CAP/0:X1,-128 Sacado lo incremental de eje X1
CAP/0:X1,256 Activo lo complementario con movimiento sobre eje X1
CAP/0:X1,-256 Sacado lo complementario con movimiento sobre eje X1
CAP/0:X1,4096 Activo la corrección sobre eje X1, si el tablero y' cargada,
CAP/0:X1,-4096 Sacado la corrección sobre eje X1
CAP/0:X1,8192 Activo la gestión sobre eje X1 de "COLLISION ERROR"
CAP/0:X1,-8192 Sacado la gestión sobre eje X1 de "COLLISION ERROR"
CAP/0:X1,16384 Activo la gestión sobre eje X1 de "SERVO ERROR"
CAP/0:X1,-16384 Sacado la gestión sobre eje X1 de "SERVO ERROR"
CAP/0:X1,32768 Activo la gestión sobre eje X1 de "POSITION ERROR"
CAP/0:X1,-32768 Sacado la gestión sobre eje X1 de "POSITION ERROR"
CAP/0:X1,65536 Activo la gestión sobre eje X1 de "ALTO COUNTER ERROR"
CAP/0:X1,-65536 Sacado la gestión sobre eje X1 de "ALTO COUNTER ERROR"
CAP/0:X1,131072 Activo la gestión sobre eje X1 de "DRIVER_OK ERROR"
CAP/0:X1,-131072 Sacado la gestión sobre eje X1 de "DRIVER_OK ERROR"
CAP/0:YM,262144 Intercambiado un máster GANTRY caracterizado con eslavas
CAP/0:YM,-262144 Restablecido como de caracterización el GANTRY

INFO suplementario (127)
CAP/127:X1,1 Activo la gestión del KP de modo SQUARE del eje X1

CAP/127:X1,-1 Puesto la gestión del KP de modo LINEAL del eje X1
CAP/127:X1,2 Activo la gestión del KD de modo SQUARE del eje X1
CAP/127:X1,-2 Puesto la gestión del KD de modo LINEAL del eje X1
CAP/127:X1,4 Activo la gestión del KD2 de modo SQUARE del eje X1
CAP/127:X1,-4 Puesto la gestión del KD2 de modo LINEAL del eje X1
CAP/127:X1,8 Activo la gestión del KI de modo SQUARE del eje X1
CAP/127:X1,-8 Puesto la gestión del KI de modo LINEAL del eje X1

Por algunos valores de caracterización dando como CÓDIGO
el valor correspondiente en OR con 80H también es modificado
el valor sobre estática; o sea el valor también queda después de REPOSICIÓN
Los códigos son:
(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, Sirvo Error
(27) (155, Position Error
(28) (156, Prescripción Máxima (Volts)

(0) (128, INFO para los siguientes sottocodici
(256) MSK_INTEGRATIVO_MOV
(512) MSK_DYNAMIC_OFFSET
(8192) COLLISION ERROR
(26384) SIRVO ERROR
(32768) POSITION ERROR
(65536) ALTO COUNTER ERROR
(131072) DRIVER_OK_ERROR

(127) (128, INFO suplementario para los siguientes sottocodici
(1) MSK_KP_SQUARE
(2) MSK_KD_SQUARE
(3) MSK_KD2_SQUARE
(4) MSK_KI_SQUARE

CAP/9:X1,50 Puesto la compensación relativa sobre eje X1: es desplazado
él cero con respecto de la compensación absoluta de 50 en positivo.
, Si el eje fuera a cero después de esta instrucción se convierte en -50,

Puesto que lo es desplazado cero con respecto de la compensación absoluta, y no con respecto del cero relativo, cada cambio no tiene cuenta de aquel precedente.
Luego para volver a cero absoluto basta con dar un cambio con valor 0.

CAP/50:X1,14,X1,17 Habilita el buffer a memorizar EPSI por X1 y VOLTIO por X1.
En este caso el valor que sigue el eje y' el código RAV correspondiente. Se pueden tamponar de 1 a 8 ases. Son válidos los siguientes códigos:
1 = cuotas ases reales
2 = cuotas ases teóricas
6 = cuotas ases reales absolutos
7 = cuotas ases teóricas absolutas
14 = error de interpolación
17 = voltio a DAC
18 = voltio a DAC absoluto

CAP/51:X1,0 Inhabilita la memorización sobre buffer
El eje y el valor no son usados.

```
     ---------------------------------------------------------------
     - DIS/1:>
     - DIS/2:>
PRINCIPIO -
     - MOV/Y,0
     - TCH/Y,100:10:L1,Y:TOCCATO
     - DIS/1:NON TOCADO EN IDO
     - TMM/2000
     - JMP/RITORNO
TOCADO -
     - DIS/1:VALORE, L1
     - TMM/2000
VUELTA    -
     - TCH/Y,0:1:L1,Y:SENTITO
     - DIS/2:NON OÍDO EN VUELVO
     - TMM/2000
     - JMP/INIZIO
OÍDO -
     - DIS/2:VALORE, L1
```

- TMM/2000
- JMP/INIZIO

Change Position Loop

Cambia el loop de posición del eje
El tipo de "loop" depende del switch a1:

- CPL/a1:A1

a1 = tipo de "loop", switch:)
-2 = sacado sumisión + Enable
-1 = sacado sumisión (DAC)
0 = sacado sumisión (SAC) (Estándard,
1 = repuesto sumisión + Enable
2 = Phase axis

A1 nombre del eje

Activando el "loop" -1, DAC, se puede mandar el eje a loop abierto con la instrucción:
- CAP/4:.. , - CAP/17:.. Y - CAP/18:.. .

Activando el "loop" 0, SAC, se puede mandar el eje a loop abierto con la instrucción - SAC /..

Ejemplo de empleo de RAV y CAP

```
;       VISUALIZACIÓN 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 de ZR
        - CAL/,HEX.PP,L1,1,40
        - DIS/2:Device de ZR
        - CAL/,HEX.PP,L2,2,40
;       - RAV/101:L3,Y
;       - RAV/102:L4,Y
;       - DIS/3:Status_R de Y,L3
;       - DIS/4:ErrorCode_R de Y,L4
;       - RAV/101:L5,XR
;       - RAV/102:L6,XR
;       - DIS/5:Status_R de XR,L5
;       - DIS/6:ErrorCode_R de XR,L6
;       - RAV/100:L7,SYSTEM
;       - DIS/7:DDD,L7
        - TMM/10
        - JMP/AA

;       VISUALIZACIÓN EN ESADECIMALE
;
;               L1 = VALOR, 0 - 2**32,
;               L2 = RAYA
;               L3 = COLUMNA INICIAL
;
        - 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
        - JMP/AA
13      - 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

;       * * * * * * * * * * * POSICIONAMIENTOS EJE XL * * * * * * * * * * *
;                       MEDICIÓN POSICIONAMIENTOS
                - 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      ; Complementario con movim.
UUU             - SFP/50000,7500,7500
                - SET/L1,0,L2,700,L3,700,L10,10,L20,0
;               L1 = POSICIÓN DE SALIDA
;               L2 = POSICIÓN FINAL
;               L3 = INCREMENTO DE POSICIÓN
;               L10 = NÚMERO MEDICIONES ADELANTE O ATRÁS
;               L20 = TIEMPO ESTACIONA EN MICROSEGUNDOS
                - 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: POSICIÓN =, 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: POSICIÓN =, 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: POSICIÓN =, L1
                - TMM/L20
TTT             - RWA/L1,1
                - DIS/0:TEMPO =, L1
```

3.8 Ases ROLLOWER

Para definir un eje como ROLLOVER y' necesario settare
Kind_axis_1 = el bit 0X00000008, ROLLOVER 0 +)
y al mismo tiempo cargar con el valor del módulo
el campo Revolution =, que no debe ser cero.

La visualización del eje y' siempre incluida entre
0 y +MODULO; para visualizar la cuota absoluta usar
- RAV/42:

Por estándard los movimientos van en la dirección mandado,
hasta a alcanzar la posición final, que y' fecha en
coordinados rollover.
Usando la instrucción - CAP/100:A1,32 todos los movimientos
qué siguen son hechos recorriendo lo espacio menor.
No es recorrido un valor> de MÓDULO también dando
um movimiento muy grande.
Esta instrucción queda activa hasta tiene que no venga
anulada de - CAP/100:A1,-32 o del relance de
AxesBrain. También la reposición no cambia el estado.

Ej. eje con módulo 360 visualizzante 60; mandando
3670 (360*10+70) el eje se lleva a 70 recorriendo 10.

Por tanto' si es leída la cuota con - RAV/42: y luego
movido el eje a esta cuota ello queda firme. Éste
sólo utilizando la función "recorrida menor" (- CAP/100:A1,32,

Recordar que no utilizando la función "recorrida
menor" (- CAP/100:A1,32, los movimientos recorren todo
el espacio mandado.

Ej. eje con módulo 360 visualizzante 60; mandando
3670 (360*10+70) el eje se lleva a 70 recorriendo 3610.
También existe el eje ROLLOVER SEÑALADO, que se puede'
activar con Kind_axis_1 = el bit 0X00000010, ROLLOVER + - MÓDULO,
En este caso las cuotas varían entre + - módulo
y no funciona - CAP/100:A1,32. Además no siendo
verdaderamente útil, podría ser una prestación
de anular.

3.9 OVER TRAVEL

EL OVER TRAVEL puede ser definido en caracterización poniéndole en Positive_over_travel = y Negative_over_travel = los valores de fina carrera referidos al cero absoluto. Luego si Offset_axis es cambiado = y' necesario también cambiar los límites de over travel porque' es cambiado la compensación absoluto.
Estos límites son activados por la puesta a cero ases; moviendo en manual el eje se para sobre el límite que se quiere superar. Los demás movimientos no parten si se supera el límite de corrida, pero da una señal de error.
Si ambos los límites son cero, no es considerado el límite de OVER TRAVEL.

Animas "hasta" a OVER TRAVEL
NOTAS por los movimientos lanzados por HANDLING_MOV
Poniendo en el campo MODIFYER_MOV de la estructura de los movimientos el valor 0x4000 (or con los otros bites del campo) los movimientos son ejecutados hasta al límite si éste es superado. No son tratados de este modo los movimientos en continuo y circulares y HOM.

Utilizando los campos:
Number_PositiveOverTravel_switch_port =
FalseTrue_logic_PositiveOverTravel_switch =
Number_NegativeOverTravel_switch_port =
FalseTrue_logic_NegativeOverTravel_switch =
también son sentidos los micros de fina carrera. Ellos son administrados en el siguiente modo:
1, caso normal que para prescindir de la puesta a cero ases es señalado error con caída de sumisión si quieren superarlos, la dirección del movimiento es probada con el corresponsal micro de más allá de carrera,

2, caso con Kind_axis_0 = que tiene el bit 0x00000040 settato; a prescindir de la puesta a cero ases el movimiento en manual cuando siente lo micro decelera y se para sin nadie señal.
La misma cosa sucede con los movimientos "hasta" a OVER TRAVEL
Por todos los demás movimientos vale el control OVER TRAVEL con los límites en caracterización; si es encontrado el micro de más allá de carrera es señalado error con caída de sumisión.

3.10 Ases sólo sometidos en movimiento

El eje normalmente INHABILITADO y' un eje que cuando y' paro no
y' sometido y si tiene el freno y' frenado. Con movimiento ello
él autoasserve liberando el eventual freno hasta a movimiento
concluido.
El eje puede ser caracterizado tal settando en
Kind_axis_0 = el bit 0x00000080

También de part-program se puede' attivare/disattivare sobre el eje
la función de normalmente inhabilitado con las siguientes instrucciones:

1, CAP/100:A1,512 de este modo el eje y' enviado
normalmente inhabilitado, pero después de una reposición
vuelve en el estado de caracterización
2, CAP/100:A1,-512 de este modo el eje no y'
normalmente inhabilitado, pero después de una reposición
vuelve en el estado de caracterización
3, CAP/258:A1,512 de este modo el eje y' enviado
normalmente inhabilitado, pero también después de una
reposición
queda en este estado
4, CAP/258:A1,-512 de este modo el eje no y'
normalmente inhabilitado, pero también después de una
reposición
queda en este estado

CAPÍTULO 4

4 Configuración nombre ases por interfaz manual

Para poder administrar los recursos de sistema en su totalidad con los ciclos de elaboración es necesario configurar los siguientes parámetros.
En el entorno "Arregla" la página manual permite de mover los ases de modo manual.
Y necesario pero configurar los ases sobre la interfaz agrupándolos.
En el expediente de sistema "Homemoveman.ini" en la sesión **[ConfiguraAssi]** y posible configurar la situación de los ases deseada.

[ConfiguraAssi]

NumeroAssi=Numero ases de configurar
NumeroGruppi=Numero grupos de agrupar

¡Asse_01 = Nombre eje le visualizado en el "Combo Box!"
¡Nombre eje físico configurado en en el expediente "sistema.txt!"
¡Numero Grupo!
¡Escrita Botón +!
¡Escrita Botón -!
l eje lineal o bien **m** eje mandril
….
¡Asse_xx = Nombre eje le visualizado en el "Combo Box!"
¡Nombre eje físico configurado en en el expediente "sistema.txt!"
¡Numero Grupo!
¡Escrita Botón +!
¡Escrita Botón -!
l eje lineal o bien **m** eje mandril

¡Como se ha podido verse usa el separador! para dividir los campos

Ejemplo

[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

CAPÍTULO 5

5 Ciclos siempre activos

Para poder administrar los recursos de sistema en su totalidad con los ciclos de elaboración es necesario configurar los siguientes parámetros.

[ParametriAUTOMAZIONE]
StringaLancio = \ AxesBrainSTUDIO \ PARTPROG\AUTO.PP
NomePartProgramLancio = AUTO.PP

Notas:
Con la caracterización estándar aquí enseñada, el sistema se encamina automáticamente el Part lanzando program "auto.pp", o bien con otro nombre dado por el desarrollador del Sistema,
Siendo como este Part Program editabile un normal Part program, si se quiere que el Sistema se encamina con un recorrido diferente y no accesible del entorno, hace falta definir este recorrido particular en el "StringaLancio ="

Ejemplo: **StringaLancio=\programmi\applicazione\partprog\start.pp**

CAPÍTULO 6

6 Corrección ases

Para poder administrar los recursos de sistema en su totalidad con los ciclos de elaboración es necesario configurar los siguientes parámetros.

- Lineal corrección lineal ass, de cuadratura, de balanceo y de cabeceo
- Parrilla corrección con parrilla de compensación en el plan

6.1 Lineal

La corrección eje y' activada leyendo los valores de corrección
sobre el expediente precisado en el expediente "**sistema.txt":**

sección: [ParametriGenerali]
voz: FileCorrezione=Correz.txt

El expediente tiene que tener el siguiente tamaño:

PHISICAL_AXIS_NAME, CORRECTED = "YM" INPUT = "YM",

-527.00000 = 0.00000
-514.30000 = -0.00104
.....

.....
31.80000 = -0.00880
44.50000 = -0.01265

CORRECTED = y' el nombre del eje físico sobre que actúa la corrección
INPUT = y' el nombre del eje físico de que son leídas las posiciones
de corrección, valor a mano izquierda de =)

Si el valor de izquierda de INPUT = soy el nombre de dos ases
, separados por una coma, se usa el tablero atado a lo primero
o según eje según el último eje movido. Naturalmente
para tener un funcionamiento correcto occore que dos tableros
de correcciones hagan referencia a un único eje corregido
y qué gil ases input comparezcan en ambas pero con orden
invertido.

El valor a mano izquierda de la señal = y' la posición referida al
cero marker, en coordinada escalera, sin compensación absoluta.
Para conseguir el valor en coordinada escalera hace falta sumar al
cotiza encoder leídos la compensación absoluta escrita en caracterización

El valor a la derecha de la señal = y' el error realzado en aquellos
pica escalera. El error y' conseguido sustrayendo a la posición
láser (valor verdadero) la posición encoder leído, valor encoder,

/ / THEORETICAL_POSITION: POSITION ENCODER HOJEADA + COMPENSACIÓN ABSOLUTE CHARATTERIZATION
/ / ERROR_POSITION: POSITION LÁSER (TRUE) - POSITION ENCODER HOJEADA

Para ejecutar el láser hace falta inhabilitar la corrección a los ases interesados.
Este se puede conseguir en los siguientes modos:
1, cambiar el nombre al expediente de corrección antes de lanzar el movimentazione; en este caso ninguna corrección y' activa.

2, cambiar el nombre al eje sobre el expediente de corrección con un nombre inexistente antes de lanzar el movimentazione; en este caso solo los ases modificados no tienen corrección.
3, inhabilitar la corrección con la instrucción CAP/0:YM,-4096,YS,-4096
En este caso solo los ases YM y YS están sin corrección.
La corrección sacada de este modo puede ser rehabilitada o dando la instrucción CAP/0:YM,4096,YS,4096 o con "REPOSICIÓN."

El error de posición del láser y' de transformarle en coordinado hombre (ej. de micrómetros en mm) con la propia señal.

Las correcciones son acumulativas sobre el eje correcto; este de la posibilidad' de también corregir el eje en función de errores función de otros ases.

6.2 Parrilla

Corrección con parrilla de compensación en el plan

Las correcciones en el espacio actúan sobre una terna de ases cartesianos
El orden de los ases correctos y' aquel enviado en CORRECTED
El orden de los ases usados y' aquel enviado en INPUT

Cada raya de datos tiene el siguiente sentido:
valor 1 = posición eje 1 INPUT
valor 2 = posición eje 2 INPUT
valor 3 = posición eje 3 INPUT, si previera,

separador =

error 1 = error de asignar al eje 1 CORRECT
error 2 = error de asignar al eje 2 CORRECT
error 3 = error de asignar al eje 3 CORRECT, si previera,

Pueden haber 2 o 3 ases INPUT y 2 o 3 ases CORRECT
El número de ases CORRECT y' independiente del número de
ases INPUT

no y necesario que el tablero sea equispaziata pero los valores
sobre las rayas, sobre las columnas y sobre las altitudes tienen que ser
respetados por todos los puntos.
El número de puntos del tablero y': 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

CAPÍTULO 7

7 CALIBRADOS ASESPID, PROPORCIONAL, COMPLEMENTARIO, DERIVATIVO

7.1 DATOS PERSONALES ES PREMISAS

Son previstos dos muchos modos para someter los ases que llamaremos:

- sumisión en VELOCIDAD
- sumisión en PAREJA

De costumbre es utilizado la sumisión en VELOCIDAD, con el que se manda a la UNIDAD DE DISCO del motor una señal, generalmente incluido entre + / - 10V.) cuyo valor representa, por analogía, la velocidad a la que debe, instante por instante, girar el motor para realizar la velocidad de desplazamiento prevista por el eje en cuestión en el ciclo de posicionamiento.
La sumisión en VELOCIDAD es prevista con conexión de tipo Analógico e/o digital, Sercos, etcétera,

La sumisión en PAREJA, dice también "en CORRIENTE", utilizado en casos particulares con accionamientos que lo solicitan, no teniendo interiormente el cálculo de los coeficientes PID,
La descripción del sentido de los parámetros por la sumisión en PAREJA es pospuesta a un específico párrafo dedicado.

Para seleccionar el tipo de pilotaje hace falta predisponer la oportuna máscara esadecimale, enseñada y descrita sucesivamente, denominada **Kind_Axis_0** poniendo el bit de peso 1 de la cifra significativa

= 0 (cero) por la sumisión en VELOCIDAD, predispuesto por estándard,
= 1 (uno) por la sumisión en PAREJA

CONOCIDOS PRELIMINARES

- Todos los valores de calibrado usan como divisor entre las cifras enteras y los decimales el punto, **0.005** y no la coma (**0,005)**
- Las unidades de medida dimensional (mm) inches, grados, rpm, etcétera, definidas por cada Eje en el parámetro STEP, es unívocas para todos los parámetros dimensionales de aquel eje y vendrán de ahora en en fin llamadas UDM, **Unidad** de **Medida Dimensionale)nell'intero** documento. Pertanto,in aplicaciones particulares, son posible, queriendo, asignar unidad de medida diferente a los individuales Ases.
- Algunos calibrados pueden ser modificáis dinámicamente de Part program por las instrucciones **CAP, Change Axis Parameter,**

7.2 CARACTERIZACIÓN ASES ANALÓGICOS EN VELOCIDAD

7.2.1 CALIBRADOS GENERALES DE SISTEMA RELATIVOS ATUTTI ÉL ASES

GeneralStatus=0x00000000 Disfraza general de sistema-Valor de default=0
, Sólo utilizado en casos particulares-no modificar,

Time_collision_latency Tiempo, en sec.) de mantenimiento de los ases bajo sumisión cuando un eje, en Error, es parado. - Valor de default=0,2

7.2.2 CARACTERIZACIÓN ASES FÍSICOS - PREDISPOSICIONES

Per ogni Asse Analogico si hanno a disposizione, e devono essere caratterizzati, tutti i parametri seguenti

Por cada Eje Analógico se tienen a disposición, y tienen que ser caracterizados, todos los parámetros siguientes

[X] Nombre o símbolo de individuación de un Eje
, alfanumérico max 7 carácteres,

i=PHYSICAL_AXIS_NAME(1,"X")
Reconducir el mismo nome/simbolo dado al eje.
El número que precede el nombre, en el ejemplo =1, es uno
indicación auxiliar del nombre, que puede ser utilizada o menos, pero que en todo caso, si existe, es reconocida por la programación a la igual del nombre del eje, por ejemplo, la elección del eje sobre que obrar puede ser definida por el resultado de un cálculo,

b={ Indicación de Sistema-no MODIFICAR

Description = "Eje X" Nombre que compadre sobre las pantallas, libero,da definir, como por ejemplo: X, Transversal, Rotatorio, etcétera,

Kind_axis = "ARM" Indicación de Sistema-no MODIFICAR

Kind_axis_0=0x00000000 Máscara de los calibrados del **Tipo de Eje.**
Valor de estándard = **4000**

Valores previstos y caratterizzabili: NO UTILICES VALORES PREVISTOS

0x00000001 - Sumisión Eje en PAREJA, Valor de estándard = **0 -** VELOCIDAD,
0x00000010 - KI con funcionamiento no lineal, más grande con error bajo,
Valor de estándard = **0 (**KI con funcionamiento lineal)
0x00000020 - La primera sumisión del eje después de haber dado, o repuesto, potencia ocurre de modo gradual. Eso es particularmente útil para permitir un correcto arreglo garajista de los Ases Gantry después de un apagamiento de la máquina.
0x00000040 - El eje decelera sobre lo micro de fina carrera, si existe, sólo en Manual. Útil cuando se mueven los Ases antes de haber hecho los Ceros
0x00000080 - La capacitación del eje, Enable, viene automáticamente fecha y sacada al principio y al final de un movimiento. Útil para aquellos Ases que necesitan bloqueo a fin animo.

0x00000800 - Ejecución OMO a golpe.
0x00001000 - Ejecución OMO solo sobre el Marker
0x00002000 - Ejecución OMO solo sobre lo Micro.

0x00004000 - El eje sólo se mueve después de OMO Valor de estándard = **4000**

Kind_axis_1=0x00070000 Máscara de los tipos de Eje - Valor de estándard **1F0000**
Este valor de estándard predispone el control de los errores de
Collision, Sirve, Position, Alto Counter, Unidad de driver ok.
Valores previstos y caratterizzabili: NO UTILICES VALORES PREVISTOS

0x00000004-Eje con capacitación Sirvo después del 1° movimiento - Valor de estándard = **0**
0x00000008-Eje Rollover, rotativo, con reposición a 360°, o bien a otro valor, a ejemplo como N° de estaciones. Si =0 el eje no se borra pasando por 360° y sigue contando 361, 362 etcétera
0x00000040 - Eje "Traslado" con dinámica por Traslado elevadores y anillo de posición floja
Valor de estándard = **0.**
0x00000080-Eje con tramo de aceleración S-CURVA si el bit de peso **80** es igual a 1.
Valor de estándard = **0,** la aceleración predispuesta es por tanto sinusoidal,

0x00000100-Eje con tramo de desaceleración S-CURVA si el bit de peso **100** es igual a 1.
Valor de estándard = **0,** la desaceleración predispuesta es por tanto sinusoidal,
0x00000200-Eje con sistema de puesta en fase, Motor lineales,-Valor de estándard = **0**
0x00010000-Activación del error de COLLISION. Si pusiera = 0 el collision no viene controlada-Valor de estándard = **1**
0x00020000-Activación del SIRVO ERROR A EJE PARO. Si pusiera = 0 el sirvo error no es controlado-Valor de estándard = **1**
0x00040000-Activación del POSITION ERROR (EJE EN MOVIMIENTO). Si pusiera = 0 el posicionamiento no es controlado-Valor de estándard = **1**
0x00080000-Activación de la protección de ALTO COUNTER ERROR. Si pusiera = 0 ella protección no actúa-Valor de estándard = **1**
0x00100000-Capacitación a la señal de UNIDAD DE DISCO VALE. Si pusiera = 0 no es controlado el
señal de Drver Ok-Valor de estándard = **1**
0x00800000-EJE MÁSTER DE UN SISTEMA DE ASES GANTRY
Si = 0 no es un Eje Gantry. Si =1 es un Eje Máster de un sistema Gantry, dónde, como descrito en el párrafo "NOTAS SOBRE LA GESTIÓN DE LOS ASES GANTRY", serán dadas todas las informaciones relativas a la caracterización y al funcionamiento GANTRY

- Valor de estándard = **0**

0x01000000-EJE MÁSTER CON ESLAVAS QUE TIENE QUE SEGUIR, si no va en Error.
Si = 0 el eje eslavas sigue como logra, o sea con sus parámetros de calibrado, pero no va en Error. Si = 1 El eje eslavas tiene que seguir estrechamente el Máster. En caso no logre, va en error
- Valor de estándard = **0**

0x02000000-EJE MÁSTER CON ESLAVAS QUE SEGUE IL (Cálculo) TEÓRICO
Si = 0 las eslavas sigue el Encoder del Máster - Valor de estándard = **0**

0x08000000-Capacitación PID COMPLEMENTARIO TAMBIÉN EN MOVIMIENTO
Si = 0 sólo a eje firme-Valor de estándard = **0**
0x10000000-SÓLO POR EJE STEPPER - BOOST ACTIVO EN ACELERACIÓN
Valor de estándard = **0**
Nota-*Usted calibrado de los Ases Stepper es tratado en adecuado capítulo*

Kind_board_TransducerPosition=40 N° de reconocimiento del tipo de fichas Ases utilizáis sobre los que reside el canal Encoder. El valor de estándard = **40** corresponde a las fichas PCI 2000 - no MODIFICAR

NOTA-Usted 3 caracterizaciones siguientes sólo son referidas a la ficha PCI 2000.

Board_number_TransducerPosition=0 Posición física del Back-plane en que se encuentra la ficha sobre que reside el canal Encoder de este Eje, que está caracterizando, El valor, incluido entre 0 y N, número máximo de fichas inseribili en la unidad Base selecta, es predispuesta en fase de prueba. NO MODIFIQUES

NOTA - En las Unidades Base MICROSYS 4120 la numeración de 0 a N parte de la ficha más lejana de la UCP, como enseñado en figura a lado.

Address_port_TransducerPosition=0x00000000 **NO UTILIZADO**

Physical_number_TransducerPosition=1÷6 Numero físico del canal Encoder de la ficha Ases asociados con el eje que está caracterizando. Se utiliza de costumbre el canal 1 por el primer Eje, el 2 por el segundo, etcétera pero eso no es estrechamente obligatorio, por cuyo es necesario efectuar esta caracterización.
NOTA - *Por estándard el N. del transductor Encoder es asociado con el correspondiente N. del eje en fase de prueba de la unidad de Gobierno*

Kind_board_DAC=40 Tipo de ficha Convertidor DAC usado
Valor pretarato = 40 por las fichas PCI 2000 - no MODIFICAR

NOTA-Usted 3 caracterizaciones siguientes sólo son referidas a la ficha cod. PCI 2000.

Board_number_DAC=0 Posición física del Back-plane en que se encuentra la ficha sobre que reside el Convertidor DAC de este Eje, que está caracterizando, El valor, incluido entre 0 y N, número máximo de fichas inseribili en la unidad Base selecta, es predispuesta en fase de prueba. NO MODIFIQUES
NOTA - *En las Unidades Base MICROSYS 4120 la numeración de 0 a N parte de la ficha más lejana de la UCP, como enseñado en la figura a lado.*

Address_port_DAC=0x00000000 **NO UTILIZADO**

Physical_number_DAC=1÷6 Numero físico del Convertidor DAC de la ficha Ases asociados con el eje que está caracterizando. Se utiliza de costumbre el canal 1 por el primer Eje, el 2 por el segundo, etcétera pero eso no es estrechamente obligatorio, por cuyo es necesario efectuar esta caracterización.
Por estándard el N. del Convertidor DAC es asociado con el correspondiente N. del eje en fase de prueba de la unidad de Gobierno
NOTA-Si de un Eje, DAC + ENCODER, se usa sólo el canal de medida, como por ejemplo para conectar una Octavilla, el **Phisical_number_DAC** tiene que ser puesto **= 0**

NOTA - de costumbre *el valor asignado a esta caracterización es el mismo le asignado al* **Physical_number_TransducerPosition=1÷6**

Number_SERVO_switch_port = "EnableX" Nombre lógico de la salida de la señal de UNIDAD DE DISCO ENABLE de este eje, en calibrado,

FalseTrue_logic_SERVO_switch=1 Nivel lógico de la señal de ENABLE , 1 o bien cero - si mucho de 1 o bien de cero el sistema no administra el Enable, Valor de estándard = **1**

Number_DRVOK_switch_port=”DriverOKX" Nombre lógico del input de la señal de DRIVER VALE de este eje

FalseTrue_logic_DRVOK_switch=1 Nivel lógico de la señal de Driver VALE , 1 o bien cero - si mucho de 1 o bien de cero el sistema no administra el Driver Ok, Valor de estándard = **1**

Level_TCH=0 Cada Eje tiene una entrada LATCH conectado con el correspondiente canal Encoder. La estrecha conexión del HW permite de conseguir strobature sincrónicos con la adquisición de la relativa posición del Encoder.
La entrada Latch es habilitado Level_TCH=1 poniendo e inhabilitado poniéndolo = **0.**
Cada Eje puede strobare valores de modo independiente de los otros.
NOTA - Queriendo al mismo tiempo strobare la posición de más Ases, (por ejemplo con la señal de un Tastatore, se tiene que llevar la señal eléctrica en paralelo sobre todos los Input Latch asociado con los ases correspondientes, dándolo mismo valor de caracterización a los Level_TCH de los ases interesados, el valor de asignar tiene que ser elegido entre 2 y 64, Valor de estándard = **0**

Revolution=360.0 Reposición de la medida de un eje rotativo a 360°.
Cuando el eje alcanza los 360°, recomenza a contar de 0 (cero). Para habilitar este funcionamiento hace falta

poner la máscara Kind_Axis_1 = 0x00000008. Si el bit de peso 8 es puesto = **0**, la cuenta no él resetta a 360° pero continua 361, 362 etcétera
Nota-Si se tiene que contar una división diferente de 360°, como por ejemplo las posiciones de un cambio útil formadas por 105 estaciones, se tararán la máscara = 8 pero en el valor de Revolution se escribirá 105, eso significa que después de la estación 104 se presentará las Cero,

7.2.3 VALORES ES COEFICIENTES POR SUMISIÓN DINÁMICOS ASES

KP=20 **Coeficiente de Proporcionalidad** entre el valor del error y el valor de la Velocidad. , Error = Valor de la posición artificiosa-, menos, Valor de la posición real del eje,
NOTA-*Una verificación significativa del bueno calibrado de un Eje es la partidaria:*
Con todos los otros coeficientes puestos = cero (KI) KD, KD2, MOLT_VFF, MOLT_AFF, moviendo el eje a velocidad constante, el error de persecución tiene que ser exactamente = VEL, mm/sec.) / KP
Valor de estándard = **20**

KI=0 **Coeficiente de Integración**, que tiende a reducir los derivas de posición del eje con respecto de la posición artificiosa.
Por ejemplo, a Eje firme, recobra la compensación del accionamiento, que puede variar por causas diferentes en el tiempo, ES buena norma tararlo después de haber borrado la compensación del eje por el parámetro **Volt_offset** y haber minimizado la compensación sobre el accionamiento.
Valor de estándard = **0**

VoltMaxIntegratedError=1 **Nivel de error máximo acumulado hasta** el que es habilitado la intervención del coeficiente KI. Por valores superiores se tiene en todo caso la intervención de KI limitada al valor programado. El valor es expresado en Voltio. Valor de estándard = **1**

KD=0 **Coeficiente Derivativo -** Sirve a reducir la variación del error de persecución.
En regulación de VELOCIDAD este parámetro tiene poca influencia. Valor de estándard = **0**

KD2=0 NO LE UTILIZADO EN REGOLAZ. DE VELOCIDAD

Valor de estándard = **0**

Molt_VFF=0

Porcentaje del valor de la señal de velocidad que es transmitida al accionamiento del eje. El valor de calibrado estará próximo a 100, y será alcanzado poniendo a punto todo el servosistema.
NOTA-*El valor de estándard del VFF es puesto = **0** para permitir de notar el curso del error de posición que, con VFF = 0, hace mover el eje.*

Molt_AFF=0

DE COSTUMBRE NO LE UTILIZADO EN REGOL:
DE
VELOCIDAD Valor de estándard = **0**

KS=0.0005

Coeficiente de Apagamiento a eje firme.
El valor le es expresado en UDM, en la unidad de medida dimensional elección con STEP por este Eje, y efectúa un redondeo sobre el valor de la señal de error, (redondeo sobre los valores artificiosos, evitando que se produce un error acumulativo debido al cálculo, y cortinas a apagar eventuales oscilaciones sobre la cuota final.
Este apagamiento es particularmente útil para la estabilización de Ases ligeros e/o sin roce como ases a rollos, idrosostentati o pneumosostentati. Calibrado típico = ½ del valor de la resolución, Valor UDM de 1 step Encoder,
Valor de estándard = **0.0005**

Step=-0.001

Este parámetro sirve a calcularlo desplazamiento efectivo de un eje, en UDM, por cada step del Encoder. El desplazamiento puede ser expresado directamente con el valor o bien con la fórmula de cálculo. La señal-, menos, permite de invertir el hacia de cuenta si resulta contrario al hacia del movimiento. La fórmula de insertar es:

, Paso vida / Relación de trasm.) / N° step Encoder

En el ejemplo 1 el resultado del cálculo es el partidario: , 10.16/5)/4000 = **0.000508** mm.
y es evidente que es mejor introducir la fórmula que no introduzcas directamente el valor, que no debe ser redondeado.

En el ejemplo 2, independientemente de la realización mecánica, igual al antes, el valor del Step es = **0.0005 (**aquel de la raya).
En este caso será más conveniente introducir directamente el valor del paso de la raya.

El valor de estándard = **-0.001 ha** sido elegido porque:

A menudo las rayas utilizadas tienen paso 0.001mm.
Se recuerda que se usa el punto y no la coma para separar los decimales
Se recuerda que el valor puede ser señalado, la falta de señal es entendida = señal +)

Volt_max =10

Este parámetro, tarabile de 0 a 10, define el valor máximo a cuyo borde la amplitud de la señal de referencia determinada al accionamiento del motor.
NOTA - *Durante la puesta a punto de un Eje es bien en todo caso limitar la velocidad punta alcanzable poniendo un límite inferior, reconduciéndolo a 10 por la puesta a punto final.*
Valor de default=10

Volt_offset=0.0

Calibrado del valor de la compensación para tener detención el eje en posición, compensando la compensación del accionamiento. El valor puede ser positivo o negativo. En este caso deberá ser hecho preceder de la señal-, menos.
Valor de estándard = **0.0**

KC_Parameter=3000/10*(20/1,

Coeficiente de Conversión, que define la velocidad del eje, cadena cinemática, eléctrica y de medida comprendió, aplicando al accionamiento una referencia de velocidad igual a **1Volt.**
Calcular como de ejemplo 1 precedente, relativo a STEP,:

Cómo Valor de estándard se ha preparado la fórmula **3000/10*(20/1,** porque es un valor normal con los siguientes parámetros:

Motor de **3000** rpm con nivel analógico a **10Volt**
Paso vida = **20mm.**
Relación de transmisión = **1 (**motor en eje sobre la vid)

Con la referencia de velocidad al valor máximo, 10V.), resulta una velocidad de rápido max = 60.000mm/min.
El valor, artificioso en el ejemplo, puede ser insertado como valor de KC pero se aconseja utilizar la fórmula predispuesta como estándard porque adaptando algunos valores durante la caracterización, no se ha forzado a rehacer todas las veces el cálculo e insertar el nuevo resultado

Vel_max=45000

Velocidad punta, (expresa en UDM / min.) de definir por el eje en calibrado.
NOTA - Usted velocidad punta no tiene que ser superior al valor de KC Parameter equis 10, 10V = valor máximo de referencia aplicable al accionamiento,
Usted conocidos que si es escrito un valor superior a aquel máximo, se acumula error de persecución porque el cálculo sobra la posición instantánea alcanzable del eje.

Acc_max=1000.0 Los valores de aceleración y desaceleración son expresadas en
Dec_max=1000.0 **UDM / sec2** y son definidos por el constructor de la máquina.
Valor de estándard = **1000.0, UDM** / sec2,
NOTA-*Poder definir valores diferentes por Aceleración y Desaceleración permite de poder bajar ligeramente la desaceleración cuando, obrando con valores particularmente altos, se hubiera en presencia de overshoot sobre la cuota final. De otra manera se perjudicaría el tiempo ciclo porque también se humillaría la aceleración*

Time_acc_nla=-1 Estos calibrados definen el **tipo de tramos de Aceleración**
Time_dec_nla=-1 **y de Desaceleración**, que pueden ser de tres tipos
Lineal-por tramos veloces
S-curva o trapezioidale-por tramos más dulces de aquellos lineales, conseguidas insertando un tiempo adicional, a los tiempos de aceleración e/o de desaceleración conseguido con tramos lineales debidos a la caracterización de **Acc_max** y **Dec_max,**
Sinusoidales-por tramos veloces pero la máquina tiene movimientos más dulces.

Valores de calibrado de Time_acc_nla / _dec_nla

Por Tramo Lineal asignar 0 (Fig.1)

Por Tramos S-CURVA (Fig.2)
Acceleraz. , Kind_Axis_1 con bit **80 =1,**
Deceleraz. , Kind_Axis_1 con bit **100 =1,**
asignar-1 por el tramo natural
, Aceleración max = a la programada,
asignar tiempo en sec. , 0.02 = 20msec.) ,
Aceleración max adaptado por el cálculo,

Por Tramos Sinusoidales (Fig.3)
Acceleraz. , Kind_Axis_1 con bit **80 =0,**
Deceleraz. , Kind_Axis_1 con bit **100 =0,**
asignar-1 por el tramo natural,
Aceleración max = a la programada,
asignar tiempo en sec. , 0.02 = 20msec.)
, Aceleración max adaptado por el cálculo,

NOTA - Para optimizar el ciclo es posible utilizar tipos diferentes

7.2.4 CONTROL Y ERRORES ASES

ServoError_standby=2.0

Calibrado del Sirvo error a eje firme. (UDM)
Sirve a proteger l ' eje de eventuales derivas o esfuerzos mecánicos que tiendan a desplazarlo de la posición bajo controlo en posición.
El valor, expreso en UDM (Unidad de Medida definida con STEP)., es modificable de programa con la instrucción **CAP/26:** Valor de estándard = **2.0**

ServoError_motion=15.0

Calibrado del Sirvo error a eje en movimiento
(diferencia instantánea entre la cuota artificiosa y aquella real).
El valor, de definir en fase de puesta a punto del eje, también es modificable de Part program con la instrucción **CAP/25:**
De costumbre el valor es orden de algunos mm.
Viene pero predispuesto un valor de estándard =15.0 **UDM.per** permitir de mover los ases sin ir en sirvo error aunque no son tarados todavía bien, protegiéndolos en todo caso de posibles fugas por errores de cableado u otro.

Position_error=0.5

Define la **tolerancia de posicionamiento** y es
le expresado en UDM, tamaños de programación = mm, grados, etc. El valor, estrechamente atado a las características de la máquina, de costumbre es elegido no muy estrecho, porque la bondad del posicionamiento no depende de este valor sino de aquel del servosistema y de la mecánica.
Y en todo caso modificable de Part program con la instrucción **CAP/27:**
En caso de que fuera solicitada una tolerancia muy estrecha, por ejemplo sobre un interasse de agujeros u otro, la tolerancia es modificable de programa.
Valor de estándard = **0.5** UDM.

Cycles_positioning=2.0 **Tiempo max, en sec.) de posicionamiento** dentro de ella tolerancia, a partir de la llegada en posición del cálculo. Vencido el tiempo (tiempo muerto), si el eje no entra en tolerancia, nace una señal de error. Valor de estándard = **2.0**

Cycles_wait=0.02 **Tiempo, en sec.) durante el que el eje debe quedar en la tolerancia** de posicionamiento. Sirve a no intercambiar el cruce de la posición final con el efectivo posicionamiento en tolerancia. Debe ser minimizado porque es un tiempo que siempre se suma a aquel de posicionamiento. Valor de estándard = **0.02**

VelStopCounter=1.0 Velocidad real del eje, expresa en UDM/min, bajo al que el sistema averigua la coherencia entre referencia determinada a los Accionamientos, expresado por el valor VoltStopCounter y la velocidad a que el eje está moviendo. Si es notada incoherencia, eje lento y mando alto, parte el tiempo "Delay alto counter" al final del que nace el relativo error.
Y modificable de Part program con la instrucción **CAP/31:** Valor de estándard = **1.0**

VoltStopCounter=0.5 Nivel de la Referencia de Velocidad, expreso en Voltio, superado el que el control "StopCounter." es efectuado. Permite es decir de intervenir antes de aumentar excesivamente la tensión al motor, con el eje aparentemente torpe a mover. Y modificable de Part program con la instrucción **CAP/30:** Valor de estándard = **0.5**

DelayStopCounter=0.2 Ventana de tiempo, expreso en sec. dentro del que el eje tiene que salir de las condiciones de Alto Counter definido por los 2 valores **VelStopCounter** y **VoltStopCounter** sobre descrito
Permite de controlar, por ejemplo, que no sea interrumpido el canal de cuenta Encoder, o bien si el eje, llegado en proximidad de la meta, no logra alcanzarla por duros garajistas, impedimentos o averías. Este control es habilitado por el valor de la máscara **Kind_Axis_1.** = 0x00080000
Transcurrido el tiempo, es dado el mando de ALTO COUNTER ERROR
El tiempo es modificable de Part program con la instrucción **CAP/29:** Valor de default=0.2

TimeOutDriverOk=1.0 Tiempo, en sec. dentro del que tiene que llegar el señal de DRIVER VALE si en uno de los controles cíclicos efectuado automáticamente por el sistema, la señal de Driver Ok es encontrado no activo.
El Temporizador es habilitado por el valor de la máscara Kind_Axis_1=0x00100000 Valor de estándard = **1.0**

NOTA-*Algunas UNIDADES DE DISCO no devuelven la señal de UNIDAD DE DISCO VALE. En este caso el valor diella disfraza será = 0*

7.2.5 AZZERAMENTO ASSI, FINE CORSA E BACKLASH

Number_OMO_switch_port = "HomeX" **Nombre lógico del input** de la señal de Homing de este eje

FalseTrue_logic_OMO_switch=1 **Nivel lógico de la señal de Homing**
, 1 o bien cero. Si mucho de 1 o bien de cero el sistema no administra el input,
Valor de estándard = **1**

Space_OMO=-10000 **Hacia de búsqueda de lo Micro de Homing**
También es caracterizada la distancia máxima practicable por el eje dentro del que lo Micro tiene que ser encontrado indudablemente. Sirve como protección en caso de sregolazioni mecánicos u otro.
El valor de la distancia le está en **UDM** eil hacia de la búsqueda es definida por la señal + / -
Valor de estándard = **-10000**

Space_OMO_Micro_OFF = Espacio máximo que el eje puede cumplir en sentir el liberación de lo Micro de cero.

Space_OMO_Marker = Espacio máximo que el eje puede cumplir en buscar la muesca del Marker.

Vel_OMO=1000 **Velocidad de búsqueda de lo Micro de Homing**
Le expresada en UDM/min.
Valor de estándard =1000

Vel_MARKER=500 **Velocidad de búsqueda del Marker (Index)**
Le expresada en UDM/min.
Valor de estándard =500

Offset_axis=-0.0 Representa la distancia física, expresa en UDM, entre ella posición dónde se quiere poner lo"Zero eje" y el Marker de cero del Encoder rebuscado sobre lo micro de cero. El valor puede ser positivo(senza señalo, o negativo, precedido por la señal -) según dónde se encuentra lo micro de cero.
Valor de estándard = -0.0

Positive_over_travel=0.0 Cuota de fina carrera positiva con respecto del Cero Eje , expresa en UDM. Valor de estándard =0.0

Negative_over_travel=0.0 Cuota de fina carrera rechazo con respecto de él Cero Eje, expresa en UDM. Valor de estándard =0.0
Si ambos los valores de Positivos y Negativos over travel son = 0 el control es inhabilitado, los Ases no tienen fina carrera,

Number_PositiveOverTravel_switch_port = " **Nombre lógico del input** de la señal de lo Micro de Fina Carrera positiva del eje de insertar entre las comillas.
, A veces, para poder mover la máquina en Manual antes de hacer los orígenes, son predispuestos los Micros de fina carrera a salvaguardia de errores de maniobra.

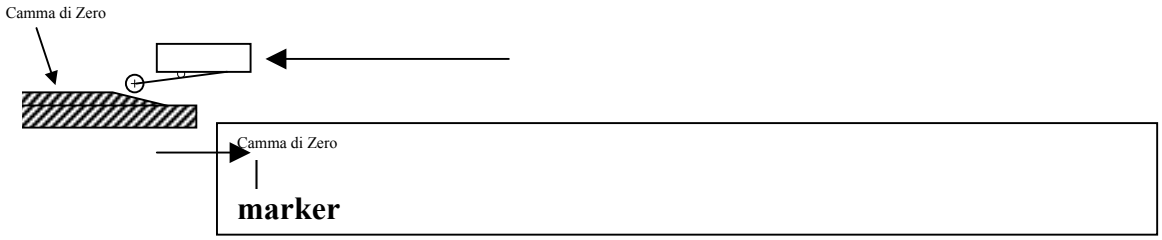
FalseTrue_logic_PositiveOverTravel_switch=2 **Nivel lógico de la señal de Fina Carrera Positiva.**
, 1 o bien cero. Si mucho de 1 o bien de cero el sistema no administra el input,
Valor de estándard = **2,** porque de costumbre no existen este Micros,

Number_NegativeOverTravel_switch_port = " **Nombre lógico del input** de la señal de lo Micro de Fina Carrera rechazo del eje de insertar entre las comillas.

FalseTrue_logic_NegativeOverTravel_switch=2 **Nivel lógico de la señal de Fina Carrera Rechazo.** Valor de estándard = **2,** porque de costumbre no existen este Micros,

Backlash=-0 Valor de compensación del juego mecánico, realzado experimentalmente y expreso en UDM.

El valor puede ser positivo o bien negativo, precedido por la señal menos. La señal tendrá que ser negativo si la dirección del eje en la fase de búsqueda del marker de cero es positiva y viceversa, como enseñado en figura siguiente, Valor de estándard **= -0**



7.2.6 GESTIÓN DEL FRENATURA ASES

Number_BRAKE_switch_port = "

Nombre lógico de la salida de la señal de BRAKE de mando del freno de este eje.

FalseTrue_logic_BRAKE_switch=2

Nivel lógico de la señal de BRAKE, 1 o bien cero - si mucho de 1 o bien de cero el sistema no administra el freno,
Valor de estándard = **2,** porque de costumbre los ases no son dotados con freno,

Time_BRAKE_Lock=1

Tiempo en sec. empleado por el freno para actuar. Por todo esto tiempo el eje queda bajo control. , caso típico un eje vertical que tiene que ser tenido en posición si no puede caer, Este tiempo sólo es calculado si el freno, BRAKE, es caracterizado.
Valor de estándard = **1(sec**.)

Time_BRAKE_Enable=1

Tiempo en sec., de antelación de la capacitación (Enable)
del eje antes de dar el mando de desbloqueo del freno. , garantía que el eje esté bajo control antes de desenfrenarlo, Este tiempo sólo es calculado si el freno, BRAKE, es caracterizado. Valor de estándard = **1(sec**.)

Time_BRAKE_Unlock=1

Tiempo en sec. empleado por el freno para desbloquearse. Por todo esto tiempo el eje queda bajo control en posición. Transcurrido este tiempo el control es habilitado a mover el eje.
Este tiempo sólo es calculado si el freno, BRAKE, es caracterizado. Valor de estándard = **1(sec**.)

7.2.7 ANTICOLLISIONE

El calibrado de los valores de anticollisione necesita algunas explicaciones y ejemplos iniciales.
Cada Eje que puede chocar mecánicamente con otros tiene que ser protegido en ambas las direcciones, derecha e izquierda o bien sobre y bajo etcétera,
Llamaremos genéricamente **Eje X** aquel en calibrado, **Axis 1** y **Axis 2** aquellos con que **el eje X no** tiene que chocar.
Para tarar el anticollisione del eje X, hace falta por tanto dar al Sistema las siguientes informaciones:
Axis_1_name_anticollision = "nombre dell 1 escrito entre las comillas
Distance_1_anticollision=0.0 distancia de seguridad de anticollisione hacia el eje 1
Kind_axis_1_anticollision=0 tipo (modalidad) de anticollisione que se quiere actuar
Axis_2_name_anticollision = "nombre dell 2 escrito entre las comillas
Distance_2_anticollision=0.0 distancia de seguridad de anticollisione hacia el eje 1
Kind_axis_2_anticollision=0 tipo (modalidad) de anticollisione que se quiere actuar

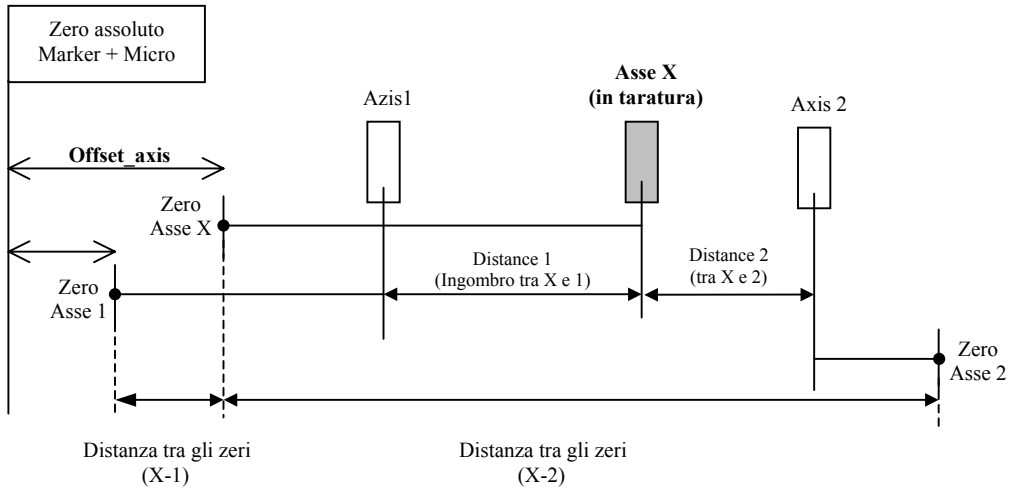
Tendrán que también ser rellenados los mismos campos para los Ases 1 y 2, que irán a su vez defendida por colisión con el eje X y con eventuales otros ases, con los que el eje X no puede chocar porque se encuentra entre las 1 y el 2.

El valor de anticollisione, expresa en UDM será igual a:

Distance = Distancia entre los ceros, del eje X y del otro Eje,-, menos, Estorbo entre los dos Ases

Nota-Si los Ases tienen el mismo origen, el valor de Distance es igual al valor del estorbo con señal-, menos,

La figura siguiente enseña las distancias que considerar



Quando si tarerà l'Asse 1 dovrà essere impostato lo stesso valore di Distance tra l'attuale Asse 1 y el eje X, y tan incluso por el eje 2.

Tipo de anticollisione

Kind_axis_1_anticollision=0 tipo (modalidad) de anticollisione que se quiere actuar

El valor de programar es dado por la suma decimal de los valores asignada a los tamaños **A., B, C, D, Y** como de esquema siguiente:

A Código de aproche entre los Ases. Puede asumir los siguientes valores:

Valores decimales del tamaño A

Eje X concorde con Eje 2 ,Asse X se acerca al eje 2 moviendo en positivo,
1

Eje X concorde con Eje 1, Eje X si acerca al eje 1 moviendo en negativo,

AEje X discorde de Eje 1, Eje X se acerca al eje 1 moviendo en positivo,
2
3
4)

AEje X discorde de Eje 1 Eje X se acerca al eje 1 moviendo en negativo

5

B Modalidad de acercamiento entre los Ases. Puede asumir los siguientes valores:
AsseX se acerca al eje 1 hasta a la posición de colisión (**Distance)**
AsseX no se acerca pero espera de tener el entero campo libre

C Modalidad de persecución entre los Ases. Puede asumir los siguientes valores:
AsseX sigue el eje 1 sin chocar, si 1 se aleja, X mueve siguiendo,
AsseX no sigue el eje 1 pero espera de tener el campo libre

D Modalidad de cálculo de "Distance." Puede asumir los siguientes valores:
la realización del anticollisione se basa en las posiciones reales (Movimentazione)

Y Cálculo del anticollisione después de OMO. Puede asumir los siguientes valores:
El cálculo del "**Distance" es** hecho después de cualquiera modalidades de puesta a cero ases **OMO, aunque** los ases no han sido borrados sobre el Marker del Encoder
El cálculo del "**Distance"** sólo es hecho después de la modalidad de puesta a cero ases
sobre el Marker del Encoder

Ejemplo de caracterización:

En el caso se quiera controlar el anticollisione con:

A	Ases concordes con acercamiento negativo	= **2**
B	haciendo acercar los Ases	= **8**
C	haciéndolos seguir	= **0**
D	"Distance" calculando sobre cuotas reales	= **0**
Y	sólo después de sólo haber hecho los ceros sobre lo Micro	= **64**

Valor decimal de asignarle a **Kind_Axis_collision = 74**

7.2.8 ASES ESLAVAS-GANTRY

EJE ESLAVAS - Máscara Kind_Axis_1=0x00800000 (bit a Cero)

Axis_name_slave = " Nombre del eje con funzionamento"Slave" con respecto de éste que está caracterizando

K_follower_slave=0.0 Coeficiente porcentual de persecución dell.

GANTRY - Máscara Kind_Axis_1=0x00800000 (bit a Uno)

Axis_name_slave = " Nombre del eje con funzionamento"Slave" con respecto de éste que está caracterizando

K_follower_slave=0.0 En funcionamiento Gantry asume el sentido de coeficiente de reparto de los errores de posición instantánea entre los dos ases Máster y eslavas.
Si = 0 no hay reparto. Si = 100, los errores son repartidos igualmente entre los ases.
, Si por ejemplo uno de los ases viene un po ralentizado por un duro garajista u otro, mientras se acelera lo para reajustarse, se entibia un po el otro hasta minimización de los errores. Eso garantiza un más buen mantenimiento instantáneo de la alineación de los ases.

VoltGantry=0.0 Diferencia en Voltio de las señales de referencia de Velocidad a los 2 Drives de un eje GANTRY no rígido, declarado en Kind_Axis_1 = 0x01800000, Alineación administrada por dos motores sincronizados y no de un sistema mecánicamente rígido,
Este valor garantiza que, para mantener los ases alineados, no se supera cierto esfuerzo. La diferencia entre las 2 señales de referencia no puede superar nunca el valor programado.
Valor de estándard = **0.0**

DeltaOffset_SlaveGantry=0.0

Distancia entre él cero (marker) del Máster con respecto de él cero (marker) de las eslavas de un sistema GANTRY no rígido. La distancia, expresa en UDM, sirve al sistema para tener cuento distancia entre los correspondientes Ceros para tener los ases alineados.
Valor de estándard = **0.0**

7.2.9 CALIBRADO MOTORES LINEALES

Antepongan

Para definir un Eje como "Lineal", hace falta settare la máscara:

Kind_axis_1 = al valor **0x00000200**-Eje con sistema de puesta en fase, Motor lineales,

con las reglas desplegadas en el párrafo CONCEPTOS PRELIMINARES SOBRE LAS MÁSCARAS

Cuando se utilizan motor lineales no existe obviamente ningunos

relación de transmisión entre motor y Eje.

Además la raya óptica de encuesta de la posición del eje es físicamente la misma utilizada por el accionamiento por la gestión de las fases del motor.

Pues el calibrado del **KC** ya no podrá eesere definido por la fórmula **KC_Parameter=3000/10*(20/1,** válida por motor rotatorios, pero del nuevo **KC_Parameter=velocità asse(1)/volt**

Si por ejemplo el eje tiene que mover a 45 mt/min. con 10Volt, el valor de calibrado será **KC_Parameter** = **4500 (**45.000mm./10)

(1) La velocidad siempre le es expresada en UDM/min. En la descripción se ha usado la expresión "45mt./min porque es el más usual, pero aquella correcta habría sido 45000UDM/min.

Puesta en fase Motor

A diferencia de los Ases a motor rotatorio, los ases con Motor lineal necesitan de costumbre una "Búsqueda" de la fase para localizar la posición relativa entre envolvimientos e imanes. Esta operación tiene que ocurrir antes de cualquier movimiento, comprendido la búsqueda de lo micro de Cero porque, no conociendo la fase sobre dicho, no es posible hacer mover el motor.

La búsqueda de la fase es una tarea del accionamiento, por el que los siguientes mandos han sido predispuestos sucesivamente descrito.

Number_PHASE_switch_port = "	Nombre de la salida utilizado para encaminar el proceso de "búsqueda" de la fase, por los accionamientos que lo solicitan,
FalseTrue_logic_PHASE_switch=2	Definición del estado lógico de la salida de que sobre (1 o bien 0) Valor de estándard = **2** porque no definido a priori.

Delay_PHASE=15 Tiempo máximo de espera que acabas el proceso de búsqueda de la fase. Valpre de estándard = **15,** sec.)

Existen los dos casos siguientes:
Accionamientos que devuelven la señal de DRIVER VALE al final del proceso de búsqueda fase. Si el proceso acaba antes y el accionamiento señala "fase ejecutada VALE", por el input de DRIVER VALE, el sistema toma enseguida bajo control el eje sin esperar el vencer tiempo. Si en cambio la señal de DRIVER VALE no llega dentro del tiempo, es señalado error.
Accionamientos que no devuelven la señal de DRIVER VALE. En este caso el sistema espera en todo caso el vencer tiempo y luego toma bajo control el eje. Si la fase no fue investigada dentro del tiempo, el eje se moverá mal y nacerá un error de movimentazione (collision) etcétera,

Volt_PHASE=0 **Atención-**Este calibrado sólo sirve para los Accionamientos que solicitan de definir el valor de la tensión que dar a los envolvimientos del motor durante la búsqueda de la fase.
De costumbre los accionamientos no solicitan este valor, que debe en estos casos ser tassativamente = cero. Valor de estándard = **0**

7.3 CALIBRADO EN CORRIENTE

7.3.1 PREDISPOSICIONES

Selección de la modalidad en Pareja, settando la Máscara al valor 0x00000001

Kind_axis_0=0x00000001 Máscara de los calibrados del **Tipo de Eje** con sumisión en Pareja

7.3.2 COEFICIENTES ES SENTIDOS POR EL CALIBRADO EN COR

KP=20 **Coeficiente de Proporcionalidad** entre el valor del error y el valor de la Corriente de mando.
Valor de estándard = **20**

KI=0 **Coeficiente de Integración**, que tiende a reducir los derivas de posición del eje con respecto de la posición artificiosa.
Por ejemplo, a Eje firme, recobra la compensación del accionamiento, que puede variar por causas diferentes en el tiempo, ES buena norma tararlo después de haber borrado la compensación del eje por el parámetro **Volt_offset** y haber minimizado la compensación sobre el accionamiento.
Valor de estándard = **0**

VoltMaxIntegratedError=1 **Nivel de error máximo acumulado hasta** el que es habilitado la intervención del coeficiente KI. Por valores superiores se tiene en todo caso la intervención de KI limitada al valor programado. El valor es expresado en Voltio. Valor de estándard = **1**

KD=0

Coeficiente Derivativo-E indispensable para tener bajo control el eje juntamente al KP. Típicamente tiene valor acerca de doble de aquel del KP.
Usted conocidos que más el accionamiento es veloz en la respuesta, más el valor de KD se acerca al de KP
Valor de estándard = **0,** porque se usa de costumbre el calibrado en Velocidad.
Acordarse de porlo al valor =40 primero dios iniciar los calibrados en Pareja

KD2=0

Minimiza el error debido a la Aceleración, teniéndola cuanto más posible par a aquel programada.
No usado en calibrado de Velocidad. Valor de estándard = **0**

Molt_VFF=0

Porcentaje del valor de la señal de Velocidad que es transmitida al accionamiento del eje. **El valor de calibrado estará próximo a cero** porque en este caso no se controla la Velocidad. Valor de estándard = **0**
NO MODIFIQUES

Molt_AFF=0

Porcentaje del valor de Aceleración que sirve a minimizar el error de persecución en fase de Aceleración y Desaceleración, tan como actúa normalmente la señal de Velocity Feed Forward en los calibrados de Velocidad,
De costumbre no utilizado en regulación de VELOCIDAD. Valor de estándard = **0**

KC_Parameter=3000/10*(20/1,	**Coeficiente de Conversión**, que define la aceleración del eje, cadena cinemática, eléctrica y de medida comprendió, expresa en UDM/sec2, conseguida aplicando al accionamiento una referencia de Corriente equivalente a un valor de tensión a la salida del DAC paras a **1Volt.** **ATENCIÓN-**El valor de estándard "**3000/10*(20/1,"** vale por calibrado del eje en Velocidad que de costumbre es el más utilizado. Tiene que pero ser borrada la fórmula e insertado el valor por calibrado en Pareva

7.4 CALIBRADO STEP-MOTORS

Number_DIRECTION_switch_port = "

Nombre lógico de la salida de la señal de **DIRECTION** de este eje

FalseTrue_logic_DIRECTION_switch=1

No MODIFIQUES EL VALOR Nivelo lógico de la señal **DIRECTION**
, Si = **1** el eje mueve en un verso, mientras si = **0** mueve en verso opuesto.
Valor de estándard = **1**

Number_STEP_switch_port = "

Nombre lógico de la salida de la señal **STEP** de este eje.

En seguida son enumerados los parámetros por modificar y que cambian sentido cuando un eje es tarado para pilotar Step Motors.

Kind_board_TransducerPosition=40

N° de reconocimiento del tipo de fichas Ases utilizáis sobre los que reside el canal Step Motor..
El valor de estándard = **40** corresponde a las fichas PCI 2000.
Existen los casos seguenti:Se el sistema preve el empleo de Step Motor dotados de Encoder por la lectura de los steps efectuados, funcionamiento a anillo cerrado, dejar el valor **40** y elegir el canal Encoder en el calibrado del parámetro:
Physical_number_TransducerPosition=1÷6
en cuánto uno de los canales se utiliza Encoder de los Ases Analógicos. El relativo DAC puede por tanto sólo ser usado como salido analógica por un Eje Mandril a anillo abierto u otro.

Si el sistema preve el empleo de Step Motor sin Encoder, a anillo abierto, modificar el valor en:

1002 si el canal Step Motor está sobre la Ficha 6 Ases Analógicos PCI 2000
1003 si está SOBRE LA Ficha 24+24 I/O

Board_number_TransducerPosition=0

Posición física del Back-plane en que se encuentra la ficha sobre que reside el canal de pilotaje Step Motor de este Eje, que está caracterizando, El valor, incluido entre 0 y N, número máximo de fichas inseribili en la unidad Base selecta, es predispuesta en fase de prueba. NO MODIFIQUES
NOTA - En las Unidades Base MICROSYS SERIE 4000 la numeración de 0 a N parte de la ficha más lejana de la UCP

Kind_board_DAC=40

Tipo de ficha con Salida PULSE usado.
Insertar el valor:
1002 si el canal Step Motor está sobre la Ficha 6 Ases Analógicos PCI 2000
1003 si está sobre la Ficha 24+24 I/O

Board_number_DAC=0

Posición física del Back-plane en que se encuentra la ficha sobre que reside la salida PULSE de este Eje, que está caracterizando, El valor, incluido entre 0 y N, número máximo de fichas inseribili en la unidad Base selecta, es predispuesta en fase de prueba. NO MODIFIQUES
NOTA - *En las Unidades Base MICROSYS de la serie 4000 la numeración de 0 a N parte de la ficha más lejana de la UCP, como enseñado en la figura a lado.*

Address_port_DAC=0x00000000

NO UTILIZADO

Physical_number_DAC=1÷3

Numero físico de la salida PULSE de la ficha sobre que reside el pilotaje del eje que está caracterizando.

¡ATENCIÓN! Si el eje Step Motor está **sin Encoder**, el número (1) 2, 3, selecto tendrá que ser el mismo indicado en el campo:
Physical_number_TransducerPosition=1÷3
Tendrán que también ser iguales entre ellos los números indicados en los campos:
Board_number_TransducerPosition
Board_number_DAC
Por consiguiente tendrán que también estar entre ellos pares los números, **1002** o bien **1003,** indicados en los campos:
Kind_board_TransducerPosition
Kind_board_DAC

7.4.1 CALIBRADOS POR SUMISIÓN DINÁMICOS ASES STEPPER

KP=20 **Coeficiente de Proporcionalidad** entre el valor del error
y el valor de la Velocidad. , Error = Valor de la posición artificiosa-, menos, Valor de la posición real del eje,
Para Ases con Motor Step, KP sólo actúa a **Eje firme.**
Valor de estándard = **20**

KD=0 **Frecuencia máxima de la faja baja, expresa en Hz,**
, y de Sumisión a Eje Firme.
Define la frecuencia de pilotaje del Motor Step sobre al que el sistema tiene que acelerar constantemente hasta la faja alta, cuya frecuencia mínima es expresada por el coeficiente **KD2.** La zona de frecuencias intermedias entre la faja baja y la faja alta es aquel en que el Motor tiene un agujero de pareja.. , Valores de sacar de la Fecha - Sheet del motor,
Si tarara =0 el motor se mueve sin control de faja.. Valor de estándard = **0**

KD2=0 **Frecuencia mínima de la faja alta,** Vedi **KD,**
Valor de estándard = **0**

KS=0.0005 **Coeficiente de Apagamiento a eje firme.**
El valor le es expresado en UDM, en la unidad de medida dimensional elección con STEP por este Eje, y evita que se oscilas sobre la cuota final por el fallido logro de un número entero de pasos.
El calibrado típico es un valor igual al desplazamiento físico del eje correspondiente a 1(uno, paso del motor.
Nota-En los casos de motorización a anillo cerrado, con Encoder, el valor de la resolución expresado por **KS puede** ser diferente de la resolución del Encoder.
Valor de estándard = **0.0005, de modificar como sobre dicho, antes de iniciar los calibrados dinámicos.**

KM=0.01 **Valor, en UDM, del desplazamiento debido a 1 paso del Motor.**
El calibrado típico es un valor igual al desplazamiento físico del eje.
Nota-de costumbre **KM** es igual a **KS**
Valor de estándard = **0.01, de modificar como sobre dicho, antes de iniciar los calibrados dinámicos.**

Volt_max=10 Este parámetro, por los Motor Step, define el valor de la frecuencia máxima, en Hz, de la faja superior al que el motor está capaz de funcionar. , el valor mínimo de la faja es expresado por **KD2,**
Valor de default=10 **de modificar segundo fecha sheet del motor.**

Volt_vel=10 **NO MODIFIQUES EL VALOR.** Si la dirección del Motor no es concorde a la del Encoder, invertir la señal

Step=-0.001 Con Ases Stepper dotado con Encoder, este parámetro sirve a calcular el desplazamiento efectivo de un eje, en UDM, por cada step del Encoder. El desplazamiento puede ser expresado directamente con el valor o bien con la fórmula de cálculo, como para los Ases analógicos. Con Ases sin Encoder, reconducir el valor asignado a **KM.**
La señal-, menos, permite de invertir el hacia de cuenta si resulta contrario al hacia del movimiento. Valor de estándard = **-0.001**

Vel_max=45000 **Velocidad punta**, (expresa en UDM / min.) de definir por el eje en calibrado.
NOTA - Usted velocidad punta, representada por la frecuencia máxima a cuyo el motor puede funcionar es definida por la fórmula
Vel_max=KM * Volt_max * 60.
Usted conocidos que si es escrito un valor superior a aquel máximo artificioso, el sistema procede en todo caso a la velocidad punta permitida. En realidad por tanto este calibrado es necesario si se quiere borde la velocidad, frecuencia, principio de funcionamiento.
Valor de estándard = **45000 de modificar con base en el valor típico del Motor**

Number_MARKER_switch_port = " **Nombre lógico del input** de la señal usada como Marker por los motores sin Encoder.
(Input de costumbre realizado por aspillera y fotocélula)

FalseTrue_logic_MARKER_switch=2 **Nivel lógico de la señal del Marker.**
, 1 o bien cero. Si mucho de 1 o bien de cero el sistema no administra el input,
Valor de estándard = **2**

FalseTrue_logic_STEP_switch=0

NO MODIFIQUES EL VALOR
Nivel lógico de la señal **STEP.**
Valor de estándard = **0**

Number_BOOST_switch_port = "

Nombre lógico de la salida de la señal de **BOOST** de este eje. Esta señal se activa automáticamente con dos muchas modalidades según el calibrado de la máscara **Kind_axis_1 =**

- **Si la máscara tiene el bit Kind_axis_1=0x10000000,** settato=1, la salida sólo conmuta en la fase de aceleración.
  Si Kind_axis_1=0x00000000, sitio =0, la salida conmuta al principio del movimiento y queda activo por toda la duración del desplazamiento.
  Estas señales son utilizadas por algunos tipos de accionamientos que levantan automáticamente la corriente en aceleración o bien por todo el movimiento y la bajan a motor firme.

FalseTrue_logic_BOOST_switch=2

Nivel lógico de la señal de **BOOST**
, 1 o bien cero. Si mucho de 1 o bien de cero el sistema no administra la salida,
Valor de estándard = **2**

7.5 TRAZA SOBRE COMO PROCEDER EN EL CALIBRADO ASES

En primer lugar hace falta controlar que el eje tenga las cuotas con la dirección y la unidad de medida, UDM, intencionales. Si el resultado no es corregido cambiar el valor de **Step** =, actuando sobre la señal por la dirección y sobre el valor por la unidad de medida. Este control no depende del tipo de sumisión usada y se puede hacer al aire libre con el loop de posición (eje sometido).
Antes de cerrar el loop de posición averiguar que **Volt_max no** tenga un valor alto, al objetivo de evitar desastrosas fugas del eje. Los valores cautelados os dependen del eje, pero se usan de costumbre valores del orden de 1 voltios.
Antes de cerrar el loop de posición averiguar que los parámetros relativos al PID, **KP KI KD KD2 AFF VFF,** sea todo a cero a exclusión del **KP** por las sumisiones en velocidad y de los **KP KD** por las sumisiones en pareja. Por la sumisión en velocidad se puede iniciar con un **KP** de 20; por aquellos en pareja con un **KP** de 10 y **KD** de 20.
Cerrando el loop de posición, después de haber controlado que las cuotas sean corregidas, se pueden tener los siguientes casos:

1, el eje queda sometido. Eso significa que **KC_Parameter** tiene la señal justa y su valor no y muy incorrecto.

2, el eje escapa y va en error de sumisión. Eso puede significar que **KC_Parameter** tiene la señal equivocada o que su valor y muy incorrecto. Tratar de corregir **KC_Parameter** eventualmente bajando los parámetros del PID, hasta a que el eje quede sometido.

notar que el eje pudiera parecer sometido si el error de posición queda a cero; por tanto y bien "turbar el eje" tratando de desplazarlo manualmente, para que actúe la realimentación del PID. Un auxilio se puede tener de "Calibrado Ases" en el entorno "Arregla", dónde es posible ver el mando del eje en calibrado, que tiene que quedar a un valor estable u oscilar reaccionando a las perturbaciones determinadas al eje.

Cuando el eje queda sometido, hace falta levantar **Volt_max** para poder dar mandos a la sumisión suficiente a hacerlo mover, aunque todavía limitados, ej. 5 voltios,
A este punto los procedimientos divergen según las sumisiones.

7.5.1 Sumisión en velocidad

Para este tipo de sumisión hace falta cerciorarse de haber tarado bien **KC_Parameter**, que define el valor de transformación entre los voltios del DAC y la correspondiente velocidad conseguidas por el eje. Este parámetro puede ser conseguido por valores conocidos o deducido. Si el eje en cuestión es un eje rotatorio un método es hacer ejecutar un Part-program, que utilizando el eje sin control, open loop, asigna una tensión a la sumisión y a premidera la velocidad. El Part-program es del tipo enseñado en seguida:

```
;               ASIGNA UNA TENSIÓN A. EL EJE A1
;               Y VISUALIZA DE ELLO LA VELOCIDAD'
;
                - TSK/VEL_A1
                - CPL/-1:A1
;               - SDO/nome freno,1      ; Instrucción de usarse con ases frenados
                - 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
```

Asignando un voltio, la velocidad define el valor de **KC_Parameter, aparte** la señal que depende de la dirección de rotación del motor con respecto de la dirección de cuenta del encoder,
El empleo de este Part-program y condicionado por el tipo de eje, que tiene que ser rotatoria o en todo caso con un largo desplazamiento. Además el eje no tiene que ser frenado, de otro modo a la abertura del loop de posición, el freno entra en función, y si no es desenfrenado, con **SDO/nome freno,1, se** amenaza de quemar el motor. Naturalmente el Part-program debe ser adaptado con el nombre del eje por tarar.

Otro método consiste en hacer ejecutar un Part-program que hace mover el eje a cierta velocidad, y con el solo parámetro KP utilizado, ve PID, es controlado el error de persecución que tiene que ser:

Error =, Velocidad eje / 60, / KP

```
;               CONTROLO ERROR DE PERSECUCIÓN DEL EJE A1
;
                - TSK/ERR_A1
                - SFP/3000,500,500      ; Feed, Acc, Dec
AA              - MOV/A1,500   ; Mueve eje A1 a 500
                - MOV/A1,0              ; Mueve eje A1 a 0
                - JMP/AA
                - END

PartProgram[ERR_A1]
AA              - RAV/14:L1,A1
                - DIS/1:ERRORE A1,L1
                - JMP/AA
                - RET
```

Naturalmente el Part-program debe ser adaptado con el nombre del eje por tarar y la velocidad de asignar al eje, velocidad que tiene que ser alcanzada y mantenida porque la prueba funciona para ases en velocidad constante.

Si el error se aleja mucho de aquel artificioso cambiar el **KC_Parameter** o actuar sobre la renta de la sumisión en caso de que sea posible. la realimentación del PID.

Un auxilio se puede tener de "Calibrado Ases" en el entorno "Arregla", dónde es posible ver el mando y el error de persecución del eje que está moviendo.

En el ejemplo anterior, considerando una velocidad de 3000 UDM/min. y un KP de 10 el valor de conseguirse es: , 3000/60)/10 = 5

7.5.2 Sumisión en pareja

También para este tipo de sumisión hace falta cerciorarse de haber tarado bien **KC_Parameter**, que define el valor de transformación entre los voltios del DAC y la correspondiente aceleración conseguidas por el eje. Este parámetro puede ser conseguido por valores conocidos o deducido. Además en estas sumisiones es necesario que estén presentes los parámetros
KP y **KD**, de otro modo el eje no se puede someter, oscila con el solo **KP.** Para deducir el valor, en módulo, de **KC_Parameter** se puede proceder en los siguientes modos:
Este método se puede usar con ases lineales. Y necesario hacer ejecutar un Part-program que asigna una tensión en Voltio al convertidor y medir con un dinamometro la fuerza en Kg que desarrolla el motor sobre el eje. Si se conoce la masa en Kg del eje el valor de asignar será **KC_Parameter** =, 10000 * Fuerza / Masa, / Voltio

El Part-program de usar es:

```
;           ASIGNA UNA TENSIÓN A. EL EJE A1
;
            - CPL/-1:A1
;           SDO/nome freno,1        ; Instrucción de usarse con ases frenados
            - LET/L1,0
AA          - KYB/VOLT A1,L1
            - CAP/17:A1,L1
            - JMP/AA
            - END
```

Naturalmente el Part-program debe ser adaptado con el nombre del eje por tarar y tienen que ser programados valores bajos de tensión (+-0.1) en crecer hasta a que el eje se mueva porque de otro modo el eje también adquiriría una alta velocidad con valores pequeños de tensión, es originada una aceleración y no una velocidad.
Otro método consiste en hacer ejecutar un Part-program que hace mover el eje con cierta aceleración.
Con el auxilio de "Calibrado Ases" en el entorno "Arregla", es posible ver el mando determinado con la aceleración programada.
KC_Parameter = (Aceleración) / Voltio mando

```
;           CONTROLO ERROR DE PERSECUCIÓN DEL EJE A1
;
            - SFP/3000,500,500      ; Feed, Acc, Dec
AA          - MOV/A1,500   ; Mueve eje A1 a 500
            - MOV/A1,0              ; Mueve eje A1 a 0
            - JMP/AA

            -END
```

7.5.3 Consideraciones finales

Cuando el **KC_Parameter es** arreglado hace falta también retocar los otros valores del PID acordándose que valores altos devuelven el eje inestable, pero con valores bajos los errores de posición son más altos. Por tanto hace falta buscar valores bastante altos, pero no tanto de hacer inestable el eje.
Con las sumisiones en velocidad e indispensable **KP**, importante **VFF** y migliorativo **KI**
Con las sumisiones en pareja son indispensables **KP** y **KD**, importantes **KI** y **AFF.**
La relación **KD / KP puede** variar de 1 a 5; generalmente y alrededor de 2. Un método para buscar los valores que programar es hacer eseguir el Part-program siguiente, variando con "editor" en el entorno "automación" los valores de testare.La bondad del movimiento y el tiempo empleado dan un idea correcto empleo de los parámetros

```
;       * * * * * * * * * * POSICIONAMIENTOS EJE A1 * * * * * * * * * * *
;
                - CAP/0:A1,32768        ; Position Error habilitado
                - CAP/27:A1,0.008       ; Valor 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
;
;               L1 = POSICIÓN DE SALIDA
;               L2 = INCREMENTO DE POSICIÓN
;               L3 = NÚMERO MEDIDAS
;
                - MOV/A1,L1
                - KYB/INIZIO IDA
                - 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 POR POSICIÓN =, L1/L10

                -END
```

CAPÍTULO 8

Visión

El paquete VisAlgo trabaja por solicitudes de funciones de análisis de la imagen que efectúan operaciones de:

- Filtro y manipulación del imagina
- Análisis luz y extracciones automáticas de umbrales
- Blobs sobre rectángulos de interés
- Extracciones de contornos sobre intersección de figuras sobre rectángulos de interés
- Extracciones de bordos
- Medición de contornos
- Extracciones de entes sobre los contornos
- Medición sobre los entes

Para activar las funciones de visión con telecámaras reales es necesario configurar el expediente "**Vision.ini"**, con los siguientes parámetros.

[ParametriVisione]

NumeroTelecamere=0
Progressiva=0
SchedaDoppia=0

APÉNDICE A

Enumero Errores

Error	1	Función todavía activa
Error	2	Nombre asse/mandrino/IO no correcto
Error	3	OVERFLOW Tabelal nombres ases
Error	4	OVERFLOW Tablero ases
Error	5	Tipo eje incongruente
Error	6	FALTA EJE FÍSICO POR SISTEMA VIRTUAL
Error	7	EJE HELP NO UTILIZABLE (CYL_TRASLANTE)
Error	8	OVERFLOW Tablero JOB de movimentazione
Error	9	JOB de movimentazione DEFINIDO
Error	10	OVERFLOW ÁREA movimentazione
Error	11	PARÁMETROS Start continúo ERRADOS
Error	16	PARÁMETROS Movimentazione ERRADOS
Error	17	DEMASIADOS TPE CONTEMPORÁNEAS
Error	18	EJE YA' USADO EN OTROS JOBS
Error	19	EJE FUERA DE CAMPO DE TRABAJO
Error	20	RELUZCO FINAL <> RAYO INICIAL
Error	21	JOB NO DEFINIBLE
Error	22	DEMASIADOS ASES, MAX 8,
Error	23	NO HAY MÁS' TEMPORIZADORES
Error	24	ASES EN MOVIMIENTO
Error	25	ESTADO INCORRECTO
Error	26	ERROR DE SISTEMA EN EL EMPLEO DEL JOB
Error	27	EJE FRENADO
Error	28	CLOSE U OPEN POSITION LOOP INÚTIL
Error	29	FALTA ÁREA ESLAVAS
Error	30	MUESTREO NO ACTIVA
Error	31	DEMASIADOS PARÁMETROS
Error	32	FALTA SUMISIÓN ACTIVA
Error	33	DEMASIADAS LÁMINAS I/O CONFIGURADAS
Error	34	LÁMINA I/O EQUIVOCADO
Error	35	BIT NO SOBRE LÁMINA
Error	36	OVERFLOW DEL BIT LÓGICO
Error	37	BIT I/O EQUIVOCADO
Error	38	DEMASIADAS LÁMINAS ANALÓGICAS CONFIGURÁIS
Error	39	LÁMINA ANALÓGICA EQUIVOCADA

Error	40	CANAL ANALÓGICO NO SOBRE LÁMINA
Error	41	OVERFLOW DEL CANAL ANALÓGICO, LÓGICO,
Error	42	CANAL ANALÓGICO EQUIVOCADO
Error	43	SELECTA DESCONOCIDA
Error	44	TIEMPO CADUCADO (Time OUT)
Error	45	INTERPOLACIÓN SPLINE ERRADO
Error	47	PUESTA A CERO EJE HECHO, MANCA HOMING,
Error	48	OVERFLOW TABLE SPINDLE
Error	49	MANDRIL EN ACTIVIDAD' SAMPLE
Error	50	MANDRIL EN ROTACIÓN
Error	51	SUMISIÓN INHABILITADA
Error	52	COTIZA CORRECCIÓN ERRADA, IGUAL,
Error	53	OVERFLOW ÁREA CORRECCIÓN
Error	54	BUFFER CIRCULAR NO ACTIVO
Error	55	OVERFLOW COUNTER EN EL EJE STEPPER
Error	56	COUNTER NO DEFINIDO EN EL EJE STEPPER
Error	58	LE ANIMO EN DEADLOCK
Error	59	TASK EN ABORT (TKM)
Error	60	OVERFLOW TKM
Error	61	EJE SOBRE MICRO DE MÁS ALLÁ DE CARRERA POSITIVA
Error	62	EJE SOBRE MICRO DE MÁS ALLÁ DE CARRERA NEGATIVA
Error	63	FASE ACABADA
Error	1001	HANDLE INCORRECTO
Error	1002	DRIVER OCUPADO
Error	1003	INICIALIZACIÓN ERRADA
Error	1004	ANOMALÍA EN FASE DE PUESTA A CERO
Error	1006	DEMASIADOS HANDLER DE MOVIMENTAZIONE ACTIVA

Error Driver Fault	0x5	Configuration error - Decision Table
	0x7	Calibration error
	0x10E	Fiber not connected to digital driver

0x111

Digital driver assh't hardware enable

0x113	Fiber rx Frame error
0x114	Fiber rx Parity error
0x116	Cero command refused for movement
0x117	Lost cero axis

APÉNDICE B

Ejemplo “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=}