



Albatros

3.2.3

Control numérico



Tecnologie e Prodotti per l'Automazione

Esta documentación es propiedad de TPA Srl y cualquier uso no autorizado está prohibido. TPA Srl se reserva el derecho de modificar el contenido en cualquier momento.

Índice

1	Introducción	1
1.1	Cómo utilizar este manual	1
1.2	Las ventanas de trabajo	1
2	Composición del sistema	2
2.1	Derechos de acceso al sistema	2
2.2	Soporte multilingüe	2
2.3	Arquitectura típica de un sistema	3
2.4	Organización y configuración lógica	4
2.5	Dispositivos	5
3	Cuadro Sinóptico	7
3.1	Uso del Cuadro Sinóptico	7
3.2	Cómo intervenir en el Cuadro Sinóptico	7
3.3	Cómo intervenir en los Dispositivos	7
3.4	Movimiento Manual de los Ejes	7
4	Parámetros Tecnológicos y Herramientas	9
4.1	La ventana Parámetros Tecnológicos	9
4.2	La ventana Parámetros Herramientas	10
5	Diagnóstico	12
5.1	La ventana de Diagnóstico	12
5.2	Composición de la ventana de Diagnóstico	12
5.3	Representación de los Dispositivos	12
5.4	Interactuar con los Dispositivos	13
5.5	Lista de teclas para navegar dentro de una estructura de árbol	14
5.6	Correctores de linealidad	14
5.7	Panel de calibrado de los ejes	14
6	Errores y Advertencias	18
6.1	Introducción	18
6.2	Errores de sistema	19
6.2.1	Errores ocasionados por la gestión de los ejes	19
	1 NombreEje: vínculo codificador incorrecto	19
	2 NombreEje: movimiento no finalizado	19
	3 NombreEje: servoerror	19
	4 NombreEje: superior al límite positivo	19
	5 NombreEje: superior al límite negativo	20

	10 NombreEje: La ejecución en tiempo real es más rápida que la construcción del perfil	20
6.2.2	Errores ocasionados por la gestión de E/S remotas	20
	2049 Receptor número: configuración incorrecta	20
	2050 Receptor número: desconectado	20
	2051 Receptor número: revinculado	20
	2052 Receptor número: error en relectura salida no conectada número NúmeroSalida	20
	2054 Receptor número: tipo incorrecto	20
	2055 Receptor número: inicializado	21
	2056 Receptor número: error alimentación +24 Vcc	21
	2057 Error alimentación GreenBUS	21
	2058 Receptor número: error en relectura TipoDispositivo NombreDispositivo	21
	2059 Fracasada la prueba de la dual port memory del transmisor	21
	2060 Error en la inicialización del transmisor	21
	2061 Error en la transmisión del firmware al transmisor	22
	2062 Error en la transmisión de la configuración al transmisor	22
	2063 Error en la transmisión de la configuración al receptor	22
	2064 Receptor número: versión firmware no compatible	22
	2065 Receptor número: Error en una comunicación asíncrona	22
	2066 Receptor número: Error genérico	22
	2067 Receptor número: Error durante la transmisión de la configuración	22
	2068 Receptor número: Error interno n. NúmeroError	23
	2069 Receptor número: Error alimentación +24 Vcc banco número	23
6.2.3	Errores ocasionados por la gestión MECHATROLINK-II	23
	2308 Tarjeta NúmeroTarjeta: Inicialización fracasada por valor incorrecto de un parámetro de configuración	23
	2341 Tarjeta NúmeroTarjeta: El número de servo-accionamientos excede el máximo admitido	23
	2342 Tarjeta NúmeroTarjeta: La dirección hardware de servo-accionamiento Servo supera el valor máximo permitido	23
	2349 Tarjeta NúmeroTarjeta: El servo-accionamiento Servo no está conectado	24
6.2.4	Errores ocasionados por la gestión CanBUS	24
	2761 Nodo número: desconectado	24
	2762 Nodo número: revinculado	24
	2763 Error de falta de transmisión	24
	2764 Nodo número: Error de falta de recibir	24
	2765 Nodo número: Inicializado	24
	2766 Condición de falla en la interfaz CAN	25
	2767 Error pérdida estado CANopen	25
	2768 Nodo número: Error de falta recepción PDO	25
	2769 Nodo número: Error de recepción de un nodo no configurado	25
	2770 Nodo número: Configuración incorrecta	25
	2771 Nodo número: Error durante la comunicación SDO	25
	2772 Tiempo de expiración en el ciclo CAN de interrogación de los nodos	25
	3073 Nodo número: Error de Emergencia n. NúmeroError	25
	3074 Nodo número: Error CAN genérico n. NúmeroError	26
	3088 Tarjeta CAN número: Nodo NúmeroNodo: Error durante la comunicación SDO n. NúmeroError – descripción	26
6.2.5	Errores ocasionados por la gestión bus EtherCAT	26
	3329 Error durante la inicialización del socket de comunicación	26
	3330 Error durante la análisis de la red EtherCAT	26
	3331 Error en la configuración del buzón de correo de transmisión	26
	3332 Error en la configuración del buzón de correo de recepción	26
	3333 Tarjeta EtherCAT número: Error en el tipo de expansiones del nodo NúmeroNodo	27
	3334 Error al configurar los PDOs	27
	3335 Nodo NúmeroNodo en alarma (NúmeroError)	27

	3336 Tarjeta EtherCAT: El número de expansiones del nodo NúmeroNodo es erróneo	28
	3337 Tarjeta EtherCAT: Nodo NúmeroNodo: Desconectado	28
	3338 Tarjeta EtherCAT: Nodo NúmeroNodo: Revinculado	28
	3340 Tarjeta EtherCAT: El nodo NúmeroNodo no respondió a la solicitud (Código)	28
	3341 Tarjeta EtherCAT: El nodo NúmeroNodo no existe	29
	3342 Cable desconectado	29
	3343 Tarjeta EtherCAT número: Nodo NúmeroNodo no pasa al estado SAFE-OPERATIONAL (Código)	29
	3344 Tarjeta EtherCAT número: Nodo NúmeroNodo no pasa al estado OPERATIONAL (Código)	29
	3345 Tarjeta EtherCAT: Comunicación inestable	29
	4400 Demasiados ejes activos en FASTREAD (Función:NombreFunción línea:NúmeroLínea)	29
6.2.6	Errores ocasionados por la inicialización	29
	769 Configuración software incorrecta	29
	770 Número de IRQ configurado incorrecto	30
	772 Error en la lectura del área de memoria intermedia durante la inicialización	30
	773 Alcanzado el máximo número de ejes en configuración	30
	774 El tiempo real de los ejes no está en ejecución	30
	775 Tiempo insuficiente para la ejecución GPL	30
	776 Tiempo excesivo de ejecución del tiempo real	30
	777 Guardián expirado	30
	778 El código de Main del firmware está bloqueado	31
	1025 Tarjeta NúmeroTarjeta: No responde al comando	31
	1026 Tarjeta NúmeroTarjeta: Error al enviar el firmware a la tarjeta de ejes	31
	1028 Tarjeta NúmeroTarjeta: Firmware no presente	31
	1029 Tarjeta NúmeroTarjeta: Main bloqueado	31
	1031 Tarjeta NúmeroTarjeta: Error en inicialización	31
	1032 Tarjeta NúmeroTarjeta: La prueba de la dual port memory ha fracasado	31
	1033 Tarjeta NúmeroTarjeta: El código de arranque del firmware no está en ejecución	32
	1035 Tarjeta NúmeroTarjeta: No presente	32
	1037 Tarjeta NúmeroTarjeta: La apertura de la dual port memory ha fracasado	32
	1039 Tarjeta NúmeroTarjeta: Guardián expirado	32
	1040 Tarjeta NúmeroTarjeta: Error de alimentación +24 Vcc	32
	1047 Tarjeta NúmeroTarjeta: Configuración de software no permitida	32
	1052 Tarjeta NúmeroTarjeta: Se está ejecutando el código de arranque	32
	1053 Tarjeta NúmeroTarjeta: Guardián ejes expirado	32
	1055 Guardián expirado en la tarjeta NúmeroTarjeta	33
	1056 Tarjeta NúmeroTarjeta: Error de alimentación interfaz CAN	33
	1057 Tarjeta NúmeroTarjeta: Error interno n. NúmeroError	33
6.2.7	Errores ocasionados por la gestión de la memoria	33
	1281 Error en la asignación de memoria en el área de heap	33
	1286 Error al gestionar el heap	33
	1287 Demasiadas desasignaciones de memoria por el heap	33
	1289 Error en la creación de las variables globales	33
	1290 Error en la dimensión de las variables no volátiles	33
	1291 Error en la dimensión de las variables de sola lectura	34
6.2.8	Errores ocasionados por fallos	34
	1559 Seguimiento de punto de interrupción	34
	1569 Código operativo del microprocesador no válido	34
	1586 Valor INTEGER dividido por cero	34
	1600 Desbordamiento en el resultado de una operación de coma flotante	34
	1601 Subdesbordamiento en el resultado de una operación de coma flotante	34
	1602 Argumento no válido para una operación de coma flotante	34
	1603 Valor en coma flotante dividido por cero	35

1604	Resultado incorrecto de una operación de coma flotante	35
1605	Se ha utilizado un valor en coma flotante incorrecto	35
1728	Se ha intentado acceder a una dirección no válida	35
1735	Excepción genérica	35
1736	Datos no alineados	35
1801	Alarma temperatura	35
1802	Alarma ventilador	35
1803	La frecuencia de la CPU es inestable	36
6.2.9	Errores ocasionados por funciones GPL	36
4097	El dispositivo TipoDispositivo NombreDispositivo no está configurado	36
4098	La variable global NombreVariable no existe	36
4099	Función NombreFunción no encontrada	36
4101	Gestión incoherente del eje NombreEje	36
4105	Instrucción no ejecutable en el eje NombreEje	36
4106	El remoto referido al eje paso a paso NombreEje no está vinculado	36
4107	Instrucción SYSOK con argumentos incorrectos	37
4108	El eje NombreEje: Cota final más allá de los límites software	37
4110	Velocidad incorrecta	37
4111	Aceleración eje NombreEje negativa	37
4112	Deceleración eje NombreEje negativa	37
4114	Eje NombreEje: Restablecimiento a entrada rápida no realizado	37
4115	Eje NombreEje: marca de cero no encontrada	37
4353	Código operativo instrucción desconocida (Función:NombreFunción línea:NúmeroLínea)	38
4354	Operación matemática incorrecta (Función:NombreFunción línea:NúmeroLínea)	38
4355	Dirección incorrecta de matriz o vector (Función:NombreFunción línea:NúmeroLínea)	38
4356	Instrucción RET no invocada de CALL (Función:NombreFunción línea:NúmeroLínea)	38
4357	Variable local inexistente (Función:NombreFunción línea:NúmeroLínea)	38
4358	Etiqueta de salto inexistente (Función:NombreFunción línea:NúmeroLínea)	38
4359	Argumento macro incorrecto (Función:NombreFunción línea:NúmeroLínea)	39
4360	Error en asignación de memoria durante la ejecución (Función:NombreFunción línea:NúmeroLínea)	39
4361	Demasiadas tareas activas (Función:NombreFunción línea:NúmeroLínea)	39
4362	Formato matriz incorrecto (Función:NombreFunción línea:NúmeroLínea)	39
4363	Demasiadas instrucciones ONINPUT activas (Función:NombreFunción línea:NúmeroLínea)	39
4364	Eje ya ocupado en una referencia local (Función:NombreFunción línea:NúmeroLínea)	40
4365	Instrucción ONINPUT activada en la misma entrada (Función:NombreFunción línea:NúmeroLínea)	40
4366	Demasiadas instrucciones ONFLAG activas (Función:NombreFunción línea:NúmeroLínea)	40
4367	Instrucción ONFLAG activada en el mismo indicador (Función:NombreFunción línea:NúmeroLínea)	40
4368	Intento de escritura de una variable de tipo SolaLectura (Función:NombreFunción línea:NúmeroLínea)	40
4369	Demasiados ejes maestros activos (Función:NombreFunción línea:NúmeroLínea)	40
4370	Demasiados ejes esclavos activos (Función:NombreFunción línea:NúmeroLínea)	41
4372	Uso erróneo de una instrucción (Función:NombreFunción línea:NúmeroLínea)	41
4373	Imposible ejecutar la lectura de la velocidad de alimentación (Función:NombreFunción línea:NúmeroLínea)	41
4374	Demasiadas instrucciones de tipo IPC en ejecución (Función:NombreFunción línea:NúmeroLínea)	41
4375	FASTREAD ejecutada sobre ejes de tarjetas diferentes (Función:NombreFunción línea:NúmeroLínea)	41

4378 Instrucción no habilitada (Función:NombreFunción línea:NúmeroLínea)	42
4379 Esta instrucción no se puede utilizar en funciones lanzadas por Interrupción (Función:NombreFunción línea:NúmeroLínea)	42
4380 Demasiados solicitudes de escritura en el área de memoria intermedia (Función:NombreFunción línea:NúmeroLínea)	42
4381 No se puede utilizar una línea serial no abierta (Función:NombreFunción línea:NúmeroLínea)	42
4382 No se puede abrir una línea serial ya abierta (Función:NombreFunción línea:NúmeroLínea)	42
4383 Se intentó abrir demasiados procesos auxiliares (Función:NombreFunción línea:NúmeroLínea)	42
4384 El proceso auxiliar no está en ejecución (Función:NombreFunción línea:NúmeroLínea)	42
4385 Se intentó abrir un proceso auxiliar de otra tarea (Función:NombreFunción línea:NúmeroLínea)	43
4391 Error durante la activación del SYSOK (Función:NombreFunción línea:NúmeroLínea)	43
4394 Demasiados errores de ciclo (Función:NombreFunción línea:NúmeroLínea)	43
4395 Demasiados mensajes (Función:NombreFunción línea:NúmeroLínea)	43
4397 Desbordamiento de pila (Función:NombreFunción línea:NúmeroLínea)	43
4398 Subdesbordamiento de pila (Función:NombreFunción línea:NúmeroLínea)	43
4399 Parámetro fuera de intervalo (Función:NombreFunción línea:NúmeroLínea)	43
4865 Falta la definición de la máquina para la interpolación (G216 o G217)	43
4866 Falta la definición de los índices de la configuración máquina seleccionada (M6)	44
6.2.10 Errores ocasionados por el controlador de las comunicaciones CNCTPA	44
16385 Módulo desconectado	44
16386 Módulo vinculado	44
16387 Módulo revinculado	44
16388 Módulo inicializado	44
16389 El módulo ha interrumpido la conexión	44
16641 El firmware del control no responde a los comandos	45
16642 TpaSock no responde a los comandos	45
16643 El sistema operativo no permite utilizar RTX	45
16645 Error al enviar el código firmware	45
16646 No ha sido posible reiniciar el código firmware	45
16897 RTX no está instalado	45
16898 El usuario no tiene derechos de administrador	45
16899 Dimensión incorrecta de la RAM del módulo	46
16900 La dirección IP del módulo es incorrecta	46
16901 El módulo ya está vinculado a otra instalación	46
16902 El módulo no está configurado	46
16903 Las configuraciones del firewall impiden la comunicación	46
16904 Tarjeta de red no presente o deshabilitada	46
16905 Falta el código firmware del control	46
16906 Versión RTX incompatible con el código de firmware del control	47
16907 Versión del sistema operativo no compatible con el código firmware del control	47
17153 TipoTarjeta: Falta el código firmware del transmisor GreenBUS	47
17154 TipoTarjeta: Falta la porción de código de firmware del transmisor GreenBUS	47
17155 TipoTarjeta: Error en el envío del código de arranque del transmisor GreenBUS	47
17156 TipoTarjeta: Error en el envío del código de Main del transmisor GreenBUS	47
17157 TipoTarjeta: Falta el código de arranque	47
17158 TipoTarjeta: Falta el código de Main	48
17159 TipoTarjeta: Error en el envío del código de arranque	48
17160 TipoTarjeta: Error en el envío del código de Main	48
17409 No se ha conseguido enviar el ejecutable auxiliar	48

17410	No se ha conseguido poner en ejecución al ejecutable auxiliar	48
17667	NombreDLL: No se ha conseguido ejecutar el código firmware	48
17668	NombreDLL: No ha sido posible conseguir el puntero a la RAM compartida	48
17921	No se ha conseguido enviar NODETPA	49
17922	NODETPA no se ha reiniciado	49
17923	NODETPA no está en ejecución	49
18177	NODETPA trató de acceder a una dirección no válida	49
6.3	Advertencias genéricas	49
6.3.1	Albatros empieza la ejecución	49
6.3.2	Albatros termina la ejecución	49
6.3.3	La computadora entra en modalidad de suspensión	49
6.3.4	La computadora sale de la modalidad de suspensión	49
6.3.5	Apagado del ordenador	50
6.3.6	Nivel de acceso actual	50
6.3.7	Actualización software de los módulos	50
6.3.8	Enviar la configuración a los módulos	50
7	Configuración del Sistema	51
7.1	Introducción	51
7.2	Configuración de los Dispositivos	51
7.2.1	Introducción	51
7.2.2	Dispositivo genérico	51
7.2.3	Salida digital	52
7.2.4	Entrada analógica	52
7.2.5	Eje	52
	Datos de Base	52
	Parámetros de Movimiento	52
	Parámetros de Interpolación	53
	Otros Parámetros	53
	Parámetros de Referencia	53
	Niveles de Acceso	54
	Concatenación Ejes	54
	Correctores de Linealidad	54
7.3	Configuración Lógica	54
7.3.1	Configuración de la instalación	54
7.3.2	Configuración de los grupos	55
7.4	Configuración Física	57
7.4.1	Configuración de sistema	57
7.4.2	Configuración Hardware	57
	Configuraciones predeterminadas	58
	Configurar un nodo de un bus TPA	58
	Configurar un nodo de un bus CAN	60
	La tarjeta de control del bus	60
	El nodo CAN	60
	Insertar un nodo nuevo	60
	Configurar un nodo	61
	Características de la gestión EtherCAT en Albatros	61
	Premisa	61
	Configuración del hardware EtherCAT	61
	Descripción de un PDO	62
	Modificar un PDO de un accionamiento	62
	PDO adicionales	63
	Adquisición automática nodos EtherCAT	64
7.4.3	Configuración Virtual-Físico	64

7.4.4	Mapas de cableo	65
7.5	Lista de teclas para navegar dentro de una estructura de árbol	66
8	Instrumentos de desarrollo	67
8.1	Editor GPL	67
8.1.1	Funciones del editor GPL	67
	Uso de las expresiones regulares	69
8.1.2	Insertar mensaje	70
8.1.3	Cifrado	70
8.1.4	Lista de teclas a selección rápida disponibles	71
8.2	Librerías	73
8.3	Depuración	74
8.3.1	El depurador	74
8.3.2	Tareas en ejecución	74
8.3.3	Todas las tareas	75
8.3.4	Llamadas a función	75
8.3.5	Puntos de interrupción	75
8.3.6	Contenido de variable	76
8.3.7	Lista de teclas a selección rápida disponibles	77
8.4	Inicialización del control	77
8.4.1	Conexiones de red	77
8.4.2	Diagnóstico hardware	77
	Topología red EtherCAT	78
	Visualización y modificación de objetos en los nodos	78
8.5	Test	79
8.5.1	Guardar variable global	79
8.5.2	Ejecutar función	79
8.5.3	Importar mensajes	79
8.5.4	Nota para el usuario en el archivo de informe de las alarmas	81
8.6	Herramientas	81
8.6.1	Personalizar...	81
8.7	Browser	83
8.7.1	El browser	83
8.7.2	Buscar identificador...	83
8.7.3	Lista de teclas a selección rápida disponibles	84
9	Programas accesorios	85
9.1	XConfMerge: programa para la combinación de archivos de configuración	85
9.2	XParMerge: programa para la combinación de dos archivos de parámetros	86
10	Lenguaje GPL	87
10.1	Conceptos de base	87
10.1.1	Introducción al lenguaje GPL	87
10.1.2	Convenciones y terminología	87
10.1.3	Las variables	89
	Tipos de datos	89
	Conversión de datos	91
	Declaración y visibilidad de las variables	91
	Modificadores	92

	Asignación de RANGE	93
	Derechos de Lectura/Escritura	93
10.1.4	Las Constantes	93
	Constantes predefinidas con valor preconfigurado	94
	Constantes predefinidas con valor preconfigurado al iniciar Albatros	95
10.1.5	Palabras clave	95
10.1.6	Las funciones	96
10.1.7	Los parámetros de tipo dispositivo	98
10.1.8	El Multitasking	99
10.1.9	Las Comunicaciones	100
10.1.10	Variables que usar en la programación	101
10.1.11	Los Ejes	101
10.1.12	Correctores de linealidad	103
10.1.13	Gestión de los mensajes en lengua	104
10.1.14	Gestión de los errores de sistema	104
10.2	Funciones especiales	104
10.2.1	Personalización movimiento ejes	104
10.2.2	Funciones estándar de desplazamiento y calibrado	107
10.2.3	Función OnUIEnd#	109
10.2.4	Función OnUIPlugged#	110
10.2.5	Función OnUIUnplugged#	110
10.3	Instrucciones	110
10.3.1	Convenciones	110
10.3.2	Tipologías de las instrucciones del lenguaje GPL	110
10.3.3	Entradas/Salidas	116
	GETFEED	116
	INPANALOG	116
	INPFLAGPORT	117
	INPPORT	117
	MULTIINPPORT	117
	MULTIOUTPORT	117
	MULTIRESETFLAG	118
	MULTIRESETOUT	118
	MULTISETFLAG	118
	MULTISETOUT	118
	MULTIWAITFLAG	118
	MULTIWAITINPUT	119
	OUTANALOG	119
	OUTFLAGPORT	120
	OUTPORT	120
	RESETFLAG	120
	RESETOUT	120
	SETFLAG	120
	SETOUT	121
	WAITFLAG	121
	WAITINPUT	121
	WAITPERSISTINPUT	122
10.3.4	Ejes	122
	CHAIN	122
	CIRCABS	123
	CIRCINC	124
	CIRCLE	124
	COORDIN	125
	DISABLECORRECTION	126
	EMERGENCYSTOP	126
	ENABLECORRECTION	127

ENDMOV	127
FASTREAD	128
FREE	128
HELICABS	128
HELICINC	129
JERKCONTROL	130
JERKSMOOTH	130
LINEARABS	130
LINEARINC	131
MOVABS	131
MOVINC	132
MULTIABS	133
MULTIINC	134
NORMAL	134
RESRIFLOC	134
SETINDEXINTERP	135
SETLABELINTERP	135
SETPFLY	135
SETPFLYCHAINSTRAT	136
SETPZERO	136
SETPZEROCHAINSTRAT	136
SETQUOTE	137
SETQUOTECHAINSTRAT	137
SETRIFLOC	137
SETTOLERANCE	137
START	139
STARTINTERP	139
STOP	139
SWITCHENC	140
WAITACC	140
WAITCOLL	140
WAITDEC	141
WAITREG	141
WAITSTILL	142
WAITTARGET	142
WAITWIN	142
Parámetros Ejes	142
Lectura\Escritura	142
DEVICEID	142
GETAXIS	143
Movimiento punto a punto	148
SETACC	148
SETDEC	148
SETDERIV	149
SETFEED	149
SETFEEDF	149
SETFEEDFA	149
SETINTEG	149
SETMULTIFEED	150
SETPROP	150
SETSLOPE	150
SETVEL	150
Movimiento interpolado	151
LOOKAHEAD	151
SETACCI	151
SETACCLIMIT	151
SETACCSTRATEGY	152

	SETAXPARTYPE	152
	SETCONTORNATURE	152
	SETDECI	153
	SETDERIVI	153
	SETFEEDFAI	153
	SETFEEDI	153
	SETFEEDFI	154
	SETINTEGI	154
	SETPROPI	154
	SETSLOPEI	154
	SETSLOWPARAM	155
	SETVELI	155
	SETVELILIMIT	155
	Movimiento coordinado	156
	SETFEEDCOORD	156
	SETOFFSET	157
	Movimiento concatenado	157
	RATIO	157
	SETDYNRATIO	158
	Parámetros genéricos	158
	DYNLIMIT	158
	ENABLESTARTCONTROL	159
	NOTCHFILTER	159
	RESLIMNEG	159
	RESLIMPOS	160
	SETADJUST	160
	SETBACKLASH	160
	SETBIGWINFACTOR	162
	SETDEADBAND	162
	SETENCLIMIT	163
	SETINDEXEN	163
	SETINTEGTIME	163
	SETIRMPP	163
	SETLIMNEG	164
	SETLIMPOS	164
	SETMAXER	164
	SETMAXERNEG	164
	SETMAXERPOS	165
	SETMAXERTYPE	165
	SETPHASESINV	166
	SETREFINV	167
	SETRESOLUTION	167
10.3.5	Contadores	167
	DECOUNTER	167
	INCOUNTER	167
	SETCOUNTER	168
10.3.6	Temporizadores	168
	HOLDTIMER	168
	SETTIMER	168
	STARTTIMER	168
10.3.7	Variables, Vectores y Matrices	169
	CLEAR	169
	FIND	169
	FINDB	169
	LASTELEM	169
	LOCAL	170
	MOVEMAT	170

	PARAM	171
	SETVAL	171
	SORT	171
10.3.8	Cadenas	172
	ADDSTRING	172
	CONTROLCHAR	172
	LEFT	172
	LEN	173
	MID	173
	RIGHT	173
	SEARCH	173
	SETSTRING	174
	STR	174
	VAL	174
10.3.9	Comunicaciones	174
	CLEARRECEIVE	174
	COMCLEARRXBUFFER	175
	COMCLOSE	175
	COMGETERROR	175
	COMGETRXCOUNT	175
	COMOPEN	176
	COMREAD	176
	COMREADSTRING	176
	COMWRITE	177
	COMWRITESTRING	177
	RECEIVE	177
	SEND	182
	SENDIPC	188
	WAITIPC	188
	WAITRECEIVE	189
10.3.10	Matemática	189
	ABS	189
	ADD	189
	AND	190
	ARCCOS	190
	ARCSIN	190
	ARCTAN	190
	COS	191
	DIV	191
	EXP	191
	EXPR	192
	LOG	193
	LOGDEC	193
	MOD	193
	MUL	194
	NOT	194
	OR	194
	RANDOM	195
	RESETBIT	195
	ROUND	195
	SETBIT	196
	SHIFTL	196
	SHIFTR	197
	SIN	199
	SQR	199
	SUB	200
	TAN	200

	TRUNC	200
	XOR	201
10.3.11	Multitasking	201
	ENDMAIL	201
	ENDREALTIMETASK	201
	ENDTASK	201
	GETPRIORITYLEVEL	202
	GETREALTIME	202
	GETREALTIMECOUNT	202
	HOLDTASK	202
	RESUMETASK	202
	SENDMAIL	203
	SETPRIORITYLEVEL	203
	STARTREALTIMETASK	203
	STARTTASK	204
	STOPTASK	204
	WAITMAIL	204
	WAITTASK	205
10.3.12	Gestión de Flujo	205
	CALL	205
	DELONFLAG	205
	DELONINPUT	205
	FCALL	205
	FOR/NEXT	206
	FRET	206
	GOTO	207
	IF/IFVALUE/IF-THEN-ELSE	207
	IFACC	208
	IFAND	208
	IFBIT	209
	IFBLACKBOX	209
	IFCHANGELEVEL	210
	IFCOUNTER	210
	IFDEC	211
	IFDIR	211
	IFERRAN	211
	IFERROR	212
	IFFLAG	213
	IFINPUT	213
	IFMESSAGE	214
	IFOR	214
	IFOUTPUT	215
	IFQUOTER	215
	IFQUOTET	216
	IFRECEIVED	216
	IFREG	217
	IFSAME	217
	IFSTILL	217
	IFSTR	217
	IFTARGET	218
	IFTASKHOLD	218
	IFTASKRUN	219
	IFTIMER	219
	IFVEL	219
	IFWIN	220
	IFXOR	220
	ONERRSYS	221

	ONFLAG	221
	ONINPUT	222
	REPEAT/ENDREP	222
	RET	223
	SELECT	223
	TESTIPC	224
	TESTMAIL	224
10.3.13	Varias	225
	CLEARERRORS	225
	CLEARMESSAGES	225
	DEFMSG	225
	DELAY	226
	DELERROR	226
	DELMESSAGE	227
	ERROR	227
	IFDEF/ELSEDEF/ENDDEF	229
	MESSAGE	231
	SYSFAULT	232
	SYSOK	232
	TYPEOF	233
	WATCHDOG	233
10.3.14	MECHATROLINK-II	233
	MECCOMMAND	233
	MECGETPARAM	234
	MECGETSTATUS	235
	MECSETPARAM	236
10.3.15	Bus de campo estándar	237
	AXCONTROL	237
	AXSTATUS	238
	CNBYDEVICE	239
	READDICTIONARY	240
	WRITEDICTIONARY	240
10.3.16	EtherCAT	240
	ACTIVATEMODE	240
	ECATGETREGISTER	241
	ECATSETREGISTER	241
	GETPDO	241
	SETEOE	241
	SETPDO	242
10.3.17	Tarjetas TMSbus con gestión CAN	242
	GETCNSTATE	242
	GETSDOERROR	242
	GETMNSTATE	242
	RECEIVEPDO	243
	SENDPDO	243
	SETNMTSTATE	243
10.3.18	Simulación	243
	DISABLE	243
	DISABLEFORCEDINPUT	244
	ENABLE	244
	ENABLEFORCEDINPUT	244
	RESETFORCEDINPUT	244
	SETFORCEDANALOG	245
	SETFORCEDINPUT	245
	SETFORCEDPORT	245
10.3.19	BlackBox	245
	ENDBLACKBOX	246

	PAUSEBLACKBOX	246
	STARTBLACKBOX	246
10.3.20	ISO	247
	ISOG0	247
	ISOG1	247
	ISOG9	248
	ISOG90	248
	ISOG91	248
	ISOG93	248
	ISOG94	249
	ISOG216	249
	ISOG217	249
	ISOM2	250
	ISOM6	250
	ISOSETPARAM	251
	KINEMATICEXPR	252
10.3.21	Instrucciones no más disponibles	254
10.3.22	Instrucciones no utilizables en evento	254
10.4	Ejemplos	256
10.4.1	Puesta a cero en Interrupt	256
10.4.2	Servidor de desplazamiento ejes	257
10.4.3	Ciclo de Main con gestión errores	258
10.4.4	Operaciones con las cadenas	259
10.4.5	Ejecución Secuencial/Paralela	260
10.4.6	Rutina de Puesta a cero	260
10.4.7	Movimientos ISO	261

1 Introducción

1.1 Cómo utilizar este manual

Esta guía describe las funciones del control numérico Albatros.

La guía se ha estructurado para guiar al operador en la comprensión del sistema y de su uso.

Los puntos principales de cada sección son:

- las ventanas y las herramientas de Albatros.
- la descripción de la Arquitectura de un sistema Albatros.
- cómo visualizar los dispositivos e intervenir en los mismos con funciones manuales y de diagnóstico, utilizando el Cuadro Sinóptico.
- cómo visualizar y modificar los Parámetros Tecnológicos y Geométricos y los Parámetros Herramientas.
- cómo visualizar los dispositivos e intervenir en los mismos con funciones manuales y de diagnóstico.

Para evitar que el presente manual resulte demasiado extenso, se recomienda al lector que haga referencia a los manuales del Sistema Operativo Windows, para que pueda aprender perfectamente los mecanismos de uso del ratón, los menús, las barras de herramientas y todas las funciones operativas clásicas de Windows.

1.2 Las ventanas de trabajo

Las ventanas de trabajo son de vario tipo, dependiendo del tipo de operación que efectuar, y pueden mantenerse abiertas contemporáneamente.

Las ventanas están divididas en los siguientes tipos:

Ventana	Descripción
<i>Principal</i>	ventana principal de Albatros. Permite invocar funciones y es el contenedor de las otras ventanas, cuyo contenido depende de la aplicación específica que las mismas representan.
<u>Sinóptico</u>	contiene una representación gráfica de la máquina o de piezas de la misma, permitiendo intervenir en las mismas.
<u>Parámetros Tecnológicos</u>	permite visualizar y modificar los parámetros tecnológicos y de máquina.
<u>Parámetros Herramientas</u>	permite visualizar y modificar los parámetros.
<u>Diagnóstico</u>	permite visualizar el estado de los dispositivos y, allá donde está permitido, también intervenir en los mismos.
<u>Errores de sistema</u>	ventana con la lista de los últimos errores de sistema que se verificaron. También es posible visualizar los errores de ciclo y los mensajes.
<u>Configuración del Sistema</u>	consiente visualizar y modificar los dispositivos lógicos y físicos de la máquina.


2 Composición del sistema

2.1 Derechos de acceso al sistema

Albatros prevé cuatro niveles de acceso al sistema:

- **Usuario:** es el nivel con las mayores restricciones de acceso. No es posible modificar ninguna de las definiciones relativas a los dispositivos. Este nivel se utiliza para ejecutar las mecanizaciones y las operaciones comunes de la máquina. Cuando se inicia el sistema, se activa automáticamente este nivel de acceso.
- **Asistencia:** es el nivel utilizado para efectuar el mantenimiento ordinario de la máquina. El operador debería poder modificar algunos parámetros de configuración, pero no aportar modificaciones a la estructura de la máquina.
- **Constructor:** es el nivel utilizado para configurar las instalaciones y las máquinas. A este nivel se pueden aportar casi todas las modificaciones posibles. Lo utiliza la persona encargada del desarrollo.
- **Tpa:** es el máximo nivel de acceso al sistema. Sirve para proteger el acceso a algunas definiciones sumamente delicadas, cuya modificación requiere un profundo conocimiento de Albatros. Su uso es muy raro. La contraseña de acceso a este nivel debe ser requerida directamente a TPA.

Para acceder al sistema con un nivel de acceso superior al de Usuario o para llevar el sistema a nivel de Usuario después de haber aportado modificaciones con un nivel superior, se debe ingresar la correspondiente contraseña.

Para llamar la ventana donde se ingresa la contraseña se debe utilizar la combinación de teclas **Ctrl + *** (asterisco). En otro caso, a la derecha de la **Barra de tareas** de Windows se encuentra el icono . Haciendo clic con el botón derecho del ratón sobre el icono es posible visualizar un menú en el que aparece la opción **Change pass level**.

La ventana que se abre se presenta de esta manera:



Ventana de acceso

A este punto se debe ingresar la contraseña y se debe presionar el botón **[Aceptar]** para confirmar. En lugar de los caracteres que componen la contraseña se visualizan unos caracteres "*", de modo que nadie pueda leer la contraseña tecleada.


Una vez ingresada la contraseña, se entra directamente al correspondiente nivel de acceso. Para tener una confirmación del nivel de acceso se puede seleccionar la opción **Acerca de Albatros** desde menú **Ayuda**.

Cuando la contraseña insertada no está correcta, se visualiza el error "¡Atención! ¡Contraseña Incorrecta!".

2.2 Soporte multilingüe

Albatros soporta la visualización del texto en distintos idiomas.

Cambio de idioma

El cambio de idioma puede ser efectuado desde cualquier [nivel de acceso](#) al sistema. Para cambiar la selección del idioma hay que utilizar la combinación de teclas **Ctrl + /** o hacer clic sobre el icono  desde la "**barra de tareas**" de Windows.

En la ventana que se abre seleccionar la lengua deseada y presionar el botón **[OK]**.

El cambio de idioma no se ejecuta inmediatamente, sino al siguiente reinicio de Albatros.

2.3 Arquitectura típica de un sistema

Ya que muchos aspectos, en las representaciones gráficas y en la estructura de datos básicos de la Máquina, dependen altamente de la tipología de la Máquina, además de alguna información general, la presente Guía describe la composición de un sistema típico a modo de ejemplo.

Las indicaciones reales, los esquemas y las páginas gráficas del sistema verdadero dependen evidentemente de la aplicación específica, y, por lo tanto, las proporciona el propio Constructor de la Máquina Herramienta.

El control numérico Albatros está constituido por un PC supervisor, que expone la interfaz del Operador-Máquina y da un número variable de Módulos (de 1 a 16), para la conducción y el control de todos los recursos operativos de la Máquina Herramienta o del Sistema.

Las versiones pueden ser dos:

Monomódulo constituida por un único módulo conectado directamente al bus del PC.

Multimódulo compuesta por un mínimo de 1 a un máximo de 16 módulos y prevista normalmente para aplicaciones en Instalaciones o Líneas de máquinas; la unidad PC en este caso está físicamente separada de los Módulos que pueden estar situados en lugares diferentes de la Línea o de la Instalación.

En ambas arquitecturas, los módulos están constituidos por una o varias tarjetas de ejes para el control directo de los Ejes de Máquina y la gestión lógica de los dispositivos de Entrada/Salida.

En la versión monomódulo las tarjetas de ejes están instaladas directamente en el PC supervisor, en la versión multimódulo están instaladas en una PC industrial (Clipper) conectada al PC Supervisor a través de una red Ethernet. En la figura que sigue se reproduce un esquema de conexión entre el PC Supervisor y un módulo remoto (Clipper). Asimismo, figuran las principales actividades desempeñadas por los varios componentes.

SISTEMA DE CONEXIÓN DE UN MÓDULO REMOTO



PC Supervisor
con Microsoft Windows

Interfaz Albatros

Red Ethernet



Control Numérico
con Windows Embedded
Windows CE o RTX

Comunicaciones GPL
Interpolaciones



Tarjeta de control
del BUS de campo

Estimulo E/S
Control Ejes

Dispositivos remotos inteligentes pilotan dispositivos de E/S y ejes (remoto TRS-AX) directamente en la máquina. Dichos dispositivos leen las líneas de Entrada Digitales (ON/OFF) o Analógicas, regeneran las líneas de Salidas Digitales o Analógicas y están conectados a los Módulos mediante el Greenbus (bus en serie RS485 - 1 Mbaud), el bus CAN y Ethercat.

El funcionamiento de Albatros en máquina está protegido por la presencia de una clave hardware USB, configurada por TPA.

2.4 Organización y configuración lógica

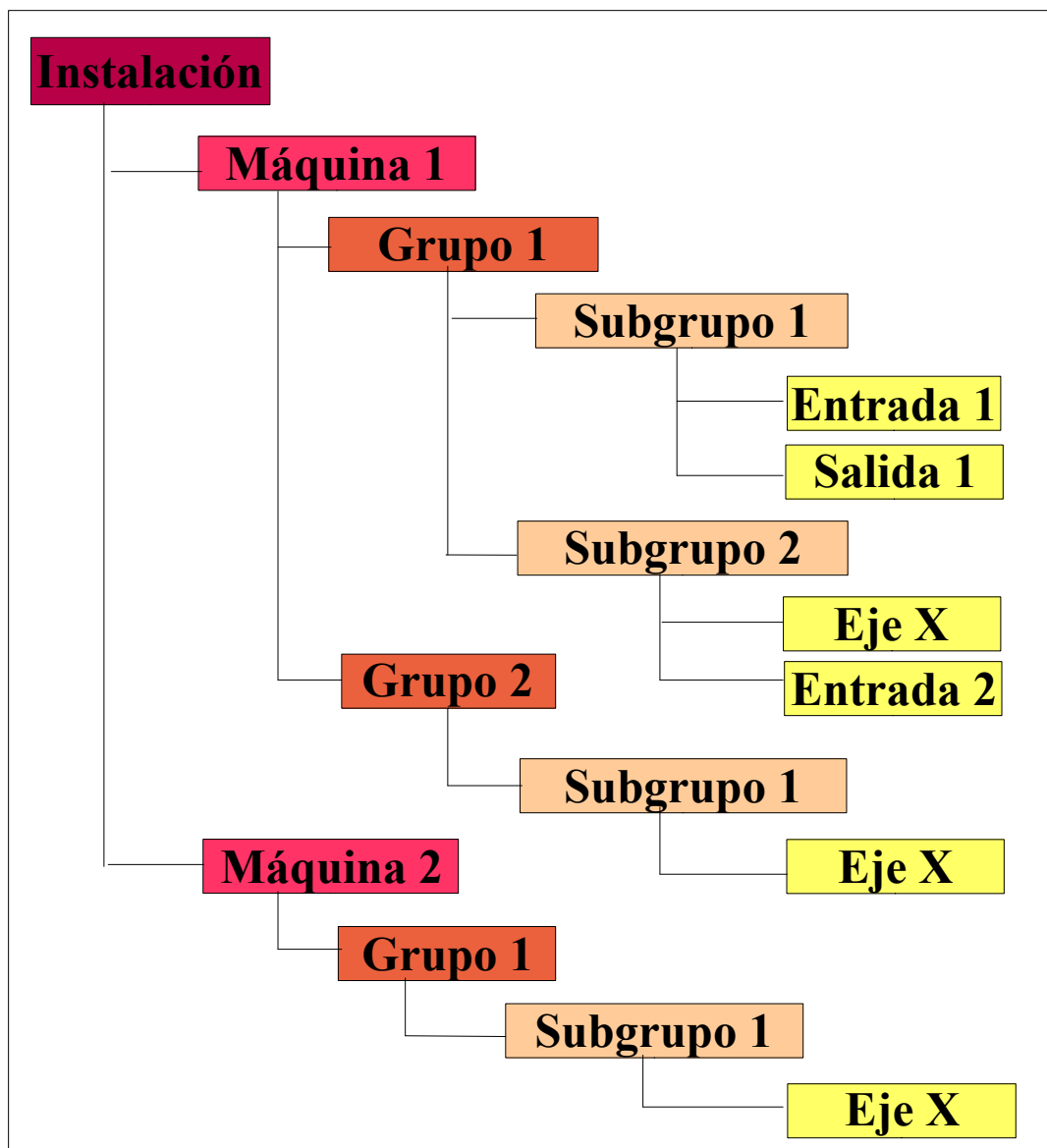
En el sistema Albatros la estructura descriptiva de la instalación o de la máquina herramienta está organizada en un archivo tecnológico con estructura jerárquica.

Este planteamiento se debe a la necesidad de conservar, a nivel de datos de configuración y de modalidad de acceso, la eventual estructura modular de las máquinas, clasificándola en términos de agregación dinámica de módulos diferentes, agregados y dispositivos, que se pueden activar o desactivar según las diferentes configuraciones posibles.

Conforme a esta estructura lógica, en el caso más general y complejo, el nivel jerárquico superior se compone de:

- | | |
|------------------------|--|
| 1. Instalación | simplemente es un conjunto de máquinas. Es decir, el conjunto de los elementos operativos gestionados por el Control numérico. La instalación está siempre presente, incluso en el caso de una única máquina, y no debe ser definida explícitamente. |
| 2. Máquina | se observa desde un punto de vista "lógico" como un conjunto de dispositivos (ejes, temporizadores, etc.) y de ciclos de control, es decir de un código escrito en GPL que pone en acto los algoritmos de control de la máquina misma. Generalmente en una máquina hay un elevado número de dispositivos que se organizan en grupos. |
| 3. Grupos | son "contenedores" que permiten organizar los componentes de una máquina según criterios lógicos. Por ejemplo, se puede definir un grupo "ejes" que contiene a todos los ejes de la máquina, los interruptores de fin de carrera, las cíclicas que efectúan la puesta a cero de los ejes, etc. |
| 4. Subgrupos | representan una especialización adicional de un grupo. Por ejemplo, el grupo "ejes" podría ser subdividido en "ejes-digitales" y "ejes-paso-a-paso". |
| 5. Dispositivos | constituyen el nivel más bajo de la jerarquía. Son una representación lógica de los componentes eléctricos y mecánicos de la máquina y son independientes del hardware subyacente. |

En la figura abajo se esquematiza la estructura de una supuesta instalación compuesta por dos máquinas:



Ejemplo de organización jerárquica de una instalación.

ATENCIÓN: Se puede dar el caso que los Grupos no estén subdivididos en Subgrupos, y estén formados directamente por Dispositivos.

Para acceder a algunas funciones como Diagnóstico, Configuración, Parámetros Tecnológicos en el caso de una instalación con varias máquinas es necesario seleccionar la máquina de la cual se desean visualizar los datos.








2.5 Dispositivos

Los dispositivos se pueden agrupar en dos categorías: dispositivos físicos y dispositivos lógicos. En el sistema, todos los dispositivos se distinguen por un nombre que identifica su uso.

Dispositivos físicos













Por dispositivos físicos se entienden todos los elementos que actúan sobre las partes eléctricas o neumáticas de la máquina o que detectan el estado; estos son:

Símbolo	Dispositivos	Función
---------	--------------	---------

	<i>Entrada Digital</i>	detecta el estado "Encendido" o "Apagado" de un dispositivo. Por ejemplo, el interruptor de seguridad de una puerta.
	<i>Salida Digital</i>	activa o desactiva un dispositivo, situándolo en el estado de "Encendido" o "Apagado". Por ejemplo, se puede utilizar para gobernar una electroválvula.
	<i>Entrada Analógica</i>	detecta el valor de una tensión de entrada en el borne correspondiente. Por ejemplo, el valor generado por una dínamo taquimétrica.
	<i>Salida Analógica</i>	impone una tensión de salida en el borne correspondiente. Por ejemplo, se puede utilizar para gobernar un inversor.
	<i>Puerto de Entrada</i>	se compone de 8 líneas de entrada digitales.
	<i>Puerto de Salida</i>	se compone de 8 líneas de salida digitales.
	<i>Eje</i>	gestiona el movimiento de un eje eléctrico. Es posible administrar ejes de distinto tipo: con control analógico, con control digital, motores paso a paso, ejes de cómputo (solo lectura codificador).

Dispositivos lógicos

Los dispositivos lógicos son elementos que intervienen exclusivamente dentro de los programas de trabajo y no están asociados a una parte física:

Símbolo	Dispositivos	Función
	<i>Temporizador</i>	elemento para contar el tiempo. La unidad de medida es el segundo. Tiene una resolución de 4 ms. Puede asumir solamente valores positivos y el valor máximo representable es de 8.589.934 segundos (aprox. 99 días) (con tiempo real de 250 Hz). Su valor se almacena en la memoria no volátil de la tarjeta de ejes.
	<i>Contador</i>	elemento para contar las operaciones. Puede asumir un valor comprendido entre -2.147.483.648 y +2.147.483.647. Su valor se almacena en la memoria no volátil de la tarjeta de ejes.
	<i>Bit Indicador</i>	elemento que puede asumir un valor de "encendido" o de "apagado".
	<i>Interruptor indicador</i>	son banderas especiales que pueden ser asociadas a algunos pulsadores de la barra de instrumentos, como Por ejemplo, la bandera de Inicio.
	<i>Puerto Indicador</i>	se compone de 8 líneas de bites indicadores.
	<i>Variable</i>	variable global tipo <i>entero</i> del código GPL.
	<i>Variable</i>	variable global tipo <i>char</i> del código GPL.
	<i>Variable</i>	variable global tipo <i>flotante</i> del código GPL.
	<i>Variable</i>	variable global tipo <i>doble</i> del código GPL.
	<i>Variable</i>	variable global tipo <i>string</i> del código GPL.
	<i>Variable</i>	variable global tipo <i>array</i> del código GPL.
	<i>Variable</i>	variable global tipo <i>matriz</i> del código GPL.

3 Cuadro Sinóptico

3.1 Uso del Cuadro Sinóptico

Durante el funcionamiento de la máquina, es posible abrir la ventana del *Cuadro Sinóptico* que permite visualizar el estado de los dispositivos más significativos.

Las informaciones presentadas por los cuadros sinópticos coinciden con las de la ventana de diagnóstico. Sin embargo, mientras en esta última, las informaciones se presentan con una estructura tipo árbol (que incluye a todos los dispositivos de la máquina), los sinópticos permiten presentar las informaciones gráficamente (visualizando, por ejemplo, una imagen de la máquina y presentando las cotas de los ejes en proximidad de estos). Además, permiten seleccionar las informaciones más significativas y agrupar las otras pantallas secundarias que se pueden llamar por el usuario.

3.2 Cómo intervenir en el Cuadro Sinóptico

Con fines diagnósticos, el Usuario puede seleccionar las páginas que integran el sinóptico haciendo doble clic con el ratón en una de las áreas de la máquina, circunscritas en la imagen por un rectángulo punteado, también llamadas "zonas calientes".

Para apuntar a una "zona caliente", a un dispositivo o a un eje, sólo hay que desplazar el puntero del ratón sobre la imagen del objeto deseado. Mientras se mueve el ratón, en la Barra de estado aparece el nombre del dispositivo por el que está pasando la flecha del ratón.

Además, el puntero del ratón toma formas diferentes según el tipo de objeto hacia el que está apuntando, indicando de esta forma una de las siguientes acciones permitidas en ese objeto. Estas son:



lupa

si está sobre una "zona caliente"



mano

si está sobre un dispositivo de salida



cursor de texto

si está sobre una casilla en la que es posible seleccionar un valor.

3.3 Cómo intervenir en los Dispositivos

Se interviene en los Dispositivos apuntando con el ratón al dispositivo deseado, completando la acción como se detalla a continuación. (El tipo de acción depende del tipo de dispositivo).

Modo de representación	Acción	Dispositivo
<i>Icono dispositivo</i>	apuntar y hacer clic	Salida digital Interruptor indicador Bit indicador
<i>Casilla de configuración valor</i>	apuntar, hacer clic y seleccionar el valor	Salida analógica Puerto de salida Puerto indicador Cota eje Temporizador Contador




3.4 Movimiento Manual de los Ejes

Para acceder a la función de movimiento manual de los ejes, se deben poseer los oportunos [derechos de acceso](#). Estos los asigna el constructor de la máquina.

Para interactuar con uno de los ejes es suficiente un doble clic con el ratón en el campo de visualización de las cotas del eje deseado. Se consigue así la apertura de la ventana para el movimiento del eje. En el caso de ejes de tipo Virtual, Paso a Paso y Cuenta, el cuadro presenta un número de informaciones reducidas. Por ejemplo, si el eje es de tipo Cuenta solo se visualizan los valores de Cota Real y de Velocidad.



En la ventana hay dos zonas que contienen:



Zona de visualización



- Tres casillas que contienen la visualización de la *Cota real* del eje [mm], de su *Velocidad actual* [m/min] y del *Error de anillo* o de persecución.
- Dos pulsadores de selección que indican el *Estado* del eje (*Libre* = de anillo abierto, por ejemplo, debido a un error de sistema, *Normal* = de anillo cerrado, en el estado de control de posición normal). Estos botones también permiten establecer el estado.
- La indicación durante el movimiento del *Estado* del eje (por ejemplo, Aceleración).
- Dos pulsadores para ejecutar en la *Dirección negativa*  o *positiva*  el desplazamiento del eje.
- El botón  para la Parada en cualquier instante del movimiento del eje durante los desplazamientos en el modo Absoluto y Paso.

Zona Movimiento




- Dos casillas para seleccionar una *Cota Negativa* y una *Cota Positiva*, que se ejecutarán en el modo *Absoluto*.
- Una casilla para programar la *Velocidad* para imponer al eje durante los movimientos manuales.
- Tres pulsadores para seleccionar el modo en que se efectuará el movimiento, a escoger entre: *Jog*, de cota *Absoluta* o *Paso*.
- Una casilla para programar el valor del *Paso* para efectuar en el modo *Paso*.

Para mover un eje se deberán fijar oportunamente los parámetros descritos. Se deberá elegir la modalidad de movimiento y se deberá presionar el botón  (para mover el eje en la dirección positiva) o el pulsador  (para mover el eje en la dirección negativa).

En modalidad *Jog* el eje se moverá hasta que el botón  o el botón  quede apretado.

En modalidad *Paso* el eje se moverá de la cantidad especificada en el cuadro de "paso" cada vez que se pulsa el botón  o el botón .

En la modalidad *Absoluto* el eje alcanzará directamente la cota establecida en la casilla Cota Positiva o Cota Negativa.

Como alternativa a los botones ,  y  es posible utilizar las teclas "+" (o Ctrl+P), "-" (o Ctrl+M) y "barra espaciadora" del teclado.

4 Parámetros Tecnológicos y Herramientas

4.1 La ventana Parámetros Tecnológicos

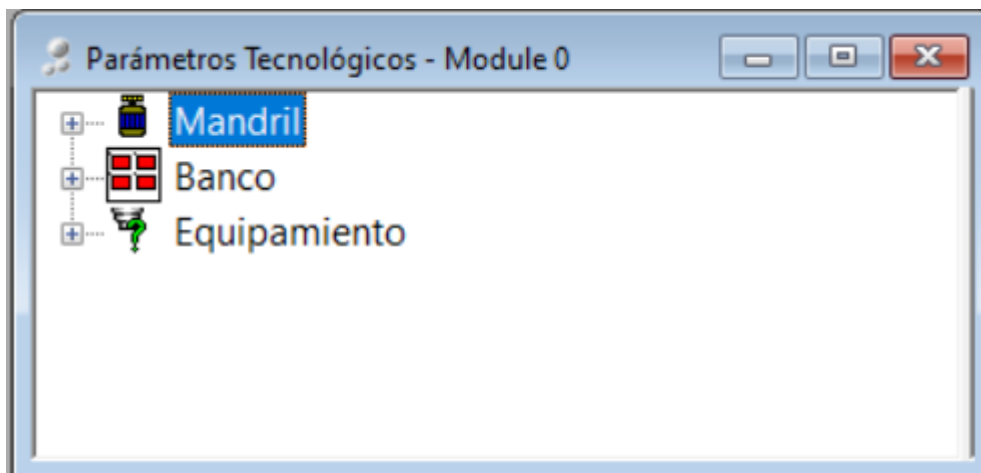
El archivo de Parámetros Tecnológicos permite memorizar las informaciones de carácter geométrico y tecnológico de una máquina. Estas informaciones son necesarias al control numérico para poder administrar correctamente el funcionamiento de la máquina.

Se abre desde el menú **Archivo->Abrir Parámetros Tecnológicos**.

En general los Parámetros Tecnológicos están organizados por Grupos/Subgrupos (en general los grupos y los subgrupos de los Parámetros Tecnológicos no están relacionados a los grupos y subgrupos donde están organizados los dispositivos de la máquina). Las modalidades de visualización las define el Fabricante de la Máquina y se diferencian en función de la aplicación específica.

Por lo general los valores indicados los establece el Fabricante al calibrar la Máquina y normalmente no pueden, si bien tan solo de vez en cuando, ser modificados por el Usuario final. Por lo general los valores indicados los establece el Constructor al calibrar la Máquina y normalmente no pueden, si bien tan solo de vez en cuando, ser modificados por el Usuario final. Por consiguiente, puede ser que algunos datos estén sujetos a protección por contraseña, para evitar modificaciones accidentales que podrían perjudicar el funcionamiento correcto del sistema.

La ventana concerniente los Parámetros Tecnológicos presenta en una estructura en árbol todos los Grupos y Subgrupos de parámetros que componen el archivo, como se detalla en esta figura:



Estructura del archivo de Parámetros Tecnológicos

La ventana contiene, visualizados en una estructura en árbol, algunos Grupos con los Subgrupos de parámetros correspondientes. La estructura tipo árbol puede ser expandida o reducida actuando sobre los pulsadores y y de cada nodo. La apertura y el cierre de las partes que componen el árbol pueden efectuarse también con las teclas: +, - flecha de dirección derecha e izquierda.

Cómo intervenir en los Parámetros Tecnológicos

Una vez abierto el árbol del Grupo/Subgrupo deseado, se accede a la página que contiene los datos.

Los datos se pueden presentar tipo tabla, o bien en casillas de texto o de selección, depende del tipo de datos y de cómo han sido definidos por el Constructor.

Si se aportan modificaciones en los datos, entonces será necesario apretar el pulsador **[OK]** para que las modificaciones se vuelvan permanentes.

Equipamiento

Un caso particular de los datos correspondientes a una máquina es el de las herramientas. Típicamente las informaciones relativas al conjunto de herramientas con las cuales está dotada la máquina (equipamiento), se memorizan en el archivo de Parámetros Tecnológicos. Los datos correspondientes a las herramientas, en cambio, se memorizan en el archivo de Parámetros Herramientas.

Para definir el equipamiento de la máquina, por lo tanto, será necesario relacionar las informaciones de los dos archivos. Si la aplicación lo prevé, por lo tanto, será posible invocar desde el archivo de Parámetros Tecnológicos las informaciones del archivo de Parámetros Herramientas. Típicamente la relación se pone en acto por medio de un pulsador con un icono similar al siguiente.



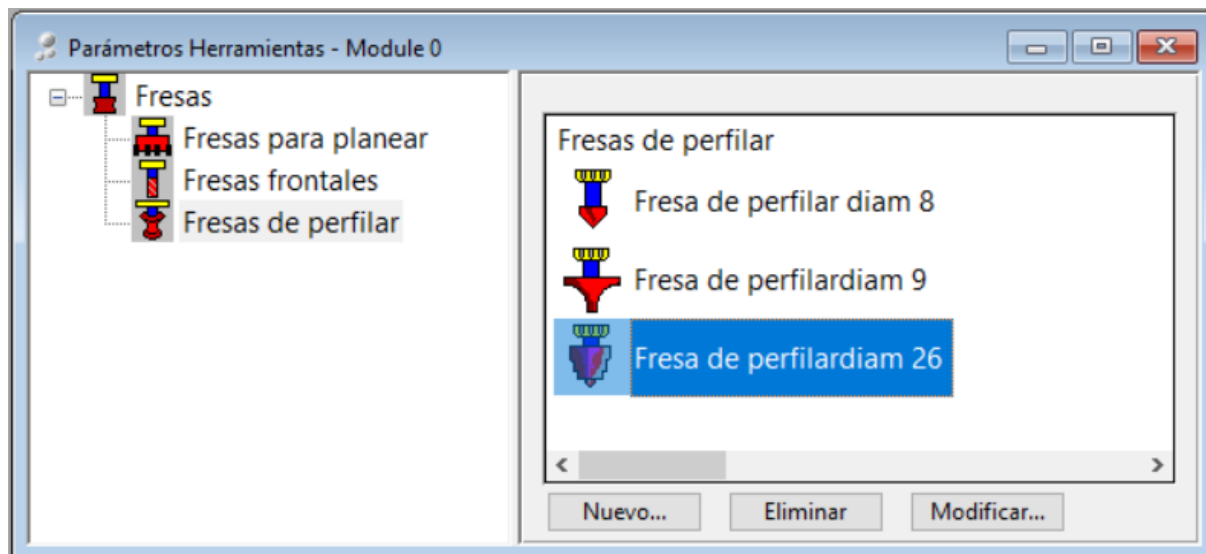
Seleccionando el icono y efectuando un clic doble con la tecla izquierda del ratón, se abre una ventana que contiene la lista de herramientas definidas en el archivo de Parámetros Herramientas de donde es posible seleccionar la herramienta deseada. Una vez efectuada la selección el icono del botón cambia y se visualiza el que está asociado a la herramienta específica.

Además, es posible visualizar los datos relativos a la herramienta efectuando sobre el icono un clic doble con la tecla derecha del ratón.

4.2 La ventana Parámetros Herramientas

Las ventanas de los Parámetros Herramientas se abren desde el menú **Archivo->Abrir Parámetros Herramientas**.

Las paramétricas de herramientas, configuradas por el Constructor según los trabajos que la máquina puede efectuar, por lo general están organizadas como se detalla en la figura a continuación:



Ejemplo de una ventana Parámetros Herramientas.

La ventana Parámetros Herramientas se compone de dos zonas:

- la *zona izquierda* contiene, visualizados en una estructura en árbol, los Grupos con los Subgrupos correspondientes de herramientas. La estructura tipo árbol puede ser expandida o reducida actuando sobre los pulsadores y de cada nodo. Por ejemplo, se puede tener un Grupo Fresadoras compuesto por Subgrupos de fresadoras con características diferentes entre sí, cuales fresadoras perfiladas, fresadoras con ajuste horizontal, etc. Cada uno de estos subgrupos tiene asociada una o varias herramientas cuyas características se asignan en un cuadro de diálogo definido por el fabricante. Las herramientas que hay en cada subgrupo se visualizan en la zona derecha de la pantalla.
- la *zona derecha*, con el título del nombre del Subgrupo seleccionado, contiene el listado de Herramientas que forman parte del Subgrupo. No es obligatorio que las herramientas definidas en esta etapa estén en la máquina. La asociación entre herramienta y posición en la máquina (equipamiento) normalmente se realiza en el archivo de parámetros tecnológicos.

Cómo intervenir en los Parámetros Herramientas

Las operaciones de introducción, modificación y borrado de herramientas del archivo se llevan a cabo a través de los *botones* que se hallan en la parte baja de la ventana:

- [Nuevo...]** permite introducir en el Subgrupo una nueva herramienta. Abre el cuadro de diálogo "Nueva Herramienta" donde es posible asignar los siguientes datos:
- *Descripción*: es un mensaje que identifica a la herramienta. La descripción puede ser elegida entre las que están en el listado, siempre que ya no haya sido asignada a otra herramienta, o bien se puede definir una nueva.
 - *Imagen*: es un icono que identifica a la herramienta. Puede ser elegida entre las que ya están en el listado, o bien puede ser invocada desde una carpeta utilizando el pulsador [Imagen]. La herramienta se introduce respetando el orden alfabético de las descripciones.
- [Eliminar]** permite eliminar una herramienta del Subgrupo, previa confirmación; la descripción correspondiente no se elimina y queda disponible para otra herramienta

[Modificar...] permite sustituir la *descripción* o la *imagen* a la herramienta seleccionada; aparece la misma ventana que se ha descrito en el comando **[Nuevo...]**.

5 Diagnóstico

5.1 La ventana de Diagnóstico

Durante la fase ejecutiva de la máquina, se puede abrir la ventana *Diagnóstico* que permite al Usuario controlar el estado de funcionamiento de la máquina, supervisado el estado lógico de las señales de E/S digitales, el valor de las E/S analógicas, el de los contadores y temporizadores y el movimiento de los ejes. En función de los [derechos de acceso](#) asignados por el constructor, además, es posible modificar el estado de los dispositivos.

Se puede, si está permitido por el nivel de acceso, lograr en tiempo real:

- la visualización del estado ON/OFF de todas las señales de Entrada y Salida digital
- la posibilidad de activar y desactivar todas las señales de Salidas digitales
- la visualización de los valores actuales (en el intervalo +/-10V) de las entradas Analógicas
- la posibilidad de asignar un valor (en el intervalo +/-10V) a todas las salidas Analógicas
- la posibilidad de asignar un valor (en el intervalo +/-10V) a todas las salidas Analógicas
- la posibilidad de mover un eje manualmente, seleccionando la velocidad, el valor de Paso o la Cota absoluta final, visualizando la posición real, la velocidad y el error de anillo
- visualización y modificación de las variables globales

En los próximos párrafos se describirán en detalle los dispositivos y las variables globales y su representación gráfica.

ATENCIÓN: En la ventana diagnóstico se visualizan exclusivamente los dispositivos habilitados para el nivel de acceso corriente.

5.2 Composición de la ventana de Diagnóstico

Con referencia a la estructura "Grupos/Subgrupos" ya definida en el capítulo [Composición del sistema](#), es imposible acceder a los dispositivos que luego se muestran en una estructura.

Como encabezamiento de la estructura encontramos el grupo representado por el icono



, seguido por el Nombre y un Comentario.

La estructura se expande o reduce haciendo un *clic* sobre el botón o . La apertura y el cierre de las partes que componen el árbol pueden efectuarse también con las teclas: +, -, flechas direccionales derecha e izquierda.

Al abrir un Grupo, aparecen en el árbol

- la "Lista dispositivos" del Grupo, representada por el símbolo
- los eventuales Subgrupos que componen el Grupo.

Al abrir un Subgrupo, aparecen los dispositivos que lo componen.

5.3 Representación de los Dispositivos

Por cada dispositivo visualizado se muestra los datos siguientes:




- el Símbolo gráfico;
- el Estado o el Valor actuales;
- el Nombre;
- el Comentario.

A continuación, se muestra la representación gráfica de los dispositivos, el tipo de dispositivo y cual magnitud del mismo se visualiza en tiempo real.

El estado de las entradas y de las salidas digitales y de las banderas está representado gráficamente por un LED que toma un cierto color, según que la entrada esté activa o no activa.

En caso de Puertos, es decir más líneas representadas contemporáneamente (8), aparecerá una fila de LED; la primera línea del grupo está representada por el LED más a la derecha, la última por el LED más a la izquierda.

Dispositivo	Símbolo	Estado	Visualización en tiempo real
-------------	---------	--------	------------------------------

Entrada digital			estado: Activo = VERDE, No activo = GRIS
Salida digital			estado: Activo = ROJO, No activo = GRIS
Entrada analógica		22.000	valor actual
Salida analógica		22.000	valor numérico actual en Voltios
Puerto de entradas			estado de cada línea (como Entrada digital). Estado: Activo = VERDE, No activo = GRIS
Puerto de salidas			estado de cada línea (como Salida digital). Estado: Activo = ROJO, No activo = GRIS
Eje		100.000	posición absoluta actual
Temporizador		12.000	valor actual en segundos
Contador		58	valor numérico actual
Bit indicador			estado: Activo = AMARILLO, No activo = GRIS
Interruptor indicador			estado (como Bit indicador). Estado: Activo = AMARILLO, No activo = GRIS
Puerto indicador			estado de cada línea (como Bit indicador). Estado: Activa = AMARILLO, No Activa = GRIS
Variable global		2	variable global tipo entero del código GPL
Variable global		127	variable global tipo char del código GPL
Variable global		50.00000000	variable global tipo flotante del código GPL
Variable global		200.00000000	variable global tipo doble del código GPL
Variable global		Area	variable global tipo string del código GPL
Variable global		[256]	variable global tipo array del código GPL
Variable global		[10][3]	variable global tipo matriz del código GPL

5.4 Interactuar con los Dispositivos

Con finalidad de diagnóstico se puede interactuar con un dispositivo para leer su estado o modificar su valor. Lo cual no es posible con algunos tipos de dispositivo, como los de entrada y los protegidos por el Constructor, en este caso un mensaje informará al Usuario en caso de que el mismo intentara efectuar operaciones en el dispositivo.

Una vez seleccionado el dispositivo, presionar *dos veces con el ratón* o bien presionar la tecla **Entrar**, o la **Barra Espaciadora**, para acceder a la ventana que permite modificar el estado o el valor del dispositivo.

De tratarse de una **Salida digital** o de un **Bit indicador**, no aparecerá ninguna ventana, sino se modificará inmediatamente el estado del dispositivo. El buen funcionamiento de la salida señala la variación del color del LED que indica su estado.

En caso de **Puerto de Salida** habrá que apuntar con el ratón sobre el LED que corresponde con la salida deseada y presionar dos veces para modificar el estado.

Lo mismo dígame para los **Interruptores indicadores** y el **Puerto del indicador**.

En cuanto a las **Salidas analógicas**, los **Temporizadores** y los **Contadores** aparece un cuadro de diálogo que visualiza el valor actual y permite programar el nuevo valor que se desea que el dispositivo asuma

inmediatamente.

La modalidad de interacción con un **Eje** se detalla en el punto [Movimiento Manual de los Ejes](#).

5.5 Lista de teclas para navegar dentro de una estructura de árbol

Tecla	Descripción
Flecha arriba	mueve la selección a la línea que se encuentra directamente encima o
Flecha abajo	debajo de la línea actual
Flecha derecha	expande en un nivel la rama seleccionada y, si ya expandida, expande la selección en la rama siguiente
Flecha izquierda	cierra la rama seleccionada y, si ya cerrada, desplaza la selección en la rama anterior
+	expande en un nivel la rama seleccionada
-	cierra la rama seleccionada
*	expande todos los niveles de la rama seleccionada
Ctrl+Alt+Mayús y Entrar	visualiza las tablas de los correctores de linealidad asociadas a un eje.
Ctrl+Alt+Mayús y clic con el botón izquierdo	Si se activa la combinación de teclas mientras un eje en el árbol de los dispositivos de un módulo está seleccionado, se visualizan todos los correctores asociados a un eje, como si fueran una matriz, donde las columnas son los ejes asociados (la primera columna es la de los auto correctores) y las filas son los valores de corrección. Los valores que pueden haber sido modificados se asumen durante el movimiento del eje, pero no se almacenan en disco.

5.6 Correctores de linealidad

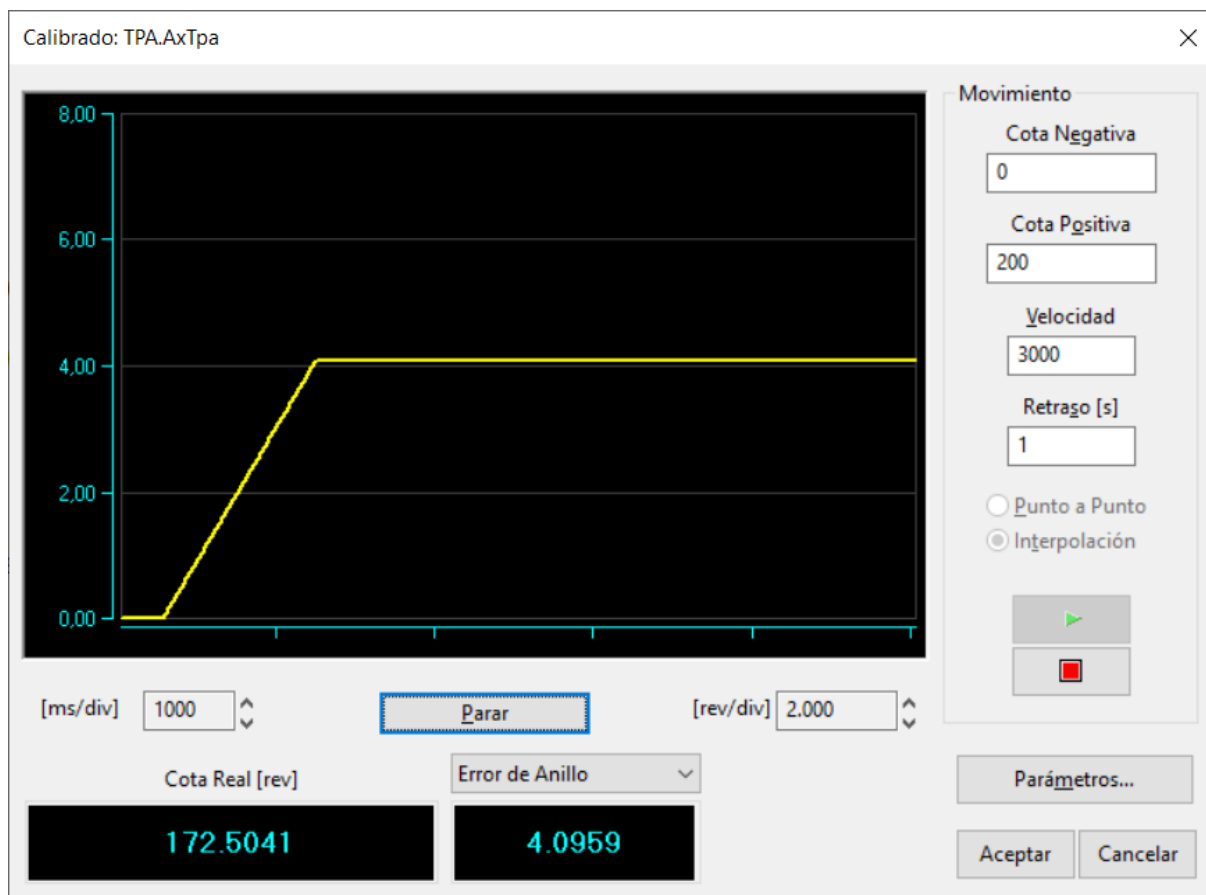
En el diagnóstico se pueden ver y modificar los correctores de linealidad del eje seleccionado, abriendo el menú contextual y seleccionando la opción [Correctores de linealidad](#). La opción está visible sólo si se definieron correctores de linealidad por el eje en la configuración. Como alternativa al menú, se puede utilizar la combinación de teclas **[Ctrl+Mayús+Entrar]**.

5.7 Panel de calibrado de los ejes

El panel de calibrado de los ejes es un instrumento que permite modificar los parámetros de configuración de un eje y simultáneamente mover el eje y exhibir el comportamiento en un osciloscopio virtual.

Para acceder al panel de calibrado de ejes se debe tener un [nivel de acceso](#) al sistema mayor o igual a "Constructor". Se accede al panel de calibrado desde la modalidad diagnóstico o manual efectuando un clic doble sobre el eje que se desea calibrar mientras se mantiene presionada la tecla **[Mayús]** o desde menú contextual seleccionando la opción "Calibrado".

De esta manera, se accede al panel de calibrado que se ve en la figura que sigue:



Para controlar el comportamiento del eje con la variación de los parámetros se lo hace mover de modo continuo entre dos alturas límites denominadas **Cota Positiva** y **Cota Negativa**. Aparte de estos parámetros se deberá establecer la **Velocidad** de movimiento del eje. En las primeras fases del calibrado se recomienda utilizar un valor bajo de velocidad. Asimismo, se puede fijar un **Retraso** que aplicar entre dos movimientos sucesivos.

En la ventana osciloscopio se visualizará el trazado del error de anillo o de una de las otras magnitudes del eje. Es posible, análogamente a los osciloscopios de banco, escalar oportunamente el gráfico de modo tal de adaptarlo a la dimensión de la ventana o para resaltar sus detalles. Mediante el ratón y las teclas de control, se puede rever el último minuto de calibrado, visualizar uno o dos cursores para hacer medidas y verificaciones en el dato mostrado, agrandar un área del gráfico para analizar detalles del dato mostrado, variar el desplazamiento y la escala tanto en las abscisas como en las ordenadas. Asimismo, se puede bloquear, presionando el pulsador **Parar**, el deslizamiento del gráfico para observarlo tranquilamente sin tener que detener el eje.

Aparte del gráfico, se visualizan en dos casillas la cota real (a la izquierda) y la cota indicada en el gráfico (a la derecha). Esta puede ser programada con el cuadro combinado colocado arriba de la casilla de visualización.

Para modificar los parámetros del eje hay que presionar el botón **[Parámetros...]**, este activa la ventana en donde es posible modificar rápido la mayoría de los parámetros del eje. Una vez aportada una modificación a uno o más parámetros será posible actuarla presionando el botón **[Aplicar]**. Este botón permite ver de inmediato el efecto de la modificación en el comportamiento dinámico del eje. Presionando **[Aceptar]** las modificaciones continúan a ser empleadas, mientras presionando **[Cancelar]** se restauran los valores presentados antes de la abertura de la ventana **[Parámetros...]**.

Los parámetros sobre los cuales típicamente habrá que trabajar son:

- Coeficiente **Proporcional**
- Coeficiente **Integral**
- Coeficiente **Derivativo**
- **Feed Forward**: porcentaje de la velocidad corriente entregado directamente al accionamiento (independientemente del error de anillo)
- **Feed Forward Acel.**: porcentaje de la referencia de velocidad entregado directamente al accionamiento durante las fases de aceleración y desaceleración del eje (como agregado a feed forward)
- **Aceleración**: duración de la rampa de aceleración

- **Deceleración:** duración de la rampa de desaceleración

El cursor en el área gráfica

El cursor es una herramienta que permite medir y visualizar unos datos de la pista. Es una línea vertical del color asociado a la pista seleccionada. Se puede desplazar dentro del gráfico con las teclas cursor o con el ratón. En una información sobre herramientas conectada con la línea vertical se pueden visualizar valores diferentes, seleccionables por un menú emergente, que se puede llamar haciendo clic con la tecla derecha del ratón cerca del cursor.

En el menú emergente se pueden elegir las opciones:

- **Canal:** muestra la lista de las pistas visualizadas en el gráfico y señala con una marca de verificación a la que se refiere el cursor. Es también posible elegir una nueva pista para asociarla al cursor.
- **Estilo:** muestra una lista de datos que pueden ser visualizados en la ventana de la información sobre herramientas.
- **Valor X-Y:** visualiza el valor sobre el eje x y sobre el eje y del punto de la pista donde está el cursor.
- **Valor X:** visualiza el valor sobre el eje x del punto de la pista donde está el cursor.
- **Valor Y:** visualiza el valor sobre el eje y del punto de la pista donde está el cursor.
- **Período:** visualiza el valor de la distancia entre dos cursores visualizados, cuando se hace la medición a lo largo del eje X.
- **Pico-pico:** visualiza el valor de la distancia entre dos cursores visualizados, cuando se hace la medición entre dos puntos a lo largo del eje Y.
- **Frecuencia:** visualiza el inverso de la distancia a lo largo del eje X.
- **Opciones:** configura la modalidad de visualización del cursor y de la información sobre herramientas asociadas
- **Utilizar el color del canal:** si habilitado, el cursor está dibujado con el color del dato muestreado al cual se refiere, de otra manera la elección del color con el cual está visualizado el cursor es casual.
- **Ocultar sugerencia en el despacho:** si habilitado, la información sobre herramientas se visualiza sólo hasta que se mantiene pulsado el botón izquierdo del ratón, después se oculta.
- **Voltear alineación:** en general, la información sobre herramientas se visualiza a la derecha del cursor. Si la opción está habilitada, el cursor se visualiza a la izquierda del cursor.
- **Ajustar a punto:** si habilitado, coloca el cursor sólo sobre los valores muestreados.

Calibrado del eje

El calibrado de un eje es una operación delicada que se debe realizar con suma atención y cautela.

Mediante la opción CalibSampleTime en la sección [Albatros] en Tpa.ini se puede modificar el periodo de muestreo de los datos de un eje para le ventana de calibrado. El valor está expresado en mili-segundos y no puede ser inferior a la frecuencia control ejes ni mayor de 100.

Antes de efectuar el calibrado del eje desde el panel es necesario definir todos los parámetros en configuración y se deberá definir el fondo de escala de la velocidad del accionamiento. El valor de tensión con el cual en Albatros corresponde la velocidad máxima es de 9 Voltios.

Para no dañar la máquina usando parámetros errados, conviene definir una velocidad baja, por ejemplo, igual al 10% de la velocidad máxima prevista para el eje. De este modo incluso si se define una ganancia elevada no se obtendrán reacciones demasiado violentas del eje.

En general se efectúa primero el calibrado para los movimientos punto a punto y luego el calibrado para los movimientos interpolados.

La primera operación, si no se hizo en la configuración, es la definición de los tiempos de aceleración y deceleración. Cuanto mayores serán estos tiempos, menor será la aceleración a la cual será sometido el eje.

Luego se define una ganancia mínima que permite el movimiento del eje. Esto sirve para controlar el correcto calibrado del accionamiento. Albatros en efecto está configurado para entregar una referencia de 9 Voltios en correspondencia de la máxima velocidad configurada en la configuración del eje. Por ejemplo, si se mueve el eje a una velocidad igual al 10% de la máxima y si el accionamiento está correctamente calibrado, se leerá una tensión de referencia igual al 10% de la máxima, o sea 0,9 Voltios. Si no se obtiene esta tensión de referencia será necesario modificar el fondo de escala del accionamiento.

Cuando se ha calibrado el accionamiento se comienza a aumentar de a poco y siempre con suma precaución, la ganancia del anillo de posición. En cada oportunidad se debe controlar que no se manifiesten condiciones de sobre-elongación o inestabilidad. La velocidad en esta fase siempre deberá ser mantenida menor o igual al 10% de la velocidad máxima. En esta fase es siempre oportuno analizar profundamente el perfil de velocidad

obtenido con el osciloscopio virtual, amplificando todo lo que fuera posible la imagen de modo tal para resaltar los detalles.

Cuando el eje logra un comportamiento que sea al mismo tiempo dispuesto y estable, se podrá aumentar, un poco por vez, la velocidad, controlando el comportamiento del eje en cada oportunidad y adaptando la ganancia en el caso que no fuera satisfactoria. La ganancia y la velocidad jamás se deberán aumentar bruscamente. Condiciones de calibrado aparentemente estables a bajas velocidades pueden no serlo a velocidades más altas.

Una vez determinado el valor óptimo de la Ganancia, de ser necesario, se podrá aumentar progresivamente los coeficientes Integrativo y Derivativo y en seguida el Feed Forward para disminuir el error de anillo y llevarlo dentro de valores aceptables desde el punto de vista de precisión del eje. El Feed Forward permite abatir, casi en su totalidad, el error de anillo durante la fase de régimen del movimiento más no durante las fases de aceleración y desaceleración. Para disminuir aún más el error de anillo en estas fases se podrá aumentar el Feed Forward Aceleración. En general valores incluso muy bajos de este parámetro son suficientes para conseguir un buen resultado.

Por lo que concierne el calibrado de un eje para los movimientos interpolados se pueden utilizar los valores ya determinados para los movimientos punto a punto, sin embargo, se debe de tener en cuenta la presencia de los otros ejes de la máquina. En particular para conseguir una buena precisión durante los movimientos interpolados es necesario balancear los errores de anillo de los ejes. Por lo tanto, se deberá determinar el eje con el mayor error de anillo (a la misma velocidad) y "empeorar" (con respecto a los parámetros de interpolación) el calibrado de los otros de modo que los errores de anillo sean idénticos.

6 Errores y Advertencias

6.1 Introducción

Albatros gestiona eventos de error y advertencias

Errores de sistema

Estamos hablando de los errores que el sistema Albatros puede detectar automáticamente, tanto durante la ejecución de los programas, cuanto durante las operaciones de mantenimiento y diagnóstico de la instalación.

Dichos errores son de varia naturaleza y van de problemas de gestión de un eje a problemas que se pueden producir durante la ejecución del programa.

Los errores del sistema pueden ser manejados dentro de los programas de trabajo por medio de la instrucción [ONERSYS](#), en caso contra de su ocurrencia se termina la ejecución se acaba la ejecución de programas en el módulo en el que se produjo el error. La aclaración de cada error de sistema está presentada en las páginas siguientes en este manual.

Errores de ciclo

Son errores que ocurren durante la ejecución del programa, y que aún permiten generalmente que ese continúe después la eliminación del error mismo. Se pueden generar errores de ciclo a través de la instrucción GPL [ERROR](#).

Mensajes

Son mensajes de advertencia que se producen en situaciones particulares de la ejecución del programa, o notificaciones de solicitud de intervención por el usuario, pero que no paran la ejecución del programa mismo. Se pueden generar mensajes a través de la instrucción GPL [MESSAGE](#).

Notificaciones

Son notificaciones genéricas, no relativas a los ciclos de máquina, y que son escritas por Albatros. No se visualizan en Albatros. La aclaración de cada notificación está en las páginas siguientes en este manual.

Barra de errores

Sobre la barra de errores está mostrado el último error de sistema que se ha producido en orden cronológico, junto con el último error de ciclo y el último mensaje.

Los **errores de sistema** se señalan con color rojo.

Los **errores de ciclo** se señalan con color rojo. Se trata de errores que se verifican durante la ejecución del programa más que generalmente permiten su continuación, previa remoción del mismo error.

Los **mensajes** se señalan con color verde.

Máquina 1: Módulo reconectado	00002
Máquina 1: Setpoint de los ejes aún no ejecutado	00001
Máquina 1: Inicializando...	00001
00000 00001	NUM

Barra de errores

Los errores que se han producido a partir del sistema se muestran en una ventana que se puede abrir con un doble clic del ratón en la *Barra de los errores* o bien desde el menú **Visualizar**. En esta ventana, además, se visualizan informaciones adicionales acerca de los errores de sistema.

La ventana está dividida en las siguientes áreas: En la **parte alta** se visualizan las siguientes informaciones:

- Tipo: identifica el tipo de error (error de sistema, error de ciclo y mensaje).
- Hora&Fecha: es la hora y la fecha en la que se ha producido el error.
- Descripción: es la descripción del error.
- Código: es el número del mensaje de error.
- Tarea: es el nombre de la tarea que ha causado el error (no presente en la Barra de errores).

Haciendo *doble clic* en una de estas columnas, la información se ordena según el contenido de la columna.

En la **parte de abajo** encontramos las casillas:

- Error de ciclo: si se ha activado se muestran también estos errores.
- Mensajes: si se ha activado se muestran también los mensajes.
- Todos: si está activado, se muestran los mensajes de todos los módulos del sistema, que corresponden al tipo de información que se está visualizando.

- Casilla nombre módulo: muestra el nombre del módulo en el que se están efectuando las visualizaciones, permitiendo seleccionar, en caso de un sistema con más módulos, el módulo cuya información se quiere visualizar

Finalmente encontramos los **botones** de comando que son:

- **[Eliminar todos]** borra toda la información de la memoria visualizada, sin eliminar el archivo.
- **[Eliminar]** permite borrar la información actual de la memoria, sin eliminarla del archivo.
- **[Aceptar]** permite cerrar la ventana.

Almacenamiento de los errores y de las notificaciones en un archivo de informe

Todos los errores se almacenan en un archivo para reproducir un historial de ellos mismos. Se trata de un archivo de texto en formato TSV. El nombre del archivo es MONTH (nombre del mes en curso).TER. Con el conjunto de Albatros se provee de un programa ViewRER que carga y monstra los archivos .TER.

6.2 Errores de sistema

6.2.1 Errores ocasionados por la gestión de los ejes

1 NombreEje: vínculo codificador incorrecto

Causa:

Con el eje parado, se ha producido una diferencia entre la cota teórica y la cota real del eje, mayor de 1024 pasos de codificador.

Esto suele ocurrir, durante la puesta en servicio del eje, cuando están invertidas las fases del codificador. Durante el funcionamiento normal se verifica cuando con el accionamiento apagado se mueve manualmente el eje sin haberlo colocado previamente en la posición LIBRE o bien cuando el eje, a causa de una calibración incorrecta, es sujeto a una excesiva sobre-elongación en la fase de llegada a su altura (overshoot). Después de este error, se restablece la señal de referencia y el eje se coloca en estado LIBRE.

Solución:

Durante la puesta en servicio del eje, controle la conexión de las fases de codificador del eje en cuestión (eventualmente activar la opción inversión de fases de codificador en la configuración del eje). Controle el calibrado del eje con la correspondiente modalidad de Diagnóstico.

2 NombreEje: movimiento no finalizado

Causa:

Al final del desplazamiento, después de 5 segundos a partir del final del movimiento teórico, la diferencia entre la cota teórica y la cota real del eje ha resultado mayor que la ventana indicada en Configuración. Simplemente puede depender del hecho que el accionamiento está apagado o no está habilitado o se puede deber a una mala regulación del desplazamiento del accionamiento. También puede ser debido a juegos mecánicos en el eje o a una ganancia del anillo de posición excesivamente baja.

Solución:

Verifique que el accionamiento esté encendido y habilitado. Verifique la calibración del eje y ajuste el desplazamiento del accionamiento del eje en cuestión.

3 NombreEje: servoerror

Causa:

Durante cualquier tipo de movimiento, la diferencia entre la cota teórica y la cota real del eje es superior al error máximo indicado en Configuración o del definido con la instrucción [SETMAXER](#). Normalmente se debe a un mal ajuste de la ganancia del anillo de posición o bien del calibre de velocidad del accionamiento, o bien a una inercia del eje excesiva.

Solución:

Compruebe el ajuste de la ganancia y el calibre de velocidad del accionamiento. Compruebe el funcionamiento correcto del codificador y del motor/accionamiento. Además, compruebe eventuales bloques mecánicos.

4 NombreEje: superior al límite positivo

Causa:

La cota teórica del eje ha sobrepasado la cota límite positiva seleccionada en Configuración o definida con la instrucción [SETLIMPOS](#).

Solución:

Corrija en el programa la cota que excede la cota límite positiva o vuelva a definir los límites de altura del eje.

5 NombreEje: superior al límite negativo

Causa:

La cota teórica del eje ha sobrepasado la cota límite negativa seleccionada en Configuración o definida por la instrucción [SETLIMNEG](#).

Solución:

Corrija en el programa la cota que excede la cota límite negativa o vuelva a definir los límites de altura del eje.

10 NombreEje: La ejecución en tiempo real es más rápida que la construcción del perfil

Causa:

La ejecución en tiempo real del perfil de movimiento es más rápida que la generación GPL del mismo perfil. El lookahead se vacía más rápidamente de lo que se llena. El error puede ser debido a dos causas generalmente concomitantes:

- le velocidad de interpolación está demasiado alta en comparación con las dimensiones de los tramos a recorrer
- los tramos a recorrer son demasiado cortos.

Solución:

Verifique que la velocidad de interpolación programada no esté demasiado alta con respecto a las dimensiones de los tramos a recorrer y verifique que los tramos de interpolación a recorrer no estén demasiado cortos.

6.2.2 Errores ocasionados por la gestión de E/S remotas

2049 Receptor número: configuración incorrecta

Causa:

El receptor remoto ha recibido una configuración diferente a la detectada en el campo. Esto puede ocurrir, si el remoto no es igual al remoto elegido en la configuración hardware de Albatros. Por ejemplo, el receptor remoto es un Albre16 y en Albatros se configuró un remoto Albre24 (GreenBus v3.0), o bien un TRS-IO con un número incorrecto de expansiones TRS-IO-E (GreenBus v4.0).

Solución:

Verifique la Configuración Hardware.

2050 Receptor número: desconectado

Causa:

El receptor remoto no responde a los comandos del transmisor.

Solución:

Compruebe la alimentación del receptor y la conexión en serie.

2051 Receptor número: revinculado

Causa:

La conexión entre transmisor y receptor se ha restaurado.

2052 Receptor número: error en relectura salida no conectada número NúmeroSalida

Causa:

La salida digital indicada está en modalidad protección o cortocircuito, de todos modos, se encuentra en un estado diverso del previsto por el control. La salida no está asociada a ningún dispositivo lógico en Configuración Virtual-Físico, por lo tanto, es síntoma de una incongruencia entre la Configuración y el efectivo cableo de la máquina.

Solución:

Verifique la Configuración Virtual-Físico. Remueva el cortocircuito o verifique que la carga aplicada no supere los límites máximos (consulte la documentación técnica).

2054 Receptor número: tipo incorrecto

Causa:

Durante la fase de inicialización de los remotos se ha verificado, a una cierta dirección, un receptor diferente al que había sido especificado en la configuración.

Solución:

Controle que la Configuración Hardware corresponda a las definiciones de los receptores remotos.

2055 Receptor número: inicializado

Causa:

El receptor se ha conectado al transmisor después de una desconexión debido a falta de electricidad.

2056 Receptor número: error alimentación +24 Vcc

Causa:

La alimentación de campo (+24 Vcc) de un dispositivo de E/S remoto no está activa o no funciona correctamente.

Solución:

Verifique el funcionamiento de la alimentación +24 Vcc.

2057 Error alimentación GreenBUS

Causa:

La alimentación del bus que conecta los receptores de E/S remoto con el control no funciona correctamente. Esta alimentación tiene que tener un valor nominal de +12 Vcc y es suministrada por el control.

Solución:

Verifique la presencia de la alimentación del GreenBUS, verifique los cables GreenBUS. Apague y vuelva a encender. Podría ser necesario sustituir la tarjeta de control.

2058 Receptor número: error en relectura TipoDispositivo NombreDispositivo

Causa:

El estado de la salida especificada no corresponde al establecido. Esto puede depender de la presencia de un cortocircuito, de la intervención de la protección (carga excesivo) o simplemente falta de alimentación. La salida especificada puede ser una salida digital, una salida analógica, una salida de control eje. El tipo de salida está especificado en la visualización del error.

Solución:

En caso de que se trate de una salida digital, verifique la alimentación +24 V (lado campo), remueva la eventual situación de cortocircuito o de excesiva absorción de la salida (consulte la documentación técnica). En caso de que se trate de una salida analógica o de una salida de control ejes, verifique la presencia y el valor de la tensión programada en la salida (tester u osciloscopio), remueva el eventual cortocircuito o la excesiva absorción de la salida (consulte la documentación técnica).

2059 Fracasada la prueba de la dual port memory del transmisor

Causa:

Se produjo un error durante las pruebas realizadas en fase de inicialización de la tarjeta de ejes. En particular ha fracasado la inicialización del transmisor GreenBus (micro-controlador i296). Puede depender de una mala configuración de las direcciones de E/S y IRQ de la tarjeta o de un conflicto con otros periféricos presentes en el sistema. También puede ser la consecuencia de un daño de la tarjeta de ejes.

Solución:

Verifique la configuración de la tarjeta, verifique que no haya conflictos con otros periféricos. Si se utiliza un receptor remoto, retransmita el firmware al módulo. Los técnicos calificados pueden efectuar una prueba Hardware de la dual port memory del micro-controlador i296. Si el problema persiste, póngase en contacto con el constructor.

2060 Error en la inicialización del transmisor

Causa:

Se ha verificado un error durante las pruebas realizadas en fase de inicialización de la tarjeta de ejes. En particular ha fallado el envío del firmware al transmisor GreenBUS (micro-controlador i296). Puede depender de una errada configuración de las direcciones de E/S y IRQ de la tarjeta o de un conflicto con otros periféricos presentes en el sistema. Finalmente, puede ser la consecuencia de un daño de la tarjeta de ejes.

Solución:

Verifique la configuración de la tarjeta, verifique que no haya conflictos con otros periféricos. Si se utiliza un receptor remoto, retransmita el firmware al módulo. Los técnicos calificados pueden efectuar una prueba Hardware de la dual port memory del micro-controlador i296. Si el problema persiste, póngase en contacto con el constructor.

2061 Error en la transmisión del firmware al transmisor

Causa:

Se ha verificado un error durante las pruebas realizadas en fase de inicialización de la tarjeta de ejes. En particular ha fallado el envío, al transmisor GreenBUS (micro-controlador i296), de la configuración de los receptores de E/S remoto.

Solución:

Verifique la configuración hardware, si se utiliza un receptor remoto, retransmita el firmware al módulo. Los técnicos calificados pueden efectuar una prueba Hardware de la RAM del micro-controlador i296. Si el problema persiste, póngase en contacto con el constructor.

2062 Error en la transmisión de la configuración al transmisor

Causa:

Se ha verificado un error durante las pruebas realizadas en fase de inicialización de la tarjeta de ejes. En particular ha fallado la inicialización de los receptores de E/S remoto.

Solución:

Verifique la configuración hardware, si se utiliza un receptor remoto, retransmita el firmware al módulo. Los técnicos calificados pueden efectuar una prueba Hardware de la RAM del micro-controlador i296. Si el problema persiste, póngase en contacto con el constructor.

2063 Error en la transmisión de la configuración al receptor

Causa:

Ha ocurrido un error durante la inicialización de un módulo remoto.

Solución:

Compruebe la configuración hardware. Los técnicos cualificados pueden efectuar un test Hardware del módulo remoto. Si el problema ocurre nuevamente, póngase en contacto con la asistencia técnica.

2064 Receptor número: versión firmware no compatible

Causa:

El receptor remoto está provisto de una versión de firmware no compatible con el firmware del control.

Solución:

Compruebe la instalación del control. Si el problema persiste, póngase en contacto con la asistencia técnica.

2065 Receptor número: Error en una comunicación asíncrona

Causa:

Ha ocurrido un error o una falta de respuesta durante la comunicación de un comando a un control remoto (GreenBUS v 4.0).

Solución:

Verifique las conexiones y la alimentación del GreenBUS. Si el problema persiste, póngase en contacto con la asistencia técnica.

2066 Receptor número: Error genérico

Causa:

Ha ocurrido un error genérico durante la comunicación de un evento o de un alarme desde un remoto (GreenBUS v 4.0).

Solución:

Verifique las conexiones y la alimentación del GreenBUS. Si el problema persiste, póngase en contacto con la asistencia técnica.

2067 Receptor número: Error durante la transmisión de la configuración

Causa:

Ha ocurrido un error de comunicación durante la transmisión de los datos de configuración a un remoto (GreenBUS v 4.0).

Solución:

Verifique las conexiones y la alimentación del GreenBUS. Apague y vuelva a encender. Si el problema persiste, póngase en contacto con la asistencia técnica.

2068 Receptor número: Error interno n. NúmeroError**Causa:**

Ha ocurrido un error interno en el remoto indicado.

Solución:

póngase en contacto con el constructor.

2069 Receptor número: Error alimentación +24 Vcc banco número**Causa:**

La alimentación de campo (+24 Vcc) de un grupo de salida conectado al mismo terminal eléctrico no está activa o no funciona correctamente.

Solución:

Compruebe el funcionamiento de la alimentación +24 Vcc.

6.2.3 Errores ocasionados por la gestión MECHATROLINK-II**2308 Tarjeta NúmeroTarjeta: Inicialización fracasada por valor incorrecto de un parámetro de configuración****Causa:**

Probablemente en configuración virtual ningún servo-accionamiento ha sido conectado con una tarjeta con bus MECHATROLINK-II (dispositivo físico).

Solución:

Verifique las conexiones físicas en Configuración Virtual-Físico.

2341 Tarjeta NúmeroTarjeta: El número de servo-accionamientos excede el máximo admitido**Causa:**

A una tarjeta con bus MECHATROLINK-II se conectó un número elevado de servo-accionamientos con respecto a la configuración programada.

Solución:

Verifique en configuración de sistema el valor de Frecuencia Control Ejes.

En la tabla son presentados los valores correctos que hay que configurar según el número de servo-accionamientos gestionados por la tarjeta.

Tarjeta	Frecuencia Control Ejes (Hz)	Número máximo servo-accionamientos
AlbMech	1000	8
AlbMech	<=500	16
DualMech Mono	1000	8
DualMech Mono	500	20
DualMech Mono	250	30
DualMech	1000	16
DualMech	500	40
DualMech	250	60

2342 Tarjeta NúmeroTarjeta: La dirección hardware de servo-accionamiento Servo supera el valor máximo permitido**Causa:**

En una tarjeta con bus MECHATROLINK-II ha sido conectado un eje (dispositivo lógico) a una dirección hardware (dispositivo físico) superior al número de servo-accionamientos que se pueden gestionar desde la tarjeta.

Solución:

Verifique en configuración de sistema el valor de Frecuencia Control Ejes. En la tabla son presentados los valores correctos que hay que configurar según el número de servo-accionamientos gestionados por la

tarjeta.

Tarjeta	Frecuencia Control Ejes (Hz)	Número máximo servo-accionamientos
AlbMech	1000	8
AlbMech	<=500	16
DualMech Mono	1000	8
DualMech Mono	500	20
DualMech Mono	250	30
DualMech	1000	16
DualMech	500	40
DualMech	250	60

Verifique en Configuración Virtual-Físico la conexión entre dispositivo lógico y dispositivo físico. Por ejemplo, si el número máximo de servo-accionamientos es 8, la conexión entre dispositivo lógico y dispositivo físico tiene que ser dentro de los primeros 8 ejes (desde Ax1 hasta Ax8).

2349 Tarjeta NúmeroTarjeta: El servo-accionamiento Servo no está conectado

Causa:

La conexión física al servo-accionamiento de la tarjeta indicada está interrumpida

Solución:

Controle los cableados del bus MECHATROLINK-II y el servo-accionamiento.

6.2.4 Errores ocasionados por la gestión CanBUS

2761 Nodo número: desconectado

Causa:

El nodo CAN indicado no está conectado al bus de campo que se refiere a la tarjeta indicada, aunque está presente en la configuración.

2762 Nodo número: revinculado

Causa:

El nodo CAN indicado está nuevamente conectado al bus de campo que se refiere a la tarjeta indicada.

2763 Error de falta de transmisión

Causa:

Error en la tarjeta indicada. No se pudo efectuar la transmisión de uno o más paquetes al nodo CAN indicado

Solución:

Póngase en contacto con el constructor.

2764 Nodo número: Error de falta de recibir

Causa:

No se pudo recibir uno o más paquetes desde el nodo CAN indicado.

Solución:

Compruebe la conexión y la alimentación del dispositivo CAN indicado. Compruebe el cableado de toda la línea CAN.

Compruebe la conexión de la línea al control numérico. Compruebe la coherencia entre las configuraciones de protocolo del dispositivo CAN indicado con respecto a las configuraciones del transmisor que está en el control numérico (velocidad en baudios, dirección, configuraciones específicas del protocolo aprobado).

2765 Nodo número: Inicializado

Causa:

El nodo CAN indicado ha sido conectado al bus de campo. Luego ha sido inicializado de manera correcta.

2766 Condición de falla en la interfaz CAN

Causa:

Se señala una anomalía en la alimentación interna del dispositivo de interfaz CAN.

Solución:

Póngase en contacto con el constructor.

2767 Error pérdida estado CANopen

Causa:

Debido a un problema grave, el transmisor CAN ya no está en estado operativo (Operational).

Solución:

Póngase en contacto con el constructor.

2768 Nodo número: Error de falta recepción PDO

Causa:

El control numérico no ha recibido el PDO, que estaba esperando del nodo CAN.

Solución:

Compruebe:

- la fuente de energía del nodo.
- que el nodo no está en modo pre-Operational.
- los datos del PDO y del bus CAN, configurados en Albatros.

2769 Nodo número: Error de recepción de un nodo no configurado

Causa:

En la red CAN se ha descubierto la existencia de un nodo no descrito en la configuración hardware de Albatros.

Solución:

Compruebe la dirección hardware del nodo y la dirección del nodo configurado en la configuración hardware. Compruebe que el nodo haya sido declarado en la configuración hardware, si no, es necesario agregarlo.

2770 Nodo número: Configuración incorrecta

Causa:

La descripción de los datos de RPDO y TPDO es incorrecta.

Solución:

Corrija la descripción de los datos de los PDOs de transmisión y de recepción en configuración Hardware.

2771 Nodo número: Error durante la comunicación SDO

Causa:

El nodo CAN indicado no contesta en una comunicación asíncrona (SDO).

Solución:

Compruebe el estado de enlace del nodo. Si el problema persiste, póngase en contacto con el constructor.

2772 Tiempo de expiración en el ciclo CAN de interrogación de los nodos

Causa:

Error de tiempo de expiración en el ciclo CAN de interrogación de los nodos.

Solución:

Modifique el valor del tiempo del muestreo configurado en la configuración Hardware.

3073 Nodo número: Error de Emergencia n. NúmeroError

Causa:

El dispositivo CANopen buscó una situación de error del nodo, especificada por el código visualizado. El código de error es un número hexadecimal.

Estas son situaciones de error que pertenecen al nodo individual y están conformes al estándar CiA DS 301 – protocolo EMERGENCY.

Solución:

Vease la documentación sobre el nodo.

3074 Nodo número: Error CAN genérico n. NúmeroError

Causa:

Ha ocurrido un error interno en el nodo indicado. El código de error es un número hexadecimal.

Solución:

Póngase en contacto con el constructor.

3088 Tarjeta CAN número: Nodo NúmeroNodo: Error durante la comunicación SDO n. NúmeroError – descripción

Causa:

En una instrucción READDICTIONARY o WRITEDICTIONARY no se pudo realizar ninguna de las peticiones de lectura/escritura SDO. Las instrucciones erróneas pueden deberse, por ejemplo, a la solicitud de lectura en un objeto CANOpen no implementado en el dispositivo de referencia; o bien, puede ser vinculado a la escritura, en un registro CANOpen, de un dato no compatible con el tipo de objeto (por ejemplo, intento de escritura de una cadena en un objeto de tipo Integer). El código de error suministrado está conforme a las especificaciones DS402 y más allá del código numérico se suministrará también la descripción textual. El código de error es un número hexadecimal.

Solución:

Compruebe que los parámetros de Velocidad de Baudios, Tiempo de muestreo, etc., configurados en configuración hardware y los parámetros de eventuales instrucciones [READDICTIONARY](#) y/o [WRITEDICTIONARY](#) en el código GPL.

6.2.5 Errores ocasionados por la gestión bus EtherCAT

3329 Error durante la inicialización del socket de comunicación

Causa:

El firmware no pudo comunicar con la tarjeta de red.

Solución:

Si la tarjeta ha sido configurada en un sistema RTX, compruebe que los archivos .ini en la subcarpeta FW de Albatros estén escritos de manera correcta. Para comprobar la sintaxis de los archivos, véase el manual de instalación de RTX en Albatros (InstallationRTXGuide.pdf).

3330 Error durante la análisis de la red EtherCAT

Causa:

Durante el escaneo preliminar de la red EtherCAT, el máster no recibió respuesta de algunos o de todos los esclavos configurados o bien la configuración no corresponde a la efectiva red EtherCAT presente en el campo.

Solución:

Compruebe el cableado entre máster EtherCAT y esclavo. Compruebe las descripciones de los dispositivos en configuración hardware. Para encontrar el error, se puede utilizar como ayuda la ventana de Diagnóstico Hardware. Aquí se visualizan los nodos presentes y, si configurados de manera incorrecta, aparte del dispositivo encontrado, se visualiza el nombre del dispositivo esperado.

3331 Error en la configuración del buzón de correo de transmisión

Causa:

El nodo EtherCAT no respondió al comando dado por el Máster. Causas posibles: comunicación ausente, nodo averiado...

Solución:

Compruebe cableado y funcionamiento remoto.

3332 Error en la configuración del buzón de correo de recepción

Causa:

El nodo EtherCAT no respondió al comando dado por Máster. Causas posibles: comunicación ausente, nodo averiado...

Solución:

Compruebe cableado y funcionamiento remoto.

3333 Tarjeta EtherCAT número: Error en el tipo de expansiones del nodo NúmeroNodo

Causa:

El tipo de expansiones configuradas en un nodo EtherCAT en configuración hardware no corresponde al tipo de expansiones realmente presentes. (Por ejemplo, en configuración hardware se definió un TRS-CAT con una expansión TRS-IO-E, mientras que en el sistema hay un TRS-CAT con una expansión TRS-AN-E).

Solución:

Compruebe que los dispositivos descritos en configuración hardware correspondan a los presentes.

3334 Error al configurar los PDOs

Causa:

El nodo EtherCAT, para el cual se ha intentado de configurar los PDOs, no está presente en la red o está estropeado.

Solución:

Compruebe que la configuración de la red EtherCAT descrita en configuración de Albatros corresponda a la configuración física de la red.

3335 Nodo NúmeroNodo en alarma (NúmeroError)

Causa:

El nodo indicado está en una situación de alarma

Solución:

Compruebe el código de alarma en la tabla siguiente:

Código alarma	Descripción
0x0001	Unspecified error (Error no especificado)
0x0002	No memory (Ninguna memoria)
0x0011	Invalid requested state change (Estado de cambio solicitado inválido)
0x0012	Unknown requested state (Estado solicitado desconocido)
0x0013	Bootstrap not supported (Arranque no soportado)
0x0014	No valid firmware (Firmware inválido)
0x0015	Invalid mailbox configuration (Configuración buzón de correo inválida)
0x0016	Invalid mailbox configuration (Configuración buzón de correo inválida)
0x0017	Invalid sync manager configuration (Configuración sync manager inválida)
0x0018	No valid inputs available (Ninguna entrada válida disponible)
0x0019	No valid outputs (Salidas inválidas)
0x001A	Synchronization error (Sincronización error)
0x001B	Sync manager watchdog (Controlador de secuencia sync manager)
0x001C	Invalid Sync Manager Types (Tipos sync manager inválidos)
0x001D	Invalid Output Configuration (Configuración salida inválida)
0x001E	Invalid Input Configuration (Configuración entrada inválida)
0x001F	Invalid Watchdog Configuration (Configuración controlador de secuencia inválida)
0x0020	Slave needs cold start (Slave necesita inicio frío)
0x0021	Slave needs INIT (Slave necesita INIT)
0x0022	Slave needs PREOP (Slave necesita PREOP)
0x0023	Slave needs SAFEOP (Slave necesita SAFEOP)
0x0024	Invalid input mapping (Asignación entrada inválida)
0x0025	Invalid output mapping (Asignación salida inválida)
0x0026	Inconsistent settings (Configuraciones inconsistentes)
0x0027	Free-Run not supported (Free-Run no soportado)
0x0028	Synchronization not supported (Sincronización no soportada)

0x0029	Free-Run needs 3 buffer mode (Free-Run necesita 3 modos búfer)
0x002A	Background watchdog (Controlador de secuencia de fondo)
0x002B	No valid inputs and outputs (Salidas y entradas inválidas)
0x002C	Fatal Sync error (Error sync irrecuperable)
0x002D	No Sync error (Ningún error sync)
0x0030	Invalid DC SYNCH Configuration (Configuración DC SYNCH inválida)
0x0031	Invalid DC Latch Configuration (Configuración DC Bloqueo temporal inválida)
0x0032	PLL Error (Error PLL)
0x0033	Invalid DC IO Error (Error DC IO inválido)
0x0034	Invalid DC Timeout Error (Error DC Tiempo de expiración inválido)
0x0035	DC Invalid Sync Cycle Time (DC Sync tiempo de ciclo inválido)
0x0036	DC Sync0 Cycle Time (DC Sync0 tiempo de ciclo)
0x0037	DC Sync1 Cycle Time (DC Sync1 tiempo de ciclo)
0x0041	MBX_AOE
0x0042	MBX_EOE
0x0043	MBX_COE
0x0044	MBX_FOE
0x0045	MBX_SOE
0x004F	MBX_VOE
0x0050	EEPROM no access (No acceso EEPROM)
0x0051	EEPROM error (Error EEPROM)
0x0060	Slave restarted locally (Slave reiniciado localmente)

3336 Tarjeta EtherCAT: El número de expansiones del nodo NúmeroNodo es erróneo

Causa:

El número de expansiones configuradas en un nodo EtherCAT en Albatros no corresponde al número de expansiones presentes. (Por ejemplo, en configuración hardware se definió un TRS-CAT con dos expansiones TRS-IO-E, mientras que en el sistema hay una sola una expansión).

Solución:

Compruebe que los dispositivos descritos en configuración hardware correspondan a los presentes.

3337 Tarjeta EtherCAT: Nodo NúmeroNodo: Desconectado

Causa:

El nodo EtherCAT no devolvió ninguna respuesta al control. El nodo podría estar desconectado de la red o podría estar apagado.

Solución:

Compruebe el cableo y que el nodo tenga alimentación.

3338 Tarjeta EtherCAT: Nodo NúmeroNodo: Revinculado

Causa:

El nodo EtherCAT indicado ha sido conectado de nuevo a la red y ha empezado de nuevo a responder a las solicitudes del control.

3340 Tarjeta EtherCAT: El nodo NúmeroNodo no respondió a la solicitud (Código)

Causa:

El nodo EtherCAT, consultado a través del servicio SDO, no respondió a la solicitud. Además, si el dispositivo ha proporcionado un código de cancelación, ese sale en el mensaje de error indicado (código). Es posible que este error aparezca durante la ejecución de las instrucciones [SETPZERO](#), [SETPFLY](#) y [FASTREAD](#) en ejes EtherCAT, ya que estas utilizan la comunicación SDO para configurar el accionamiento. En este caso específico, el mensaje de error, además de los códigos de cancelación estándar, puede contener los códigos siguientes:

- 1: Tiempo de espera caducado: el nodo no respondió dentro de 250 ms.

- 00000005: índice interno en el buzón de correo incorrecto.
- 00000006: dato recibido no conforme a la solicitud.

Solución:

Póngase en contacto con el constructor.

3341 Tarjeta EtherCAT: El nodo NúmeroNodo no existe**Causa:**

A un nodo inexistente se le intentó hacerle ejecutar un comando.

Solución:

Compruebe el número del nodo en la instrucción GPL que ha creado el error. Conecte el nodo.

3342 Cable desconectado**Causa:**

El cable EtherCAT no está conectado al control.

Solución:

Controle que el cable esté correctamente conectado al control y que no esté dañado.

3343 Tarjeta EtherCAT número: Nodo NúmeroNodo no pasa al estado SAFE-OPERATIONAL (Código)**Causa:**

El nodo EtherCAT indicado no pasó al estado OPERATIONAL.

Solución:

Póngase en contacto con el constructor y le informe sobre el número de error indicado (código), que representa el código ALstatuscode.

3344 Tarjeta EtherCAT número: Nodo NúmeroNodo no pasa al estado OPERATIONAL (Código)**Causa:**

El nodo EtherCAT indicado no pasó al estado OPERATIONAL.

Solución:

Póngase en contacto con el constructor e le informe sobre el número de error indicado (código), que representa el código ALstatuscode.

3345 Tarjeta EtherCAT: Comunicación inestable**Causa:**

La comunicación a la red EtherCAT es inestable debido a interferencias, cables o nodos defectuosos.

4400 Demasiados ejes activos en FASTREAD (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se intentó utilizar una instrucción FASTREAD con un número de ejes más grande que lo permitido. Es necesario tener en cuenta que los ejes equipados con codificador agregado equivalen a un doble eje (véase la instrucción [SWITCHENC](#)).

Solución:

Reduzca el número de ejes con codificador agregado utilizados.

6.2.6 Errores ocasionados por la inicialización**769 Configuración software incorrecta****Causa:**

La configuración de hardware del módulo remoto no corresponde a la configuración software especificada en la configuración del sistema.

Solución:

Verifique la congruencia entre los parámetros hardware del módulo remoto y los parámetros software.

770 Número de IRQ configurado incorrecto

Causa:

No ha sido establecido correctamente el IRQ de la tarjeta de ejes en la configuración de Módulo. Normalmente se está ante la presencia de un conflicto hardware con otros periféricos presentes en el sistema.

Solución:

Verifique en las definiciones del BIOS de la tarjeta madre que el IRQ utilizado por la tarjeta de ejes esté reservado "Legacy ISA". Verifique que no haya otros periféricos que utilicen el mismo IRQ asignado a la tarjeta de ejes. Si fuera posible, modifique el IRQ del periférico que entra en conflicto con la tarjeta de ejes, sino modifique el IRQ de la tarjeta de ejes.

772 Error en la lectura del área de memoria intermedia durante la inicialización

Causa:

Se ha verificado un error durante las pruebas realizadas en fase de inicialización de la tarjeta de ejes. En particular ha fallado la prueba de la RAM intermedia (Dallas). Puede depender de una mala configuración de las direcciones de E/S e IRQ de la tarjeta o de un conflicto con otros periféricos presentes en el sistema. También puede ser la consecuencia de un daño de la tarjeta de ejes.

Solución:

Verifique la configuración hardware. Los técnicos calificados pueden efectuar una prueba Hardware de la RAM del micro-controlador i296. Note que la prueba Hardware de la RAM conlleva el borrado de todos los datos memorizados en la misma. En la RAM intermedia se conservan los valores de algunos dispositivos, por ejemplo, los contadores, temporizadores y desplazamientos de los DAC de los ejes. Antes de efectuar la prueba, se deben guardar estos valores. Si el problema persiste, póngase en contacto con el constructor.

773 Alcanzado el máximo número de ejes en configuración

Causa:

Se ha tratado de configurar un número de ejes superior al número máximo admitido.

Solución:

Reduzca el número de ejes configurado. Para más información, póngase en contacto con el constructor.

774 El tiempo real de los ejes no está en ejecución

Causa:

El firmware de gestión de los ejes ha sido inicializado pero no funciona correctamente. Normalmente se está ante la presencia de un conflicto hardware con otros periféricos presentes en el sistema.

Solución:

Verifique que no haya conflictos con otros periféricos, modifique la configuración de los periféricos que causan el conflicto o los quite del sistema.

775 Tiempo insuficiente para la ejecución GPL

Causa:

La ejecución de una tarea de tiempo real ocupa demasiado tiempo ciclo. Este error es generado cuando una tarea de tiempo real no termina dentro del tiempo de comienzo del tiempo real de los ejes siguientes (por ejemplo, ha sido creado un ciclo infinito).

Solución:

Modifique el código GPL de manera tal de reducir la duración de la tarea de tiempo real.

776 Tiempo excesivo de ejecución del tiempo real

Causa:

La ejecución de una tarea de tiempo real ocupa demasiado tiempo ciclo. El tiempo de ejecución resulta un poco superior al límite máximo consentido.

Solución:

Modifique el código GPL de manera tal de reducir la duración de la tarea de tiempo real.

777 Guardián expirado

Causa:

El firmware está bloqueado.

Solución:

Póngase en contacto con el constructor.

778 El código de Main del firmware está bloqueado**Causa:**

El firmware se ha bloqueado por más de 5 tiempo reales.

Solución:

Póngase en contacto con el constructor.

1025 Tarjeta NúmeroTarjeta: No responde al comando**Causa:**

Durante la inicialización se ha detectado la presencia de una tarjeta ejes, pero esta no responde correctamente a los comandos.

Solución:

Los técnicos calificados pueden efectuar una prueba Hardware de la tarjeta de ejes. Si el problema persiste, póngase en contacto con el constructor.

1026 Tarjeta NúmeroTarjeta: Error al enviar el firmware a la tarjeta de ejes**Causa:**

No fue posible enviar el firmware a la tarjeta.

Solución:

Póngase en contacto con el constructor.

1028 Tarjeta NúmeroTarjeta: Firmware no presente**Causa:**

Los firmware presentes en la tarjeta no son correctos para el tipo de tarjeta detectado.

Solución:

Enviar el firmware correcto.

1029 Tarjeta NúmeroTarjeta: Main bloqueado**Causa:**

El firmware de la tarjeta no empezó su ejecución.

Solución:

Póngase en contacto con el constructor

1031 Tarjeta NúmeroTarjeta: Error en inicialización**Causa:**

Ha ocurrido un error durante el procedimiento de inicialización de la tarjeta de ejes.

Solución:

Controle y resuelva las causas de los errores de sistema que se produjeron antes de la ocurrencia del error corriente. Después inicialice el sistema.

1032 Tarjeta NúmeroTarjeta: La prueba de la dual port memory ha fracasado**Causa:**

Se ha verificado un error durante las pruebas realizadas en fase de inicialización de la tarjeta de ejes. En particular ha fracasado la inicialización de la dual port memory. En general esto se debe a un conflicto hardware con otros periféricos presentes en el sistema más también puede depender de un daño de la tarjeta.

Solución:

Verifique la configuración de la tarjeta, verifique que no haya conflictos con otros periféricos. Si se utiliza un módulo remoto, retransmita el firmware al módulo. Los técnicos calificados pueden efectuar una prueba Hardware de la dual port memory. Si el problema persiste, póngase en contacto con el constructor.

1033 Tarjeta NúmeroTarjeta: El código de arranque del firmware no está en ejecución**Causa:**

El firmware de puesta en servicio de la tarjeta fue inicializado pero no funciona correctamente. Normalmente se está ante un conflicto hardware con otros periféricos presentes en el sistema.

Solución:

Verifique la configuración de la tarjeta, verifique que no haya conflictos con otros periféricos. Si se utiliza un módulo remoto, retransmita el firmware al módulo. Los técnicos calificados pueden efectuar una prueba Hardware de la dual port memory que hay en la tarjeta. Si el problema persiste, póngase en contacto con el constructor.

1035 Tarjeta NúmeroTarjeta: No presente**Causa:**

Se ha verificado un error durante las pruebas realizadas en fase de inicialización de la tarjeta de ejes. En particular no ha sido detectada la tarjeta.

Solución:

Verifique que la tarjeta esté presente en el sistema y que no esté dañada. Los técnicos calificados pueden efectuar una prueba Hardware de la tarjeta. Si el problema persiste, póngase en contacto con el constructor.

1037 Tarjeta NúmeroTarjeta: La apertura de la dual port memory ha fracasado**Causa:**

La apertura de la dual port memory de la tarjeta ha fracasado.

Solución:

Los técnicos calificados pueden efectuar una prueba Hardware de la tarjeta de ejes. Póngase en contacto con el constructor.

1039 Tarjeta NúmeroTarjeta: Guardián expirado**Causa:**

El firmware de la tarjeta ejes NúmeroTarjeta está bloqueado.

Solución:

Póngase en contacto con el constructor.

1040 Tarjeta NúmeroTarjeta: Error de alimentación +24 Vcc**Causa:**

La alimentación de campo (+24 Vcc) de las salidas no está presente o no funciona correctamente.

Solución:

Verifique el funcionamiento de la alimentación +24 Vcc de campo.

1047 Tarjeta NúmeroTarjeta: Configuración de software no permitida**Causa:**

El dispositivo recibió una configuración que no es compatible con el presente hardware o habilitado. Por ejemplo, se requiere la configuración de un eje no habilitado o no presente en el dispositivo.

Solución:

Verifique la congruencia entre los parámetros hardware y software de la tarjeta.

1052 Tarjeta NúmeroTarjeta: Se está ejecutando el código de arranque**Causa:**

La tarjeta está configurada en modalidad provisional y está ejecutando el código de arranque.

Solución:

Póngase en contacto con el constructor.

1053 Tarjeta NúmeroTarjeta: Guardián ejes expirado**Causa:**

Ha ocurrido un error grave durante la ejecución del firmware de la tarjeta del control ejes. Los ejes están desactivados y la posible señal de SYSOK se baja. No restauren el sistema.

Solución:

Póngase en contacto con el constructor.

1055 Guardián expirado en la tarjeta NúmeroTarjeta

Causa:

El firmware de la tarjeta NúmeroTarjeta está bloqueado.

Solución:

Póngase en contacto con el constructor.

1056 Tarjeta NúmeroTarjeta: Error de alimentación interfaz CAN

Causa:

En el dispositivo de transmisión en la línea CanBus en uso en la tarjeta indicada ha sido descubierto un error de alimentación. Esto puede depender de un cortocircuito, de un error de cableado del bus o de un daño de la tarjeta.

Solución:

Compruebe el cableado de toda la línea Can. Compruebe la conexión de la línea con el control numérico. Se deshaga del eventual cortocircuito. En el caso de que la comunicación no se restaure, póngase en contacto con el constructor.

1057 Tarjeta NúmeroTarjeta: Error interno n. NúmeroError

Causa:

Error en el hardware del nodo.

Solución:

Póngase en contacto con el constructor.

6.2.7 Errores ocasionados por la gestión de la memoria

1281 Error en la asignación de memoria en el área de heap

Causa:

La memoria RAM disponible es menor a la solicitada, por ejemplo, por una matriz global.

Solución:

Reduzca el tamaño de las variables globales asignadas en la RAM.

1286 Error al gestionar el heap

Causa:

Error en la gestión de la memoria por el firmware.

Solución:

Póngase en contacto con el constructor.

1287 Demasiadas desasignaciones de memoria por el heap

Causa:

Error en la gestión de la memoria por el firmware.

Solución:

Póngase en contacto con el constructor.

1289 Error en la creación de las variables globales

Causa:

Ha sido definido un número demasiado alto de [variables globales](#), o se han definido matrices globales de dimensiones demasiado grandes.

Solución:

Reduzca el número de variables globales o la dimensión de las matrices.

1290 Error en la dimensión de las variables no volátiles

Causa:

Se ha definido un número demasiado alto de variables no volátiles, o se han definido matrices no volátiles de dimensiones demasiado grandes.

Solución:

Reduzca el número de variables no volátiles o la dimensión de las matrices no volátiles.

1291 Error en la dimensión de las variables de sola lectura**Causa:**

Se ha definido un número demasiado alto de variables de sola lectura, o se han definido matrices de sola lectura de dimensiones demasiado grandes.

Solución:

Reduzca el número de variables de sola lectura o la dimensión de las matrices de sola lectura.

6.2.8 Errores ocasionados por fallos**1559 Seguimiento de punto de interrupción****Causa:**

Error grave del firmware.

Solución:

Póngase en contacto con el constructor.

1569 Código operativo del microprocesador no válido**Causa:**

El microprocesador ha encontrado una instrucción desconocida. Se puede deber tanto a problemas hardware de la PC como a un daño de los ficheros que contienen el firmware de Albatros.

Solución:

En el caso de módulo local, verifique que los ficheros no estén dañados y, en su caso, reinstale Albatros. Para los módulos Clipper, efectuar una actualización del firmware. Efectuar una prueba del hardware de la PC en particular de la RAM. Si el problema persiste, póngase en contacto con el constructor.

1586 Valor INTEGER dividido por cero**Causa:**

Se ha intentado dividir un INTEGER por cero.

Solución:

Compruebe en las funciones GPL que todas las divisiones estén correctas.

1600 Desbordamiento en el resultado de una operación de coma flotante**Causa:**

El resultado de una operación entre FLOAT sobrepasa las capacidades del destinatario:

$\pm 3,402823E+38$	para los floats
$\pm 1,79769313486231E+308$	para los dobles

Solución:

En las funciones GPL, verifique si los cálculos con coma flotante estén correctos.

1601 Subdesbordamiento en el resultado de una operación de coma flotante**Causa:**

El resultado de una operación entre FLOAT es inferior a las capacidades del destinatario:

$\pm 1,401298E-45$	para los floats
$\pm 4,94065645841247E-324$	para los dobles

Solución:

En las funciones GPL, verifique si los cálculos con coma flotante están correctos.

1602 Argumento no válido para una operación de coma flotante**Causa:**

En una operación float se ha utilizado un operando no de tipo float.

Solución:

Compruebe en las funciones GPL que los cálculos float estén correctos.

1603 Valor en coma flotante dividido por cero

Causa:

Se ha intentado dividir un float o un doble por cero. Está generado también en el caso en el cual se ejecuta un logaritmo de cero.

Solución:

Compruebe en las funciones GPL que todas las divisiones estén correctas.

1604 Resultado incorrecto de una operación de coma flotante

Causa:

El resultado de una operación entre float no está correcto.

Solución:

Compruebe en las funciones GPL que los cálculos float estén correctos.

1605 Se ha utilizado un valor en coma flotante incorrecto

Causa:

Se ha utilizado un valor con coma flotante más pequeño que el mínimo representable:

$\pm 1,401298E-45$	para los floats
$\pm 4,94065645841247E-324$	para los dobles

Solución:

Compruebe en las funciones GPL que los cálculos estén correctos.

1728 Se ha intentado acceder a una dirección no válida

Causa:

El programa ha ejecutado un acceso a una zona de memoria no válida.

Solución:

Compruebe la congruencia de las variables globales/locales, si el problema sigue, señale la anomalía.

1735 Excepción genérica

Causa:

Se ha verificado una excepción no reconocida.

Solución:

Póngase en contacto con el constructor.

1736 Datos no alineados

Causa:

Error grave del firmware.

Solución:

Póngase en contacto con el constructor.

1801 Alarma temperatura

Causa:

La temperatura de la CPU del control ha superado los límites máximos consentidos.

Solución:

Compruebe que no estén presentes problemas de ventilación o causas de sobrecalentamiento. Si el problema ocurre nuevamente, póngase en contacto con el constructor.

1802 Alarma ventilador

Causa:

El ventilador de la CPU del control no funciona de manera correcta. El problema puede causar en poco tiempo el sobrecalentamiento de la CPU.

Solución:

Póngase en contacto con el constructor.

1803 La frecuencia de la CPU es inestable

Causa:

La frecuencia de trabajo de la CPU no es estable.

Solución:

Póngase en contacto con el constructor.

6.2.9 Errores ocasionados por funciones GPL

4097 El dispositivo TipoDispositivo NombreDispositivo no está configurado

Causa:

Una instrucción GPL ha utilizado un dispositivo no configurado, o bien que no tiene un vínculo Virtual-Físico. Puede ser ocasionado por todas las instrucciones a las que se pasa un dispositivo por parámetro.

Solución:

Compruebe en las configuraciones del control que todos los dispositivos utilizados por las funciones tengan un vínculo Virtual-Físico.

A continuación, retransmita las configuraciones a la tarjeta.

4098 La variable global NombreVariable no existe

Causa:

A una instrucción GPL se ha pasado como tema una variable global que no se ha definido. Generalmente ocurre cuando el control no ha sido inicializado correctamente.

Solución:

Vuelva a compilar todo el código GPL y vuelva a inicializar el control.

4099 Función NombreFunción no encontrada

Causa:

Se ha efectuado una llamada a una función que no existe. Puede ocurrir cuando, luego de una modificación del código GPL, no se ha ejecutado una inicialización del control.

Solución:

Vuelva a compilar todo el código GPL y vuelva a inicializar el control.

4101 Gestión incoherente del eje NombreEje

Causa:

Se ha producido un paso de estado ilegal en un eje. Para los pasos de estado, consulte la documentación correspondiente.

El error puede ser ocasionado por todas las instrucciones que controlan los ejes, normalmente se produce en estos casos:

- si se intenta interpolar, coordinar con un eje que está efectuando un movimiento punto a punto (o viceversa).
- si se ejecuta una instrucción Chain, SetPFly, SetPZero en un eje que está en modo transparente.
- si se intenta interpolar, bobinar, coordinar en un eje que es esclavo de otro.

Solución:

Controle que todos los movimientos de ejes hayan terminado con una instrucción de espera en cota, sobre todo si los ejes alternan movimientos de vario tipo (punto a punto, interpolación, etc.).

4105 Instrucción no ejecutable en el eje NombreEje

Causa:

Se ha tratado de ejecutar una instrucción en un eje que no la soporta. Por ejemplo, una instrucción de interpolación en un eje paso a paso.

Solución:

Corrija el código GPL.

4106 El remoto referido al eje paso a paso NombreEje no está vinculado

Causa:

Se ha tratado de operar en un eje paso a paso que no está conectado al control.

Solución:

Verifique la conexión del remoto que controla al eje.

4107 Instrucción SYSOK con argumentos incorrectos

Causa:

Se ha ejecutado una instrucción SYSOK con argumentos incorrectos. Se verifica si una o varias de las salidas digitales pasadas como temas a la instrucción no están configuradas correctamente.

Solución:

Verifique el código GPL y la configuración Virtual-Físico.

4108 El eje NombreEje: Cota final más allá de los límites software

Causa:

Se ha tratado de mover un eje más allá de los límites fijados en la configuración o por el código GPL.

Solución:

Corrija el programa de elaboración que ha causado el error. En su caso, corrija el código GPL o la configuración del eje.

4110 Velocidad incorrecta

Causa:

Se ha tratado de asignarle a un eje una velocidad nula o negativa.

Solución:

Corrija el código GPL.

4111 Aceleración eje NombreEje negativa

Causa:

Se ha tratado de asignarle a un eje una aceleración negativa.

Solución:

Corrija el código GPL.

4112 Deceleración eje NombreEje negativa

Causa:

Se ha tratado de asignarle a un eje una desaceleración negativa.

Solución:

Corrija el código GPL.

4114 Eje NombreEje: Restablecimiento a entrada rápida no realizado

Causa:

La puesta a cero de la medida en entrada rápida (punto de consigna al vuelo) no ha sido completada de manera correcta. Este procedimiento permite poner a cero la medida de un eje en movimiento en el instante en que la entrada rápida correspondiente cambia de estado. Si el eje termina el movimiento en curso, sin que se haya verificado la conmutación de la entrada, es generado el error de sistema. La causa puede ser una mala programación de los parámetros de movimiento del eje o un problema de cableo de la entrada rápida.

Solución:

Verifique el código GPL que implementa el punto de consigna al vuelo, verifique el cableo de la entrada rápida.

4115 Eje NombreEje: marca de cero no encontrada

Causa:

La puesta a cero del codificador no ha sido completada correctamente. Este procedimiento permite poner a cero la cota de un eje en movimiento en el instante en que es detectada la muesca de cero del codificador. Si el eje alcanza la cota de búsqueda de muesca sin que esta esté detectada, entonces es generado el error de sistema. La causa puede ser una mala programación de los parámetros de movimiento del eje o un problema de cableo de la señal de muesca (fase C del conector eje).

Solución:

Verifique el código GPL que implementa el punto de consigna en la muesca, verifique el cableo del eje.

4353 Código operativo instrucción desconocida (Función:NombreFunción línea:NúmeroLínea)**Causa:**

En la ejecución de una función GPL se ha producido una instrucción ilegal. En general es síntoma de daño de los ficheros que contienen el código GPL compilado. Si se ha actualizado el software del control, averigüe que se haya vuelto a compilar el código GPL. La versión anterior del software del control podría, de hecho, contener las instrucciones no más soportada por la versión nueva.

Solución:

Vuelva a compilar todo el código GPL e inicialice el control. Si el problema persiste, póngase en contacto con el constructor.

4354 Operación matemática incorrecta (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Una instrucción GPL ha tratado de ejecutar una operación matemática equivocada, por ejemplo, una división por cero. O bien algunos datos ingresados en las instrucciones GPL son incongruentes. Típicamente este error lo generan las instrucciones de movimiento en interpolación puesto que es la parte del Firmware que ejecuta más cálculos matemáticos.

Solución:

Verifique que los parámetros pasados a las instrucciones de interpolación estén correctos. Si el problema persiste, informe del mal funcionamiento al constructor.

4355 Dirección incorrecta de matriz o vector (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Una instrucción GPL ha intentado acceder a un elemento de vector o de matriz que sobrepasa el tamaño máximo. Por ejemplo, ha intentado acceder al elemento 10 de un vector de 5 elementos. Puede ser ocasionado por todas las instrucciones que aceptan como parámetro un vector o una matriz.

Solución:

Compruebe que todos los índices de matriz y vector pasados a las instrucciones cumplan con el tamaño del vector y de la matriz.

4356 Instrucción RET no invocada de CALL (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha ejecutado una instrucción RET sin que en la memoria de retención estuviese la correspondiente dirección de retorno. La causa más común es haber declarado un sub-procedimiento antes de la instrucción FRET de salida de la función sin haberla protegido con un GOTO que impida la ejecución accidental. O bien se ha efectuado un salto no deseado dentro de un sub-procedimiento.

Solución:

Controle el flujo del programa GPL. Preferentemente sitúe los sub-procedimientos al final del cuerpo de las funciones (después de la instrucción FRET).

4357 Variable local inexistente (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Una instrucción GPL ha intentado acceder a una variable local que no se ha asignado.

Solución:

Compile de nuevo y retransmita todas las funciones a la tarjeta. Si el problema persiste, señale el mal funcionamiento.

4358 Etiqueta de salto inexistente (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Una instrucción GPL ha efectuado un salto a una etiqueta de salto inexistente. Puede ser ocasionado por GOTO, CALL, FCALL, todas las IF.

Solución:

Compile de nuevo y retransmita todas las funciones a la tarjeta. Si el problema persiste, señale el mal funcionamiento.

4359 Argumento macro incorrecto (Función:NombreFunción línea:NúmeroLínea)

Causa:

A una instrucción GPL se han pasado temas no válidos. Puede ser ocasionado por todas las instrucciones. De todas formas, en la mayoría de los casos, el sistema GPL intenta corregir automáticamente dicha situación, ejecutando conversiones automáticas de tipo (cast), que sin embargo suponen pérdida de tiempo. El error se produce cuando dichas conversiones no son posibles y normalmente en los siguientes casos:

- instrucciones que intervienen en dispositivos específicos (SETTIMER, SETCOUNTER) a las que se pasa uno de tipo diferente.
- instrucciones que intervienen en bits a los que se pasa un número en coma flotante (AND, OR, etc.).
- instrucciones que intervienen en matrices o vectores a los que se pasa una variable (SORT, MOVEMAT, etc.).
- instrucciones que intervienen en cadenas a las que no se pasan cadenas.

El error se produce aunque se intenta ejecutar una instrucción en una tarjeta que no gestiona esta instrucción. (Por ejemplo, una instrucción [SENDPDO](#) o una instrucción [RECEIVEPDO](#) en una tarjeta que difiere de TMSCan o TMSCan+).

Solución:

Corrija el código GPL.

4360 Error en asignación de memoria durante la ejecución (Función:NombreFunción línea:NúmeroLínea)

Causa:

La función GPL ha intentado asignar una zona de memoria para usos internos, pero no ha encontrado memoria disponible.

Es posible que la señalización indique una situación normal, debido, por ejemplo, a un excesivo número de tareas que se están ejecutando simultáneamente o bien a variables globales dimensionadas de manera exagerada.

Solución:

Controle el dimensionamiento de las variables globales e intente reducir su tamaño. Verifique que no se estén ejecutando demasiadas tareas simultáneamente y, en ese caso, las reduzca.

4361 Demasiadas tareas activas (Función:NombreFunción línea:NúmeroLínea)

Causa:

Se han intentado ejecutar más de 256 tareas contemporáneamente.

Solución:

Reduzca el número de tareas activas simultáneamente.

4362 Formato matriz incorrecto (Función:NombreFunción línea:NúmeroLínea)

Causa:

Una instrucción que interviene en matrices ha encontrado un formato no válido. Las instrucciones que pueden ocasionar este error de sistema son las siguientes:

- MOVEMAT en caso de que el formato de la matriz fuente no corresponda con el formato de la matriz de destino.
- CLEAR si se quiere restablecer una línea de matriz que no existe.
- GETAXIS en caso de que el formato de la matriz, pasado por parámetro, no corresponda con aquel que la instrucción se espera (consulte la documentación del lenguaje GPL).

Solución:

Controle dentro de la tarea que ha ocasionado el error las instrucciones mencionadas aquí a lado. Controle en especial modo que las matrices pasadas a la MOVEMAT tengan el mismo número de columnas del mismo tipo, y que la matriz pasada a la GETAXIS tenga el formato correcto.

4363 Demasiadas instrucciones ONINPUT activas (Función:NombreFunción línea:NúmeroLínea)

Causa:

Se han activado más de 128 instrucciones ONINPUT.

Solución:

Reduzca el número de las ONINPUT.

4364 Eje ya ocupado en una referencia local (Función:NombreFunción línea:NúmeroLínea)**Causa:**

El error se refiere a la activación de las ternas de ejes con traslación rotatoria para ejecutar en más ejes cartesianos.

Se ha intentado ejecutar una SETRIFLOC pasando a la instrucción un eje que ya estaba ocupado en una terna de ejes de referencia. Se produce también en caso de que se ejecute una RESRIFLOC en un eje que no estaba ocupado en ninguna terna de ejes. El último caso es que existan más ternas de referencia disponibles (como máximo pueden ser 32).

Solución:

Controle que las ternas pasadas mediante la SETRIFLOC no tengan ejes en común.

Controle las RESRIFLOC.

Controle también que estén las instrucciones de espera en cota antes de la RESRIFLOC.

Note que mientras no haya finalizado la interpolación, la RESRIFLOC no se ejecuta.

4365 Instrucción ONINPUT activada en la misma entrada (Función:NombreFunción línea:NúmeroLínea)**Causa:**

A una instrucción ONINPUT se ha pasado la misma entrada por más de una vez.

Solución:

Controle que no haya dos ONINPUT a las que se pase como parámetro la misma entrada.

4366 Demasiadas instrucciones ONFLAG activas (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se han activado más de 128 instrucciones ONFLAG.

Solución:

Reduzca el número de las ONFLAG.

4367 Instrucción ONFLAG activada en el mismo indicador (Función:NombreFunción línea:NúmeroLínea)**Causa:**

A una instrucción ONFLAG se ha pasado el mismo indicador por más de una vez.

Solución:

Controle que no haya dos ONFLAG a los que se pase como parámetro el mismo indicador.

4368 Intento de escritura de una variable de tipo SolaLectura (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha intentado acceder en escritura a una variable de tipo sola lectura.

Las variables de tipo sola lectura siempre son globales y se encuentran en la flash del mando. Se señalan como "estáticas" en el editor de las variables globales. Si se intenta escribir en una de estas variables globales se produce dicho error de sistema.

Además, el error se produce en caso de que se utilicen variables que se encuentran en la RAM dotada de memoria intermedia ("no volátil") como tema de algunas instrucciones en escritura.

Por ejemplo, en la instrucción COORDIN la variable que pasa a indicar en la línea de procesamiento debe ser en RAM.

Solución:

Controle todas las variables estáticas y no volátiles.

4369 Demasiados ejes maestros activos (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha intentado activar como maestros más de cuatro ejes contemporáneamente.

Este error solo se produce en ejecución de la instrucción de CHAIN.

Solución:

Reduzca el número de ejes maestros.

4370 Demasiados ejes esclavos activos (Función:NombreFunción línea:NúmeroLínea)

Causa:

Se ha intentado activar más de ocho ejes esclavos de un único eje maestro. Este error solo se produce en ejecución de la instrucción de CHAIN.

Solución:

Reduzca el número de ejes esclavos.

4372 Uso erróneo de una instrucción (Función:NombreFunción línea:NúmeroLínea)

Causa:

Este error se genera en una de las situaciones siguientes:

1. Si se usa una instrucción para la gestión del buzón de correo (sendmail, waitmail, endmail, ifmail) o una instrucción para la gestión de la CIP (sendipc, testipc, waitipc) dentro de una función llamada por una educación Errsys, OnInput o OnFlag.
2. Si se usa una instrucción IfError o una instrucción IfMessage sin tener habilitada la gestión de alarmas de estado.
3. Si se usa la instrucción Controlador de secuencia sin que esté presente la tarjeta TMSWD.
4. Los parámetros definidos en una instrucción de interpolación (linearinc, linearabs, circle, circinc, circabs, helicinc y helicabs) no son consistentes. Por ejemplo, el número de los ejes que han sido declarados es diferente del número de acciones declaradas o se han declarado más ejes de los que la instrucción puede manejar.

Solución:

A continuación, están descritas las soluciones para cada causa señalada:

1. Mueva la instrucción que provoca el error en otra función o remover la instrucción.
2. Averigüe que esté habilitada la gestión de alarmas de estado. En el archivo tpa.ini en la sección [ALBATROS] a la opción AlarmsHaveState se le debe dar valor 1.
3. Remueva la instrucción Controlador de secuencia o conseguir una tarjeta TMSWD.
4. Controle que los parámetros de la instrucción GPL estén correctos. Con cada eje tiene que corresponder una cota. El número de ejes declarados en la instrucción no debe ser mayor del número de ejes que la función puede manejar. Por ejemplo, la instrucción LINEARABS puede manejar hasta 6 ejes. Si entre los parámetros se declaran más de 6 la instrucción genera el error de sistema.

4373 Imposible ejecutar la lectura de la velocidad de alimentación (Función:NombreFunción línea:NúmeroLínea)

Causa:

Fue utilizada la instrucción [GETFEED](#) en una tarjeta TmsBUS o TmsCAN que no es máster.

Solución:

Compruebe en configuración Hardware que la tarjeta donde está relacionada la velocidad de avance sea máster.

4374 Demasiadas instrucciones de tipo IPC en ejecución (Función:NombreFunción línea:NúmeroLínea)

Causa:

Se ha superado el límite máximo de 16 instrucciones IPC en ejecución simultánea.

Solución:

Modifique el código GPL.

4375 FASTREAD ejecutada sobre ejes de tarjetas diferentes (Función:NombreFunción línea:NúmeroLínea)

Causa:

Se ha tratado de ejecutar una instrucción FASTREAD pasándola como parámetros ejes que no están todos conectados a la misma tarjeta.

Solución:

Modifique oportunamente el código GPL o la configuración Virtual-físico.

4378 Instrucción no habilitada (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha tratado de utilizar una instrucción cuya ejecución no está habilitada. Probablemente la clave hardware no está insertada correctamente o falta.

Solución:

Inserte la clave hardware correctamente. Si el problema persiste, póngase en contacto con el constructor.

4379 Esta instrucción no se puede utilizar en funciones lanzadas por Interrupción (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha tratado de utilizar una instrucción no admitida en una función lanzada por interrupción. Las funciones lanzadas por interrupción son las pasadas como parámetro en las instrucciones ONERRSYS, ONINPUT y ONFLAG.

Solución:

Modifique el código GPL. Consulte [el listado de instrucciones no utilizables en interrupción](#).

4380 Demasiados solicitudes de escritura en el área de memoria intermedia (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha tratado de ejecutar demasiadas operaciones de escritura en la memoria tamponada simultáneamente (la memoria tamponada está caracterizada por un tiempo de acceso relativamente elevado).

Solución:

Verifique las instrucciones que efectúan operaciones de escritura en las variables alojadas en la memoria tamponada: contadores, temporizadores, matrices y variables declaradas "nonvolatile".

4381 No se puede utilizar una línea serial no abierta (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha tratado de ejecutar una instrucción que actúa sobre un puerto serial antes de haber ejecutado, para el mismo puerto, la instrucción COMOPEN.

Solución:

Modifique el código GPL.

4382 No se puede abrir una línea serial ya abierta (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha tratado de ejecutar una instrucción COMOPEN en un puerto serial ya abierto con la misma instrucción.

Solución:

Modifique el código GPL.

4383 Se intentó abrir demasiados procesos auxiliares (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha tratado de abrir más de 4 procesos auxiliares simultáneamente.

Solución:

Modifique el código GPL.

4384 El proceso auxiliar no está en ejecución (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha tratado de acceder a un proceso auxiliar que no está ejecutándose.

Solución:

Modifique el código GPL.

4385 Se intentó abrir un proceso auxiliar de otra tarea (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se ha tratado de abrir un proceso auxiliar desde una tarea que no es la que ha lanzado la ejecución. Un proceso auxiliar puede ser utilizado solo para la tarea que ha comenzado la ejecución.

Solución:

Modifique el código GPL.

4391 Error durante la activación del SYSOK (Función:NombreFunción línea:NúmeroLínea)**Causa:**

La activación de la señal de SYSOK no terminó correctamente. En general puede depender de un mal funcionamiento del transmisor del Greenbus en la tarjeta de ejes.

Solución:

Los técnicos calificados pueden efectuar una prueba Hardware de la dual port memory del micro-controlador i296. Si el problema sigue presentándose, póngase en contacto con el constructor.

4394 Demasiados errores de ciclo (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Están activos más de 2000 errores de ciclo.

Solución:

Corrija el código GPL limitando el número de señales.

4395 Demasiados mensajes (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Están activos más de 2000 mensajes.

Solución:

Corrija el código GPL limitando el número de señales.

4397 Desbordamiento de pila (Función:NombreFunción línea:NúmeroLínea)**Causa:**

La memoria de retención de una función GPL ha superado el límite máximo de 2 Kbytes.

Solución:

Vuelva a compilar el código GPL y controle en el informe del compilador la ocupación estimada de memoria de retención de la función que ha causado el error de sistema. Disminuya, por lo tanto, el número de variables locales y de parámetros pasados a las funciones (sustituyéndolos, por ejemplo, con variables globales). Disminuya el número de CALL.

4398 Subdesbordamiento de pila (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Puede ocurrir sólo por un error grave del Firmware, por ejemplo, en el caso de una gestión errada de los parámetros de una función o de las variables locales.

Solución:

Póngase en contacto con el constructor.

4399 Parámetro fuera de intervalo (Función:NombreFunción línea:NúmeroLínea)**Causa:**

Se le ha asignado a una variable GPL o a un dispositivo un valor no incluido en el intervalo admitido.

Solución:

Corrija y vuelva a compilar el código GPL.

4865 Falta la definición de la máquina para la interpolación (G216 o G217)**Causa:**

Se ha intentado mover los ejes con una interpolación ISO o se han establecido los índices de configuración, sin haber definido antes los índices de las matrices de configuración y los ejes que constituyen la máquina.

Solución:

Corrija y vuelva a compilar el código GPL, utilizando las instrucciones [ISOG216](#).

4866 Falta la definición de los índices de la configuración máquina seleccionada (M6)**Causa:**

Se ha intentado mover los ejes con una interpolación ISO sin haber definido antes los índices de las matrices de configuración de la máquina.

Solución:

Corrija y vuelva a compilar el código GPL, utilizando la instrucción [ISOM6](#).

6.2.10 Errores ocasionados por el controlador de las comunicaciones CNCTPA**16385 Módulo desconectado****Causa:**

La conexión entre el PC supervisor y un módulo ha sido interrumpida.

Las posibles causas son:

- falta de alimentación del módulo remoto
- interrupción incluso temporal de la conexión de los cables Ethernet debido a un falso contacto en los conectores o a un daño en los mismos cables
- falta de alimentación o funcionamiento defectuoso del hub Ethernet (de estar presente)
- bloqueo del firmware del módulo remoto debido a un daño en los ficheros de configuración
- restablecimiento de la CPU del módulo remoto debido a sobrecalentamiento o disturbios EM

Solución:

Verifique que el módulo esté encendido. Verifique los cables y los conectores Ethernet. Actualice el firmware a bordo del módulo remoto. Verifique que este no esté sometido a sobrecalentamiento debido a ventilación insuficiente y que no esté sometido a disturbios EM.

Si el problema persiste, póngase en contacto con el constructor.

16386 Módulo vinculado**Causa:**

Un módulo remoto se ha conectado al PC supervisor luego de la fase de inicialización de Albatros. Durante la puesta en servicio Albatros trata de conectar todos los módulos previstos por la Configuración de Sistema, esta fase dura aproximadamente 4 segundos. Los módulos que se conectan posteriormente provocan el error de sistema.

16387 Módulo revinculado**Causa:**

Un módulo remoto se ha vuelto a conectar al PC supervisor luego de una desconexión. Este error, por lo tanto, es siempre consecutivo al error 16385 "Módulo desconectado".

16388 Módulo inicializado**Causa:**

Un módulo remoto ha sido reinicializado durante el normal funcionamiento. Esto implica que el módulo se haya desconectado y vuelto a conectar al PC supervisor. Este error, por lo tanto, es siempre consecutivo al error 16385 "Módulo desconectado".

Este error indica que tuvo lugar el restablecimiento del módulo, debido, por ejemplo, a una falta de alimentación.

16389 El módulo ha interrumpido la conexión**Causa:**

Un módulo remoto ha interrumpido la conexión con Albatros. Esto pasa cuando el módulo no recibe comandos o interrogaciones por parte del PC Supervisor por un largo tiempo. Este error indica, por lo tanto, un problema (fuerte deceleración o bloque) en el PC Supervisor.

Solución:

Controle que en el PC Supervisor no se encuentren programas que provoquen el bloqueo o la deceleración del sistema. Deshabilite el protector de pantallas en el PC Supervisor. Si el problema persiste, contacte al constructor de la máquina.

16641 El firmware del control no responde a los comandos

Causa:

Se ha verificado un error durante la inicialización del sistema. En particular el firmware no responde correctamente. El problema puede depender de una avería de los archivos de firmware.

Solución:

Pruebe a restablecer el sistema y eventualmente a reinstalar Albatros. Si el problema persiste, póngase en contacto con el constructor.

16642 TpaSock no responde a los comandos

Causa:

Se ha verificado un error durante la inicialización del sistema. En particular el software de comunicación con los módulos remotos no responde correctamente. El problema puede depender de un problema de los archivos.

Solución:

Intente restablecer el sistema y eventualmente a reinstalar Albatros. Si el problema persiste, póngase en contacto con el constructor.

16643 El sistema operativo no permite utilizar RTX

Causa:

El sistema operativo instalado en la PC no permite el uso de RTX y, como consecuencia, no permite el correcto funcionamiento de las versiones de Albatros que prevén su presencia.

Solución:

Actualice el sistema operativo de la PC. Consulte los requisitos mínimos de sistema en el manual de instalación de Albatros (InstallationGuide.pdf).

16645 Error al enviar el código firmware

Causa:

Se ha verificado un error durante la inicialización del sistema. En particular ha fallado el envío a un módulo de un archivo de firmware.

Solución:

Intente restablecer el control. Si el problema persiste, póngase en contacto con el constructor.

16646 No ha sido posible reiniciar el código firmware

Causa:

Se ha verificado un error durante la reinicialización del sistema. En particular ha fallado en el inicio del firmware después que este había sido parado.

Solución:

Intente restablecer el sistema. Si el problema persiste, póngase en contacto con el constructor.

16897 RTX no está instalado

Causa:

La versión de Albatros instalada requiere que en la PC esté instalado RTX, pero este no ha sido detectado.

Solución:

Instale RTX o si ya está presente, lo reinstale. Consulte el manual de instalación de RTX Albatros (InstallationRTXGuide.pdf).

16898 El usuario no tiene derechos de administrador

Causa:

Albatros ya ha sido iniciado por un usuario falto de los derechos de Administrador de la PC. Los derechos de Administrador son necesarios para el correcto funcionamiento de Albatros.

Solución:

Cerrar la sesión de trabajo en curso y acceder al sistema como "Administrator" o como otro usuario con derechos de administrador.

16899 Dimensión incorrecta de la RAM del módulo

Causa:

La cantidad de RAM detectada en el módulo remoto no corresponde con la cantidad prevista. Por lo general este error indica una avería del hardware.

Solución:

Si el problema persiste, póngase en contacto con el constructor.

16900 La dirección IP del módulo es incorrecta

Causa:

Ha sido detectado un módulo remoto cuya dirección IP no pertenece a la subred del PC supervisor. Por lo tanto, Albatros no puede comunicar correctamente con el módulo remoto.

Solución:

Verifique que las configuraciones del servicio AlBDHCP y de la tarjeta de red de la PC estén correctas. Consulte el manual de instalación de Albatros (InstallationGuide.pdf).

16901 El módulo ya está vinculado a otra instalación

Causa:

Ha sido detectado un módulo remoto que resulta estar conectado con otro PC supervisor. Ello puede depender de la presencia en la red de otra PC en la cual Albatros está en ejecución y que utiliza el mismo módulo. Puede depender también por un incorrecto funcionamiento del software de comunicación del módulo Clipper.

Solución:

Verifique que el módulo remoto no sea utilizado por otro PC supervisor. Reinicie el módulo. Si el problema persiste, póngase en contacto con el constructor.

16902 El módulo no está configurado

Causa:

Ha sido relevado un módulo que no resulta estar configurado en "Configuración de Sistema".

Solución:

Configurar el módulo.

16903 Las configuraciones del firewall impiden la comunicación

Causa:

Ha sido relevado un firewall instalado en el PC que impide la comunicación entre Albatros y los módulos remotos.

Nota: Albatros puede identificar la presencia del firewall de Windows Xp y no de otros firewalls, por ejemplo, los que están incluidos en unos softwares antivirus.

Solución:

Modifique las configuraciones del firewall o desactivarlo.

16904 Tarjeta de red no presente o deshabilitada

Causa:

No ha sido detectada una tarjeta de red utilizable para la conexión a los módulos remotos.

Nota: el hecho que sea detectada una tarjeta no es garantía de que esta esté correctamente configurada y conectada.

Solución:

Verifique la presencia y la correcta configuración de la tarjeta de red. Si el problema persiste, póngase en contacto con el constructor de la máquina.

16905 Falta el código firmware del control

Causa:

Albatros no encuentra un archivo de firmware en el disco duro del PC. El problema se puede manifestar en seguida de la cancelación accidental de los archivos de firmware o bien en seguida de una actualización incorrecta.

Solución:

Verifique que los archivos contenidos en la carpeta FW de la instalación de Albatros estén presentes y sean de la versión correcta. Póngase en contacto con el constructor de la máquina.

16906 Versión RTX incompatible con el código de firmware del control

Causa:

Ha sido detectada una versión de RTX no compatible con el firmware instalado.

Solución:

Instale la versión de RTX correcta o actualice el firmware. Póngase en contacto con el constructor de la máquina.

16907 Versión del sistema operativo no compatible con el código firmware del control

Causa:

La versión del sistema operativo del módulo remoto no es compatible con el firmware instalado.

Solución:

Instale en el módulo remoto la versión correcta de sistema operativo o bien actualice el firmware. Póngase en contacto con el constructor de la máquina.

17153 TipoTarjeta: Falta el código firmware del transmisor GreenBUS

Causa:

No ha sido detectado un archivo de firmware en la cartilla FW. Por lo general, el problema depende de una cancelación accidental del archivo o de una instalación incompleta o dañada.

Solución:

Reinstale Albatros después de haber efectuado una copia de seguridad del sistema. Consulte al constructor de la máquina.

17154 TipoTarjeta: Falta la porción de código de firmware del transmisor GreenBUS

Causa:

El archivo que contiene el código firmware del transmisor GreenBus está presente en la carpeta FW, pero resulta dañado o incompleto.

Solución:

Reinstale Albatros después de haber efectuado una copia de seguridad del sistema. Póngase en contacto con el constructor de la máquina.

17155 TipoTarjeta: Error en el envío del código de arranque del transmisor GreenBUS

Causa:

Se ha verificado un error durante la reinicialización del sistema. En particular ha fallado el envío a un módulo de un archivo de firmware.

Solución:

Intente restablecer el control. Si el problema persiste, póngase en contacto con el constructor.

17156 TipoTarjeta: Error en el envío del código de Main del transmisor GreenBUS

Causa:

Se ha verificado un error durante la inicialización del sistema. En particular ha fallado el envío a un módulo de un archivo de firmware.

Solución:

Intente restablecer el control. Si el problema persiste, póngase en contacto con el constructor.

17157 TipoTarjeta: Falta el código de arranque

Causa:

No ha sido detectado un archivo de firmware en la cartilla FW. Por lo general, el problema depende de una cancelación accidental del archivo o de una instalación incompleta o dañada.

Solución:

Reinstale Albatros después de haber efectuado una copia de seguridad del sistema. Póngase en contacto con el constructor de la máquina.

17158 TipoTarjeta: Falta el código de Main**Causa:**

No ha sido detectado un archivo de firmware en la cartilla FW. Por lo general el problema depende de una cancelación accidental del archivo o de una instalación incompleta o dañada.

Solución:

Reinstale Albatros después de haber efectuado una copia de seguridad del sistema. Póngase en contacto con el constructor de la máquina.

17159 TipoTarjeta: Error en el envío del código de arranque**Causa:**

Se ha verificado un error durante la reinicialización del sistema. En particular ha fallado el envío a un módulo de un archivo de firmware.

Solución:

Pruebe a restablecer el control. Si el problema persiste, póngase en contacto con el constructor.

17160 TipoTarjeta: Error en el envío del código de Main**Causa:**

Se ha verificado un error durante la reinicialización del sistema. En particular ha fallado el envío a un módulo de un archivo de firmware.

Solución:

Pruebe a restablecer el control. Si el problema persiste, póngase en contacto con el constructor.

17409 No se ha conseguido enviar el ejecutable auxiliar**Causa:**

El error se puede verificar durante la actualización del firmware de un módulo remoto. Puede ser causado por un momentáneo incorrecto funcionamiento de la red o bien por el desperfecto del firmware del módulo remoto. El mensaje de error puede incluir un código de error.

Solución:

Intente apagar y encender de nuevo el módulo remoto y repita el proceso de actualización. Si el problema persiste, póngase en contacto con el constructor.

17410 No se ha conseguido poner en ejecución al ejecutable auxiliar**Causa:**

Se ha verificado un error durante la inicialización del sistema. En particular no ha sido posible ejecutar un programa auxiliar. El mensaje de error incluye, además, el nombre del programa auxiliar y eventualmente un código de error.

Solución:

Pruebe a restablecer el sistema. Si el problema persiste, póngase en contacto con el constructor.

17667 NombreDLL: No se ha conseguido ejecutar el código firmware**Causa:**

Se ha verificado un error durante la inicialización del sistema. En particular no ha sido posible ejecutar el código firmware. "NombreDLL" corresponde al componente que ha generado el error.

Solución:

Intente restablecer el sistema. Si el problema persiste, póngase en contacto con el constructor.

17668 NombreDLL: No ha sido posible conseguir el puntero a la RAM compartida**Causa:**

Se ha verificado un error durante la inicialización del sistema. En particular no ha sido posible abrir el canal de comunicación con el firmware. "NombreDLL" corresponde al componente que ha generado el error.

Solución:

Intente restablecer el sistema. Si el problema persiste, póngase en contacto con el constructor.

17921 No se ha conseguido enviar NODETPA

Causa:

El error se puede verificar durante la actualización del firmware de un módulo Clipper. Puede ser causado por un momentáneo incorrecto funcionamiento de la red o bien por el desperfecto del firmware del módulo Clipper. El mensaje de error puede incluir un código de error.

Solución:

Intente apagar y encender de nuevo el módulo remoto y repita el proceso de actualización. Si el problema persiste, póngase en contacto con el constructor.

17922 NODETPA no se ha reiniciado

Causa:

El error se puede verificar durante la actualización del firmware de un módulo Clipper. Puede ser causado por un momentáneo incorrecto funcionamiento de la red o bien por el desperfecto del firmware del módulo Clipper. El mensaje de error puede incluir un código de error.

Solución:

Intente apagar y encender de nuevo el módulo remoto y repita el proceso de actualización. Si el problema persiste, póngase en contacto con el constructor.

17923 NODETPA no está en ejecución

Causa:

Ha sido detectado en la red un módulo Clipper cuyo software de comunicación no está en ejecución. Por lo general es síntoma de un incorrecto funcionamiento del software de comunicación. El mensaje de error puede incluir un código de error.

Solución:

Intente apagar y encender de nuevo el módulo remoto. Si el problema persiste, póngase en contacto con el constructor.

18177 NODETPA trató de acceder a una dirección no válida

Causa:

El software de comunicación de un módulo Clipper ha generado un error. El mensaje de error puede incluir un código de error.

Solución:

Intente apagar y encender de nuevo el módulo remoto. Si el problema persiste, póngase en contacto con el constructor.

6.3 Advertencias genéricas

6.3.1 Albatros empieza la ejecución

Indica el inicio de Albatros y muestra unos datos útiles sobre la versión del programa y el entorno de ejecución.

6.3.2 Albatros termina la ejecución

Indica que Albatros está terminando la ejecución.

6.3.3 La computadora entra en modalidad de suspensión

Indica que el ordenador está entrando en el modo de suspensión. A partir de este momento, Albatros no puede más responder a solicitudes y mensajes que provienen de los ciclos GPL.

6.3.4 La computadora sale de la modalidad de suspensión

Indica que el ordenador recién ha salido del modo de suspensión. Albatros vuelve a ejecutar sin reinicialización.

6.3.5 Apagado del ordenador

Indica que el ordenador va a ser apagado mientras que Albatros está todavía en ejecución.

6.3.6 Nivel de acceso actual

Indica un cambio del nivel de acceso a las funciones de Albatros, generalmente para hacer el mantenimiento o cambiar los ciclos o la configuración.

6.3.7 Actualización software de los módulos

Indica que se ha solicitado la actualización del software y del firmware presentes en los controles remotos.

6.3.8 Enviar la configuración a los módulos

Indica que se ha solicitado la actualización de la configuración y de los ciclos presentes en los controles remotos.

7 Configuración del Sistema

7.1 Introducción

Ya hemos visto, en el capítulo concerniente la composición del sistema, como un sistema Albatros esté compuesto por uno o más módulos que componen un equipo y como cada uno de estos esté organizado en una estructura jerárquica.

Para configurar una máquina desde el punto de vista de Albatros es necesario seguir una secuencia de operaciones en las cuales se van a configurar los diferentes niveles lógicos y el Hardware subyacente.

La orden de máxima que seguir para configurar un sistema es la siguiente:

- [Configuración de Módulo](#)
- [Definición de Grupos y Subgrupos](#)
- [Configuración de los Dispositivos](#)
- [Configuración de Sistema](#)
- [Configuración Hardware](#)
- [Configuración Virtual-Físico](#)

Básicamente, las Configuraciones de Módulo, Grupo y Máquina definen la estructura lógica de la máquina, las Configuraciones de Sistema, Hardware y Virtual-Físico definen la estructura a nivel físico.

Veremos en detalle cada uno de estos puntos en los próximos párrafos.

7.2 Configuración de los Dispositivos

7.2.1 Introducción

En el capítulo relativo a la composición del sistema Albatros han sido presentados los tipos de dispositivos que pueden aparecer en un módulo. En seguida los dispositivos son nuevamente presentados, pero desde el punto de vista de su configuración.

Para cada tipo existe un número máximo de dispositivos que pueden ser programados como está precisado en la siguiente lista:

Tipo de dispositivo	Número máximo
Entrada analógica	128
Salida analógica	128
Entrada digital	4096
Salida digital	4096
Puerto de entrada	512
Puerto de salida	512
Eje	240
Temporizador	128
Contador	128
Bit indicador	1024
Interruptor indicador	256
Puerto indicador	256

7.2.2 Dispositivo genérico

La mayoría de los dispositivos requieren los mismos parámetros de configuración. En seguida es presentada la configuración de una Entrada Digital, las mismas consideraciones valen también para:

- Bit indicador
- Interruptor indicador
- Salidas analógicas
- Puerto de entrada
- Puerto de salida
- Puerto indicador
- Temporizadores
- Contadores

Para configurar un dispositivo entre los que están enumerados arriba, hay que especificar las siguientes configuraciones:

- **Nombre:** nombre del dispositivo que presenta al máximo 40 caracteres.
- **Comentario:** breve descripción del dispositivo puede ser traducida en diferentes idiomas, no son admitidos espacios.
- **Accesos en Lectura:** especifica el nivel de acceso mínimo requerido para que el dispositivo sea visualizado en las ventanas de Diagnóstico y en los Cuadros Sinópticos.

- **Accesos en Escritura:** especifica el nivel de acceso mínimo requerido para que el dispositivo sea visualizado en las ventanas de Diagnóstico y en los Cuadros Sinópticos.
- **Público:** especifica si el estado del dispositivo puede ser leído y modificado por el código GPL que no pertenece al grupo en el cual el dispositivo está colocado.

7.2.3 Salida digital

La salida digital tiene un parámetro más respecto a los dispositivos estándar: Monoestable.

Para configurar una salida digital hay que especificar las siguientes configuraciones:

- **Nombre:** nombre del dispositivo que presenta al máximo 40 caracteres.
- **Comentario:** breve descripción del dispositivo puede ser traducida en diferentes idiomas.
- **Monoestable:** si seleccionado configura la salida como monoestable, o sea cuando la salida es programada como ON vuelve automáticamente como OFF después de 200 ms.
- **Accesos en Lectura:** especifica el nivel de acceso mínimo requerido para que el dispositivo sea visualizado en las ventanas de Diagnóstico y en los Cuadros Sinópticos.
- **Accesos en Escritura:** especifica el nivel de acceso mínimo requerido para poder modificar el estado del dispositivo.
- **Público:** especifica si el estado del dispositivo puede ser leído y modificado por el código GPL que no pertenece al grupo en el cual el dispositivo está colocado.

7.2.4 Entrada analógica

La entrada analógica tiene un parámetro más respecto a los dispositivos estándar: El tipo de tensión en entrada.

Para especificar una entrada analógica hay que especificar las siguientes configuraciones:

- **Nombre:** nombre del dispositivo que presenta al máximo 40 caracteres.
- **Comentario:** breve descripción del dispositivo puede ser traducida en diferentes idiomas, no son admitidos espacios.
- **Tipo:** permite seleccionar el intervalo de tensiones leídas en entrada.
- **Accesos en Lectura:** especifica el nivel de acceso mínimo requerido para que el dispositivo sea visualizado en las ventanas de Diagnóstico y en los Cuadros Sinópticos.
- **Accesos en Escritura:** especifica el nivel de acceso mínimo requerido para poder modificar el estado del dispositivo.
- **Público:** especifica si el estado del dispositivo puede ser leído y modificado por el código GPL que no pertenece al grupo en el cual el dispositivo está colocado.

7.2.5 Eje

Datos de Base

Los datos de base que hay que especificar son:

- **Nombre:** nombre del dispositivo que presenta al máximo 40 caracteres.
- **Descripción:** breve descripción del dispositivo, puede ser traducida en diferentes idiomas, no son admitidos espacios.
- **Resolución:** resolución del encoder, depende de las características del encoder y de la unidad de medida especificada. Hay que tener presente que las tarjetas ejes de Albatros cuentan como impulso tanto los frentes de subida como los frentes de bajada de ambas fases encoder (un encoder de 2500 impulsos giro será, por lo tanto, visto como un encoder de 10000 impulsos giro).
- **Tipo Eje:** tipo de eje. Son previstos los tipos **Analógico** (controlado en analógico), **Paso a Paso, Digital, Cuenta** (sólo lectura encoder), **Virtual**.
- **Unidad de Medida:** unidad de medida utilizada para expresar las cotas del eje. Desde ellas dependen todas las magnitudes derivadas, se recomienda programarla antes de cualquier otro parámetro.
- **Inversión Fases:** permite compensar vía software una eventual inversión de cableo de las fases encoder.
- **Inversión Referencia:** permite invertir la referencia de velocidad del eje. Si utilizado junto con la inversión fases permite invertir la dirección del eje (si el cableo está correcto).
- **Habilitación Muesca:** disponible solo para ejes de cuenta, pone a cero automáticamente la cota cuando es relevada la muesca encoder.

Parámetros de Movimiento

Parámetros utilizados para el movimiento punto-punto del eje.

- **Velocidad máxima:** velocidad máxima del eje.
- **Aceleración:** tiempo de duración de la rampa de aceleración.
- **Deceleración:** tiempo de duración de la rampa de deceleración.
- **Velocidad mínima:** programable solo para ejes paso a paso, velocidad que el motor alcanza en un único paso.

- **Tipo de rampa:** tipo de rampa de aceleración y deceleración. No programable para motores paso-a-paso.
- **Proporcional:** coeficiente proporcional del controlador PID del anillo de posición.
- **Integradora:** coeficiente integral del controlador PID del anillo de posición.
- **Derivada:** coeficiente derivado del controlador PID del anillo de posición.
- **Feed Forward:** porcentual de alimentación enviada. Permite un correspondiente abatimiento del error de anillo con igual velocidad.
- **Feed Forward Acel.:** porcentual de alimentación enviada de aceleración. Permite el abatimiento del error de anillo residuo (no abatido por la alimentación enviada) durante las fases de aceleración y deceleración del eje.
- **Muestreos Integradora:** Programa el número de muestreos del error de anillo utilizados para el cálculo de la componente integral. Los valores válidos están entre 1 y 200. El valor predeterminado es 50. Véase instrucciones GPL [SETINTEGTIME](#).

Parámetros de Interpolación

Parámetros utilizados para el movimiento en interpolación del eje.

A excepción de la velocidad mínima, tienen el mismo significado de los parámetros vistos en las configuraciones de Movimiento. Solo que estos están utilizados para los movimientos interpolados.

Nota: los valores de aceleración y deceleración, programados en los parámetros de interpolación, no pueden ser inferior a los valores correspondientes en los parámetros de movimiento.

Otros Parámetros

- **Velocidad Manual:** específica la velocidad máxima de configuración utilizable en los movimientos manuales. No será nunca superior a la velocidad máxima programada.
- **Servoerror Dinámico:** habilita o inhabilita el error servo dinámico. El valor predeterminado es servoerror dinámico deshabilitado, por lo que permanece habilitado el error servo umbral. Véase la instrucción GPL [SETMAXERTYPE](#).
- **Velocidad de referencia y Error de anillo:** estos dos valores sirven para calcular la razón lineal real de las dos magnitudes en la máquina. Para que los valores sean tenidos en cuenta, ambos deben ser positivos y distintos de cero, y el campo **Servoerror Dinámico** necesita estar habilitado.
- **Espera eje parado:** habilita o inhabilita la función de recuperación de rebosamiento. Introduce un tiempo de espera de 50 ms al final de cada movimiento.
- **Timeout desplazamiento Eje:** Los valores válidos están incluidos entre 0 y 1024. Véase la instrucción GPL [ENABLESTARTCONTROL](#).
- **Límite acoplamiento codificador incorrecto:** Los valores programados son exprimidos en la unidad de medida con la cual es expresada la resolución del eje. Los valores programables tienen estar entre 128/resolución del eje y 16384/resolución del eje. El valor predeterminado se calcula sobre la base de un número de pasos par a 1024, o sea 1024/resolución del eje.
- **Límite Servoerror Positivo:** valor máximo del error de bucle para el bucle de regulación en dirección positiva.
- **Límite Servoerror Negativo:** valor máximo del error de bucle para el bucle de regulación en dirección negativa.
- **Límite Eje Positivo:** valor máximo de la carrera del eje en dirección positiva.
- **Límite Eje Negativo:** valor máximo de la carrera del eje en dirección negativa.
- **Ventana de llegada en cota positiva:** tolerancia sobre la cota de llegada en dirección positiva.
- **Ventana de llegada en cota negativa:** tolerancia sobre la cota de llegada en dirección negativa.

Parámetros de Referencia

- **Referencia:** valor de la tensión de referencia a la cual corresponde la velocidad máxima.
- **Ajuste Automático:** habilita o deshabilita el cálculo de recuperación del desplazamiento automático. Normalmente se encuentra habilitado.
- **Desplazamiento Inicial:** Valor al cual programar el desplazamiento inicial de referencia. El valore tiene que estar entre -10 y 10. El valor predeterminado es 0.
- **Frecuencia filtro de Notch:** Valor de frecuencia a filtrar. El valor tiene que estar entre 0 y 500.
- **Tensión Mínima:** Programa los parámetros de tensión mínima para el eje indicado. El valore negativo tiene que estar entre -10 y 0, el valor positivo entre 0 y +10. Véase instrucción GPL [SETDEADBAND](#).
- **Umbral:** Programa los valores de umbral. Son siempre menores o iguales a los respectivos valores de tensión mínima, por lo tanto, el valor negativo de umbral tiene que estar entre 0 y el valor de tensión mínima negativa. El valor máximo de umbral tiene que estar entre 0 y el valor de tensión mínima positiva.

Niveles de Acceso

- **Accesos en Lectura:** especifica el nivel de acceso mínimo requerido para que el eje sea visualizado en las ventanas de Diagnóstico y en los Cuadros Sinópticos.
- **Accesos en Escritura:** especifica el nivel de acceso mínimo requerido para poder modificar el estado del eje.
- **Público:** especifica si el estado del eje puede ser leído y modificado por un código GPL que no pertenece al grupo en el cual el eje está colocado.

Concatenación Ejes

Parámetros de concatenación ejes. Son los coeficientes del controlador PID que compensa las diferencias de error de anillo entre el eje maestro y los ejes esclavos.

- **Proporcional:** coeficiente proporcional
- **Integradora:** coeficiente integral
- **Derivada:** coeficiente derivado

Correctores de Linealidad

Configuración de los correctores de linealidad del eje. Los correctores permiten compensar tanto los errores de posicionamiento de un eje, debido a las vaguedades del trabajo de la mecánica del eje (auto-correctores) y como los errores debidos al efecto dado por los demás ejes de la máquina (correctores cruzados) típicamente ligado a la flexión de la estructura. Los correctores no son habilitados de manera automática más deben de ser habilitados en la ventana de modificación de los valores de corrección (botón **[Modificar...]**) y activados a través del código GPL con la instrucción [ENABLECORRECTION](#).

- **Intervalo de Corrección:** permite programar la distancia entre una corrección y la siguiente. El número de medidas será dado por la longitud del eje dividida por la longitud del intervalo de corrección.
- **Nombre archivo correctores:** permite programar el nombre del archivo en el cual los valores de corrección serán almacenados. Se trata de un archivo ASCII en el cual los valores son separados por el carácter ";". Esto permite su modificación con un normal editor de texto. La extensión del archivo no debe de ser especificada, es asignada automáticamente la extensión ".csv" (comma separated values).
- **Datos de Corrección:** permite especificar la lista de los ejes por los cuales el eje corriente genera corrección. El eje corriente está siempre incluido en la lista, o sea el auto-corrector está siempre presente. Es posible especificar hasta otros 5 ejes. Para alcanzar un eje hay que seleccionarlo en la lista a la izquierda y presionar el botón **[>> Agregar]**. Para quitar un eje hay que seleccionarlo en la lista a la derecha y presionar el botón **[Quitar <<]**. Para especificar los valores de corrección ha que seleccionar un eje de la lista a la derecha y presionar el botón **[Modificar...]**, se abrirá una ventana con una tabla en la cual insertar los valores de corrección.

ATENCIÓN: Existe un límite máximo de **235** correctores de linealidad, gestionados por el sistema, para cada eje. Como consecuencia, dada la longitud del eje, el intervalo de medida deberá tener una longitud mayor o igual a la ducentésima trigésima quinta parte de la longitud del eje. Por ejemplo, para un eje con 2500 mm de longitud habrá que configurar un intervalo de corrección mayor o igual a 10,63 mm. Además, existe un límite al valor máximo de una corrección individual, esta tendrá que ser inferior a 1024 pasos de encoder, por ejemplo, para un eje con resolución de 256 pasos/mm la corrección máxima es de ± 4 mm.

7.3 Configuración Lógica

7.3.1 Configuración de la instalación

Para definir una nueva máquina o para modificar una ya existente hay que acceder a la página gráfica de Configuración Módulo. La Configuración Módulo es la configuración de los módulos que componen la instalación.

La apertura del ambiente de Configuración es posible si se está al nivel de acceso fabricante o superior.

Acceso a la Configuración

Seleccione desde el menú **Archivo** la opción **Abrir Configuración**.

Si en la instalación no hay ya módulos configurados, se abre automáticamente la Configuración de Módulo, si no se abre la Configuración de Máquina. En este caso para pasar a la Configuración Módulo se puede seguir de la siguiente manera:

Seleccione desde el menú **Edición** la opción **Configuración Módulo**

Para agregar un módulo a la instalación es suficiente seleccionar el botón **[Nuevo]**. El botón **[Modificar]** permite modificar los datos de un módulo ya existente, el botón **[Eliminar]** permite cancelar un módulo, el botón **[Cerrar]** permite salir de la configuración de instalación.

Los datos que identifican una máquina, y que habrá que especificar, son:

- el número de módulo: un número entero progresivo que, si no se especifica, es asignado por el sistema
- una breve descripción.

Hay unos datos que conciernen el Hardware:

- **Frecuencia Control Ejes:** indica la frecuencia con la que se intercambian periódicamente los datos entre el control numérico y los dispositivos conectados a ese a través de los bus de campo.
- **Número Canales Interpolación:** indica el número máximo de canales de interpolación (es decir, los grupos de ejes que ejecutan un movimiento de tipo interpolado) que se pueden administrar simultáneamente.
- **Porcentajes de uso de la CPU:** indica el porcentaje de tiempo, en relación con el período de control de los ejes (es decir, el contrario de la "frecuencia control ejes") que está reservado para la ejecución del firmware.

La misma ventana se puede abrir desde la rama del módulo de la Configuración de los grupos, la rama del Módulo de la Configuración de máquina y desde la rama del módulo de la Configuración Hardware.

7.3.2 Configuración de los grupos

La primera vez que se proyecta una máquina es necesario definir cada uno de sus componentes y escribir todos los programas de control. En muchos casos el proyecto es desarrollado a partir de una máquina ya realizada modificada según las características de la nueva máquina.

Creación de un Grupo

Para crear un nuevo grupo es necesario acceder a la página gráfica de configuración de grupos. La primera rama del árbol es el módulo, de donde derivan todos los grupos, los subgrupos y los dispositivos. Presionando la tecla **[ENTRAR]** o el botón **[Modificar]** se abre una ventana de diálogo para modificar los datos del módulo.

Seleccionar desde el menú **Edición** la opción **Grupos**.

Desde aquí se puede crear nuevos grupos, modificar o cancelar los existentes y copiar un grupo.

Lista de los comandos para crear, modificar, cancelar, copiar y pegar grupos, subgrupos y dispositivos.

Comando	Acción
Crear un nuevo grupo, subgrupo, dispositivo	[Ctrl+Entrar], Botón [Nuevo], Edición->Nuevo, menú contextual
Modificar un grupo, subgrupo, dispositivo	[Entrar], Botón [Modificar], Edición->Modificar..., menú contextual
Eliminar un grupo, subgrupo, dispositivo	[Supr], Botón [Eliminar], Edición->Eliminar, menú contextual
Habilitar o deshabilitar el uso de un grupo en máquina	Menú contextual, Botón [Habilitar]
Copiar un grupo, subgrupo, dispositivo	[Ctrl+C], Botón [Copiar], Edición->Copiar, menú contextual
Pegar un grupo, subgrupo, dispositivo	[Ctrl+V], Botón [Pegar], Edición->Pegar, menú contextual

Cuando se crea un grupo nuevo, aparece la ventana siguiente, donde se configuran:

- el nombre del grupo
- un comentario (se puede traducir en los diferentes idiomas soportados por Albatros).

Es posible, además, marcar el grupo como **Intergroup**. De todas formas, hay que configurar por lo menos un grupo como intergrupo para que esta configuración sea utilizada por Albatros para identificar el grupo "principal" de la máquina. Este es el grupo desde el cual es lanzada automáticamente en el inicio la función principal (la que tiene el mismo nombre del grupo). Este mecanismo sirve para inicializar la máquina y ejecutar en el inicio las tareas que verifican que todo funcione correctamente antes de pasar el control a quien lo utilizará.

Cuando se desactiva un grupo que tiene dispositivos conectados a los dispositivos físicos, se pide, si se quiere borrar el vínculo Virtual-Físico. Si se elige mantener los vínculos, los pin de los dispositivos físicos conectados a esas se visualizan en color gris en la representación gráfica del Virtual-Físico.

Añadidas de un subgrupo a un grupo

Para crear un subgrupo de un grupo, hay que estar posicionados en el grupo.

Si no se está interesados en crear unos subgrupos, se puede seleccionar la casilla *Lista de los Dispositivos*, y seleccionar el botón **[Aceptar]**. El nombre del subgrupo se asigna automáticamente.

Ahora es posible insertar cada uno de los dispositivos en el interior del subgrupo. El mecanismo es similar al de la creación de los subgrupos. En este caso, aparece una ventana con una lista de los dispositivos disponibles.

Seleccionado el dispositivo que nos interesa, hay que seleccionar el botón **[Aceptar]** para la confirmación.

Aparece una ventana que nos permite configurar el nombre, un comentario, y otros datos que varían en relación con el dispositivo seleccionado. Una descripción detallada de los tipos de dispositivos y de sus configuraciones será expuesta más adelante en el capítulo [Configuración de los Dispositivos](#).

Copia de un dispositivo

La función de copia de un dispositivo permite copiar un dispositivo. Ante todo, hay que seleccionar el dispositivo y en seguida el botón **[Copiar]**. Para insertar el dispositivo en la lista hay que seleccionar la rama donde pegar el dispositivo y accionar el comando **[Pegar]**. Es necesario insertar en la ventana de diálogo el nuevo nombre del dispositivo.

Copia de un subgrupo

La función para copiar un subgrupo permite copiar un subgrupo que incluye todos los dispositivos contenidos en ese. Para insertar el subgrupo se necesita seleccionar la rama donde tiene que ser pegado y activar el comando **[Pegar]**. En la ventana de diálogo es necesario insertar el nuevo nombre del subgrupo.

Copia de un grupo

La función de copia de un grupo permite copiar un grupo que comprende todos los subgrupos y los dispositivos en él contenidos. Además, es copiado el eventual sinóptico de grupo a él asociado (sinóptico cuyo nombre coincide con el del grupo).

Permite crear de manera rápida unos grupos que tienen una estructura similar a la de un grupo ya existente sin tener que volver a crear uno por uno todos los dispositivos. Para copiar un grupo hay que elegir el grupo que se quiere copiar, seleccionar el botón **[Copiar]** e insertar en el cuadro de diálogo el nombre del nuevo grupo.

La copia de dispositivos, subgrupos y grupos puede también ser ejecutada entre módulos diferentes.

Elegir los grupos que pertenecen a la máquina

Después de crear el archivo de grupos, hay que habilitar o deshabilitar los grupos realmente presentes.

Los grupos son todos presentes en máquina, si no se decide deshabilitarlos a través del botón **[Deshabilitar]** o seleccionando el mismo comando desde el menú contextual. Si un grupo está desactivado, aparece la frase **No presente** al lado del nombre del grupo.

Para visualizar solo los grupos presentes en máquina, elija **Modificar...** desde el menú **Edición**.

Para insertar un grupo nuevo, presione el botón **[Insertar]**. Así aparece una ventana que con una lista de los grupos presentes en el archivo Grupos y no insertados en máquina todavía.

A este punto, seleccione el grupo elegido antes y **lo arrastre con el ratón** en la ventana de Configuración Máquina, o seleccione el botón **[Introducir]**.

Además, a través del botón **[Quitar]**, es posible quitar un grupo existente o realizar la búsqueda del nombre de un grupo o de un dispositivo dentro del árbol que compone la máquina.


Un solo intergrupo debe estar presente en una máquina.

7.4 Configuración Física

7.4.1 Configuración de sistema

La configuración de sistema permite asociar unos recursos físicos (unidad de control) a los módulos definidos en la configuración lógica. Eso es posible en la ventana de dialogo Configuración del Sistema. Aquí está la lista de los **módulos** de la instalación, a cada uno de estos puede ser asociado un **Nodo de red**.

- **Nodo local:** Sistemas "Locales" donde el HW que implementa el control está montado directamente en el PC que constituye la interfaz usuario del sistema.
- **Nombre de un nodo de red:** Sistemas "Remotos" donde el HW que implementa el control está conectado al PC.
- **No configurado:** ninguna configuración. Es el predeterminado inicial. Si esta opción permanece, en seguida será posible asociar un módulo remoto en la ventana de dialogo **Conexiones nodos de red**.

Se pueden configurar hasta 16 módulos, y uno solo puede ser configurado como nodo local. Para asignar un módulo, seleccionar el botón **[Modificar]** o hacer un doble clic del ratón sobre el nodo de red a modificar. Abriendo el menú despegable se puede visualizar la lista de los módulos remotos disponibles, escoger de utilizar un nodo local o escoger de programar el módulo como no configurado. Para confirmar la selección, hay que seleccionar el botón .

Nota: El funcionamiento de Albatros en la máquina está protegido por una clave hardware usb, configurada por TPA.

7.4.2 Configuración Hardware

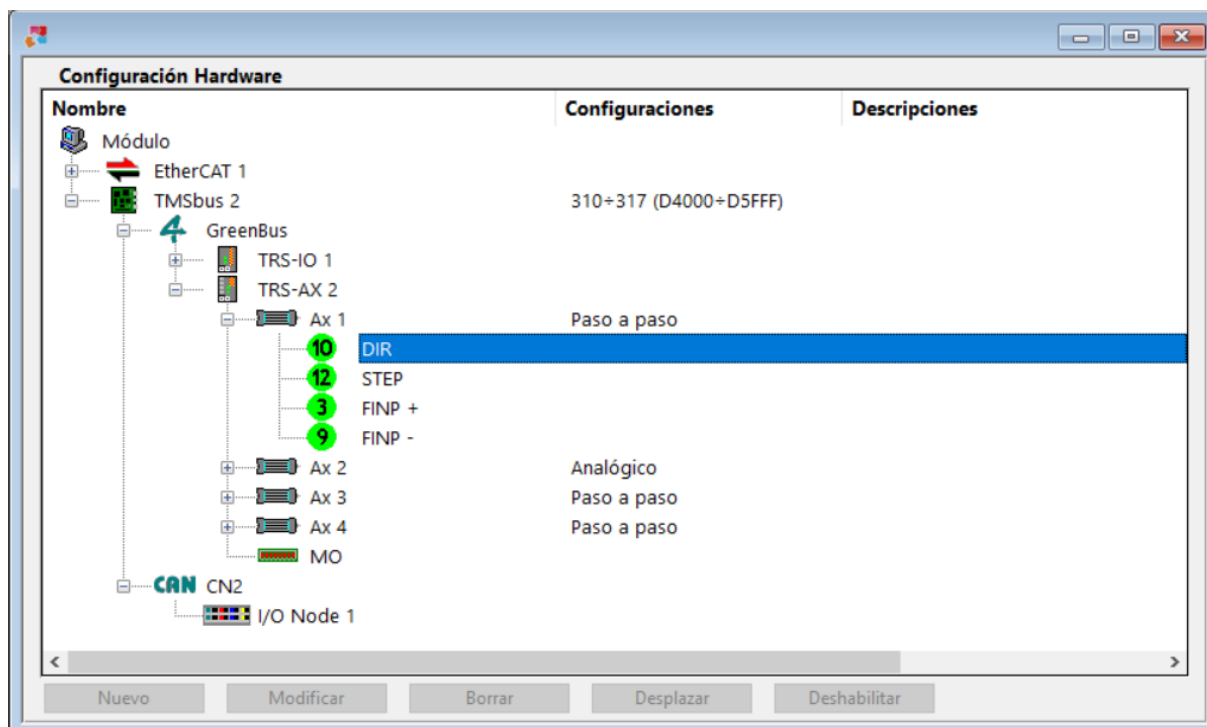
En la configuración hardware se definen las tarjetas y los nodos que van a componer el sistema. La tarjeta que ocupa la primera posición en el listado se llama Tarjeta Master.

Los tipos de tarjeta configurables son:

- TMSbus hasta dos
- TMSbus+ hasta cuatro
- TMSCombo+ hasta cuatro
- DualMech hasta cuatro
- DualMech Mono hasta cuatro
- TMScan hasta dos
- TMScan+ hasta cuatro
- AlbMech hasta dos
- EtherCAT una

Descripción de la ventana de configuración hardware

La ventana de configuración hardware se abre seleccionando desde menú la opción **Edición->Hardware**. Para insertar una tarjeta, un módulo de E/S remoto o un nodo CAN o EtherCAT, se debe pulsar el botón **[Nuevo]**. Se visualizará una ventana que permite la selección de la tarjeta o del módulo de E/S remoto y, para los buses CAN y GreenBUS, la posición donde se deben insertar. Generalmente, por cada módulo se pueden configurar hasta un máximo de 4 tarjetas y, de acuerdo al tipo de tarjeta y bus, un número variable de módulos de E/S remotos.



Configuración Hardware

En la columna **Configuraciones** se asignan los datos relativos a la tarjeta o al nodo. Con el comando **[Desplazar]** se puede desplazar una tarjeta o un nodo o una expansión de un TRS-IO o un TRS-CAT desde una posición del árbol a otra. Se pueden mover las expansiones sólo dentro del mismo nodo. Esta operación mantiene las conexiones presentes en la configuración [Virtual-Físico](#).

Además, es posible deshabilitar un nodo o una expansión de un TRS-IO o de un TRS-CAT. Esta deshabilitación hace que se mantengan las conexiones en configuración Virtual-Físico. Si el nodo pertenece a un bus GreenBUS, el nodo y los dispositivos conectados a él son totalmente ignorados por el sistema. No se produce ningún error cuando el módulo no se detecta durante la inicialización y no se produce ningún error cuando se ejecuta una instrucción GPL en un dispositivo asociado al módulo. Si el nodo, o la expansión, pertenece a la red EtherCAT, o si la expansión pertenece a un bus GreenBUS, se necesita que no esté en la red. Si en GPL se accede a un dispositivo no conectado, se produce un error de sistema.

Es necesario poner cuidado al utilizar esta funcionalidad.

Para deshabilitar un nodo o una expansión, usar el comando **[Deshabilitar]**, para volver a habilitar un nodo o una expansión, usar el comando **[Habilitar]**.

Configuraciones predeterminadas

Varias configuraciones predeterminadas están disponibles. Desde menú **Edición->Modificar tipo control...** o desde menú contextual en la rama del módulo es posible seleccionar la configuración deseada. Si la configuración es nueva, el árbol se rellena con las tarjetas y los nodos definidos, en caso contrario se efectúan comprobaciones para validar que los hardware que ya están presentes sean compatibles con el tipo de módulo seleccionado. Todo lo que no es compatible se elimina.

Configurar un nodo de un bus TPA

Los tipos de módulo de E/S remoto configurables en GreenBUS (v3.0) son:

- | | |
|-----------|---|
| • Albre8 | 8 entradas y 8 salidas digitales |
| • Albre16 | 16 canales configurables vía software como entrada o salida digital |
| • Albre24 | 24 entradas y 24 salidas digitales |
| • Albre48 | 48 entradas y 48 salidas digitales |
| • Albrem | 10 puertos en entrada y 10 puertos en salida |
| • AlbSTEP | 8 entradas y 6 salidas digitales, un motor paso-a-paso |

- AlbEV 20 o 24 electroválvulas (conector D-sub 25 pin)
- Albrea 4 entradas y 4 salidas analógicas

Los tipos de módulo remoto que se pueden configurar en GreenBUS (v4.0) son:

- TRS-AX 4 ejes analógicos o paso-a-paso
- TRS-EV-24 24 electroválvulas (conector D-sub 25 pin)
- TRS-16 16 canales configurables vía software como entrada o salida digital
- TRS-IO 16 canales que se pueden configurar por software como entrada o salida digital expansible con módulos de tipo TRS-IO-E y módulos TRS-AN-E hasta un máximo de 5 y módulos TRS-AC-E.
- TRS-IO-E 16 canales que se pueden configurar por software como entrada o salida digital utilizable sólo como expansiones de un módulo TRS-IO
- TRS-AN-E 1 entrada analógica y 1 salida analógica que se puede utilizar solo como expansiones de un módulo TRS-IO
- TRS-REM remoto genérico que se utiliza para la construcción de módulos especiales. Se pueden conectar hasta 12 puertos de entrada y 12 puertos de salida
- TRS-AC-E 1 eje de conteo y 2 entradas digitales configurables como muesca de cero y entrada rápida. La tabla siguiente describe el número máximo de TRS-AC-E configurables en un TRS-IO. Si hay 3 expansiones en total entre TRS-IO-E y TRS-AN-E, se puede configurar una sola expansión TRS-AC-E. Si está configurada una sola expansión (TRS-IO-E o TRS-AN-E), se pueden conseguir hasta 2 expansiones TRS-AC-E.

Los módulos remotos TRS-AX, TRS-IO, TRS-REM y TRS-16 son exclusivamente conectables a tarjetas TMSbus, TMSbus+ y TMSCombo+.

A cada tarjeta TMSbus y TMSbus+ se pueden conectar un máximo de 4 módulos remotos TRS-AX.

Los tipos de módulo remoto de Tpa que se pueden configurar en un EtherCAT bus son:

- TRS-CAT 16 canales que se pueden configurar por software como entrada o salida digital expansible con módulos tipo TRS-IO-E, módulos TRS-AN-E y módulos TRS-AC-E
- STAR-CAT transforma una topología de red EtherCAT lineal en una topología a estrella usando un canal en entrada y hasta 3 diferentes canales en salida

La tabla siguiente describe el número máximo de expansiones configurables en un TRS-CAT:

Número expansiones TRS-IO-E y TRS-AN-E	Número expansiones TRS-AC-E
7	0
5	1
3	2
1	3

Para los módulos remotos de tipo TRS-AX, si el número de TRS-AX insertados aumenta, disminuye el número de TRS-16 y TRS-IO que se pueden usar.

Para calcular el número máximo de módulos remotos de tipo TRS-16 y TRS-IO que pueden insertarse, es necesario aplicar la siguiente fórmula: número de otros módulos remotos = 32 - (número de TRS-AX * 4). Por ejemplo, si a una tarjeta TMSbus están conectados 3 TRS-AX, aplicando la fórmula conseguimos: número de otros remotos = 32 - (3*4), entonces pueden insertarse un máximo de 20 módulos remotos de tipo TRS-16 y/o TRS-IO.

La selección de la posición del remoto se hace necesariamente en función de la dirección configurada con el conmutador en el módulo remoto. Véase la documentación hardware de cada remoto.

Si se selecciona un remoto TRS-AX, podría ser necesario configurar el tipo de ejes gestionados.

En el esquema siguiente se describen qué tipos de eje se pueden asociar a los hardware varios.

- tarjeta AlbMech ejes digitales
- tarjeta DualMech ejes digitales
- tarjeta DualMech Mono ejes digitales
- remoto TRS-AX ejes analógicos (si el conector específico es de tipo Analógico), ejes de cuenta (si el conector específico es de tipo Analógico), ejes paso a paso (si el conector específico es de tipo Paso a paso)
- remoto AlbStep ejes paso a paso
- expansión TRS-AC-E ejes de cuenta

En las tarjetas MECHATROLINK-II se puede gestionar cada eje en control de posición o en control de velocidad (predeterminado). El tipo de control por cada eje puede ser modificado en la ventana de Configuración hardware en la columna **Configuraciones**. El número de ejes configurables varía según el valor de frecuencia control ejes configurado:

Tarjeta	Frecuencia Control Ejes (Hz)	Número máximo servo-accionamientos
AlbMech	1000	8
AlbMech	<= 500	16
DualMech Mono	1000	8
DualMech Mono	500	20
DualMech Mono	250	30
DualMech	1000	16
DualMech	500	40
DualMech	250	60

Configurar un nodo de un bus CAN

La tarjeta de control del bus

Albatros puede gestionar dispositivos sobre buses de campo CAN a través de las tarjetas TPA equipadas de conector por bus CAN. El bus CAN es presente sobre las tarjetas **TMSbus**, **TMSbus+**, **TMSCan+** y **TMSCan**.

Configurar los datos de base y los servicios

Los datos del bus CAN se definen en configuración hardware. Se selecciona el bus CAN para poder definir sus parámetros y se hace clic sobre el botón **[Modificar]**.

Los datos de base son:

- **Tiempo de muestreo (TIME)**: tiempo de muestreo en ms. No puede ser superior a 60000 (60 segundos). El valor predeterminado es 2. Sobre bus S-CAN solo se acepta el valor 2.
- **Tiempo para la comunicación síncrona del PDO (TIMEPDO)**: tiempo expresado en mseg. Indica el tiempo para la comunicación síncrona de los PDO. El valor configurado no puede ser superior al valor de TIME (no es un valor obligatorio).
- **Tiempo de espera (TIMEAFTERRESET)**: tiempo expresado en ms. Indica el tiempo de espera durante la fase inicial debido a un restablecimiento software de los nodos en red. No puede ser superior a 60000 (60 segundos).
- **Número de ciclos CAN sin respuesta (LIFETIMEFACTOR)**: es el número de ciclos CAN sin respuesta a la llamada de Node Guarding, antes de que se cree el error de nodo desconectado. El valor no puede ser superior a 100 e inferior a 1. (El valor predeterminado es 3).
- **Velocidad en baudios (BAUDRATE)**: número velocidad de la comunicación CAN en kilo bit por segundo (el valor puede ser 1000, 500, 250, 125, 100).

Los servicios pueden ser activados o desactivados seleccionando o deseleccionando la opción de verificación, que se refiere al servicio. Al campo **Extra** se le pueden asignar valores que tienen un significado establecido por el constructor de la máquina. No se ejecuta ningún control de ese valor. El valor predeterminado es 16.

El nodo CAN

Insertar un nodo nuevo

Un nodo se puede insertar seleccionado la rama CAN del árbol y haciendo clic sobre el botón **[Nuevo]**. El tipo de nodo procede del tipo de bus. Si el bus es CAN, el nodo es un E/S, si el bus es S-CAN, el bus insertado es un Servo. En el cuadro de diálogo para insertar el nodo, seleccione entre:

- **Posición**: es el número de nodo (de 1 en adelante).
- **AutoOp**: se seleccionado, el dispositivo admite el pasaje automático al estado Operacional debido a una reconexión.

Configurar un nodo

Para cada nodo es posible definir los PDOs de transmisión y de recepción. Si el bus es CanOpen, se pueden definir hasta un máximo de 8 PDOs para tarjetas como TMSBus y TMSBus+, un máximo de 4 para tarjetas TMSCan y TMSCan+. Si selecciona en el tree el nodo y se hace clic sobre el botón **[Nuevo]**. Los datos que configurar en el cuadro de diálogo son:

- **Tipo PDO:** seleccione **Transmisión** o **Recepción**, si se quiere definir un TPDO o un RPDO.
- **Dimensión:** dimensión del PDO de recepción o de transmisión.
- **COB-ID:** valor que solo puede ser definido en tarjetas TMSBus y TMSBus+. El valor se visualiza y almacena como número decimal. Para visualizar el valor en hexadecimales, es necesario seleccionar la casilla **Hexadecimal**, puesta al lado del valor mismo.
- **Asíncronico:** si activado, se configuran PDO asíncronos, es decir, aquellos que no se actualizan en cada ciclo. Esta opción solo está gestionada en tarjetas TMSCan y TMSCan+. La recepción de los PDOs asíncronos tiene lugar en el código GPL a través de la instrucción [RECEIVEPDO](#).

Características de la gestión EtherCAT en Albatros

La modalidad de comunicación es siempre DC-Synchronous. El primero nodo de la red da el clock, por este hecho es necesario asegurarse que aquel nodo ofrezca un clock preciso y estable, como se ofrece, por ejemplo, por TRS-CAT. No se puede utilizar otras modalidades, como, por ejemplo, el Free-Run. Los protocolos gestionados son: CoE (CAN application protocol over EtherCAT) y EoE (Ethernet over EtherCAT). Dentro de CoE, se gestionan los dispositivos de perfiles DS401 y DS402, con modalidad operativa predeterminada de control eje de modo *Velocidad síncrona cíclica*.

El número máximo de nodos EtherCAT es 200.

Premisa

A cada dispositivo físico EtherCAT se asocia un archivo ESI (EtherCAT Slave Information) que describe las características y las funcionalidades que el dispositivo expone. El archivo es en formato XML. Para cada dispositivo hay que haber un solo archivo ESI. En general, los archivos ESI se pueden descargar desde el sitio web del constructor. Albatros busca estos archivos en la carpeta definida en Tpa.ini en la sección [tpa] a la opción DirESIFiles. La opción predeterminada es la subcarpeta "\EtherCAT" de SYSTEM. Desde los archivos ESI Albatros toma las informaciones sobre el dispositivo, analizando todos los elementos "/Devices/Device/Type". Cada dispositivo se identifica por un Vendor ID, un Product ID y un Revision Number. Siempre desde los archivos ESI se obtienen las informaciones sobre expansiones (también llamadas módulos) de los dispositivos. Albatros encuentra las informaciones relacionadas a los tipos de expansiones al buscar en el archivo ESI del dispositivo los elementos "Modules/Module".

Configuración del hardware EtherCAT

El hardware se configura describiendo las tarjetas máster y, para cada tarjeta, la lista de los dispositivos físicos conectados sobre el bus a aquella tarjeta. Los dispositivos físicos se pueden también llamar "nodos" del bus de campo. Para EtherCAT la tarjeta máster no es una tarjeta específica de control del bus, pero se utiliza una conexión de red del módulo.

Para los módulos locales, la conexión de red debe ser entre aquellas gestionadas por RTX, mientras para los módulos remotos se utiliza una conexión específica de red del módulo, entre aquellas gestionadas por Windows CE 6.0. Para cada módulo, local o remoto, se puede configurar un solo máster.

La tarjeta máster tiene que estar insertada en la configuración hardware seleccionado la rama Módulo y haciendo clic sobre el botón **[Nuevo]**.

El nodo EtherCAT

Para insertar un nodo, seleccione la rama tarjeta del árbol y haga clic sobre el botón **[Nuevo]**. Los índices de los nodos tienen que ser consecutivos, para que a cada comando **[Nuevo]** se le agrega un nodo después de los otros. Para cada nodo, hay que definir los datos siguientes:

- **Nombre del dispositivo:** seleccione el nombre del dispositivo entre aquellos listados. Solo se visualizan dispositivos con revisión más reciente; si se quieren visualizar todas las revisiones, es necesario seleccionar la opción **Visualizar revisiones**.
- **Modalidad eje:** seleccione la modalidad operativa que utilizar para los nodos de tipo accionamiento, con arreglo a la norma DS402, con respecto al objeto 6060₁₆ "modos de funcionamiento (modes of operation)". La selección es entre **Posición síncrona cíclica** y **Velocidad síncrona cíclica**. La última representa el valor predeterminado.

- **Forzar como nodo de E/S:** si activado, fuerza el control numérico para considerar el accionamiento como un nodo E/S. Este atributo solo se aplica a nodos que tienen DS402 (servo-accionamientos).

La modalidad eje es el único dato que se puede modificar después de insertar el nodo.

Un nodo EtherCAT puede ser eliminado, desplazado, deshabilitado y copiado. Los comandos son disponibles en el menú contextual de cada rama, en el menú principal de Albatros y en unos botones presentes en la ventana de configuración hardware.

El comando **Eliminar** cancela el nodo seleccionado, todas sus expansiones y conexiones virtuales-físicas, y todos los nodos siguientes se mueven a la dirección previa.

El comando **Desplazar** desplaza el nodo seleccionado a una nueva dirección. Los nodos después la dirección elegida se desplazan también de una posición.

El comando **Copiar** almacena los datos del nodo seleccionado para usarlos en un comando Pegar sucesivo.

El comando **Pegar** inserta en el árbol de los dispositivos el nodo que se ha copiado antes. El nodo se inserta en la posición elegida por el usuario. Se desplazan todos los nodos siguientes.

La expansión EtherCAT

Para insertar una expansión, seleccionar el nodo al que se quiere añadir la expansión y clicar el botón **[Nuevo]**. Los índices de las expansiones tienen que ser consecutivos, por lo que se añade una expansión en cola a otras por cada comando **[Nuevo]**. Los elementos del PDO de las expansiones no se pueden modificar.

Descripción de un PDO

Se pueden definir hasta 16 PDO enviados por el nodo (TxPDO) y hasta 16 PDO recibidos por el nodo (RxPDO). Cada RxPDO describe un solo PDO que el nodo recibe por el maestro, o sea salidas digitales y analógicas para los nodos de E/S o velocidad destino y controlword para nodos eje. Cada TxPDO describe un solo PDO que el nodo envía al maestro, o sea entradas digitales y analógicas para los nodos de E/S o posición actual y palabra de estado para nodos eje.

Para el listado y la descripción de los PDO y de los objetos asignables en un PDO hay que hacer referencia a la documentación del dispositivo específico EtherCAT y a su archivo ESI.

Hay tres modalidades para describir los PDOs de un nodo:

- No indicar algún PDO.
De esta manera el control numérico utiliza los PDO configurados de manera predeterminada en el dispositivo.
- Solo indicar los PDOs sin proveer un listado de los objetos.
A usar cuando un CN tiene diferentes PDOs alternativos y no programables.
- Describir de manera completa el PDO, indicando el objeto de comunicación y la lista de los objetos que tienen que ser trazados en mapa.
Esta modalidad es la que permite el mayor control sobre las informaciones que se envían y se reciben por el nodo.

Cada objeto está descrito por su índice en el object dictionary del CN, opcionalmente seguido por un subíndice. Si el subíndice no está presente, se considera que sea 0.

El objeto del diccionario (object dictionary) es el corazón de cada dispositivo. Habilita el acceso a todos los tipos de datos del dispositivo, a los parámetros de comunicación y a los parámetros de configuración y elaboración de datos.

Atención: no todos los objetos del object dictionary se pueden trazar en el mapa en un PDO.

Modificar un PDO de un accionamiento

En cuanto a los nodos de servo-accionamientos, hay un PDO para cada eje, así que el enésimo TxPDO y el enésimo RxPDO del nodo hacen referencia al enésimo eje del nodo. Los primeros dos objetos de cada RxPDO y TxPDO tienen un significado y un tamaño asignados anteriormente. La tabla siguiente describe como configurar los PDO de los ejes de un mismo nodo, controlados en modalidad eje **Velocidad síncrona cíclica**.

Accionamiento	RxPDO		TxPDO	
	1° objeto 16 bit Controlword	2° objeto 32 bit Target velocity	1° objeto 16 bit Statusword	2° objeto 32 bit Actual position
1° eje	6040 ₁₆	60FF ₁₆	6041 ₁₆	6064 ₁₆
2° eje	6840 ₁₆	68FF ₁₆	6841 ₁₆	
enésimo eje	Agregar 800 ₁₆ a cada objeto del eje que lo precede.			

La tabla siguiente describe como configurar los PDO de los ejes de un mismo nodo, controlados con modalidad eje **Posición síncrona cíclica**.

Accionamiento	RxPDO		TxPDO	
	1° objeto 16 bit Controlword	2° objeto 32 bit Target position	1° objeto 16 bit Statusword	2° objeto 32 bit Actual position
1° eje	6040 ₁₆	607A ₁₆	6041 ₁₆	6064 ₁₆
2° eje	6840 ₁₆	687A ₁₆	6841 ₁₆	6864 ₁₆
enésimo eje	Agregar 800 ₁₆ a cada objeto del eje que precede.			

Para modificar los objetos de un eje, seleccione el eje en el árbol y haga clic sobre el botón **[Modificar]**. En el cuadro de diálogo se visualizan el índice de los PDOs de transmisión, el índice del PDO de recepción y la lista de los objetos configurados.

En particular, los datos del cuadro de diálogo son:

- **Índice PDO de transmisión:** es el valor del PDO de transmisión del eje. No es un valor modificable.
- **Objetos:** lista de los objetos del PDO de transmisión. Los primeros dos objetos se determinan según la modalidad eje seleccionada.
- **Lista objetos:** lista de los objetos del PDO de transmisión que se pueden utilizar. Para cada objeto, se definen, si disponibles, además del índice, también el subíndice y la descripción.
- **Índice PDO de recepción:** es el valor del PDO de recepción del eje. No es un valor modificable.
- **Objetos:** lista de los objetos del PDO de recepción. Los primeros dos objetos se determinan según la modalidad eje seleccionada.
- **Lista objetos:** lista de los objetos del PDO de recepción que se pueden utilizar. Para cada objeto, se definen, si disponibles, además del índice, también el subíndice y la descripción.

Los botones **[Arriba]** y **[Abajo]** sirven para modificar el orden de los objetos en la lista de la ventana **Objetos**.

El botón **[Agregar]** permite insertar un objeto que no está entre los presentes en la **Lista objetos**.

Para modificar el valor de un objeto, seleccione el objeto y haga doble clic con el ratón. La sintaxis para definir un objeto nuevo es la siguiente:

- **número objeto:** es el número del objeto en formato hexadecimal.
- **Subíndice:** es el número del subíndice. Hay que separarlo del número objeto a través del carácter `\'` (punto). Este es un valor opcional que, si no está definido, se asume 0 como valor predeterminado. Este es en formato decimal.
- **L:** dimensión del objeto en bits. Tiene que ser un múltiplo de 8.

Los valores no presentes se extraen de los datos del object dictionary, o se asignan valores predeterminados.

Ejemplo:

objeto completo también de subíndice y longitud: 1600.1L16

objeto sin subíndice: 1601L24

objeto simple: 1603

Ciertos valores se pueden leer además a través del GPL y de la instrucción GETAXIS, a la que se refiere. Es posible además trazar los objetos adicionales tanto por la ventana de calibración como por el osciloscopio.

De forma más general, desde GPL se puede acceder en lectura y escritura a objetos específicos dentro de un PDO a través de las instrucciones [GETPDO](#) y [SETPDO](#), a la que se refiere.

PDO adicionales

Para definir los PDOs adicionales de transmisión y recepción, seleccione el nodo del árbol y haga clic sobre el botón **[Modificar]**.

En el cuadro de diálogo, para agregar un PDO de transmisión, haga clic sobre el botón **[TPDO adicionales]**, mientras que para agregar un PDO de recepción, haga clic sobre el botón **[RPDO adicionales]**.

La ventana de diálogo para la configuración de un PDO adicional se parece al cuadro de diálogo para insertar los objetos.

En este cuadro, se puede seleccionar el índice PDO y los objetos que asociar. Si el PDO tiene unos objetos obligatorios, que no se pueden modificar, estos se pueden visualizar y todos los botones de modificación están desactivados. El nuevo PDO se agrega en el árbol después de los accionamientos. Para modificar los datos, seleccione y haga clic sobre el botón **[Modificar]**. Para cancelar un PDO adicional, hay que seleccionar el PDO en el árbol y, luego, el botón **[Eliminar]**.

Adquisición automática nodos EtherCAT

Si en la configuración hardware está el bus EtherCAT, se puede adquirir de la red los nodos conectados, a través del comando **Adquisición automática nodos**, que se puede activar desde el menú contextual. Para que el comando pueda ser ejecutado, los datos de la configuración hardware y los datos presentes en el control numérico tiene que ser alineados, y los archivos ESI que describen el nodo sobre la red tiene que estar en la carpeta definida en Tpa.ini, en la sección [tpa], entrada DirESIFiles.

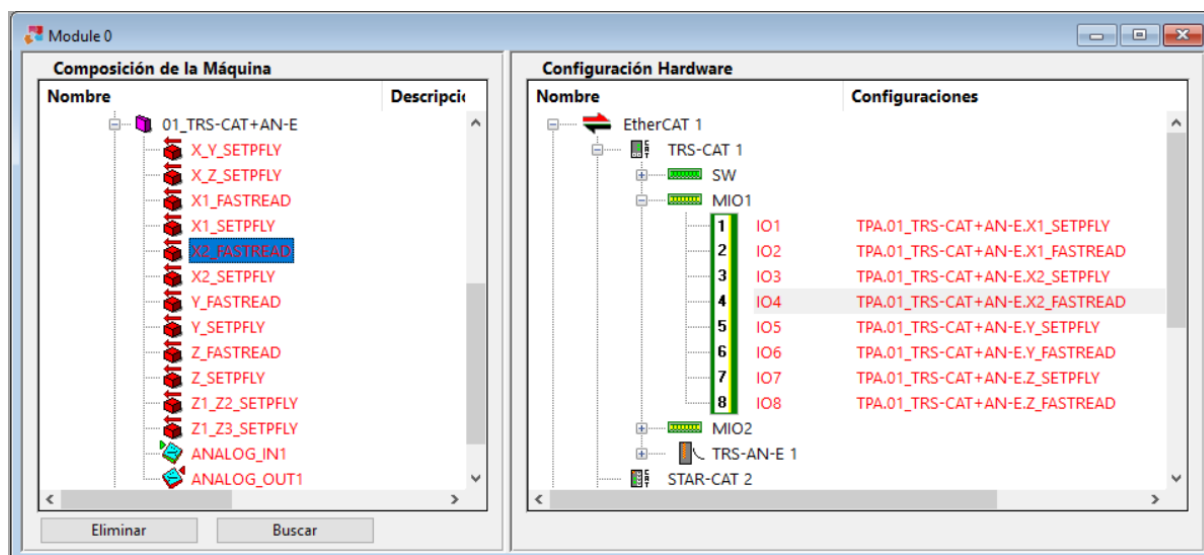
La adquisición de los nodos EtherCAT desde la red sigue las reglas enumeradas:

- ningún nodo está configurado en la configuración hardware: los datos relativos a los nodos son leídos desde la red y mostrados en configuración
- unos nodos EtherCAT ya están insertados en la configuración hardware:
 - si el nodo leído desde la red tiene la misma dirección, el mismo Vendor ID, el mismo Product ID y el mismo Revision Number que el nodo configurado, entonces se mantiene el nodo configurado y todos sus PDOs
 - si el nodo leído desde la red tiene la misma dirección que el nodo configurado per Vendor ID diferente, o Product ID diferente, entonces se elimina el nodo de la configuración con su virtual-físico, y se sustituye por el nodo de red. Si el nodo de red es un accionamiento, los PDO se leen desde el archivo ESI, en el caso contrario se leen desde el nodo de red.
 - si el nodo leído desde la red tiene la misma dirección, el mismo Vendor ID, el mismo Product ID sino Revision Number diferente que el nodo configurado, entonces se sustituye sólo el número de revisión, y se mantienen los PDOs.
 - si el nodo definido en configuración no tiene un nodo correspondiente en la red EtherCAT que ocupa la misma dirección, entonces se deshabilita en la configuración, y se mantiene el virtual-físico.

7.4.3 Configuración Virtual-Físico

La Configuración Virtual-Físico es la última fase de la configuración, y consiste en la unión de los dispositivos lógicos a los componentes hardware.

Abriendo la configuración Virtual-Físico aparecen dos ventanas: a la izquierda la de la Composición de Máquina (virtual), a la derecha la de la Configuración Hardware (físico). En ambas, todos los elementos que componen el sistema son representados de forma gráfica y en una estructura en árbol.



Configuración Virtual-Físico

Las comunicaciones Virtual-Físico ya realizadas están marcadas en la ventana "Composición Máquina", por el Nombre del dispositivo en color rojo, mientras están marcadas en la ventana "Composición Hardware" por el nombre del tipo de señal, que sigue el número del borne, ese también de color rojo.

Para cada eje de una tarjeta MECHATROLINK-II se pueden configurar 6 entradas y 1 salida digital en Virtual-Físico. Para una descripción detallada, véase el capítulo **Lenguaje GPL->Instrucciones ->MECHATROLINK-II->MECGETSTATUS**.

Si en un módulo hay el bus EtherCAT, es también posible configurar tarjetas para el bus MECHATROLINK-II,

pero con unas limitaciones: con tiempo real a 1 ms no se puede juntar más de seis ejes MECHATROLINK-II (para cada bus); con tiempo real a 2 ms el límite llega a 16 ejes.

Los dispositivos o los bornes que se deben todavía juntar son de color negro. Los dispositivos que pertenecen a un grupo deshabilitado del que quisimos guardar la configuración en virtual-físico son representados de color gris.

Las señales correspondientes a los ejes, en la ventana "Composición Hardware", están precedidas cada una de un rectángulo cuyo color corresponde al color de la vaina del hilo al interno del cable de conexión. Es posible evidenciar una conexión seleccionando un dispositivo lógico (o un componente hardware) y presionando la barra espaciadora, la conexión está evidenciada como una línea roja que une el dispositivo al componente hardware. Además, es posible mantener siempre activa la visualización de las conexiones con el comando **[Alt+Entrar]**.

Para visualizar cual dispositivo lógico está conectado al componente hardware, es necesario seleccionar el componente hardware y ejecutar un doble clic del ratón.

Para seleccionar el dispositivo lógico y el dispositivo físico que conectar, son posibles diferentes procedimientos:

primer procedimiento

- Vuelva visible en la pantalla, en la ventana "Composición Hardware", el borne físico al cual se quiere conectar el dispositivo
- Seleccione, o haga clic sobre el dispositivo lógico deseado en la ventana "Composición Máquina".

segundo procedimiento

- Seleccione, o haga clic sobre el dispositivo virtual deseado en la ventana "Composición Máquina"
- Seleccione el comando desde menú **Edición->Buscar el dispositivo físico adecuado** o la combinación de teclas **[Ctrl+espacio]**. En automático Albatros visualiza en la ventana "Configuración hardware" el primer dispositivo físico libre a lo que se puede conectar el dispositivo lógico.

tercer procedimiento posible

- Seleccione, o haga clic sobre el dispositivo virtual en la ventana "Composición Máquina"
- Seleccione el comando desde menú **Edición->Buscar el dispositivo sucesivo no vinculado** o la combinación de teclas **[Ctrl+TecNum+]** o el comando **Edición->Buscar el dispositivo anterior no vinculado** o la combinación de teclas **[Ctrl+TecNum-]**.

Para realizar la conexión entre los dos dispositivos seleccionados:

- En el dispositivo lógico que hay que ser conectado, hacer clic con el botón izquierdo del ratón y, teniéndolo presionado, arrastrar el ratón en la dirección del borne. Aparecerá una línea roja que evidencia la conexión en curso. Una vez llegados a la línea del borde, dejar el botón para concluir la operación
- seleccione el comando **iVincular!** desde menú **Edición** o la combinación de teclas **[Ctrl+L]**.

Para eliminar una conexión, hay que seleccionar el dispositivo o el componente interesado y presionar el botón **[Eliminar]** o la tecla **[Supr]**.

7.4.4 Mapas de cableo

Efectuada la conexión entre dispositivos virtuales y los correspondientes dispositivos físicos es posible imprimir unos mapas o listas en los cuales está reconducida la asociación entre dispositivos físicos y virtuales.

Para poder efectuar esta operación es necesario que en el sistema esté instalado MS-Word (versión 6 o sucesiva), Albatros disfruta de hecho de sus funciones para formatear los mapas.

El sistema, además, tiene que estar configurado correctamente. Esto significa que tienen que estar presentes en el sistema los archivos modelo utilizados para compilar los mapas. Estos son una serie de archivos con extensión ".doc" que están normalmente en la carpeta System o en otra carpeta de la instalación (típicamente la carpeta "Mapas"). Lo importante es que la carpeta donde se encuentran estos archivos se corresponda con la especificada en el archivo TPA.INI con la clave "DirMaps". Por ejemplo:

```
[TPA]
DirMaps=C:\Albatros\Mapas
```

Para imprimir los mapas de cableo hay que seleccionar un cualquier componente hardware que está en la ventana a la derecha de la [configuración Virtual-Físico](#) o que hay en la ventana de [Configuración hardware](#).

Presionado el botón "Imprimir" de la barra de instrumentos o seleccionando la opción **Imprimir** del menú **Archivo**, aparece la normal ventana de selección de las opciones de impresión. Efectuadas nuestras selecciones, hay que confirmarlas presionado el botón **[Aceptar]** y aparecerá una ventana que contiene una lista de los componentes hardware presentes en la configuración.

En esta ventana podemos seleccionar los componentes para los cuales queremos que se imprima el mapa de cableo.

Presionando en fin el botón **[Aceptar]**, se imprimen los mapas de cableo. Si se cancela la selección de la

opción **Imprimir en papel**, los mapas se guardan como documentos MS-Word en la carpeta del módulo corriente (Mod.0, etc.).

Ya que hay un alto número de páginas que pueden ser enviadas a la impresora, se recomienda efectuar una prueba imprimiendo el mapa de un solo componente hardware controlando que todo funcione. Si en lugar de los mapas se imprime una lista de dispositivos lógicos, probablemente no ha sido seleccionado un componente (por ejemplo, una tarjeta ejes o un módulo remoto) en la ventana relativa al hardware.

7.5 Lista de teclas para navegar dentro de una estructura de árbol

Tecla	Descripción
Flecha Arriba	mueve la selección a la línea que se encuentra directamente encima o debajo de la línea actual
Flecha Abajo	
Flecha Derecha	expande de un nivel la rama seleccionada y, si expandida ya, expande la selección en la rama siguiente
Flecha Izquierda	cierra la rama seleccionada y, si cerrada ya, desplaza la selección a la rama anterior
+	expande de un nivel la rama seleccionada
-	cierra la rama seleccionada
*	expande todos los niveles de la rama seleccionada


8 Instrumentos de desarrollo

8.1 Editor GPL

8.1.1 Funciones del editor GPL

El editor GPL es el instrumento que permite la creación y las modificaciones de los archivos que contienen el código GPL de Albatros. Esta función es activa solo con un nivel de contraseña igual o superior a la de Constructor. A cada archivo de funciones son asignadas unas informaciones visualizables desde el menú **Archivo->Informaciones**.

Las maneras de funcionamiento son las típicas de un editor de texto, por lo tanto, tenemos las instrucciones de **Copiar**, **Pegar**, **Buscar**, **Reemplazar**, etc. Todos estos comandos se pueden seleccionar desde el menú **Modificar**.

Deshacer	Permite, cuando es posible, anular la última operación realizada. Se restablece la situación anterior a la última operación realizada.
Repetir	Restablece la situación antes del último comando Deshacer .
Cortar	El texto o los datos seleccionados son eliminados y copiados en una memoria temporal para permitir su eventual inserción con el comando Pegar .
Copiar	El texto y el elemento seleccionado es copiado en una memoria temporal para que pueda ser insertado mediante el comando Pegar .
Pegar	El contenido de la memoria temporal es insertado, utilizando criterios diferentes según la función activa.
Eliminar	El texto o las líneas u el elemento seleccionado son eliminados. Se puede recuperar lo que ha sido cancelado seleccionando de inmediato el comando Cancelar .
Seleccionar todo	permite seleccionar todo el resto del fichero activo. Pueden aplicarse a las líneas seleccionadas los comandos Copiar , Cortar , Pegar .
Buscar...	Búsqueda de un texto en el documento corriente. Es posible establecer los criterios que se utilizarán en la fase de búsqueda: la dirección de búsqueda, la distinción entre caracteres en mayúsculas y minúsculas, la búsqueda de la palabra completa y búsqueda usando expresiones regulares.
Buscar sucesivo	Permite repetir una búsqueda anterior y mediante el comando Buscar da la posibilidad de modificar los criterios de búsqueda programados.
Sustituir...	Permite buscar un texto en el documento corriente y reemplazarlo con otro texto.
Insertar dispositivo	Inserta un dispositivo desde la lista de dispositivos. Esta función es cómoda de manera particular, cuando se trabaja con un número elevado de dispositivos, por lo tanto, puede ser difícil recordar sus nombres. Se visualizan solo los dispositivos del módulo corriente, que pueden ser llamados y todos los dispositivos públicos de otros módulos.
Insertar función	Se inserta una función o parte de una función, a partir de la posición del cursor, mediante la lectura de un archivo prototipo, escrito por el constructor de la máquina. Pueden ser escritos más archivos prototipo. El archivo prototipo es un archivo de texto, cuyo nombre debe empezar con el código GPL y la extensión TXT, Tiene que estar memorizado en la carpeta donde se almacenan las bibliotecas (normalmente system\lib). Si se definen más archivos prototipo, a la selección del comando se abre un cuadro de diálogo en el que se presenta la lista de los nombres de los archivos prototipo sin prefijo y sin extensión. Los archivos prototipo pueden contener, por ejemplo, definiciones de constantes utilizadas comúnmente, funciones de gestión de errores de sistema, funciones genéricas, códigos que implementan algoritmos para uso general y así sucesivamente. También pueden contener comentarios. Se puede crear un archivo prototipo, creando un texto seleccionado en el archivo de funciones GPL. Este comando solo está disponible como abreviado de teclado [Ctrl+Mayús+C]. Se abre un cuadro de diálogo para introducir el nombre para el fragmento de código.
Insertar mensaje	Inserta en el texto GPL el código numérico asociado al mensaje elegido. Permite insertar nuevos mensajes en el fichero de lengua.
Activar/Desactivar salto de página	Inserta o borra un salto de página  . El salto de página puede ser utilizado como un marcador para ir a posiciones importantes dentro del archivo de funciones.

Ir al siguiente salto de página

Mueve el cursor de edición a la línea del salto de página siguiente en comparación con su ubicación.

Ir al anterior salto de página

Mueve el cursor de edición a la línea del salto de página anterior en comparación con su ubicación.

```

3
4 Function AbsMovemnt
5     param axisname as axis
6     param speed as float
7     param position as double
8
9     iftarget axisname goto move
10    ifstill  axisname goto move
11    fret
12    move:
13        setvel  axisname, speed
14        movabs  axisname, position
15        waitstill axisname
16        fret
17

```

Editor GPL

La comprobación de la corrección de la sintaxis está ejecutada en fase de archivado, cuando también el texto es compilado. Es de todas formas posible un primer análisis visual de parte del programador, ya que el texto es visualizado en un color diferente según lo que representa. Por ejemplo, las instrucciones están coloradas de azul, los comentarios de verde, las etiquetas de rojo.

Se puede modificar el valor de las tabulaciones desde menú **Opciones->Tabulaciones....** Se pueden definir dos tipologías de tabulaciones:

- tabulaciones absolutas: configuran la posición inicial para las instrucciones de código GPL, la posición inicial del primer argumento de las instrucciones y la posición inicial para el comentario.
 - tabulación relativa (espacio): establece con qué número de espacios corresponde una tabulación.
- Las tabulaciones representan una ayuda para hacer más comprensible de manera visual el código GPL.

Para cada instrucción o palabra clave es disponible la ayuda en línea como soporte en la redacción de la función. Para llamar ayuda es suficiente posicionar el cursor en la instrucción y teclear **[F1]**.

En cada línea de texto se puede escribir solo una instrucción. Es posible seguir con la instrucción en la línea siguiente pulsando el carácter '_' (precedido por un espacio) como el último de la línea. Esto permite insertar unos comentarios en medio de una instrucción:

```

Mess           _
age
1000 ;código del mensaje que _
        se visualizará

```

- 3 ;casilla de sinóptico en [Entrar].
la que se visualizará

Uso de las expresiones regulares

En la ventana **Buscar** y en la ventana **Reemplazar** se puede utilizar las expresiones regulares. Una expresión regular es una secuencia de caracteres numéricos y alfanuméricos usados para buscar y reemplazar porciones de texto dentro de un texto más grande. Albatros usa la gramática de las expresiones regulares de ECMAScript. En este párrafo se describen los usos principales de las expresiones regulares en Albatros.

La expresión regular más simple è la representada por un solo carácter. Hay unas excepciones, representadas por los caracteres siguientes:

- **.** (**punto**): el punto busca cualquier carácter. Por ejemplo "A.." busca todos los casos de la letra A mayúscula seguida por dos caracteres cualesquiera.
- **[]** (**corchetes**): los corchetes permiten especificar una lista de caracteres en ellos. En el texto se buscan casos para cada carácter presente. Si el primer carácter es **^** (**acento circunflejo**) se buscaran todos los caracteres, salvo los listados entre los corchetes.

Por ejemplo:

[<>]: se buscan todos los casos del carácter < y del carácter >.

[.]AX: se buscan todas las ocurrencias que tienen la cadena "AX" dentro.

[a-d]: se buscan todos los casos de los caracteres a, b, c y d. El guion indica un conjunto de caracteres.

[\[]]: se buscan todos los casos del carácter [y del carácter].

[^+]: se buscan todos los caracteres salvo el carácter +.

- ***** (**asterisco**): se buscan todos los casos de un carácter (o conjuntos de caracteres) que se encuentran antes del asterisco. El asterisco afecta sólo el carácter antes de él: para hacerlo afectar un grupo de caracteres, se necesita utilizar las paréntesis.

Por ejemplo:

;-*: se buscan todos los casos de los caracteres ; y ;- y ;-----.

- **+** (**signo más**): busca uno u más casos del carácter antes de él. Se distingue del uso del asterisco porque el carácter antes del signo + tiene que estar siempre. El signo más afecta sólo el carácter antes de él: para hacerlo tener efecto sobre un conjunto de caracteres, se necesita utilizar las paréntesis. Vamos a reutilizar el ejemplo usado para el carácter asterisco:

;-+: se buscan todos los casos de los caracteres ;- y ;---. No se busca el carácter ; (**punto y coma**) solo.

- **?** (**signo de interrogación**): vuelve opcional el carácter antes de él, que entonces puede ocurrir como mucho una vez.

Por ejemplo:

Setfeedi?: se buscan todos los casos de la palabra Setfeed y de la palabra Setfeedi.

- **{ }** (**llaves**): precisan cuántas veces un carácter (o un conjunto de caracteres) hay que encontrarse dentro de un texto.

Por ejemplo:

ee{2}: se buscan todos los casos de dos ee consecutivas.

- **^**. (**acento circunflejo y punto**): busca el primer carácter de cada fila.
- **^** (**acento circunflejo**): busca el término solicitado sólo si está al comienzo de una fila
- **|** (**pleca**): busca los términos que se encuentran tanto antes como después del carácter '|'.
Por ejemplo:
Send|Const: se buscan todos los casos de la palabra Send y de la palabra Const.
- **** (**barra inversa**): la barra inversa tiene dos funciones:

- 1) vuelve el carácter normal en una función.
 - \b busca el borde inicial o final de la palabra
 - \B busca la palabra salvo el borde inicial
 - \d busca sólo las cifras
 - \D busca todos salvo las cifras
 - \s busca el espacio
 - \S busca todo salvo el espacio
 Ejemplos:
 - \bi busca todas las palabras en el texto que empiezan por la letra i.
- 2) vuelve un carácter especial en un carácter normal. Por ejemplo, para buscar en un texto el carácter especial **^ (acento circunflejo)** es suficiente poner la barra inversa antes.

8.1.2 Insertar mensaje

En Albatros hay dos tipos de mensajes: los mensajes de módulo y los mensajes de grupo.

El comando se puede seleccionar desde menú **Edición->Insertar Mensaje**.

Los mensajes de grupo son insertados directamente en el editor durante la escritura del código GPL, a través de la instrucción **DEFMSG**. Son visibles y utilizables solo al interno del grupo en el cual son definidos, por lo tanto, es posible tener la misma definición de mensaje en más grupos, sin que se verifique ninguna sobre-posición.

Los mensajes de módulo, a diferencia de los de grupo, pueden ser utilizados por todos los grupos. Pueden ser insertados utilizando un cuadro de diálogo que permite llamar un mensaje ya presente en el archivo de idioma o insertar uno nuevo.

Ventana de gestión de mensajes

En esta manera no hay que preocuparse de modificar el archivo. El mensaje estará insertado en el idioma actual y posteriormente será traducido en otros idiomas (utilizando el programa TpaLangs).

Todos los mensajes presentes en los archivos de idioma son enumerados en la opción **Descripción**. Para introducir un mensaje dentro de una función hay que seleccionar, una vez elegido el texto, el botón **[Modificar Texto]**.

Para modificar un mensaje existente **[Modificar]** o crear uno nuevo **[Nuevo]**, como primera operación hay que teclear la modificación o el nuevo texto y seleccionar el botón correspondiente.

8.1.3 Cifrado

En Albatros es posible volver a visualizar el texto de origen de las funciones con el cifrado.

La habilitación del cifrado tiene lugar configurando en tpa.ini la opción **Tele+=1**. El valor predeterminado es 0, que indica que el cifrado no está habilitado.

Cuando se guarda un archivo de funciones y el cifrado está habilitado, será visualizado el siguiente mensaje: "¿Desea cifrar el archivo?" Si se contesta **[Sí]**, se cifra el archivo. Es posible cifrar un archivo que previamente se guardó sin cifrar, aunque un archivo cifrado siempre seguirá siendo tal.

Con el cifrado habilitado, si la contraseña es de tipo Constructor diaria, un archivo no codificado no será cifrado.

Un archivo de funciones cifradas puede ser visualizado o modificado en Albatros sólo por el usuario que lo había guardado antes. ¡El propietario de un archivo de funciones cifradas no puede cambiar!

Si se quiere quitar el cifrado de un archivo, hay que usar un programa externo llamado SBIANCA, que se encuentra en la carpeta Bin de Albatros.

Un modo de utilizo del programa Sbianca es a través de la interfaz gráfica. Se seleccionan los archivos que se desea descifrar. Por cada archivo se muestran las propiedades **Estado** y **Credenciales**. El **Estado** puede ser **Texto sin formato** si el archivo no es cifrado, o **Cifrado** si el archivo está cifrado. La propiedad **Credenciales**

brinda informaciones sobre la visibilidad del archivo: **Legible** indica que se puede visualizar el archivo desde el nivel de contraseña actual, de lo contrario, **Bloqueado** indica que no se puede visualizar el archivo. Después de elegir los archivos, habrá que hacer clic sobre **[iDescifrar!]** para descifrarlos o **[iCifrar!]** para cifrarlos.

Otra modalidad de utilizo es desde línea de comando. Los comandos y los archivos son pasados como argumentos al programa. Si no están argumentos, se activa el modo de uso de la interfaz gráfica. La sintaxis es:

```
[-l|-e|-d] archivo1 archivo2...
```

Con:

- -l, o ninguna opción, para visualizar el estado de cada archivo
- -d para descifrar los archivos
- -e para cifrar archivos

Al final de la ejecución, el programa envía los resultados de la operación solicitada en salida. La salida del programa es en formato Markdown.

8.1.4 Lista de teclas a selección rápida disponibles



Borrar un texto

Tecla	Descripción
Retroceso	elimina un carácter a la izquierda o elimina el texto seleccionado
Ctrl+Retroceso	elimina la palabra a la izquierda
Supr	elimina un carácter a la derecha o elimina el texto seleccionado
Ctrl+T	elimina las palabras o los espacios a la derecha
Ctrl+Supr	elimina la palabra a la derecha y todos los espacios siguientes hasta el principio de la nueva palabra



Comentario de más filas de texto

Tecla	Descripción
Ctrl+';': Con los teclados italianos hay que presionar [Mayús] también	agrega o quita el carácter de comentario a las filas seleccionadas.



Posicionar el cursor

Tecla	Descripción
Flecha Arriba	desplaza el cursor en el sentido seleccionado
Flecha Abajo	
Flecha Derecha	
Flecha Izquierda	
Inicio	desplaza el cursor alternativamente al principio de la línea y al primer carácter de la línea.
Fin	desplaza el cursor al final de la línea
Ctrl+Inicio	desplaza el cursor al comienzo del documento
Ctrl+Fin	desplaza el cursor al final del documento
Ctrl+Flecha Izquierda	desplaza el cursor una palabra a la izquierda
Ctrl+Flecha Derecha	desplaza el cursor una palabra a la derecha
Ctrl+Entrar	posiciona el cursor al primer carácter de la línea siguiente



Seleccionar

Tecla	Descripción
Mayús+Inicio	selecciona hasta el principio de la línea a partir de la posición del cursor
Ctrl+Mayús+Inicio	selecciona hasta el principio del documento a partir de la posición del cursor
Ctrl+Mayús+Fin	selecciona hasta el final del documento a partir de la posición del cursor

Ctrl+Mayús+Flecha Izquierda	selecciona la palabra o los espacios a la izquierda del cursor
Ctrl+Mayús+Flecha Derecha	selecciona la palabra o los espacios a la derecha del cursor
Mayús+Re Pág	selecciona una página hacia arriba a partir de la posición corriente del cursor
Mayús+Av Pág	selecciona una página hacia abajo a partir de la posición corriente del cursor
Ctrl+W	selecciona la palabra en correspondencia de la que el cursor está posicionado
Ctrl+A	selecciona todo el documento

☐ Selección rectangular

Tecla	Descripción
Alt+ Mayús+Flecha Arriba Mayús+Flecha Abajo Mayús+Flecha Izquierda Mayús+Flecha Derecha	selecciona un bloqueo rectangular de código
o	
Alt+selección con el ratón	

☐ Tabulaciones

Tecla	Descripción
Tab	si ningún texto está seleccionado, inserta el espaciado entre caracteres en Opciones->Tabulaciones . Si han sido seleccionadas más líneas, inserta a la derecha el espaciado programado para la tabulación relativa. Si está seleccionada una sola línea el texto es borrado
Mayús+Tab	si ningún texto está seleccionado, desplaza el cursor a la izquierda del espaciado definido en Opciones->Tabulaciones . Si ha sido seleccionada, una o más líneas son insertadas a la izquierda del espaciado programado para la tabulación relativa

☐ Copiar y Pegar

Tecla	Descripción
Ctrl+C Ctrl+Insert	copia el texto seleccionado al Portapapeles
Ctrl+X Mayús+Supr	elimina el texto seleccionado y lo copia al Portapapeles
Ctrl+V Mayús+Insert	inserta el contenido del Portapapeles a partir de la posición del cursor
Ctrl+Y	elimina la línea donde el cursor está posicionado y copia el contenido en el Portapapeles

☐ Deshacer/Restaurar

Tecla	Descripción
Ctrl+X Alt+Retroceso	anula el último teclado
Ctrl+Mayús+Z	restaura el último teclado

☐ Buscar y Reemplazar

Tecla	Descripción
Ctrl+F3	busca hacia abajo en todo el documento la palabra sobre la cual el cursor está posicionado

Ctrl+Mayús+F3	busca hacia arriba en todo el documento la palabra sobre la cual el cursor está posicionado
F3	busca el término siguiente. La ventana de diálogo Buscar debe ser cerrada
Mayús+F3	busca el término anterior. La ventana de diálogo Buscar debe ser cerrada
Alt+F3	abre la ventana de diálogo Buscar y programa como texto que buscar la palabra en correspondencia de la que el cursor está posicionado

Visualizar errores de compilación

Tecla	Descripción
Doble clic sobre el error	coloca el cursor sobre la línea de la función GPL donde el error descrito ha ocurrido
F4	coloca el cursor sobre la línea de la función GPL donde a ocurrido el error posterior al último error seleccionado
Mayús+F4	coloca el cursor sobre la línea de la función GPL donde a ocurrido el error anterior al último error seleccionado

Creación de un archivo prototipo

Tecla	Descripción
Ctrl+Mayús+C	guarda el texto seleccionado en el archivo de funciones GPL. Se abre una ventana de diálogo para insertar el nombre para el fragmento de código

Gestión del listado

Tecla	Descripción
Ctrl+M	Expande o contrae el listado seleccionado

8.2 Librerías

Una librería consiste en una recolección de funciones GPL que pueden ser llamadas al interno del código GPL personalizado sin ser ligadas a una particular configuración. Las librerías son muy útiles en cuanto pueden ser fácilmente copiadas de una máquina a otra. Se evita el tener que volver a escribir código común cuando se implementan nuevas máquinas. Por ejemplo, se puede crear una librería de funciones geométricas y matemáticas.

Los archivos de librería son colocados en la carpeta `system\lib`. Son compilados cuando se ejecuta uno de los siguientes comandos: **CNC->Inicialización**, **Archivo->Compilar todo**, **Guardar** el archivo de librería o el archivo de variables globales.

Si en el código GPL de una máquina está asignado un nombre de función o de variable que ya hay en una librería, en fase de compilación tendrá siempre precedencia el de la máquina. Si el mismo nombre existe en dos librerías diferentes para identificar lo que interesa es necesario en el código GPL utilizar la siguiente sintaxis extendida: **nombrelibrería.nombrefunción**. Por ejemplo, si la función `LongitudSegmento` existe en la librería `LIBGEO` y en la librería `LIBMAT` y se quiere utilizar la función de la librería `LIBGEO`, ocurre escribir: `LIBGEO.LongitudSegmento`.

Todas las operaciones relativas a una librería están gestionadas a través del cuadro de diálogo indicado arriba. Es posible crear una **[Nueva]** librería. El nombre asignado está guardado en la lista de las librerías instaladas. Se pueden importar librerías ya existentes o bien transformar archivos de grupos en librerías, llamándolas a través de la ventana de diálogo abierta por el botón **[Importar..]**. La misma operación se ejecuta para utilizar librerías anteriormente eliminadas a través el comando **[Eliminar]**.

Para modificar el código de una librería hay que seleccionar el botón **[Modificar]**. La librería es abierta por el editor GPL. Durante la redacción de las funciones de librería hay que tener presente estas reglas fundamentales:

- no es posible acceder a dispositivos, funciones, variables propias de la configuración al interno de las que se está escribiendo la función
- es posible llamar funciones y variables públicas de otras librerías
- las funciones declaradas al interno de una librería son predeterminadamente definidas privadas. Para que puedan ser llamadas por otros archivos de función ocurre declararlas de tipo [PUBLIC](#).

La posibilidad de modificación de una librería es subordinada al nivel de acceso de quien está utilizando Albatros. Es posible asignar o modificar las autorizaciones de acceso a una librería seleccionando el botón **[Propiedades]**.

Las variables globales declaradas en una librería son visualizadas en [Diagnóstico](#) en una sección dedicada. La visualización de los elementos de la librería es ligada a los derechos de acceso de quien está utilizando Albatros.

8.3 Depuración

8.3.1 El depurador

El depurador es una función de Albatros que permite ejecutar paso a paso la secuencia de las instrucciones de una tarea GPL, permite, por lo tanto, encontrar y corregir eventuales errores lógicos y malos comportamientos del código.

Esta función solo está activada con un nivel de contraseña igual o superior a constructor.

Con la depuración es posible, por ejemplo:

- asignar puntos de interrupción
- interrumpir la ejecución de una tarea y visualizar el valor de una variable
- seguir la secuencia de ejecución de una función
- verificar el valor asumido por una variable local
- verificar que, en el caso de una instrucción If, la rama elegida sea correcta

Los comandos utilizables en modalidad de depuración pueden ser seleccionados desde el menú Depuración. Los principales son los siguientes:

Continuar	Reanuda la ejecución de una tarea bloqueada. La tarea continuará hasta que el punto de interrupción esté bloqueado de nuevo o alcance un punto de interrupción.
Reiniciar	Reinicia la depuración de la tarea seleccionada.
Interrumpir ahora	Bloquea la ejecución de la tarea que se está depurando. El cursor se posiciona en la línea en que la instrucción fue interrumpida. Cuando una tarea está bloqueada, se puede pilotar su ejecución y verificar el estado de las variables locales.
Paso a paso por instrucciones	Ejecuta cada instrucción GPL. La tarea debe haber sido interrumpida anteriormente.
Ejecutar hasta el final de la función	Ejecuta todas las instrucciones hasta la primera instrucción que sigue la función corriente.
Paso a paso por procedimientos	Ejecuta cada instrucción GPL o si la instrucción es una llamada de función ejecuta la función completa.
Ejecutar hasta el cursor	Ejecuta las instrucciones hasta la posición del cursor.
Terminar la depuración	Termina la ejecución de una. El archivo de las funciones que se están depurando se abre en modalidad edición.

Para acceder a la depuración hay que visualizar [la lista de las tareas en ejecución](#) (desde menú **Depuración->Tarea en ejecución**) o [la lista de todas las tareas](#) (desde menú **Depuración->Todas las tareas**) y luego seleccionar la tarea sobre la cual se quiere efectuar la depuración.

Antes de ejecutar la depuración es necesario que no existan errores de compilación de las funciones (por ejemplo: errores de sintaxis, variables no declaradas) y que el módulo sobre el cual se quiere efectuar la depuración haya sido inicializado correctamente.

La ventana de depuración es parecida a la del editor GPL; sin embargo, en ella no es posible aportar modificaciones al código. El fondo de la ventana es gris y la línea en ejecución es evidenciada con el color amarillo.

Atención: No es posible ejecutar al mismo tiempo la depuración de más tareas pertenecientes al mismo módulo.










8.3.2 Tareas en ejecución

Se puede seleccionar el comando desde menú **Depuración->Tarea en ejecución**. Visualiza la lista de las tareas en ejecución asociadas a una máquina o módulo. Seleccionando una tarea es posible [ejecutar la depuración](#) o terminar la ejecución, utilizando respectivamente los botones **[Depuración]** o **[Terminar]**.

8.3.3 Todas las tareas

Visualiza, en una ventana de diálogo, la lista con todas las tareas definidas en el código GPL. En el gráfico está representadas en una estructura a árbol. Cuando se selecciona una función, se abre el archivo donde esa está definida y el cursor se posiciona en correspondencia con su primera instrucción. De esa manera es posible configurar unos [Puntos de interrupción](#) aún antes del principio de la ejecución. Se puede ejecutar la depuración de cualquier tarea y función sin parámetros.

A continuación, está descrito el significado de cada símbolo empleado para la composición del árbol de ejecución de las tareas. Un símbolo particular es aquel que indica la función recursiva, o sea una función que llama en su interior la función por la que estuvo llamada.

Símbolo	Descripción
	tarea de la función principal del Intergrupo
	tarea autorun
	tarea genérica
	tarea real-time
	función de grupo
	función de grupo ejecutada por instrucciones como ONINPUT, ONFLAG
	función de librería
	función de librería ejecutada por instrucciones como ONINPUT, ONFLAG
	función recursiva

8.3.4 Llamadas a función

Durante la depuración es posible visualizar la lista de las funciones que han sido llamadas más que aún no han retornado (o sea, todas las funciones en las cuales aún no ha sido ejecutada la instrucción FRET). Aparece un cuadro de diálogo que enumera las llamadas a función que han llevado hasta la instrucción en curso. Encima a la lista están las funciones ejecutadas más recientemente.

Para observar el comportamiento de una llamada a función:

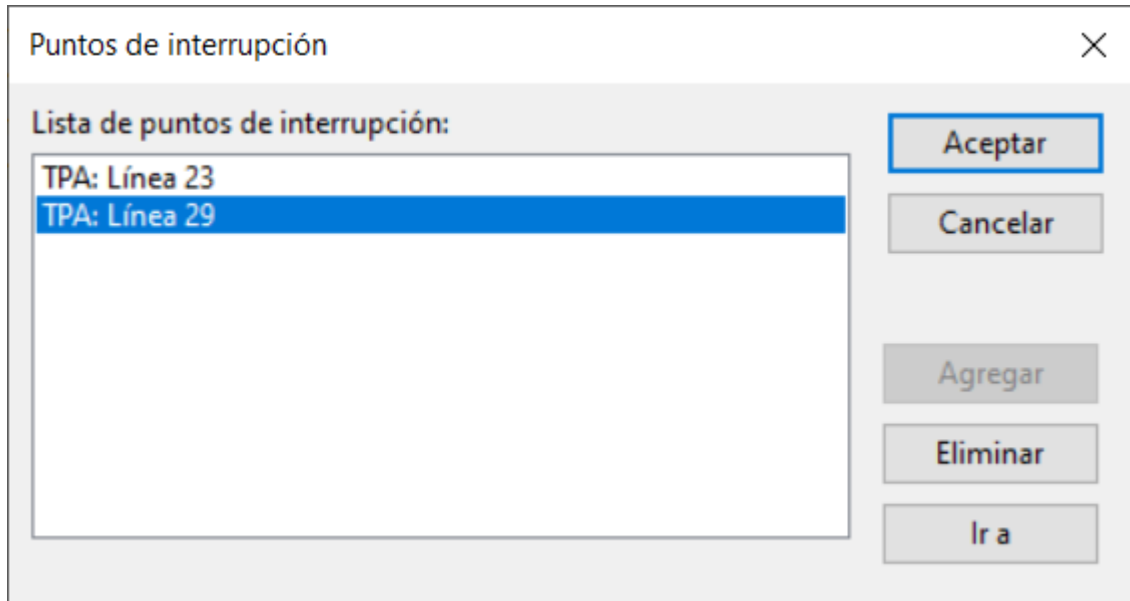
- desplace el cursor en la posición deseada en el interior de la función.
- seleccione **Depuración->Ejecutar hasta el cursor** para llevar la ejecución del programa hasta la posición deseada.
- seleccione **Depuración->Llamadas a función**, o las teclas Ctrl+K.
- desde las ventanas de diálogo Llamadas, seleccione el nombre de una función. El cursor se desplaza hacia la primera instrucción de la función seleccionada.

8.3.5 Puntos de interrupción

Un punto de interrupción permite ver de manera más detallada la secuencia de ejecución de las instrucciones, examinar o modificar variables y dispositivos, examinar la lista de las llamadas a funciones etc.

La ejecución de una tarea se bloquea cuando es alcanzada la instrucción en el cual ha sido introducido el punto de interrupción.

Los puntos de interrupción pueden ser programados tanto antes que una tarea sea ejecutada como durante su ejecución (desde menú **Depuración->Puntos de interrupción**). Es posible también eliminar unos puntos de interrupción, una vez que ya no son necesarios.



Lista de los puntos de interrupción

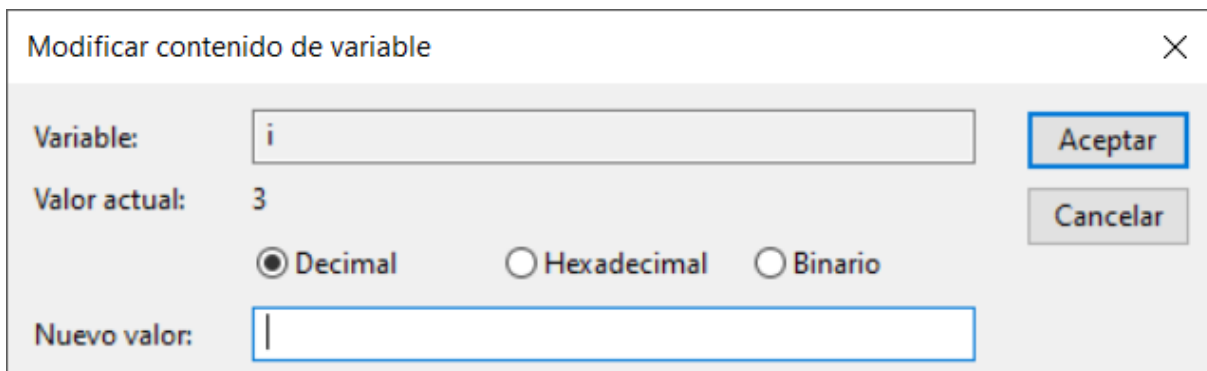
Existen situaciones en las cuales, aunque si han sido introducidos puntos de interrupción, la tarea no se bloquea porque la ejecución no alcanza nunca el punto de interrupción. En estos casos se puede bloquear la tarea con el comando **Depuración->Interrumpir ahora**. El cursor está posicionado en la instrucción GPL que estaba por ser ejecutada en el momento de la interrupción.

8.3.6 Contenido de variable

Se puede seleccionar el comando desde menú **Depuración>Contenido de variable**.

Después de haber interrumpido la ejecución de una tarea es posible visualizar:

- el valor de las variables locales declaradas en las funciones en las cuales la tarea está parada
- las variables globales
- el valor asumido por una expresión
- el estado de los dispositivos y de los parámetros de tipo dispositivo.



Visualización/Modificación del contenido de una variable

Si la variable (o el dispositivo) no es de solo lectura, su contenido es modificable: obviamente la modificación influenciará el desenvolvimiento sucesivo de la tarea.

Modificar el valor de una variable o de un dispositivo permite probar la ejecución en condiciones diferentes a la ejecución normal, corregir eventuales errores encontrados y continuar con la ejecución de las instrucciones sucesivas.

Desplazando también el ratón en la variable, en el nombre del dispositivo o bien en la constante, se puede visualizar el contenido de una variable, de un dispositivo o de una constante. Aparece una descripción donde se indica el tipo, el nombre y el valor del dato. Si se selecciona una expresión, se visualiza el resultado. Si el cursor del ratón está dentro de una selección, se usa la selección entera, en caso contrario solo la palabra en la que está el cursor del ratón. Si el cursor del ratón no está dentro de una palabra, se usa todo el argumento. Por ejemplo, para ver el valor de la matriz $Mx[3][columna]$, si el cursor del ratón está en "3" en la descripción emergente, se visualiza 3; si el cursor está en "columna" se visualiza el valor de la columna; si está en "matriz", no visualiza nada; si está sobre un corchete, visualiza el valor de $Mx[3][columna]$.

8.3.7 Lista de teclas a selección rápida disponibles

Para activar los comandos de la **Depuración** es posible seleccionar opción de menú **Depuración** u obrar directamente desde el teclado.

Las teclas que se utilizan son:

Tecla	Descripción
Ctrl+F5	Abre el cuadro de diálogo con la lista de las tareas en ejecución
Ctrl+Mayús+F5	Abre el cuadro de diálogo con la lista de todas las tareas
Ctrl+B	Abre el cuadro de diálogo para insertar o cancelar los puntos de interrupción
Ctrl+F9	Inserta o elimina un punto de interrupción en correspondencia de la línea en la que el cursor está posicionado
Ctrl+K	Abre la ventana de dialogo para visualizar la lista de las funciones que han sido llamadas, sino que aún no han retornado
Mayús+F9	Abre un cuadro de diálogo para visualizar el contenido de una variable
F8	Ejecuta la instrucción. Si esta es una función, entra en la función
Mayús+F7	Ejecuta todas instrucciones de la función
F10	Ejecuta la instrucción. Si esta es una función, la ejecuta sin entrar
F7	Ejecuta todas instrucciones hasta la instrucción en la que el cursor está posicionado. El cursor tiene que estar posicionado sobre una instrucción dentro la función
Alt+Inter	Interrumpe la ejecución del código en la última instrucción ejecutada
F5	Reanuda la ejecución del código después de una interrupción
Mayús+F5	Termina la tarea corriente y la ejecuta otra vez
Alt+F5	Termina la depuración

8.4 Inicialización del control


8.4.1 Conexiones de red

El funcionamiento de Albatros en la máquina está protegido por una clave hardware usb, configurada por TPA. Se puede seleccionar el comando desde menú **CNC->Conexiones de red**. Visualiza el estado de los módulos remotos conectados al sistema. Si un módulo no está conectado el símbolo que lo representa está marcado con una X roja.

Para cada módulo están presentes dos campos. El primer campo es el nombre del módulo y el segundo campo es el nombre de la estación de red. Habitualmente este último tiene los primeros caracteres con el nombre fijo "TPA" seguido por el número de serie del módulo remoto.

Asignación de un nodo de red (módulo remoto) a un módulo lógico

Se puede asignar un nodo de red a un módulo seleccionando con el botón del ratón la escrita "No configurado" o seleccionando el botón **[Modificar]**. Después de algunos segundos aparecerá una ventana con la lista de los módulos remotos disponibles en red (todos módulos remotos tienen que estar encendidos y deben de haber obtenido correctamente una dirección IP).

Es posible a este punto seleccionar el nodo de red que se quiere conectar al módulo lógico y confirmar la selección pulsando el botón .

Note como esta operación puede ser ejecutada con el nivel de contraseña "Asistencia", sin tener que acceder a la configuración de Sistema de Albatros que requiere, a cambio, un nivel de contraseña "Constructor".

Actualización del software de un módulo remoto

Se puede actualizar completamente el software del control almacenado en la memoria interna del módulo remoto, seleccionando el botón **[Actualizar]**. Para que Albatros actualice el módulo remoto hay que seleccionar el módulo remoto que se desea actualizar y esto tiene que conectarse y estar en comunicación con Albatros.

8.4.2 Diagnóstico hardware

Se puede seleccionar el comando desde menú **Cnc->Diagnóstico hardware**. El Diagnóstico Hardware visualiza el elenco y el estado de los módulos configurados, de las tarjetas y de los nodos remotos a ellos pertenecientes, como definidos en la configuración hardware. Si el símbolo de la tarjeta o de un nodo está marcado con una X roja, significa que esta no ha sido detectada entre el hardware presente en el control o que no ha sido posible inicializar correctamente. Si está marcada por un punto interrogativo amarillo, significa que el sistema ha detectado la presencia de una tarjeta o de un nodo, que, pero no corresponde con la tipología definida en la configuración.

Topología red EtherCAT

En la ventana del diagnóstico hardware, cuando en el *árbol* un nodo de una red EtherCAT está seleccionado, se activa el botón **[Detalles]**, que por su parte activa la visualización en formato gráfico de la topología de la red EtherCAT.

En este gráfico se visualizan las informaciones sobre los nodos físicamente presentes sobre la red: el estado de cada nodo, el estado de los ejes, si el nodo es un servoaccionamiento, y la calidad de la comunicación. Cada nodo y cada eje es representado como un rectángulo cuyo color define el estado.

Desplazando el puntero del ratón sobre el rectángulo, aparece una información sobre herramienta que describe el estado y los errores de comunicación del nodo o el estado del eje.

Visualización y modificación de objetos en los nodos

En la ventana del diagnóstico hardware, cuando en el *árbol* un nodo de una red EtherCAT está seleccionado, se activa el botón **[Object dictionary]**, para visualizar y modificar los objetos del nodo. La modificación de los datos es posible sólo al nivel **Constructor**.

Los objetos se agrupan en función de sus direcciones.

Dirección inicial	Dirección final	Nombre Área
0x0000	0x0FFF	Data type area
0x1000	0x1FFF	Communication area
0x2000	0x5FFF	Manufacture specific area
0x6000	0x6FFF	Input area
0x7000	0x7FFF	Output area
0x8000	0x8FFF	Configuration area
0x9000	0x9FFF	Information area
0xA000	0xAFFF	Diagnosis area
0xB000	0xBFFF	Service transfer area
0xC000	0xEFFF	Reserved area
0xF000	0xFFFF	Device area

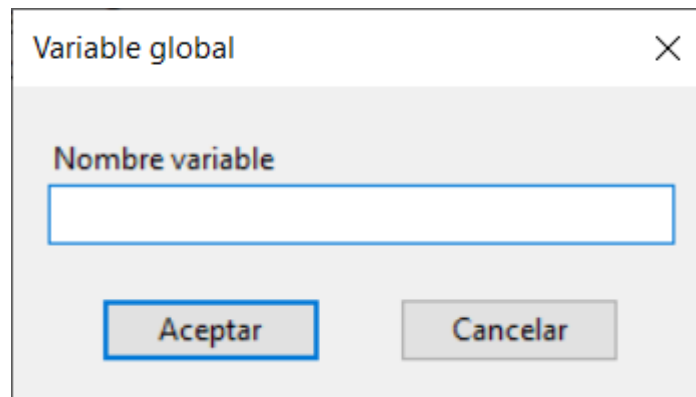
Los valores numéricos enteros son visualizados tanto en base decimal como en base hexadecimal. Cuando se modifica su valor, el número puede ser escrito también en base hexadecimal, usando la notación \$...H, y en base binaria, con \$...B (como se hace en GPL).

8.5 Test

8.5.1 Guardar variable global

Se puede seleccionar el comando desde menú **Test->Guardar variable global**.

Memorizar el contenido de una variable global en el disco en forma de un archivo de texto formateado. El archivo tiene el nombre compuesto por *nombrevariable.txt* y será guardado en la carpeta *Informe*. La operación es posible solo si el nivel de acceso en lectura de la variable global es compatible con el nivel de acceso corriente.



Almacenamiento de una variable global

8.5.2 Ejecutar función

Se puede seleccionar el comando desde menú **Test->Ejecutar función**.

Ejecuta una función de manera independiente respecto al resto del sistema. Por lo tanto, es creada una tarea que comienza su propia ejecución da la función seleccionada y que toma el nombre de ella. Son ejecutables solo las funciones que no tienen parámetros en entrada y cuyo nivel de acceso en lectura es compatible con el nivel de acceso corriente. Si la función que es ejecutada es la función principal del intergrupo, en seguida son ejecutadas también todas las tareas autorun.

8.5.3 Importar mensajes

Los mensajes de grupo asignados a través de la instrucción GPL [DEFMSG](#), son almacenados en un archivo de formato XMLNG. Los mensajes contenidos dentro del archivo pueden ser modificados, añadidos o cancelados. Para ver las modificaciones en el código GPL, se necesita utilizar el comando **Test->Importar mensajes de grupo**, que se habilita sólo cuando no hay ventanas abiertas.

Para poder ejecutar una importación todo el código GPL tiene que ser compilado sin errores. Si así no fuera, el usuario sería avisado con el mensaje "No todo el código GPL resulta compilado".

No son importados los mensajes de grupo pertenecientes a archivos [cifrados](#) (Véase el capítulo **Instrumentos de desarrollo->Editor GPL->Cifrado**) para los cuales el usuario no tiene el permiso de visualización en claro.

Solo son importados los mensajes que ya han sido definidos en el código GPL. La modificación del texto GPL no es efectuada si hay por lo menos una DEFMSG sucesiva a una instrucción IFDEF.

Durante la importación pueden ser descubiertos errores cuando:

- entre los textos de un particular mensaje de grupo, el identificativo de un idioma está presente más de una vez
- un texto cualquiera está vacío (o sea: "")
- el nombre de un grupo o de una librería es definido más de una vez.

Al terminar la importación es ejecutada la compilación de todos los módulos en que han sido modificados grupos o librerías.

A continuación, la lista de todos los idiomas que pueden ser utilizados en los archivos XMLNG:

"AFK" Afrikáans
"ARA" Árabe
"AZE" Azerí
"BAS" Baskir
"BEL" Bielorruso
"BGR" Búlgaro
"BSB" Bosnio (alfabeto latino)
"BSC" Bosnio (alfabeto cirílico)

"BRE" Bretón
"CAT" Catalán
"CHS" Chino simplificado
"CHT" Chino tradicional
"COS" Corso
"CSY" Checo
"CYM" Gaélico
"DAN" Danés
"DEA" Alemán (Austria)
"DEU" Alemán (Alemania)
"ELL" Griego
"ENG" Inglés
"ENU" Inglés (Estados Unidos)
"ESP" Español
"ETI" Estonio
"EUQ" Euskera
"FAR" Persa
"FIN" Finés
"FRA" Francés
"FPO" Filipino
"FRB" Francés (Bélgica)
"FYN" Frisón
"GLC" Gallego
"HAU" Hausa
"HEB" Hebreo
"HRB" Croata (Bosnia y Herzegovina)
"HRV" Croata (Croacia)
"HUN" Húngaro
"IBO" Igbo
"IND" Indonesio
"IRE" Irlandés
"ISL" Islandés
"ITA" Italiano
"JPN" Japonés
"KAL" Groenlandés
"KOR" Coreano
"SAH" Sakha
"KYR" Kirguís
"LVI" Letón
"LTH" Lituano
"LBX" Luxemburgués
"MNN" Mongol
"NON" Noruego Nynorsk
"NOR" Noruego Bokmål
"NLB" Neerlandés (Bélgica)
"NLD" Olandese (Países Bajos)
"OCI" Occitano
"PLK" Polaco
"PTB" Portugués (Brasil)
"PTG" Portugués (Portugal)
"RMC" Romanche
"ROM" Rumano
"RUS" Ruso
"SKY" Eslovaco
"SLV" Esloveno
"SQI" Albanés
"SRM" Serbio (alfabeto latino, Serbia)
"SRN" Serbio (alfabeto cirílico, Bosnia y Herzegovina)
"SRO" Serbio (alfabeto cirílico, Serbia)
"SRP" Serbio (alfabeto latino, Montenegro)
"SRQ" Serbio (alfabeto cirílico, Montenegro)
"SRS" Serbio (alfabeto latino, Bosnia y Herzegovina)
"SVE" Sueco
"TAJ" Tayiko
"TRK" Turco
"TTT" Tártaro
"TUK" Turcomano
"UKR" Ucraniano
"URD" Urdú
"UZB" Uzbeko
"VIT" Vietnamita

"WOL" Wólof
"XHO" Xhosa
"YOR" Yoruba
"ZUL" Zulu

8.5.4 Nota para el usuario en el archivo de informe de las alarmas

Con el comando **Insertar nota** del menú **Test** es posible insertar una notificación en el archivo de informe de las alarmas del mes presente (MONTHxx.TER). La opción del menú se habilita desde un nivel de contraseña igual o superior a asistencia.

8.6 Herramientas

8.6.1 Personalizar...

El comando se puede seleccionar desde el menú **Herramientas->Personalizar...**

Permite configurar un número máximo de 20 programas, cuya ejecución podrá ser iniciada desde el menú **Herramientas** de Albatros.

The screenshot shows a dialog box titled "Herramientas" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Estructura de Menú:** A list box containing "ViewRer.exe" which is currently selected. To the right of this list are five buttons: "Agregar...", "Aceptar" (highlighted with a blue border), "Eliminar", "Cancelar", "Mover arriba", and "Mover abajo".
- Comando:** A text input field containing "C:\Albatros\bin\ViewRer.exe".
- Texto en el Menú:** A text input field containing "ViewRer.exe".
- Argumentos:** An empty text input field.
- Solicitud de Argumentos:** A checked checkbox.
- Nivel de habilitación:** A group box containing four radio button options: "Usuario" (selected), "Asistencia", "Constructor", and "Tpa".

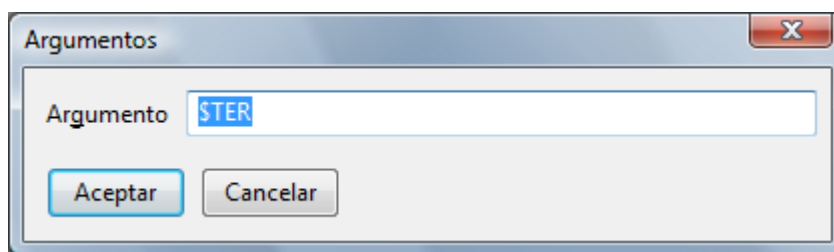
Configuración del menú Herramientas

- Estructura de Menú:** se enumeran los programas visualizados en el menú **Herramientas**.
- Comando:** nombre del programa que hay que ejecutar. Se puede definir también de manera opcional la carpeta donde está archivado el programa, si fuera diferente de la carpeta desde la que se ha ejecutado Albatros o desde las carpetas, donde el sistema operativo ejecuta la búsqueda de los archivos ejecutables (variable de entorno Windows PATH).
- Texto en el Menú:** es el nombre que se quiere que aparezca en el menú **Herramientas** para identificar el programa ejecutable.
- Argumentos:** cualquier combinación de argumentos desde línea de comando, que el programa necesita para una ejecución correcta. Es posible insertar argumentos dinámicos. \$Ter Por ejemplo, utilizando la cadena \$TER en ejecución de ViewRER, se abre el archivo de informe del mes actual.

Aquí está la lista de los argumentos:

\$File	Nombre ruta completo del archivo corriente.
\$FileName	Nombre archivo y extensión del archivo corriente.
\$FileDir	Disco y carpeta del archivo corriente.
\$Ter	Ruta completa del archivo de informe errores del mes en curso.
\$DirModule	Disco y carpeta que contienen MODx del archivo corriente.
\$Module	Número módulo del archivo corriente.
\$Bin	Disco y carpeta que contienen los ejecutables de Albatros.
\$TpaIni	Ruta completa del archivo de inicialización tpa.ini
\$ReqDirModule	Ruta (disco y carpetas) del módulo de Albatros. En caso de configuración de varios módulos se abre una ventana de selección del módulo.
\$ReqModule	Número del módulo de Albatros. En caso de configuración de varios módulos se abre una ventana de selección del número de módulo.
\$ReqFile	Nombre del archivo. Se abre una ventana de solicitud nombre archivo. El archivo seleccionado será pasado entre los argumentos del programa que será ejecutado.

- Solicitud de Argumentos:** si seleccionado, cada vez que solicita la ejecución del programa, se presenta un diálogo para insertar los argumentos diferentes de los configurados en la opción Argumentos, que pueden variar según las modalidades de iniciación del programa.



Especifica argumentos de iniciación de un programa

- Nivel de habilitación:** configura el nivel de visualización del programa en el menú Herramientas. A programas de test o modificación de datos de Albatros normalmente se asigna un nivel constructor. A programas para la edición de mecanizaciones de máquina se asigna un nivel de habilitación usuario.

La edición de unos campos puede producirse también a través del empleo del botón **[Agregar...]**. Este abre el diálogo **Agregar Herramienta** para la selección del programa que ejecutar. Las tipologías permitidas de los archivos ejecutables son .EXE, .COM, .PIF, .BAT.

Cuando se cierra el diálogo con confirmación de los datos, el programa se inserta en la ventana **Estructura Menú**, el nombre del programa y de la carpeta a la que pertenece, a la línea **Comando**.

Los otros botones presentes son **[Eliminar]**, **[Mover arriba]**, **[Mover abajo]** y se usan respectivamente para borrar un programa y ordenar la lista de los programas.

8.7 Browser

8.7.1 El browser

El browser es una función de Albatros que utiliza las informaciones generadas por el compilador para crear una base de datos para la búsqueda rápida de los símbolos definidos en las funciones.

Esta función es activa solo con un nivel de contraseña igual o superior a constructor. Los comandos utilizables pueden ser seleccionados desde el menú **Depuración**.

Con el browser es posible:

- posicionar el cursor en la línea en la cual es definida por primera vez una función, una variable o una constante de módulo, de grupo o de librería (desde **Depuración->Ir a la definición**)
- posicionar el cursor en las líneas en las cuales está referenciada una función, un dispositivo, una variable de módulo o de grupo, una instrucción GPL (con excepción de las instrucciones FCALL y FRET). Desde menú **Depuración->Ir a la referencia**, para visualizar la referencia anterior o sucesiva seleccionar desde el menú respectivamente las opciones **Depuración->Anterior** o **Depuración->Sucesivo**)

Las variables de grupo son gestionadas solo en la ventana de edición del grupo al que pertenecen.

Para actualizar un browser cuando se pasa a una nueva versión se recomienda como primera operación guardar las variables globales y después ejecutar el comando desde **Archivo->Compilar Todo**.

Durante la fase de edición de las funciones se pierde la correspondencia entre texto y símbolos buscados y es restablecida en fase de archivado.

8.7.2 Buscar identificador...















Se puede seleccionar el comando desde menú **Depuración->Buscar identificador...**. Abre un cuadro de diálogo para insertar el nombre del símbolo que se quiere buscar en el código GPL. Según el **Tipo de búsqueda** programado es identificada la definición o la primera referencia del símbolo.

El nombre insertado puede tener las siguientes características:

- no contiene algún carácter '.' (punto): está buscado en todos los archivos de funciones
- contiene sólo un carácter '.' (punto): el nombre que precede el punto es considerado como el nombre del grupo y el símbolo es buscado solo en ese grupo. Por ejemplo, si son definidas dos funciones con nombre VisError, una en el grupo MAIN y la otra en el grupo EJES, si se escribe como identificador EJES.VisError el cursor será posicionado en la primera línea de la función VisError del grupo EJES.
- contiene dos caracteres '.' (punto): el nombre que precede el primer punto es considerado como el nombre del grupo, el que precede el segundo punto es considerado como el nombre del subgrupo y el símbolo es buscado solo en ese subgrupo.
- si termina con un carácter '*' (asterisco), son considerados todos los símbolos que comienzan con los caracteres precedentes el asterisco.

En el caso en que se verifique una ambigüedad en la búsqueda del símbolo se abre una ventana de diálogo, donde están visualizados todos los símbolos que tienen el nombre requerido. Desde aquí es posible elegir el símbolo deseado.

En seguida está descrito el significado de los símbolos particulares utilizados en la lista para la selección del identificador.

Símbolo	Descripción
	instrucción GPL
	constante de módulo o de grupo o de librería
	variable de módulo o de grupo
	variable de librería
	vector de librería
	matriz de librería
	función de librería
	mensaje de grupo
	etiqueta
	variable local
	vector local
	matriz local
	parámetro simple
	parámetro vector



parámetro matriz

8.7.3 Lista de teclas a selección rápida disponibles

Para activar los comandos del browser es posible seleccionar las opciones del menú **Depuración** u obrar directamente desde el teclado..

Las teclas que se utilizan son:

Tecla	Descripción
F2	posiciona el cursor en la línea en la cual está definido el símbolo seleccionado. Si en base de datos del navegador están definidos más símbolos que tienen el nombre elegido, se abre un cuadro de diálogo en la cual se da al usuario la posibilidad de escoger el símbolo requerido.
Mayús+F2	posiciona el cursor a la primera referencia del símbolo seleccionado. En caso de ambigüedad se abre el cuadro de diálogo para la selección del símbolo elegido.
Ctrl+F2 Ctrl+'+' o Ctrl+Re Pág	abre un cuadro de diálogo para la selección del símbolo elegido. posiciona el cursor en la referencia sucesiva (usar '+' del teclado numérico).
Ctrl+'-' o Ctrl+Av Pág	posiciona el cursor en la referencia precedente (usar '-' del teclado numérico).

9 Programas accesorios

9.1 XConfMerge: programa para la combinación de archivos de configuración

XConfMerge es una herramienta que realiza la combinación de los archivos de configuración. Se ejecuta desde línea de comandos desde la carpeta bin, ya que XConfMerge lee el archivo tpa.ini.

Los archivos leídos por XConfMerge son:

- hardware.xconf: contiene los datos del virtual-físico y de la configuración hardware (físicos)
- devices.xconf: contiene los datos de la configuración de grupos, subgrupos y dispositivos (lógicos)
- devices.xmlng: contiene los mensajes que se pueden traducir. De todos modos se consideran todos los archivos de idioma que se encuentran en la carpeta
- addresses.xdb: contiene las direcciones lógicas de los dispositivos.

Los argumentos que pasar son:

- la carpeta desde la que leer los archivos nuevos
- el número del módulo al que se refiere el personalizado. Si ningún número está indicado, se considera 0 como valor predeterminado.

La ruta desde la que se lee el archivo que actualizar, que en fin es la misma donde se escribe el archivo conseguido por el procedimiento de combinación, es extrapolada desde los datos configurados en tpa.ini.

Cuidado: los archivos que actualizar son sobrescritos y no se realiza ninguna copia de seguridad automática.

Reglas de combinación para los archivos de configuración de los grupos:

1. si en los dos archivos se encuentra el mismo grupo, subgrupo o dispositivo, se mantienen los datos y la habilitación del archivo viejo.
2. si el grupo, subgrupo o dispositivo se encuentra sólo en el nuevo archivo, los datos y las habilitaciones son copiados.
3. si el grupo, subgrupo o dispositivo se encuentra sólo en el archivo viejo, este se elimina.

Reglas de combinación para los archivos de configuración hardware y del virtual-físico:

1. si en los dos archivos se encuentra el mismo hardware, se mantienen el hardware y el virtual-físico del archivo nuevo con las habilitaciones definidas en el archivo viejo.
2. si el hardware se encuentra sólo en el nuevo archivo, se mantiene el nuevo hardware con su habilitación y el nuevo virtual-físico.
3. si el hardware se encuentra sólo en el archivo viejo, se elimina junto con el virtual-físico.

Reglas de combinación para los archivos de las direcciones lógicas:

1. si el archivo no se encuentra en la carpeta de los archivos que importar, en tal caso se mantienen las direcciones lógicas del personalizado y se asignan direcciones nuevas a los nuevos dispositivos, si los hay.
2. si el archivo está en la carpeta de los archivos que importar, en tal caso se leen y utilizan las direcciones lógicas en él.

Reglas de combinación para los archivos de mensajes:

1. No se realiza ninguna combinación, sino el nuevo archivo de mensajes es copiado en la carpeta de módulo.

Al final de la ejecución, la herramienta de combinación devuelve los valores siguientes:

0	todo OK
---	---------

1	la combinación de los archivos no se completó correctamente
2	no se definieron argumentos
3	el número de módulo dado como argumento es incorrecto
4	la carpeta dada como primer argumento no existe

9.2 XParMerge: programa para la combinación de dos archivos de parámetros

XParMerge es una herramienta que realiza la combinación de dos archivos de parámetros.

Se ejecuta desde línea de comandos desde la carpeta bin, ya que XParMerge lee el archivo tpa.ini. Los argumentos que pasar son: el nombre del archivo nuevo y el número del módulo al que se refiere.

La ruta desde la que se lee el archivo que actualizar y la donde escribir el archivo, conseguido por el procedimiento de combinación, es extrapolada desde los datos configurados en tpa.ini.

Cuidado: no se realiza ninguna copia de seguridad automática del viejo archivo de parámetros, sino es sobrescrito.

Reglas de combinación para los archivos de Parámetros Tecnológicos:

- 1) si en los dos archivos se encuentra el mismo control, se mantiene el valor del archivo viejo, pero se actualizan los otros parámetros que lo definen (desactivado, visible, nombre variable GPL.....) sacandolos desde el nuevo archivo;
- 2) si en el archivo nuevo está definido un control que no se encuentra en el archivo viejo, el control es mantenido;
- 3) si en el archivo viejo está definido un control que no se encuentra en el archivo nuevo, el control es eliminado.

Reglas de combinación para los archivos de Parámetros de Herramienta:

- 1) si el diálogo de referencia del nuevo archivo es diferente del diálogo de referencia del archivo viejo, se mantienen todos los diálogos del archivo nuevo, si los hay, y se actualizan los del archivo viejos (sacados o añadidos controles nuevos);
- 2) si el diálogo de referencia del archivo nuevo es el mismo del diálogo de referencia del archivo viejo, los diálogos del nuevo se añaden a los diálogos del viejo;
- 3) si el diálogo de referencia se encuentra sólo en el archivo viejo, este mismo y su diálogos son eliminados.

10 Lenguaje GPL

10.1 Conceptos de base

10.1.1 Introducción al lenguaje GPL

El lenguaje GPL (General Purpose Language) es el lenguaje utilizado para la creación de las funciones del sistema Albatros.

Su estructura es en parte similar a la del BASIC, pero se caracteriza por la presencia de muchas instrucciones dedicadas al control de los dispositivos.

El lenguaje está compuesto por más de 200 instrucciones, que para más comodidad están subdivididas en grupos compuestos por instrucciones que tienen una funcionalidad similar.

Además, el lenguaje es [multitasking](#), por lo tanto permite la ejecución de más tareas al mismo tiempo.

Sintaxis típica de una instrucción GPL

Las instrucciones GPL tienen toda una estructura similar, que refleja el siguiente esquema:

nombreinstrucción parámetro-1, parámetro-2, parámetro-N

El número de parámetros efectivamente presentes dependerá de la instrucción y del contexto en el cual es utilizada, en lo general el número máximo de parámetros admitido para una instrucción o función GPL es 120. En algunos casos pueden no estar presentes parámetros.

El bloqueo mínimo de código GPL es la [función](#).

Subdivisión del código en grupos

El código GPL está subdividido en bloques que reflejan la subdivisión lógica de la máquina en grupos. Para cada grupo tendremos por lo tanto un archivo con el código a él asociado. A los archivos con el código de los grupos presentes en la máquina se agregan un archivo de variables y constantes globales visibles por el código GPL de todos los grupos y las [librerías](#). Estas contienen código separado por la configuración de la máquina y por lo tanto fácilmente transportable de una máquina a otra.

10.1.2 Convenciones y terminología

Principales términos adoptados

ARGUMENTO	Es uno de los argumentos de la instrucción; puede ser definido como <i>constante</i> , <i>variable</i> , o como <i>parámetro</i> , según el tipo de instrucción; si está entre paréntesis cuadrados ([]) significa que puede ser omitido, esto comporta naturalmente una diferente manera de ejecutar la instrucción.
PALABRA CLAVE	Es un argumento para elegir entre argumentos con valor predefinido y en lo general está representado en mayúsculo; la lista de las palabras claves está presente en una página de ayuda dedicada.
PARÁMETRO	Es el argumento de una instrucción no definido al interno de la misma instrucción más que es pasado, como parámetro, a la función cuando es ejecutada; a veces es nombrado también con el término <i>argumento parametrizado</i> .
CONSTANTE	Es un argumento definido de manera fija a través el meta-comando CONST o bien que es fijado rígidamente al interno de la instrucción.
VARIABLE	Es un argumento que ha sido definido como variable global de módulo o de grupo o a través la instrucción LOCAL y puede ser organizado como variable simple o como vector o como matriz. Véase Las variables .
PARÁMETRO DE CONFIGURACIÓN	Es un argumento que ha sido definido en configuración, como, por ejemplo, los parámetros de un eje.

Argumentos que se repiten mayormente en la descripción de las instrucciones

En seguida están presentados los términos relativos a argumentos que seguido se repiten en la sintaxis de las instrucciones GPL, seguidos cada uno por la relativa descripción. En los casos en que un argumento pueda emplear valores diferentes de los que están descritos aquí abajo, su descripción es reanudada en la sección *Argumentos* de la página de ayuda de la instrucción.

nombreentrada	nombre de dispositivo de entrada digital
nombresalida	nombre de dispositivo de salida digital
nombreindicador	nombre de dispositivo interruptor indicador o bit indicador
nombrepuerto	nombre de dispositivo entrada puerto, salida puerto o indicador puerto
nombrer temporizador	nombre de dispositivo temporizador
nombrecontador	nombre de dispositivo contador
nombrefunción	nombre de una función (vale también como parámetro dispositivo en el caso de ERRSYS)
nombresubprograma	nombre de un subprograma, es equivalente a <i>etiqueta</i> a quien ocurre hacer referencia para la explicación; la llamada a subprograma se efectúa por medio de la instrucción "CALL nombresubprograma".
eje	nombre de un eje
constante	un carácter o un número entero o doble o bien una palabra clave
valor	constante o variable (el <i>tipo</i> depende de la instrucción)
variable	nombre de variable, elemento de vector o elemento de matriz
variabledispositivo	nombre de <i>parámetro dispositivo</i>
matriz	nombre de una matriz
vector	nombre de un vector
etiqueta	nombre de la etiqueta de salto o el nombre de un subprograma.
estado	estado lógico, puede ser ON o OFF o bien 1 o 0
tiempodeexpiración	cantidad de tiempo dentro del cual un cierto evento debe verificarse, o un tiempo de retraso (constante o variable)
cota	valor de la cota (constante doble o variable doble)
radio	valor del radio (constante doble o variable doble)
ángulo	valor del ángulo (constante doble o variable doble)
númgiros	número de revoluciones (constante doble o variable doble)
velocidad	valor de velocidad (constante float o variable float)
sentido	rotación en el sentido de las agujas del reloj o en el sentido contrario (variable o constante: CW o CCW)
operando	(constante o variable o nombredispositivo)
resultado	resultado de la operación (variable o nombredispositivo)
nombredispositivo	nombre de un tipo de dispositivo cualquiera (o parámetro dispositivo)
constantecad	secuencia de caracteres encerrada entre comillas (ej. "cadena")
variablecad	el nombre de un vector de caracteres, o sea, una cadena
operador	operadores de comparación, también en combinación entre ellos: > (mayor) = (igual a) < (menor) pueden ser combinados poniéndolos cerca, ej. >= (o sea mayor o igual)
tipo	tipo de constante o variable: "char" (8 bit), "integer" (32 bit), "float" (32 bit), "double" (64 bit), "string"
parámetro dispositivo	es una variable que representa un dispositivo. Los dispositivos son definidos en Configuración.

Principales términos utilizados para un eje

cota teórica (o target)	Posición corriente "teórica" programada, instante tras instante, por el control numérico, en base al algoritmo de generación del perfil de velocidad.
cota real	Posición efectiva del eje como detectado por el transductor de posición. Difiere de la cota teórica de una cantidad llamada "error de persecución" o "error de anillo".
cota final	Corresponde a la cota programada de llegada de un movimiento. El algoritmo de cálculo del perfil de velocidad lleva la cota teórica a alcanzar exactamente el valor final.
ventana llegada cota	Intervalo programable que tiene como punto central la cota teórica final: cuando la cota real se coloca a su interior, el movimiento efectivo es considerado terminado.

ventana grande de llegada cota	Ventana de llegada en cota multiplicada por un factor programable a través de la instrucción SETBIGWINFACTOR .
error de persecución	Diferencia, instante tras instante, entre cota teórica y cota real de un eje: resulta normalmente proporcional a la velocidad de traslación e inversamente proporcional a la "ganancia de anillo proporcional".
ganancia [de anillo] proporcional	Parámetro de regulación del eje, programable: determina la relación entre la velocidad corriente y el error de persecución relativo.
feed forward (avance)	Parámetro de regulación del eje, programable: determina una contribución directa (proporcional a la velocidad programada) inyectado en el comando de velocidad del accionamiento. Permite reducir, a igual velocidad e igual ganancia proporcional, el valor del error de persecución.
feed rate override	Porcentaje de la velocidad programada. Este parámetro permite reducir la velocidad de ejecución respecto a la velocidad programada, en un valor porcentual que puede variar de 0% a 100%.
tolerancia	Valor de desplazamiento por el cual el eje se desplaza de la trayectoria original en una interpolación multieje entre dos bloques de desplazamiento consecutivos.
juego mecánico	Espacio entre cava y diente de una pareja de engranajes.

10.1.3 Las variables

Las variables contienen informaciones que son utilizadas por el lenguaje GPL que en ellas deposita los valores necesarios para la ejecución del programa.

Las variables son caracterizadas por un "tipo" que refleja las características de la información que en ellas será depositada. Además, para cada variable es asociada una visibilidad que define cuales conjuntos o subconjuntos de código pueden operar (leer y escribir) en esa.

Tipos de datos

DATOS SIMPLES o ESCALARES

El GPL soporta tipos de datos simples y agregados. Los tipos de dato simples son similares a los que están disponibles en la mayor parte de los lenguajes de programación:

Char

Es un entero con signo comprendido en el intervalo [-128 ; +127] y tiene longitud de 1 byte. La declaración de una variable de tipo char se efectúa con la siguiente sintaxis:

```
NombreVariable as char
```

Integer

Es un entero con signo comprendido en el intervalo [-2147483647 ; +2147483647] y tiene longitud de 4 bytes (corresponde con el tipo long del C).

La declaración de una variable de tipo integer se efectúa con la siguiente sintaxis:

```
NombreVariable as integer
```

Float

Es un número en coma flotante comprendido en los intervalos [-3,402 E+38 ; +3,402 E+38], tiene longitud de 4 bytes (utilizado típicamente para representar las velocidades).

La declaración de una variable de tipo float se efectúa con la siguiente sintaxis:

```
NombreVariable as float
```

Double

Es un número en coma flotante comprendido en los intervalos [-1,797 E+308 ; 1,797 E+308], tiene longitud de 8 bytes (utilizado típicamente para representar las velocidades).

La declaración de una variable de tipo double se efectúa con la siguiente sintaxis:

```
NombreVariable as double
```

Estos tipos de datos pueden ser utilizados al mismo tiempo en una misma expresión. El GPL ejecuta una conversión automática sin dar un mensaje de advertencia. Por lo tanto hay que tener cuidado con la posible pérdida de información que puede verificarse cuando se utilizan tipos de datos diferentes en la misma

expresión.

En algunas situaciones la conversión no es posible. En estos casos se obtienen en general unas advertencias por el compilador o unos errores de sistema.

DATOS AGREGADOS

El array

Es un conjunto de variables de tipo simple, todas del mismo tipo, obtenido asociando un índice al nombre de la variable. El índice es encerrado entre paréntesis cuadradas. Si el array se llama, por ejemplo, "parámetros", el primer elemento del conjunto será indicado con "parámetros[1]", el segundo con "parámetros[2]" y así en adelante.

El array tiene un número fijo de elementos que debe de ser definido en el momento de la declaración. Una típica declaración de un array sigue la sintaxis:

```
parámetros[10] as integer
```

Donde parámetros[10] especifica que el nombre del array es "parámetros" y que está compuesto por 10 elementos, as integer define el tipo de dato simple de los elementos del array que en este caso es integer.

Los arrays pueden ser constituidos por datos de tipo simple o por cadenas.

Un array puede ser compuesto por un máximo de 262144 elementos.

Se puede inicializar los vectores directamente en el código GPL en el momento de la inicialización. La sintaxis GPL puede ser:

```
[READONLY] vector[númerolíneas] as integer = 1,2,3,4
[READONLY] vector[númerolíneas] as cadena = "uno","dos","tres","cuatro"
```

Las matrices

Las matrices son arrays bidimensionales o sea variables a las cuales se asignan dos índices. Podemos asociar mentalmente una matriz a una tabla de datos subdividida en líneas y columnas. Para indicar una de las casillas de la tabla podemos definir en cual línea y en cual columna de la tabla ella se encuentra. El primer índice de la matriz corresponde con el número de línea, el segundo al número de columna.

A diferencia de los array, las matrices pueden contener tipos de datos diferentes entre ellos, pero con la siguiente limitación: podemos usar un tipo de datos simple diferente para cada columna pero no podemos cambiar el tipo dentro de la columna misma.

Por ejemplo, podemos definir una matriz cuya primera columna es de tipo integer y la segunda es de tipo float. Pero no podemos tener una matriz en la cual en la primera línea tenemos un integer y un float y en la segunda línea tenemos un char y un double. Las líneas deben de ser todas iguales por lo que concierne el tipo de dato de los elementos por los cuales está compuesta.

La declaración de una matriz puede ser efectuada según una de las siguientes sintaxis:

```
offset[10] as double double double
```

```
dim_pieza[50] as float:longitud float:anchura float:espesor
```

En el segundo tipo de declaración está asignada una etiqueta o nombre simbólico a cada columna. Los nombres simbólicos de las columnas resultan muy útiles cuando se trabaja con matrices muy grandes. De hecho en estas situaciones es difícil acordar el significado de los tamaños memorizados en el interior de cada columna de la matriz. El nombre simbólico nos permite identificar de inmediato el tipo de dato con el que estamos trabajando. Por ejemplo, "Offset[1][3]" es menos claro que "Offset[1].eje_X".

Las matrices pueden contener solo datos de tipo simple. No se puede, por ejemplo, crear matrices que contienen unas cadenas. Una matriz puede estar compuesta por un máximo de 262144 líneas.

Las matrices se pueden inicializar directamente en el código GPL, al momento de su declaración. La sintaxis GPL puede ser:

```
[READONLY] nombrematriz[númerolíneas] as double double integer double = _
1.1, 2.2, 3, 0.1 _
1.2, 3.4, 5, 0.1 _
2.1, 5.6, 6, 0.1
```

Las cadenas

Las cadenas son conjuntos de caracteres o sea de datos tipo char que son gestionados de manera particular ya que representan texto que se puede leer.

Una cadena es muy similar a un array de char. La diferencia más grande está dada por la presencia de un carácter terminador que es agregado automáticamente a la cola de las cadenas. El GPL presenta, además, unas instrucciones que permiten manipular las cadenas.

Típicamente las cadenas son utilizadas para escribir unos mensajes, leíbles por el usuario, en la pantalla o en un archivo de Informe.

La declaración de una variable de tipo Cadena puede ser efectuada según una de las siguientes sintaxis:

NombreVariable as String

NombreVariable as String[20]

En el primer tipo de declaración es asignada una dimensión predeterminada de 256 bytes. En el segundo caso se define una dimensión máxima para la cadena.

Los valores de cadena son secuencias de caracteres entre ápicos dobles. Ejemplo: "Pulsar el botón".

Para insertar el carácter `"` (doble ápice) hay que insertarlo dos veces. Ejemplo: "Pulsar el botón ""Inicio""".

Para insertar los caracteres, a través de el código numérico, se escriben dentro de la cadena los caracteres `\u` seguidos por el valor numérico del carácter hexadecimal. Ejemplo: `\u20ac` es el símbolo del euro. Si se escribe `"\u20ac 15,6"` se obtiene € 15,6.

Conversión de datos

En todas las expresiones matemáticas, con excepción de la instrucción `EXPR`, los tipos de datos de los operandos se convierten al tipo de datos de la variable resultado y luego se hace la operación. Es importante prestar mucha atención a la declaración de los tipos de datos, porque pueden influir en el resultado. La tabla siguiente es un ejemplo de como los resultados pueden variar según el tipo de dato:

DIV	Operando 1 (Integer)	Operando 2 (Double)	Resultado (char)
	3	5.0	0
	5	1.9	5
	1200	107.2	Indefinido
	1200	250.0	Indefinido
DIV	Operando 1 (Double)	Operando 2 (Double)	Resultado (Double)
	3	5.0	0.6
	5	1.9	2.631
	1200	107.2	11.194
	1200	250.0	4.8

En la instrucción `EXPR`, si todos los operandos no son del mismo tipo, se realizará una conversión automática y el tipo del resultado de la operación será igual al mayor de los dos según la siguiente regla:

- char < integer
- float < doble
- char o integer < float o doble.

Tras la resolución de la expresión, el resultado se convertirá al tipo de la variable resultado.

EXPR	Operando 1 (Double)	+	Operando 2 (Integer)	/	Operando 3 (Float)	Resultado (Integer)
	900.0	+	100	/	400.0	900
EXPR	Operando 1 (Double)	+	Operando 2 (Integer)	/	Operando 3 (Float)	Resultado (Double)
	900.0	+	100	/	400.0	900.25

Declaración y visibilidad de las variables

Las declaraciones de las variables y de las constantes pueden ser efectuadas solo en algunos puntos particulares del código GPL.

Se pueden definir variables:

- Globales de módulo
- Globales de grupo

- Locales (solo variables)
- Globales de librería

El número máximo de variables globales (de módulo y de grupo) que pueden ser declaradas es 2048.

Se pueden definir unos modificadores que atribuyen a las variables unas características adicionales.

Variables globales de módulo

Las variables globales de módulo están contenidas en un particular archivo al que se accede seleccionando la opción del menú **Archivo->Abrir Variables globales**.

La declaración es efectuada, como hemos visto en los párrafos precedentes, especificando el nombre de la variable, seguido por la palabra clave "AS", seguida por el tipo de dato (o por los tipos de dato en el caso de las matrices).

Estas variables son visibles directamente por el código de todos los grupos.

Variables globales de grupo

Las variables globales de grupo son definidas al principio del código relativo al grupo. Deben de ser declaradas antes de las funciones GPL.

Estas variables son visibles directamente por todo el código contenido al interno del grupo. Además es posible extender la visibilidad de estas variables al interno del grupo declarándolas como variables "Públicas".

Las variables públicas no son accesibles directamente desde el exterior del grupo. Para poder acceder a ellas hay que utilizar sus nombres precedidos por el nombre del grupo al cual pertenecen. Por ejemplo, si se quiere modificar la variable pública "desplazamiento" del grupo "ejes" desde el código del grupo "Main" habrá que escribir "SETVAL 10 ejes.offset".

Para declarar una variable global de grupo se utiliza la misma sintaxis utilizada para las variables globales de módulo. La principal diferencia está en la definición de las variables públicas.

Para definir una o más variables públicas y privadas se utilizan las etiquetas "Public:" y "Private:". Por ejemplo:

```
Public:
  offset as double
  velocidad as float
Private:
  herramienta as integer
```

Variables locales

Las variables locales son declaradas adentro del cuerpo de una función. Deben de ser declaradas antes de cualquier otra instrucción con la excepción de la declaración de los parámetros de la función.

Las variables locales son accesibles solo adentro de la misma función.

Estas variables son creadas con valor 0 (es asignada la memoria necesaria) solo al principio de la ejecución de la misma función y son destruidas (la memoria es librada) al término de la ejecución. Las variables globales, a cambio, son creadas durante la inicialización del módulo y están siempre visibles en "Diagnóstico".

La declaración de una variable local sigue la sintaxis que ya hemos visto, pero tiene que ser precedida por la palabra clave "LOCAL".

Por ejemplo:

```
Function tarea
  local cota_centro as double
  movabs X,cota_centro
fret
```

Variables globales de librería

Las variables globales de librería son contenidas en las librerías de código GPL. Son análogas a las variables globales de grupo.

Modificadores

Modificadores: READONLY

Las variables globales de módulo y globales de grupo pueden ser declaradas como READONLY.

Una variable "solo lectura" es una variable cuyo valor no puede ser modificado desde el código GPL, más puede ser modificado "desde afuera" o sea desde el archivo de los parámetros tecnológicos de área.

El archivo de los parámetros tecnológicos es una base de datos en la cual están conservados unos valores que caracterizan la máquina que pueden variar en el largo periodo a causa de modificaciones de la misma máquina o de mantenimiento extraordinaria. Estas informaciones están normalmente insertadas en una matriz GPL durante la inicialización del control.

Ejemplos de este tipo de informaciones son los desplazamientos de las áreas de trabajo o las dimensiones y los parámetros tecnológicos de las herramientas.

Declarando estas variables como "solo lectura" nos protegemos de modificaciones accidentales de informaciones que por lo general no deben de variar durante el normal funcionamiento de la máquina.

La dimensión máxima de una variable "solo lectura" es de 128 Kbytes.

La declaración de una variable "solo lectura" debe de ser efectuada según la siguiente sintaxis:

`readonly` NombreVariable as tipo

Modificadores: **NONVOLATILE**

Las variables declaradas de clase **NONVOLATILE** están memorizadas en la RAM no volátil (dotada de batería) en vez que en la RAM normal. Como consecuencia los valores guardados en estas variables no son perdidos cuando se apaga el control numérico.

La declaración de una variable novolátil debe de ser efectuada según la siguiente sintaxis:

```
nonvolatile NombreVariable as tipo
```

Por ejemplo:

```
nonvolatile OffsetAreas[2] as double:offsetX double:offsetY double:offsetZ
```

Sólo pueden ser definidas "nonvolatile" las variables globales de grupo y de módulo.

La dimensión máxima total de las variables memorizadas en la RAM no volátil es de 65536 bytes. La dimensión máxima de una sola matriz no volátil es de 1024 bytes.

Asignación de **RANGE**

Es posible en el momento de la declaración de una variable asignarle un range de valores. De todas formas en el momento no está efectuado un control del respeto de los límites en fase de ejecución, solo está efectuado un control por el compilador en el caso en que estén asignados unos valores constantes (por ejemplo, para inicializar la variable).

La principal ventaja es por lo tanto una clase de auto-documentación del código.

La definición de los range se efectúa según la siguiente sintaxis:

```
NombreVariable Range:valmin..valmax AS tipo
```

Por ejemplo:

```
NombreHerramienta Range:1..100 as integer
```

Derechos de Lectura/Escritura

Los derechos de lectura y escritura permiten especificar el [nivel mínimo de acceso](#) al sistema necesario para visualizar (derecho de lectura) y modificar (derecho de escritura) su valor.

La sintaxis es la siguiente:

```
NombreVariable Read=S Write=M AS tipo
```

Las palabras claves utilizadas para especificar los derechos son:

- READ lectura
- WRITE escritura

Los valores asignables son:

- U o USER usuario
- S o SERVICE asistencia
- M o MANUFACTURER constructor
- T o TPA tpa

Los predeterminados ofrecidos para los valores son:

- READ lectura para asistencia (S o SERVICE)
- WRITE escritura para constructor (M o MANUFACTURER) y tpa (T o TPA)

10.1.4 Las Constantes

El GPL prevé cuatro tipos de constantes:

- Integer
- Double
- Char
- String

Las constantes de tipo Char son declaradas utilizando los superíndices como sigue:

```
Const COD = 'A'
```

Las constantes de tipo String son declaradas utilizando los dobles superíndices como sigue:

```
Const MSG = "Comienzo tarea"
```

Las constantes de tipo Integer y de tipo Doble son declaradas con la siguiente sintaxis:

```
Const PI = 3.14
Const MSGBOX = 12
```

Para las constantes de tipo integer está prevista también una notación binaria y hexadecimal:

```
Const MASK = $11001001b    ; binario
Const MASK = $F5h          ; hexadecimal
```

También las constantes de grupo y librería pueden ser públicas o privadas.

La sintaxis es similar a la de las variables.

Ejemplo:

```
Public:
Const PI = 3.14
Const MSGBOX = 12
Private:
Const MASK = $11001001b
```

ATENCIÓN: no existen constantes de tipo Float. Los valores decimales tendrán que estar necesariamente declarados como Double. A veces esto puede causar unos mensajes de advertencia de parte del compilador (cuando se utilizan instrucciones GPL optimizadas para el uso de tipos Float).

Se pueden definir las constantes como resultado de expresiones de cálculo, con la siguiente sintaxis:

```
Const a = 10
Const b = 20
Const c = a + b
```

Los operadores admitidos son los mismos utilizados en la instrucción [EXPR](#).

Constantes predefinidas con valor preconfigurado

El lenguaje GPL prevé algunas constantes predefinidas. Éstas pueden ser utilizadas directamente sin necesidad de estar definidas.

Las constantes predefinidas y los valores correspondientes son:

ON	1
OFF	0
UP	+1
DOWN	-1
POSITIVE	+1
NEGATIVE	-1
CW	1
CCW	0
TRUE	1
FALSE	0
NOWAIT	0
WAIT	1
WAITACK	2
STORE	1

NOSTORE	0
NOPLACE	0
COM1	0
COM2	1
COM3	2
COM4	3
COM5	4
COM6	5
COM7	6
COM8	7
NOPARITY	0
ODDPARITY	1
EVENPARITY	2

Constantes predefinidas con valor preconfigurado al iniciar Albatros

El lenguaje GPL incluye unas constantes predefinidas cuyo valor es definido al arranque de Albatros. Se pueden usar en la instrucción [IFDEF](#).

_ID_MODULE	número del módulo actual. El número de módulo está entre 0 y 15.
_REMOTE_MODULE	tipo de módulo. La constante vale 1 si el módulo es remoto, vale 0 si el módulo es local, no está definida si el módulo no está configurado en Configuración de sistema.
_VER_MAJOR	número de versión principal de Albatros. Si la versión de Albatros es 3.2.1, el valor de la versión principal es 3.
_VER_MINOR	número de versión secundario de Albatros. Si la versión de Albatros es 3.2.1, el valor de la versión secundario es 2.
_VER_REVISION	número de revisión de Albatros. Si la versión de Albatros es 3.2.1, el valor de la revisión es 1.
_VER_SP	cadena que describe si el Service Pack está instalado, por ejemplo "Service Pack 1f", de otro modo no está definida.
_VER_FULL	número de versión completa. En el caso de la versión 3.2.1, es \$00030201H.

10.1.5 Palabras clave

Las palabras clave son identificadores con uso reservado y no pueden ser utilizados de otra forma.

Las palabras clave disponibles son:

Todos los nombres de las instrucciones GPL	Véase la parte del manual "Instrucciones " para la descripción de todas las instrucciones GPL
Todos los tipos de dato	Véase Las variables
Los parámetros de tipo dispositivo	Véase Los parámetros de tipo dispositivo
EXIST	utilizada en la instrucción IFDEF para verificar la existencia de un grupo. Véase la instrucción IFDEF
NOTEXIST	utilizada en la instrucción IFDEF para verificar la no existencia de un grupo. Véase la instrucción IFDEF
LINKED	empleada en la instrucción IFDEF para habilitar la compilación de bloques de código si el dispositivo está vinculado en Virtual-Físico. Véase la instrucción IFDEF

UNLINKED	empleada en la instrucción IFDEF para habilitar la compilación de bloques de código si el dispositivo no está vinculado en Virtual-Físico. Véase la instrucción IFDEF
FUNCTION	Declaración de una función. Véase Las funciones
AS	Utilizada para la declaración de las variables. Véase Las variables
PUBLIC	Es un atributo de una función. Véase Las funciones
AUTORUN	Es un atributo de una función. Indica que la función es de arranque automático. Véase Las funciones
R= o READ	Es un atributo de una función o de una variable. Señala el nivel de acceso en lectura. Véase Las funciones y Las variables y Derechos de acceso
W= o WRITE	Es un atributo de una función o de una variable. Señala el nivel de acceso en escritura. Véase Las funciones y Las variables y Derechos de acceso
CONST	Permite asignar un nombre significativo, llamado constante simbólica, en lugar de un número, de un carácter o de una cadena. Véase Las variables
READONLY	Es un atributo de una variable global. Véase Las variables
NONVOLATILE	Es un atributo de una variable global. Véase Las variables
PRIVATE	Es un atributo de una función. Véase Las funciones
RANGE	Utilizada para la definición de un intervalo de valores para cada variable. Véase Las variables
USER	Es un atributo de una función o de una variable. Señala la tipología de acceso. En este caso usuario. Véase Las funciones o Las variables .
SERVICE	Es un atributo de una función o de una variable. Señala la tipología de acceso. En este caso manutención. Véase Las funciones o Las variables .
MANUFACTURER	Es un atributo de una función o de una variable. Señala la tipología de acceso. En este caso constructor. Véase Las funciones o Las variables .
TPA	Es un atributo de una función o de una variable. Señala la tipología de acceso. En este caso TPA. Véase Las funciones o Las variables .

10.1.6 Las funciones

Las funciones son el bloque mínimo de código GPL. Las instrucciones GPL no pueden ser insertadas secuencialmente en un archivo más hay que agruparlas en funciones. El número máximo de funciones declarables es 8191.

Desde el punto de vista del compilador son funciones todos los bloques de código GPL que comienzan con una línea cuya primera palabra es FUNCTION. Sin embargo no es una palabra clave que indica el final del texto de una función: la función se acaba con la línea que es anterior al inicio de otra función o con el final del archivo que contiene las funciones.

La sintaxis para la definición de una función es:

```
FUNCTION      NombreFunción Atributos
                Parámetros
                Variables locales
```


Lista de las instrucciones GPL

Una función es, además, un particular tipo de dispositivo de Albatros. Comparte con los dispositivos unas propiedades: un nombre unívoco (no traducible), un indicador de visibilidad (si el dispositivo es pública o no), un [nivel de acceso](#) en lectura y un nivel de acceso en escritura (Consulte el párrafo siguiente).

Derechos de acceso

[Derechos de acceso](#) Los derechos de acceso permiten especificar el nivel mínimo de acceso al sistema que permite la visibilidad (derecho de lectura) y la ejecución (derecho de escritura). La sintaxis es la siguiente:

```
Function NombreFunción READ=S WRITE=M
```

Los derechos son identificados por las palabras clave READ (lectura) y WRITE (ejecución) Los valores asociables, que corresponden a los niveles de acceso, son:

- U o USER usuario
- S o SERVICE asistencia
- M o MANUFACTURER constructor
- T o TPA tpa

Los predeterminados ofrecidos para los valores son:

- READ lectura para asistencia (S o SERVICE) y usuario (U o USER)
- WRITE escritura para constructor (M o MANUFACTURER) y tpa (T o TPA)

Funciones Autorun

Una función de tipo autorun es ejecutada automáticamente en la inicialización de la máquina. Las funciones autorun tienen, además, la característica de ser reiniciadas automáticamente cuando terminan a causa de un error de sistema. La sintaxis es la siguiente:

```
Function NombreFunción autorun
```

Es suficiente por lo tanto añadir el modificador "autorun" a la declaración de la función.

Funciones Public

Una función normalmente puede ser puesta en ejecución (llamada) solamente por el código que reside al interno del archivo de grupo. Para que una función pueda ser puesta en ejecución por el código GPL de otro grupo tiene que ser definida de tipo **public**. La sintaxis para definir una función public es la siguiente:

```
Function NombreFunción public
```

Es suficiente por lo tanto añadir el modificador "public" a la declaración de la función. Representan una excepción las funciones que pertenecen al intergrupo, las cuales son siempre de tipo **public**.

Funciones de Subgrupo

Una función puede ser asociada a un subgrupo simplemente anteponiendo el nombre del subgrupo al de la función. El nombre del subgrupo y el de la función deben de ser separados por un punto ".". Por ejemplo, la siguiente función pertenece al subgrupo "X" del grupo "Ejes".

```
Function X.puesta a cero
local vel as float
movabs X,100
waitstill X
Fret
```

Funciones asíncronas

Una función asíncrona es llamada automáticamente por el control numérico cuando se genera el evento al cual está ligada la función.

Las tipologías de eventos son tres:

- Cambio de estado de una entrada digital: instrucción ONINPUT
- Cambio de estado de un bit indicador o un interruptor indicador: instrucción ONFLAG
- Generación de un error de sistema: instrucción ONERRSYS

Cuando se genera el evento es llamada la función (no como tarea autónoma más en el contexto de la tarea en la cual ha sido ejecutada la instrucción ON... correspondiente) como FCALL implícita, apenas después que la instrucción corriente ha terminado la ejecución.

Típicamente las funciones asíncronas sirven para gestionar situaciones de emergencia y deben de ser muy rápidas. Por lo tanto estas funciones no pueden utilizar cualquier instrucción GPL más un subconjunto que garantice tiempos de ejecución breves.

Funciones con parámetros en entrada (paramétricas)

Una función puede tener parámetros en entrada declarados, mientras no retorna en ningún caso un valor.

Los parámetros pueden ser vistos como particulares variables locales cuyo valor es inicializado desde el externo en el instante en el cual la función es puesta en ejecución. Los parámetros son declarados con la palabra clave PARAM y siguen la misma sintaxis utilizada para las variables locales. Los parámetros deben de ser enumerados en las primeras líneas del cuerpo de la función, antes de cualquier otra instrucción y antes de las variables locales.

Existen dos maneras en que se pueden pasar los parámetros:

- **por valores:** son pasados por valor todos los tipos de dato simples, o sea CHAR, INTEGER, FLOAT y DOBLES. El pasaje por valor conlleva la creación de una copia del valor original. Las modificaciones aportadas al parámetro tienen efecto solo en el contexto de la función.
- **por referencia:** son pasados por referencia los tipos estructurados, o sea ARRAYS, MATRICES y CADENAS. El pasaje por referencia conlleva el uso de la variable de origen, de consecuencia las modificaciones aportadas al parámetro tienen efecto en el contexto de la función que llama. Esta característica puede ser disfrutada para devolver valores de retorno a la función que llama.

Típicamente una función es puesta en ejecución con la instrucción FCALL. Si se trata de una función paramétrica después del nombre de la función habrá que especificar la lista de los valores de asignar a los parámetros.

En el ejemplo que sigue hay una función paramétrica que ejecuta una tarea de perforado. Las cotas del centro del foro y la velocidad de avance del eje Z son pasadas a la función como parámetros.

```
Function Perforación
Param Qx as Double      ; cota X del centro foro
Param Qy as Double
Param vel as Float      ; velocidad de avance

Movabs      X, Qx, Y, Qy
Waitstill   X,Y
.....
Fret
```

La llamada de esta función, para efectuar, por ejemplo, un foro a las cotas (12.5, 25.7) y con velocidad de avance de 3 m por minuto, podrían ser efectuadas en la siguiente manera:

```
Fcall      Perforación 12.5, 25.7, 3.0
```

Los parámetros pasados a la función deben de corresponder por número y tipo a los que están declarados en la función llamada. La ejecución de la función que llama vuelve a comenzar al término de la ejecución de la función llamada.

Como parámetro de una función también puede ser [declarado un dispositivo](#). Esto permite escribir unas funciones de uso general, como, por ejemplo, una función de puesta a cero utilizable con todos los ejes presentes en la máquina:

```
Function PUESTA A CERO PUBLIC
param      eje as Axis
movabs     eje,100
Fret

Function MAIN
.....
Ejes.Puesta_a_cero x
Fret
```

La función puesta a cero pertenece al grupo Ejes y es declarada PUBLIC de modo que pueda ser vista también por las funciones declaradas en otros grupos. La función Main llama la función puesta a cero del grupo ejes, especificando como parámetro en entrada el eje que se quiere mover.

10.1.7 Los parámetros de tipo dispositivo

Los tipos Dispositivos son unas variables particulares que permiten hacer referencia a un dispositivo de la máquina.

Estas tipologías de dato pueden ser utilizadas **exclusivamente** en la declaración de los [parámetros de una función](#). No se puede por lo tanto declarar variables de este tipo. La definición de los nombres y de las demás características de los dispositivos quedan de pertinencia de la Configuración del sistema.

En la siguiente tabla están presentados los tipos Dispositivo y las correspondientes palabras clave que hay que utilizar para la declaración de los correspondientes parámetros.

Tipo	Palabra clave
Entrada digital	INPUTDIG

Salida digital	OUTPUTDIG
Entrada analógica	INPUTANALOG
Salida analógica	OUTPUTANALOG
Eje	AXIS
Temporizador	TIMER
Contador	COUNTER
Bit indicador	FLAGBIT
Interruptor indicador	FLAGSWITCH
Puerto indicador	FLAGPORT
Puerto de Entrada	INPUTPORT
Puerto de Salida	OUTPUTPORT
Función	FUNCTION
Dispositivo genérico	DEVICE
Task	TASK

En el siguiente ejemplo está declarado y utilizado un parámetro de tipo eje:

```
Function prueba
  Param eje as axis

  MovAbs     eje,100
  WaitStill  eje

Fret
```

10.1.8 El Multitasking

El sistema es de tipo multitasking y por lo tanto se puede haber más tareas GPL en ejecución, donde tarea se refiere a un proceso de gestión de una entidad analógica (típicamente un grupo).

Las tipologías de tareas a disposición son dos: las tareas normales y las tareas de tiempo real.

Tareas normales

El multitasking está basado en un algoritmo cooperativo basado en prioridad. Esto garantiza a todas las tareas de poder ejecutar de manera cíclica y de variar sus prioridades. El algoritmo de programación (planificación de tareas) prevé la ejecución de una instrucción para cada tarea activa (estado running). A cada tarea está asociada una prioridad que puede ser programada con la instrucción [SETPRIORITYLEVEL](#). La prioridad está identificada por un número entero comprendido entre 0 (prioridad máxima) y 255 (prioridad mínima). Para las tareas con prioridad 0 (cero) será ejecutada una instrucción a cada ciclo de programación, para las tareas con prioridad 1 será ejecutada una instrucción cada dos ciclos de programación y así sucesivamente hasta las tareas con prioridad 255 para las cuales estará ejecutada una instrucción cada 256 ciclos de programación.

La ejecución de las tareas normales es asíncrona respecto a la frecuencia de refresco de los ejes. Esto significa que no hay garantía sobre el hecho que una función GPL sea completada en el arco de tiempo que hay entre dos actualizaciones del estado de los ejes.

Una tarea es identificada por el nombre de la función GPL desde la cual comienza su ejecución.

La ejecución de una tarea puede comenzar:

- automáticamente en la inicialización del sistema: función principal del intergrupo y funciones autorun.
- en seguida de la ejecución de una instrucción [STARTTASK](#).
- en seguida del lanzamiento en modalidad manual del interfaz gráfica de Albatros.

Cada tarea está caracterizada por un estado interno:

RUNNING	la tarea está en ejecución
HOLD	la tarea está suspendida
BREAK	la tarea está interrumpida para la depuración

La jerarquía de las tareas está organizada en árbol. Cada tarea es creada por otra de la cual es hija, por lo tanto comporta que si la tarea madre termina, todas las tareas hijas serán terminadas. El número máximo de tareas en ejecución al mismo tiempo es 500.

Recuerde que un número elevado de tareas en ejecución conlleva una disminución de la velocidad a la que se ejecuta cada una de las tareas.

Si se cree que la aplicación que se va a realizar podría conllevar el uso de un número de tareas mayor de 200, es necesario utilizar un hardware adecuado como los módulos CN2128.

Tareas De Tiempo Real

Las tareas de tiempo real se diferencian de las precedentes por el hecho de que no son sujetas a un mecanismo de programación ni tienen una prioridad asociada, más son integralmente ejecutadas en cada actualización del estado de los ejes (tiempo real ejes).

Es absolutamente necesario que la ejecución de estas tareas termine dentro de un tiempo determinado porque la ejecución de las tareas GPL descritas anteriormente queda suspendida durante la ejecución de las tareas de tiempo real.

El sistema ejecuta unos controles sobre el tiempo de ejecución de las tareas de tiempo real y en el caso en que estas superen el tiempo máximo consentido es generado un error de sistema.

Por lo tanto se recomienda NO crear ciclos infinitos (por ej. con instrucciones GOTO) al interno de estas tareas, ciclos que, además, no son necesarios ya que la ejecución del código vuelve a comenzar desde el principio en cada tiempo real ejes.

Siempre para evitar tiempos de ejecución excesivos las tareas de tiempo real son sujetas a restricciones en el uso de algunas instrucciones GPL. Las instrucciones cuyo uso no está consentida son las [no utilizables en interrupt](#).

Se recomienda el uso de tareas de tiempo real solo para las actividades que deben de ser necesariamente efectuadas de manera sincrónica con la actualización de las cotas ejes. Para la mayoría de las actividades de control es más oportuno utilizar tareas normales.

Las tareas de tiempo real son iniciadas con la instrucción [STARTREALTIMETASK](#) y pueden ser interrumpidas con la instrucción [ENDREALTIMETASK](#). Es posible activar un máximo de 256 tareas de tiempo real al mismo tiempo.

Ya no vale el mecanismo de herencia, por lo tanto, si la tarea que ha iniciado una tarea de tiempo real termina, esta última queda en ejecución.

Las variables locales declaradas en la tarjeta de tiempo real son utilizadas solamente al iniciar de la tarea y en seguida mantienen el valor de la última ejecución.

Las tareas de tiempo real no son caracterizadas por los estados previstos para las tareas normales. Es posible efectuar la depuración de una tarea de tiempo real, más en este caso el sistema desclasa automáticamente la tarea a "tarea normal" por toda la duración del actividad de depuración.

Si en una tarea de tiempo real se verifica un error de sistema, la tarea es declasada a tarea normal y puesta en estado HOLD para permitir una análisis con la depuración.

10.1.9 Las Comunicaciones

Las comunicaciones entre el GPL y el mundo exterior ocurren a través de tres diferentes modalidades:

- SEND/RECEIVE
- Comunicaciones seriales
- IPC

Send/Receive

Las instrucciones [SEND](#) y [RECEIVE](#) implementan un mecanismo de comunicación orientado al mensaje.

La comunicación puede ocurrir o dentro del mismo módulo (poco ventajoso), entre más módulos de una línea o bien entre los módulos y el supervisor Albatros o bien con aplicaciones OLE.

El funcionamiento es similar al del correo electrónico; para cada mensaje está asociado un destinatario, un identificador de la información enviada (o requerida), la información misma más unas informaciones de servicio. Albatros ejerce la función de recolección y clasificación de las informaciones y en algunos casos da directamente las informaciones requeridas.

Esta modalidad de comunicación es normalmente utilizada para el envío de los programas de trabajo entre el supervisor y los módulos de control, para la sincronización de la actividad de las máquinas de una línea y para la interconexión con aplicaciones externas (servidor OLE).

Comunicaciones seriales

En el lenguaje GPL hay algunas instrucciones, por ejemplo, [COMREAD](#) y [COMWRITE](#), que permiten enviar y recibir datos a través las puertas seriales del control. Por lo tanto es posible interconectar el control con dispositivos externo como invertir, terminales o PLC. Estas instrucciones, utilizadas de manera oportuna, permiten implementar protocolos de comunicación serial como MODBUS-RTU etc.

IPC

El IPC o Inter Process Communication es una modalidad de comunicación entre procesos. En particular esta modalidad permite definir un área de memoria compartida entre dos o más procesos que puede ser utilizada para el intercambio de informaciones.

Típicamente está utilizado cuando las prestaciones dadas por el interfaz OLE de Albatros no están adecuadas.

En el lado GPL la comunicación IPC es implementada con las instrucciones [SENDIPC](#), [WAITIPC](#) y [TESTIPC](#). En caso de módulo local, los procesos externos pueden hacer referencia a las API, proveídas por RTX, o bien al componente COM **gplipc2.dll** directamente dado por TPA, que simplifica su uso.

En el caso de módulo remoto, los procesos en ejecución en el supervisor usan el componente COM **gplipc2net.dll**.

Para más informaciones, póngase en contacto con TPA.

10.1.10 Variables que usar en la programación

La mayoría de las instrucciones ha sido realizada de manera que pueda trabajar con diferentes tipos de variables (CHAR, INTEGER, FLOAT, DOUBLE). Sin embargo cada instrucción tiene un tipo de variable preferida para la cual ha sido optimizada. Para alcanzar mejores prestaciones durante la ejecución del código GPL se recomienda utilizar los tipos de variables recomendadas en la descripción de cada instrucción. Por lo general se recomienda respetar la siguiente tabla que asocia a las principales grandezas utilizadas en la programación los respectivos tipos óptimos:

<i>dimensión</i>	<i>tipo</i>
cota	<i>doble</i>
velocidad	<i>float</i>
tiempo	<i>doble</i>
contador	<i>entero</i>
valor puerto/indicador puerto	<i>entero</i>
timeout (tiempo de expiración)	<i>doble</i>
entrada/salida analógica	<i>float</i>
cosenos directores	<i>doble</i>
carácter de control cadena	<i>char</i>
aceleraciones/deceleraciones	<i>entero</i>

10.1.11 Los Ejes

Es normalmente indicado con el nombre "eje" un sistema electromecánico cuyo fin es el desplazamiento controlado de un órgano de una máquina herramienta.

Podemos describir este sistema en términos de los elementos que lo componen, estos pueden ser subdivididos en base a sus características tecnológicas.

Por lo tanto tenemos unos componentes mecánicos entre los cuales:

- estructura portante
- guías
- cojinetes
- tornillo + cojinetes a re-circulación de esferas

cuya tarea es de contrastar las fuerzas en juego, disminuir las fricciones, transformar el movimiento de rotatorio a movimiento de traslación etc.

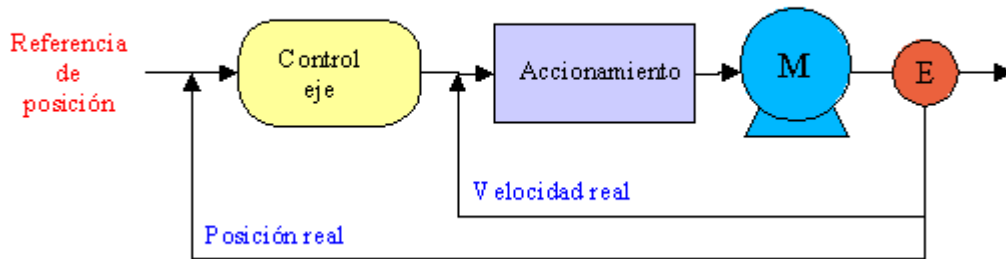
Tendremos también componentes eléctricas y electrónicas entre las cuales:

- el motor
- interruptores de fin de carrera
- el codificador
- dinamo taquimétrica

cuya tarea es de dar la potencia necesaria al movimiento y detectar el estado del sistema.

Estos elementos están entre ellos conectados de manera tal para permitir la ejecución de movimientos de manera controlada.

Esquema de control retroactivado

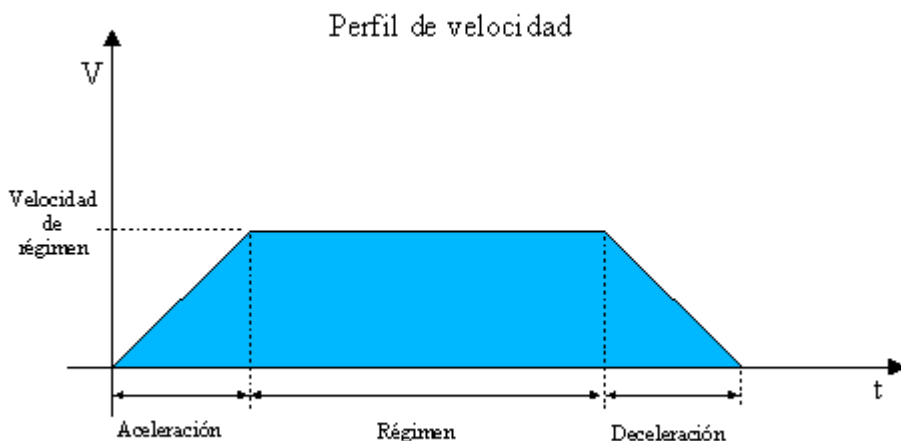


La tarea del control numérico es la de controlar la posición y el movimiento de un eje.

El movimiento de un eje puede ser descompuesto en 5 fases:

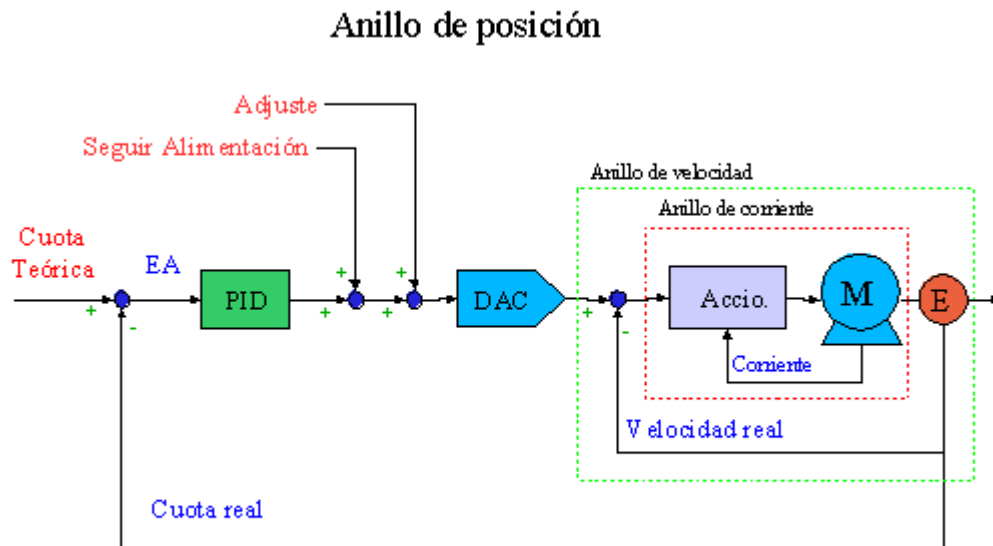
Aceleración	fase inicial durante la cual el eje aumenta progresivamente la velocidad hasta alcanzar la velocidad deseada
Régimen	fase intermedia durante la cual el eje se mueve con velocidad constante (esta fase puede no estar si el espacio por recorrer es menor del espacio recorrido en aceleración y deceleración)
Deceleración	fase durante la cual el eje reduce progresivamente la velocidad hasta 0
Ventana	fase de espera que el error de anillo se reduzca al valor indicado en la configuración como "ventana de llegada en cota"
Cota	fin del movimiento

Al final del movimiento el eje deberá de ser posicionado dentro de un intervalo llamado ventana de llegada en cota (define la tolerancia en el posicionamiento del eje). Si esto no acontece dentro de 5 segundos de la fin prevista del movimiento, el sistema genera un error de sistema de "movimiento no finalizado".



Para cada movimiento el control numérico calcula un perfil de velocidad como el que está visible en figura, luego calcula las cotas teóricas dividiendo el perfil de velocidad en intervalos temporales pares al tiempo de refresco del eje y calculando el área de cada parte. El área constituye el incremento de cota que el eje debe de alcanzar en el intervalo de tiempo necesario para respetar el perfil de velocidad visto arriba.

El control del eje es implementado a través de un controlador PID que se ocupa de "cerrar el anillo de posición", o sea de dar una referencia de velocidad al accionamiento calculado en base a la cota que se quiere alcanzar (cota teórica) y a la cota real leída por el codificador. La diferencia entre cota real y cota teórica es llamada **Error de Anillo**.



Esquema del control eje de Albatros

10.1.12 Correctores de linealidad

La tabla de los correctores de linealidad de un eje es considerada como una matriz, llamada *NombreGrupo.NombreSubgrupo.NombreEje#correctors* o *NombreGrupo.NombreEje#correctors*, y se puede usar en todas las instrucciones en las que se accede a matrices, elementos de matriz y filas de matriz.

El número de columnas de la matriz se corresponde con el número de ejes configurados en la ventana de los correctores de linealidad en configuración ejes. Los correctores automáticos están en la primera columna. Todos los valores de corrección son de tipo float. Se puede lograr el número total de filas de la matriz con la instrucción GPL [LASTELEM](#).

Es posible acceder a estas matrices tanto en lectura como en escritura y, si modificadas, los valores son usados de inmediato para corregir la cota, solamente si la corrección ha sido habilitada.

Ejemplo:

```
Function ReadCorr
local i as integer
local j as integer
local row as integer
local firstvalue as float

; lectura del primer valor de corrección automática del eje AX
firstvalue = X.Ax#correctors[1][1]
; número ejes en la ventana de los correctores de linealidad
setval 3 column
lastelem X.Ax#correctors row
; incremento de un valor constante todas las correcciones
for i 1 row
  for j 1 column
    X.Ax#correctors[1][j] = X.Ax#correctors[i][j] + 0.025
  next
next
```

fret

10.1.13 Gestión de los mensajes en lengua

Como descrito en el capítulo Composición del sistema, Albatros soporta la [visualización de los mensajes de texto en más idiomas](#).

Este soporte se ejecuta a través el uso de TpaLangs, que es un programa externo a Albatros que gestiona unos archivos de mensajes. Este programa ayuda la traducción de los mensajes en las diferentes lenguas.

Texto asociado a Errores de Ciclo y Mensajes

Un tipo particular de textos que normalmente son visualizados por Albatros son los Mensajes y los Errores de Ciclo generados por el código GPL.

Estos son normalmente definidos por quien desarrolla el código GPL durante la escritura del mismo código.

Para facilitar el trabajo del programador el editor del GPL permite insertar el texto de un mensaje directamente desde Albatros, o sea sin tener que utilizar TpaLangs.

Una segunda posibilidad para manejar los mensajes en lengua es provista por el uso de la instrucción GPL [DEFMSG](#).

10.1.14 Gestión de los errores de sistema

Cuando se verifica un [error de sistema](#) (Véase capítulo **Errores y Advertencias->Introducción**) el comportamiento normal del control ha sido de terminar todas las tareas: la gestión de los errores de sistema permite evitar la terminación de las tareas para las cuales ha sido habilitada.

Errores de sistema generados de manera predeterminada, por subdesbordamiento de pila y por desbordamiento de pila son gestionados directamente por el control sin llamar la función de gestión de los errores de sistema: la tarea es puesta en estado HOLD.

La función de gestión de los errores

Al interno del código GPL deben de ser definidas una o más funciones para el análisis del error de sistema y por la consecuente definición de oportunas acciones para poner en seguridad la máquina. La función de llamar es pasada como parámetro a la instrucción GPL [ONERRSYS](#). (Véase capítulo **Lenguaje GPL->Instrucciones->Gestión de flujo->ONERRSYS**).

Al verificarse de un error de sistema la tarea que ha generado el error es puesta en estado HOLD. Si las tareas autorun generan errores de sistema son lanzadas de nuevo solo si el error de sistema no es un FAULT. Si el error de sistema es generado sin número de tarea es puesta en HOLD la tarea corriente.

10.2 Funciones especiales

10.2.1 Personalización movimiento ejes

El interfaz gráfica del sistema Albatros prevé la posibilidad de desplazar los ejes en manual y da un soporte gráfico al calibrado de estos.

El movimiento en manual de los ejes se ejecuta por medio del panel de desplazamiento ejes, el calibrado puede ser efectuado a través del panel de calibrado. Ambos pueden ser llamado por la función de Diagnóstico y por los cuadros sinópticos.

En ambos casos el movimiento del eje es mandado por un grupo de funciones GPL cuya ejecución queda invisible al usuario.

El sistema dispone de un conjunto predefinido de estas funciones que resultan adecuadas en la mayoría de los casos. De todas formas puede ser necesario personalizarlas, por ejemplo, para introducir unas restricciones al desplazamiento de los ejes para el estado de la máquina o bien para gestionar dispositivos auxiliares como los frenos ejes.

La personalización se hace por medio de la creación de dos funciones GPL por cada eje: una para el desplazamiento en manual y otra para el calibrado. Estas funciones son opcionales, por lo tanto se utilizan si son detectadas por el sistema, en otro caso se utilizan las funciones estándar. Además, se considera también una personalización parcial de las funciones de movimiento.

Desplazamiento en manual

Las funciones de *movimiento manual* personalizadas deben respetar las siguientes específicas:

- La función tendrá que pertenecer al mismo grupo o subgrupo al cual pertenece el eje al cual se refiere.
- El nombre de la función tendrá que ser **MoveAx#nombre_eje** donde nombre_eje se debe sustituir con el nombre del eje definido en configuración. Por ejemplo, para el eje X el nombre de la función será: MoveAx#X.
- La función tendrá que considerar los siguientes parámetros:

1. **Acción solicitada.** Puede ser un movimiento a una cota absoluta, un movimiento incremental, una parada, etc. Las acciones son identificadas por un número entero, el compilador GPL prevé unas constantes predefinidas asociadas a las acciones:

<code>_MOVAXABS</code>	movimiento a cota absoluta
<code>_MOVAXINC</code>	movimiento a cota incremental
<code>_MOVAXSET</code>	configuración de la cota
<code>_MOVAXFREE</code>	configuración estado free
<code>_MOVAXNORMAL</code>	configuración estado normal
<code>_MOVAXEND</code>	reposición del estado eje al final del movimiento (no sirve para detener el eje)

2. **Resultado.** Sirve al sistema para saber si la acción requerida está gestionada por la función personalizada. Si la acción no está gestionada se utiliza la función estándar correspondiente. Se trata por lo tanto de un valor de retorno que la función personalizada debe programar y con este fin se define como un parámetro pasado por referencia (array de un solo elemento).
3. **Velocidad.** Es significativa solamente cuando la acción pedida es un movimiento y programa su velocidad.
4. **Cota.** Significativa solamente para las acciones de movimiento y de programación cota.

Ejemplo de función de desplazamiento personalizada:

```
Function MoveAx#X
  param action as integer
  param result[1] as integer
  param speed as float
  param position as double

  setval 1, result[1]

  select action
  case _MOVAXEND
    fcall EndMovement
  case _MOVAXABS
    fcall AbsMovement X, speed, position
  case _MOVAXINC
    fcall IncMovement X, speed, position
  case _MOVAXSET
    fcall PositionSet X, position
  case _MOVAXFREE
    fcall FreeAxis
  case _MOVAXNORMAL
    fcall NormalAxis
  case else
    call Unknown
  endselect

  fret

Unknown:
  setval 0, result[1]
  ret
```

Las funciones EndMovement, AbsMovement, etc. (los nombres no son vinculantes) deberán implementar la gestión personalizada de las acciones pedidas. Para facilitar el trabajo del programador mostramos las funciones estándar de desplazamiento que pueden ser útiles como traza para el desarrollo de aquellas personalizadas.

Calibrado

Las funciones de calibrado personalizadas deben respetar las siguientes específicas:

- La función tendrá que pertenecer al mismo grupo o subgrupo al que pertenece el eje al que se refiere.
- El nombre de la función tendrá que ser **CalibAx#nombre_eje**, donde nombre_eje se debe sustituir con el nombre del eje definido en configuración. Por ejemplo, para el eje X el nombre de la función será: CalibAx#X.
- La función tendrá que considerar los siguientes parámetros:
 1. **Acción solicitada.** Puede ser un movimiento punto-punto o un movimiento interpolado.

2. **Resultado.** Sirve al sistema para saber si la acción requerida está gestionada por la función personalizada. Si la acción no está gestionada se utiliza la función estándar correspondiente.
3. **Velocidad.** Velocidad de movimiento durante el calibrado
4. **Cota Positiva.** Cota positiva del movimiento alternado de calibrado.
5. **Cota Negativa.** Cota negativa del movimiento alternado de calibrado.
6. **Espera.** Espera entre un movimiento y aquel siguiente

ATENCIÓN: hay que tener presente que en algunas ocasiones acciones efectuadas en el panel de calibrado comportan la ejecución de la función de desplazamiento eje. Por ejemplo, al final del movimiento de calibrado (cuando es presionado el botón de parada) se efectúa una operación de reposición del eje por la cual se llama la función de desplazamiento eje personalizada con el parámetro acción programado a `_MOVAXEND`. De la misma manera cuando se modifica la cota del eje desde el panel de calibrado es llamada la función de desplazamiento eje con el parámetro acción programado a `_MOVAXSET`.

Ejemplo de función de calibrado personalizada:

```
Function CalibAx#X
  param action as integer
  param result[1] as integer
  param speed as float
  param PosPosition as double
  param NegPosition as double
  param waitTime as float

  setval 1,result[1]

  select action
  case _CALAXPP
    fcall PPCalibration X, speed, PosPosition, NegPosition, _
      waitTime
  case _CALAXINT
    fcall IntCalibration X, speed, PosPosition, NegPosition, _
      waitTime
  case else
    call Unknown
  endselect

  fret

Unknown:
  setval 0, result[1]
  ret
```

Las funciones PPCalibration y IntCalibration (los nombres no son vinculantes) deberán implementar la gestión personalizada de las acciones pedidas. Para facilitar el trabajo del programador mostramos las funciones estándar de calibrado que pueden ser útiles como traza para el desarrollo de aquellas personalizadas.

Interacción con la ventana de Desplazamiento eje en manual

Las funciones para la interacción con la ventana de desplazamiento manual tienen que respetar las siguientes especificaciones:

- La función tendrá que pertenecer al mismo grupo o subgrupo al que pertenece el eje al que se refiere.
- El nombre de la función tendrá que ser **MoveAx#nombre_eje#Acción** donde nombre_eje tiene que ser sustituido con el nombre del eje definido en configuración y Acción puede asumir una entre las siguientes definiciones:

OPEN	indica que el usuario acaba de abrir la ventana de desplazamiento eje
CLOSE	indica que el usuario va a cerrar la ventana de desplazamiento eje
ACTIVE	señala que la ventana de desplazamiento eje está activa
INACTIVE	señala que la ventana de desplazamiento eje no está activa
JOG	indica que está programado un movimiento para desplazamiento gestionado en tiempo real por el operador
STEP	indica que está programado un movimiento con desplazamiento de paso predefinido
ABSOLUTE	indica que está programado un movimiento con desplazamiento a coordenada establecida

Por ejemplo, si la ventana de desplazamiento eje para el eje X ha sido abierta, será llamada la función con nombre `MoveAx#X#Open`.

Modificaciones a la ventana de Desplazamiento eje en manual

Es posible agregar hasta 4 pulsadores a la ventana de desplazamiento eje. Tienen que ser definidas, en el mismo subgrupo en el cual es definido el eje interesado, funciones GPL a nombre fijo MoveAx#NombreEje#BUTTONtexto. NombreEje representa el nombre del eje interesado y texto representa el texto que aparecerá sobre el pulsador. El texto puede contener el carácter '&' para introducir un acelerador desde teclado. Si texto comienza con un número comprendido entre 1 y 4, este número es considerado como la posición en la cual será insertado el pulsador en la ventana de desplazamiento eje. El texto del pulsador se puede traducir introduciendo una DEFMSG, en el grupo donde hay el eje, que tiene como identificador MOVEAX#BUTTONtexto. La presión del pulsador personalizado comporta la ejecución de la función GPL asociada. No es ejecutada ninguna espera de fin ejecución, ni ninguna verifica de inicio ejecución de la función.

10.2.2 Funciones estándar de desplazamiento y calibrado

Las que siguen son las funciones estándar utilizadas por el panel de desplazamiento eje y del panel de calibrado.

Las funciones varían según el tipo de eje interesado: conteo, paso-paso, etc.

Estas funciones pueden ser [personalizadas](#).

Funciones estándar de desplazamiento manual**Movimiento a cota absoluta**

```

; para ejes paso-paso
Function AbsMovement
  param axisname as axis
  param speed as float
  param position as double

  ifstill      axisname goto move
  fret
move:
  setvel      axisname, speed
  movabs     axisname, position
  waitstill  axisname
  fret

; para todos los demás tipos de eje
Function AbsMovement
  param axisname as axis
  param speed as float
  param position as double

  iftarget   axisname goto move
  ifstill    axisname goto move
  fret
move:
  setvel      axisname, speed
  movabs     axisname, position
  waitstill  axisname
  fret

```

Movimiento incremental

```

; para ejes paso-paso
Function IncMovement
  param axisname as axis
  param speed as float
  param position as double

  ifstill      axisname goto move
  fret
move:
  setvel      axisname, speed
  movinc     axisname, position
  waitstill  axisname
  fret

; para todos los demás tipos de ejes

```

```
Function IncMovement
  param axisname as axis
  param speed as float
  param position as double

  iftarget    axisname goto move
  ifstill    axisname goto move
  fret

move:
  setvel    axisname, speed
  movinc    axisname, position
  waitstill axisname
  fret
```

Configuración cota

```
; para ejes cuenta
Function PositionSet
  param axisname as axis
  param position as double

  setquote    axisname, position
  fret

; para ejes paso-paso
Function PositionSet
  param axisname as axis
  param position as double

  ifstill    axisname goto set
  fret
set:
  setquote    axisname, position
  fret

; para todos los demás tipos de ejes
Function PositionSet
  param axisname as axis
  param position as double

  iftarget    axisname goto set
  ifstill    axisname goto set
  fret
set:
  setquote    axisname, position
  fret
```

Configuración estado Libre

```
Function FreeAxis
  param axisname as axis

  free    axisname
  fret
```

Configuración estado Normal

```
Function NormalAxis
  param axisname as axis

  normal    axisname
  fret
```

Funciones estándar de calibrado

Calibrado en punto-punto

```
; para ejes paso-paso
Function PPCalibration
```

```

param axisname as axis
param speed as float
param PosPosition as double
param NegPosition as double
param WaitTime as float

setvel      axisname, speed
loop:
movabs      axisname, PosPosition
waitstill   axisname
delay       WaitTime
movabs      axisname, NegPosition
waitstill   axisname
delay       WaitTime
goto        loop
fret

; para todos los demás tipos de ejes
Function PPCalibration
param axisname as axis
param speed as float
param PosPosition as double
param NegPosition as double
param WaitTime as float

setvel      axisname, speed
loop:
movabs      axisname, PosPosition
waitstill   axisname
ifquotet   axisname,<>,PosPosition goto exit
delay       WaitTime
movabs      axisname, NegPosition
waitstill   axisname
ifquotet   axisname,<>,NegPosition goto exit
delay       WaitTime
goto        loop
exit:
fret

```

Calibrado en interpolación

```

Function IntCalibration
param axisname as axis
param speed as float
param PosPosition as double
param NegPosition as double
param WaitTime as float

setveli     axisname, speed
loop:
linearabs   axisname, PosPosition
waitstill   axisname
ifquotet   axisname,<>,PosPosition goto exit
delay       WaitTime
linearabs   axisname, NegPosition
waitstill   axisname
ifquotet   axisname,<>,NegPosition goto exit
delay       WaitTime
goto        loop
exit:
fret

```

10.2.3 Función OnUIEnd#

La función "OnUIEnd#" está ejecutada, si está presente, por Albatros antes de terminar todas las tareas de un módulo. Debe ser definida el archivo de funciones del intergrupo. El tiempo máximo de ejecución de la función "OnUIEnd#" es 2 segundos, después que Albatros terminará todas las tareas.

El tiempo máximo de espera para que esta función termine la configuración se le puede configurar en Tpa.INI, en la sección [Albatros], bajo Timeout.OnUIEnd=*valor*, donde *valor* está expresado en mili-segundos y no puede ser mayor de 60000.

10.2.4 Función OnUIPlugged#

La función "OnUIPlugged#" se ejecuta cuando hay que saber, por ejemplo, en el inicio de una instalación, se Albatros ha comunicado con el módulo remoto. Esta función tiene que ser definida en el intergrupo.

10.2.5 Función OnUIUnplugged#

La función "OnUIUnplugged#" se ejecuta antes que termine la ejecución de Albatros (y entonces antes que Albatros se desconecte de un módulo). Esta función debe ser definida en el intergrupo. Albatros ejecuta esta función entre máximo 2 segundos, durante los cuales se leen

- Errores de ciclo
- Errores de sistema
- Mensajes

Al final de la ejecución Albatros termina.

El tiempo máximo de espera para que esta función termine la configuración se le puede configurar en Tpa.INI, en la sección [Albatros], bajo Timeout.OnUIUnplugged=*valor*, donde *valor* está expresado en mili-segundos y no puede ser mayor de 60000.

10.3 Instrucciones

10.3.1 Convenciones

Las páginas siguientes son organizadas en forma de tarjetas y describen, relativamente en cada instrucción:

- la sintaxis
- una descripción de los argumentos: tipo de dato y valores admitidos
- una descripción de funcionamiento
- eventuales notas
- eventuales ejemplos

Las instrucciones del mismo tipo han sido agrupadas para facilitar el aprendizaje y la consulta.

10.3.2 Tipologías de las instrucciones del lenguaje GPL

El lenguaje está compuesto por instrucciones que se pueden agrupar así:

Instrucciones para la gestión de las Entradas/Salidas

GETFEED	lee la velocidad de avance de anulación
INPANALOG	lee una entrada analógica
INPFLAGPORT	lee un puerto indicador
INPPORT	lee un puerto digital
MULTIINPPORT	lee hasta 4 puertos de salida
MULTIOUTPORT	configura hasta 4 puertos de salida
MULTIRESETFLAG	programa más indicadores a 0
MULTIRESETOUT	programa más salidas a 0
MULTISETFLAG	programa más indicadores a 1
MULTISETOUT	programa más salidas a 1
MULTIWAITFLAG	espera el estado de un bit indicador o de un interruptor indicador
MULTIWAITINPUT	espera el estado de más entradas
OUTANALOG	modifica una salida analógica
OUTFLAGPORT	modifica un puerto indicador
OUTPORT	modifica un puerto digital
RESETFLAG	programa un indicador a 0
RESETOUT	programa una salida a 0
SETFLAG	programa un indicador a 1
SETOUT	programa una salida a 1
WAITFLAG	espera el estado de un bit indicador o de un interruptor indicador
WAITINPUT	espera el estado de una entrada

[WAITPERSISTINPUT](#) espera el estado persistente de una entrada

Instrucciones para la gestión de los Ejes

CHAIN	espera el estado persistente de una entrada
CIRCABS	interpolación circular absoluta
CIRCINC	interpolación circular incremental
CIRCLE	realiza un círculo
COORDIN	movimiento ejes coordenados
DISABLECORRECTION	inhabilita la corrección lineal para el eje especificado
EMERGENCYSTOP	fuerza una parada de emergencia de los ejes
ENABLECORRECTION	habilita la corrección lineal para el eje especificado
ENDMOV	termina el movimiento de un eje
FASTREAD	lectura rápida cotas ejes
FREE	pone el eje en estado libre
HELICABS	interpolación helicoidal absoluta
HELICINC	interpolación helicoidal incremental
JERKCONTROL	habilita o deshabilita el control de los movimientos en interpolación
JERKSMOOTH	enlaza de manera continua, o sea con continuidad de aceleración y de velocidad, los perfiles de velocidad de los ejes durante los movimientos de contorneado.
LINEARABS	interpolación lineal absoluta
LINEARINC	interpolación lineal incremental
MOVABS	movimiento ejes de manera absoluta
MOVINC	movimiento ejes de manera incremental
MULTIABS	interpolación lineal multieje de manera absoluta
MULTIINC	interpolación lineal multieje de manera incremental
NORMAL	quita el estado libre de un eje
RESRIFLOC	restablece referencias iniciales
SETINDEXINTERP	asocia a un eje una variable para la cuenta de los bloqueos de interpolación ejecutados
SETLABELINTERP	asocia a un eje una variable para identificar un bloqueo de desplazamiento
SETPFLY	puesta a cero al vuelo
SETPFLYCHAINSTRAT	controla el comportamiento del eje Esclavo a frente de una instrucción setpfly en el maestro
SETPZERO	puesta a cero en la muesca de cero
SETPZEROCHAINSTRAT	controla el comportamiento del eje subordinado a frente de una instrucción setpzero en el maestro
SETQUOTE	programa la cota
SETQUOTECHAINSTRAT	controla el comportamiento del eje subordinado a frente de una instrucción setquote en el maestro
SETRIFLOC	programa referencias espaciales
SETTOLERANCE	programa los valores de tolerancia para la interpolación lineal
START	reanuda el movimiento de un eje
STARTINTERP	fuerza el inicio de una interpolación
STOP	suspende el movimiento de un eje
SWITCHENC	permite sustituir el codificador de un eje con aquel de otro eje
WAITACC	espera que el eje esté en aceleración
WAITCOLL	espera que el eje supere una coordenada desde la cual empezar a verificar la presencia de una colisión
WAITDEC	espera que el eje esté en deceleración
WAITREG	espera que el eje esté en régimen
WAITSTILL	espera que la cota final sea igual a la cota teórica
WAITTARGET	espera que el eje esté en destino
WAITWIN	espera que el eje esté en ventana

Instrucciones para la gestión de los Parámetros eje Lectura/Escritura

[DEVICEID](#) escribe la dirección lógica asociada a un dispositivo

[GETAXIS](#) lee uno o más datos de un eje

Movimiento punto a punto

SETACC	programa la aceleración
SETDEC	programa la deceleración
SETDERIV	programa el coeficiente de acción derivada
SETFEED	programa la velocidad de avance punto-punto
SETFEEDF	programa el control de las previsiones

SETFEEDFA	programa el feed forward de aceleración
SETINTEG	programa el coeficiente de acción integral
SETMULTIFEED	programa el valor porcentual de la velocidad de avance de alimentación de los ejes indicados
SETPROP	programa el coeficiente de acción proporcional
SETSLOPE	programa el tipo de rampa en los movimientos rápidos
SETVEL	programa la velocidad

Movimiento interpolado

LOOKAHEAD	programa el lookahead de interpolación
SETACCI	programa la aceleración por interpolación
SETACCLIMIT	habilita y inhabilita el cálculo automático de la velocidad de régimen de interpolación
SETACCSTRATEGY	selecciona el tipo de aceleración
SETAXPARTYPE	cambia el conjunto de parámetros eje de uso
SETCONTORNATURE	programa el ángulo de contorneado
SETDECI	programa la deceleración por interpolación
SETDERIVI	programa el coeficiente de acción derivada de interpolación
SETFEEDFAI	programa el feed forward de aceleración en interpolación
SETFEEDFI	programa el feed forward en interpolación
SETFEEDI	programa la velocidad de avance en interpolación
SETINTEGI	programa el coeficiente de acción integral de interpolación
SETPROPI	programa el coeficiente de acción proporcional de interpolación
SETSLOPEI	programa el tipo de rampa en los movimientos rápidos
SETSLOWPARAM	modifica los parámetros necesarios para calcular la velocidad de desaceleración en caso de que sea activa la funcionalidad de desaceleración en el contorneado
SETVELI	programa la velocidad de interpolación
SETVELLIMIT	programa cada componente de velocidad del eje especificado

Movimiento coordinado

SETFEEDCOORD	programa el valor porcentual de variación máxima instantánea de la velocidad de avance del eje
SETOFFSET	programa un desplazamiento de cota

Movimiento concatenado

RATIO	programa la relación de encadenamiento de un eje esclavo con respecto al propio eje maestro
SETDYNRATIO	varía de manera dinámica, durante el movimiento del eje máster, la relación de encadenamiento

Parámetros genéricos

DYNLIMIT	habilita o inhabilita el test sobre la superación de los límites eje de manera dinámica
ENABLESTARTCONTROL	habilita y programa el tiempo de espera para el control de la fallida partida o repentino paro del eje
NOTCHFILTER	configura la frecuencia de corte del filtro notch para el eje especificado
RESLIMNEG	inhabilita el límite negativo del eje
RESLIMPOS	inhabilita el límite positivo del eje
SETADJUST	programa el ajuste de un eje
SETBACKLASH	disminuye o elimina los efectos de los juegos mecánicos en la trayectoria del eje
SETBIGWINFACTOR	modifica el factor de multiplicación para el cálculo de la ventana grande sobre el eje seleccionado
SETDEADBAND	configura los parámetros de tensión mínima para el eje indicado
SETENCLIMIT	modifica el límite de enlace codificador incorrecto
SETINDEXEN	activa o desactiva sobre el eje la puesta a cero de la cota en correspondencia de la muesca de cero
SETINTEGTIME	configura el número de muestreos del error de anillo utilizado para el cálculo de la componente integral
SETIRMP	programa la velocidad de inicio rampa
SETLIMNEG	programa el límite negativo del eje
SETLIMPOS	programa el límite positivo del eje
SETMAXER	programa el valor máximo de persecución tolerado

SETMAXERNEG	programa el valor máximo de persecución tolerado (dirección negativa)
SETMAXERPOS	programa el valor máximo de persecución tolerado (dirección positiva)
SETMAXERTYPE	programa el tipo de test en el error servo
SETPHASESINV	activa o desactiva en el eje indicado la inversión fases
SETREFINV	habilita o inhabilita en el eje indicado la inversión de referencia de velocidad
SETRESOLUTION	cambia la resolución de un eje

Instrucciones para la gestión de los Contadores

DECOUNTER	disminuye un contador
INCOUNTER	incrementa un contador
SETCOUNTER	programa un contador

Instrucciones para la gestión de los Temporizadores

HOLDTIMER	bloquea un temporizado
SETTIMER	programa un temporizador
STARTTIMER	inicia el temporizador

Instrucciones para la gestión de Variables, Vectores y Matrices

CLEAR	puesta a cero de variable, vector, matriz
FIND	búsqueda de un elemento
FINDB	búsqueda de un elemento vector o matriz ordenados de manera creciente
LASTELEM	último elemento de un vector o matriz
LOCAL	declaración de variable, vector, matriz local
MOVEMAT	copia una línea de matriz en otra
PARAM	declaración de un parámetro de función
SETVAL	modifica una variable
SORT	orden de vector o matriz

Instrucciones para la gestión de las Cadenas

ADDSTRING	concatena dos cadenas
CONTROLCHAR	programa un carácter de control en una variable cadena
LEFT	extrae los primeros caracteres
LEN	lee la longitud de una cadena
MID	extrae unos caracteres
RIGHT	extrae los últimos caracteres
SEARCH	busca en una cadena
SETSTRING	modifica una variable cadena
STR	convierte desde número a cadena
VAL	convierte desde cadena a número

Instrucciones para la gestión de las Comunicaciones

CLEARRECEIVE	vacía la lista de RECEIVE que se debe satisfacer
COMCLEARRXBUFFER	vacía el búfer de recepción de un puerto serial
COMCLOSE	cierra un puerto serial
COMGETERROR	lee el código de error
COMGETRXCOUNT	lee el número de bytes presentes en el búfer de recepción
COMOPEN	abre un puerto serial
COMREAD	lee desde el puerto serial
COMREADSTRING	lee una cadena desde el puerto serial
COMWRITE	escribe en el puerto serial
COMWRITESTRING	escribe una cadena en el puerto serial
RECEIVE	recepción datos desde el externo
SEND	envío datos al externo
SENDIPC	envía una información IPC
WAITIPC	espera la llegada de una información IPC
WAITRECEIVE	recepción datos desde el externo con espera

Instrucciones matemáticas

ABS	valor absoluto
ADD	suma
AND	AND binario

ARCCOS	arcocoseno
ARCSIN	arcoseno
ARCTAN	arcotangente
COS	coseno
DIV	división
EXP	exponencial
EXPR	resuelve expresiones matemáticas
LOG	logaritmo natural
LOGDEC	logaritmo en base 10
MOD	módulo
MUL	multiplicación
NOT	NOT binario
OR	OR binario
RANDOM	genera un número casual
RESETBIT	programa a 0 un bit
ROUND	redondeo
SETBIT	programa a 1 un bit
SHIFTL	gira los bit a la izquierda
SHIFTR	gira los bit a la derecha
SIN	seno
SQR	raíz cuadrada
SUB	resta
TAN	tangente
TRUNC	truncamiento
XOR	XOR binario

Instrucciones para la gestión de Multitask

ENDMAIL	señala la fin de la ejecución de un comando
ENDREALTIMETASK	termina una tarea de tiempo real
ENDTASK	termina una tarea
GETPRIORITYLEVEL	lee el nivel de prioridad de la tarea corriente
GETREALTIME	devuelve el tiempo pasado desde el inicio del tiempo real ejes
GETREALTIMECOUNT	devuelve el número de tiempo reales transcurrido
HOLDTASK	interrumpe la ejecución de una tarea
RESUMETASK	reanuda la ejecución de una tarea
SENDMAIL	envía un comando al buzón de correos 'mail'
SETPRIORITYLEVEL	programa el nivel de prioridad para la tarea corriente
STARTREALTIMETASK	inicia una tarea de tiempo real
STARTTASK	inicia la ejecución de una tarea
STOPTASK	suspende la ejecución de una tarea y para el movimiento de los ejes asociados
WAITMAIL	recibe un comando del buzón de correos 'mail'
WAITTASK	espera que una tarea termine la ejecución

Instrucciones para la gestión de Flujo

CALL	llama a subprograma
DELONFLAG	inhabilita la gestión de la emergencia en bit indicador o interruptor indicador
DELONINPUT	inhabilita la gestión de la emergencia en entrada digital
ENDREP	acaba la repetición bloqueo con REPEAT
FCALL	llamada a función
FOR/NEXT	extensión de REPEAT
FRET	retorno de llamada a función
GOTO	salto a etiqueta
IF	test sobre una variable
IFACC	test si el eje está acelerado
IFAND	test sobre operación de AND
IFBIT	test sobre bit
IFBLACKBOX	test si el registro de la actividad de los dispositivos lógicos está activa
IFCHANGEVEL	test si el eje está en cambio velocidad
IFCOUNTER	test sobre un contador
IFDEC	test si el eje está decelerado
IFDIR	test sobre ruta eje
IFERRAN	test sobre error de anillo
IFERROR	test sobre el error de ciclo activo
IFFLAG	test sobre un indicador
IFINPUT	test sobre entrada

IFMESSAGE	test sobre el mensaje activo
IFOR	test obre operación de OR
IFOUTPUT	test sobre una salida
IFQUOTER	test sobre cota real
IFQUOTET	test sobre cota objetivo
IFRECEIVED	test sobre recepción datos
IFREG	test si el eje está correcto
IFSAME	controla que los dos argumentos se refieran al mismo dato
IFSTILL	test si el eje está parado
IFSTR	test sobre cadena
IFTARGET	test si el eje está en objetivo
IFTASKHOLD	test si la función paralela está suspendida
IFTASKRUN	test si la función paralela está ejecutada
IFTIMER	test sobre un temporizador
IFVALUE	test sobre una variable
IFVEL	test sobre velocidad eje
IFWIN	test si el eje está en la ventana
IFXOR	test sobre operación de XOR
NEXT	acaba repetición bloqueo de FOR
ONERRSYS	configura la llamada a una función sobre error de sistema
ONFLAG	emergencia sobre bit indicador o indicador interruptor
ONINPUT	emergencia sobre entrada digital
REPEAT	repeticiones de un bloqueo de instrucciones
RET	vuelta desde subprograma
SELECT	selección múltiple con salto
TESTIPC	controla la presencia de una información IPC
TESTMAIL	test y recepción de un comando

Instrucciones Varias

CLEARERRORS	borra todos los errores de ciclo del módulo
CLEARMESSAGES	borra todos los mensajes del módulo
DEFMSG	define un mensaje de grupo
DELAY	bloquea la función corriente por un cierto tiempo
DELERROR	borra un error de ciclo anterior
DELMESSAGE	borra un mensaje anterior
ERROR	envía al PC un error de ciclo
IFDEF/ELSEDEF/ENDDEF	test para la compilación condicional
MESSAGE	envía al PC un mensaje
SYSFAULT	desactiva la señal de SYSOK
SYSOK	activa la señal de SYSOK
TYPEOF	tipo del argumento
WATCHDOG	habilita, actualiza, inhabilita el controlador de secuencia desde GPL en el módulo hardware TMSWD

Instrucciones para la gestión de MECHATROLINK-II

MECCOMMAND	envía un comando al accionamiento del eje
MECGETPARAM	lee un parámetro del eje indicado
MECSETPARAM	escribe un parámetro en el eje indicado
MECGETSTATUS	lee los valores de STATUS, ALARM y IO_MON

Instrucciones para la gestión del Bus de campo estándar

AXCONTROL	configura un valor para la ControlWord
AXSTATUS	restituye el valor contenido en la StatusWord
CNBYDEVICE	restituye el número de tarjeta y nodo de un dispositivo
READDICTIONARY	lee el contenido de un objeto del diccionario
WRITEDICTIONARY	escribe el contenido de un objeto del diccionario

Instrucciones para la gestión EtherCAT

ACTIVATEMODE	programa una modalidad operativa
ECATGETREGISTER	restituye el contenido de un registro del ESC (EtherCAT Slave Controller)
ECATSETREGISTER	escribe el contenido de un registro del ESC (EtherCAT Slave Controller)
GETPDO	restituye un objeto dentro de un PDO EtherCAT
SETEOE	activa o desactiva el Sniffer
SETPDO	configura un objeto dentro de un PDO EtherCAT

Instrucciones para la gestión del bus CAN

GETCNSTATE	restituye el estado del protocolo NMT para un nodo de una tarjeta
GETSDOERROR	restituye el último error que se produjo
GETMNSTATE	restituye el estado del protocolo NMT por el nodo maestro de la tarjeta
RECEIVEPDO	lee el contenido de un PDO asíncrono
SENDPDO	escribe el contenido de un PDO asíncrono
SETNMTSTATE	configura el estado del protocolo NMT para el nodo de la tarjeta

Instrucciones para la Simulación

DISABLE	inhabilita uno o más ejes
DISABLEFORCEDINPUT	inhabilita el forzamiento de las entradas
ENABLE	habilita uno o más ejes
ENABLEFORCEDINPUT	habilita el forzamiento de las entradas
RESETFORCEDINPUT	fuerza una entrada en OFF
SETFORCEDANALOG	fuerza una entrada analógica
SETFORCEDINPUT	fuerza una entrada en ON
SETFORCEDPORT	fuerza un puerto de entradas

Instrucciones para la funcionalidad "BlackBox"

ENDBLACKBOX	acaba la funcionalidad de registro
PAUSEBLACKBOX	suspende la funcionalidad de registro
STARTBLACKBOX	inicia la funcionalidad de registro

Instrucciones para la gestión ISO

ISOG0	configura el movimiento en rápido
ISOG1	configura el movimiento interpolado
ISOG9	configura la parada forzada del movimiento
ISOG90	configura la interpretación de las cotas como cotas absolutas
ISOG91	configura la interpretación de las cotas como cotas relativas
ISOG93	configura la interpretación de la velocidad como inversión del tiempo
ISOG94	configura la interpretación de la velocidad como unidad de medida por minuto
ISOG216	define las matrices para los parámetros de la máquina
ISOG217	describe los ejes físicos y los ejes virtuales que constituyen la máquina
ISOM2	libera los ejes desde el movimiento ISO
ISOM6	selecciona los índices de las matrices de los parámetros
ISOSETPARAM	configura unos parámetros que caracterizan la fluidez del movimiento interpolado ISO
KINEMATICEXPR	configura las expresiones individuales de cinemática invertida y directa

10.3.3 Entradas/Salidas

GETFEED

Sintaxis

GETFEED **variable**

Argumentos

variable feed rate (velocidad de avance)

Descripción

Copia el valor del feed rate, leído por la tarjeta de I/O remoto, en la **variable** especificada. El valor de feedrate está incluso entre 0 y 100 y es un valor en porcentaje.

Opera en una entrada analógica no visible en configuración.

Todas las tarjetas t.p.a. que gestionan el feed rate disponen de un conector dedicado.

INPANALOG

Sintaxis

INPANALOG **nombrentranalógica, variable**

Argumentos

nombreentranalogica nombre del dispositivo de tipo entrada analógica
variable variable

Descripción

Copia el valor de la entrada analógica especificada por **nombreentranalogica** en la **variable** especificada.

INPFLAGPORT**Sintaxis**

INPFLAGPORT **nombrepuertoindicador, variable**

Argumentos

nombrepuertoindicador nombre de dispositivo tipo puerto indicador
variable variable

Descripción

Copia el estado del puerto indicador especificado desde **nombrepuertoindicador** en la **variable** especificada.

El puerto indicador es visto como una máscara de bit. A cada indicador del puerto es asociado un bit. Si un indicador está en el estado "ON", el correspondiente bit es programado a 1.

INPPORT**Sintaxis**

INPPORT **nombrepuerto, variable**

Argumentos

nombrepuerto nombre de dispositivo tipo puerto de entrada
variable variable integer o char

Descripción

Copia el estado del puerto de entrada **nombrepuerto** en la **variable** especificada.

El puerto de entrada es visto como una máscara de bit. Si una entrada del puerto está en estado "ON" el correspondiente bit es programado en 1.

MULTIINPPORT**Sintaxis**

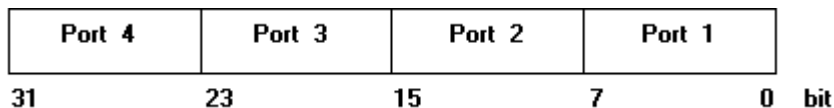
MULTIINPPORT **puerto1[,...,puerto4], variable**

Argumentos

puerto1 suministra los bits desde 0 hasta 7
puerto2 suministra los bits desde 8 hasta 15
puerto3 suministra los bits desde 16 hasta 23
puerto4 suministra los bits desde 24 hasta 31
variable variable entera que recibe los puertos de entrada

Descripción

Lee hasta cuatro puertos de entrada simultáneamente y los escribe en una **variable**. La lectura de los puertos ocurre en manera atómica y esto garantiza que se ejecute la lectura al interior del tiempo real mismo. El puerto1 corresponde al bit más bajo, mientras el puerto4 al bit más alto.

**MULTIOUTPORT****Sintaxis**

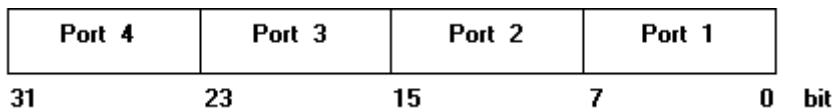
MULTIOUTPORT **valor, nombrepuerto1[,...,nombrepuerto4]**

Argumentos

valor número o variable integer de escribir en los puertos de salida
nombrepuerto1 recibe los bits desde 0 hasta 7
nombrepuerto2 recibe los bits desde 8 hasta 15
nombrepuerto3 recibe los bits desde 16 hasta 23
nombrepuerto4 recibe los bits desde 16 hasta 31

Descripción

Escribe **valor** en cuatro puertos de salida simultáneamente. La escritura de los puertos indicados ocurre de manera atómica y esto garantiza que se ejecute la escritura al interior del mismo tiempo real. Si **nombrepuerto2**, **nombrepuerto3**, **nombrepuerto4** no están especificados, el valor del relativo byte es 0.



MULTIRESETFLAG

Sintaxis

MULTIRESETFLAG **máscara, nombreindicador1[, ..., nombreindicador32]**

Argumentos

máscara máscara de los indicadores interesados - constante o variable
nombreindicador1 nombre de dispositivo tipo indicador

Descripción

De todos los indicadores **nombreindicador** (1÷32), desactiva, o sea pone en "OFF", aquellos cuyo bit en el argumento **máscara** está puesto en 1.

El bit 0 de **máscara** (el de peso más bajo) corresponde a **nombreindicador1**.

MULTIRESETOUT

Sintaxis

MULTIRESETOUT **máscara, nombresalida1[, ..., nombresalida32]**

Argumentos

máscara máscara de las salidas interesadas - constante o variable
nombresalida1 nombre de dispositivo tipo salida

Descripción

De todas las salidas **nombresalida** (1÷32), desactiva aquellas cuyo bit en el argumento **máscara** está puesto en 1. El bit 0 de **máscara** (el que tiene peso más bajo) corresponde a **nombresalida1**.

MULTISETFLAG

Sintaxis

MULTISETFLAG **máscara, nombreindicador1[, ..., nombreindicador32]**

Argumentos

máscara máscara de los indicadores interesados - constante o variable
nombreindicador1 nombre de dispositivo tipo indicador

Descripción

De todos los indicadores **nombreindicador** (1÷32), activa, o sea pone en "ON", aquellos cuyo bit en el argumento **máscara** está puesto en 1. El bit 0 de **máscara** (el de peso más bajo) corresponde a **nombreindicador1**.

MULTISETOUT

Sintaxis

MULTISETOUT **máscara, nombresalida1[, ..., nombresalida32]**

Argumentos

máscara máscara de las salidas interesadas - constante o variable
nombresalida1 nombre de dispositivo tipo salida

Descripción

De todas las salidas **nombresalida** (1÷32), activa aquellas cuyo bit en el argumento **máscara** está puesto en 1.

El bit 0 de **máscara** (el de peso más bajo) corresponde a **nombresalida1**. Si la salida es de tipo monostable es desactivada automáticamente al final de un tiempo de espera fijo de 200 milisegundos.

MULTIWAITFLAG

Sintaxis

MULTIWAITFLAG **máscara, indicador1[, ..., indicador32], estado [, tiempodeespera [, GOTO etiqueta]]**

MULTIWAITFLAG **máscara, indicador1[, ..., indicador32], estado [, tiempodeespera [, CALL nombreesubprograma]]**

MULTIWAITFLAG **máscara, indicador1[, ..., indicador32], estado [, tiempodeespera [, nombrefunción]]**

Argumentos

máscara constante o variable Máscara de los indicadores interesados

indicador1[,...indicador32] nombre de dispositivo tipo indicador

estado constante predefinida. Los valores admitidos son:
 - **ON** estado indicador activado
 - **OFF** estado indicador desactivado

tiempodeespera constante o variable Es el tiempo límite de espera

etiqueta etiqueta de salto (GOTO)

nombreesubprograma etiqueta de subprograma (CALL)

nombrefunción nombre de la función

Descripción

Espera hasta que los indicadores especificados por **indicador1...indicador32** estén en el estado indicado por el parámetro **estado** (ON/OFF).
 De todos los indicadores de verifica aquellos cuyo bit en el argumento **máscara** es habilitado (ON). El bit 0 del argumento **máscara** (aquello de peso menor) corresponde al bit definido por **indicador1**, el bit 1 corresponde al bit definido por **indicador2** y así hasta el bit definido por **indicador32**.
 El parámetro **tiempodeespera** permite configurar un tiempo de espera diferente del que está ofrecido de manera predeterminada que es de un segundo.
 Cuando hay **etiqueta** o **nombreesubprograma** o **nombrefunción**, cuando el tiempo de espera expira el programa va a **etiqueta** o llama **nombreesubprograma** o **nombrefunción**.

MULTIWAITINPUT**Sintaxis**

MULTIWAITINPUT **máscara, entrada1[, ..., entrada32], estado [, tiempodeespera [, GOTO etiqueta]]**

MULTIWAITINPUT **máscara, entrada1[, ..., entrada32], estado [, tiempodeespera [, CALL nombreesubprograma]]**

MULTIWAITINPUT **máscara, entrada[, ..., entrada32], estado [, tiempodeespera [, nombrefunción]]**

Argumentos

máscara constante o variable. Máscara de las entradas interesadas

entrada1[,...entrada32] nombre entrada

estado constante predefinida. Los valores admitidos son:
 - **ON** estado entrada activada
 - **OFF** estado entrada desactivada

tiempodeespera constante o variable. Es el tiempo límite de espera

etiqueta etiqueta de salto (GOTO)

nombreesubprograma etiqueta de subprograma (CALL)

nombrefunción nombre de la función

Descripción

Espera hasta que las entradas especificadas por **entrada1...entrada32** estén en el estado indicado por el parámetro **estado** (ON/OFF).
 De todas las entradas verifica aquellas cuyo bit en el argumento **máscara** esté habilitado (ON). El bit 0 del argumento **máscara** (el de peso más bajo) corresponde al bit definido por **entrada1**, el bit 1 corresponde al bit definido por **entrada2** y así hasta el bit definido por **entrada32**.
 Si no son especificados argumentos opcionales, después de un segundo desde el principio de la ejecución de la instrucción (tiempo de espera predeterminado), es automáticamente generado el siguiente mensaje parametrizado: "Wait inputn ON/OFF". El nombre de la entrada señalada es la de la primera entrada habilitada que no aún no ha satisfecho el estado. Si está presente el parámetro **tiempodeespera** la visualización descrita arriba es generada al terminar el tiempo de espera solicitado. Si, una vez terminado el tiempo de espera, se verifica la condición requerida, es automáticamente generado un mensaje parametrizado para borrar el que ha sido enviado precedentemente.
 Cuando está presente **etiqueta** o **nombreesubprograma** o **nombrefunción**, al terminar el tiempo de espera el programa salta a **etiqueta** o llama **nombreesubprograma** o **nombrefunción**.

OUTANALOG**Sintaxis**

OUTANALOG **nombresalidaanalógica, valor**

Argumentos

nombresalidaanalógica nombre de dispositivo tipo salida analógica o eje

valor constante o variable

Descripción

Define la tensión especificada por **valor** sobre la salida analógica o en el eje expresado por **nombresalidaanalógica**.

OUTFLAGPORT

Sintaxis

OUTFLAGPORT **nombrepuertoindicador, valor**

Argumentos

nombrepuertoindicador nombre de dispositivo tipo puerto indicador
valor constante o variable

Descripción

Copia el **valor** en el puerto indicador especificado por **nombrepuertoindicador**. El parámetro **valor** es visto como una máscara de bit. Para cada bit es asociado un indicador del puerto. Si un bit vale 1 el correspondiente indicador es programado al estado "ON".

OUTPORT

Sintaxis

OUTPORT **nombrepuerto, valor**

Argumentos

nombrepuerto nombre de dispositivo tipo puerto de salida
valor constante o variable, entero o char

Descripción

Copia el **valor** y puerto de salida **nombrepuerto**. El puerto de salida es visto como una máscara de bit. Si un bit vale 1 la correspondiente salida es programada en "ON".

RESETFLAG

Sintaxis

RESETFLAG **nombreindicador**

Argumentos

nombreindicador nombre de dispositivo tipo indicador

Descripción

Desactiva (pone en OFF) el indicador **nombreindicador**.

RESETOUT

Sintaxis

RESETOUT **nombresalida**

Argumentos

nombresalida nombre de dispositivo tipo salida digital

Descripción

Desactiva (pone en OFF) la salida **nombresalida**.

SETFLAG

Sintaxis

SETFLAG **nombreindicador**

Argumentos

nombreindicador nombre de dispositivo tipo indicador

Descripción

Activa (pone en ON) el indicador **nombreindicador**.

SETOUT

Sintaxis

SETOUT **nombresalida**

Argumentos

nombresalida nombre de dispositivo tipo salida digital

Descripción

Activa (pone en ON) la salida **nombresalida**.

Si la salida está configurada como tipo monoestable es desactivada automáticamente al termino de un tiempo de espera fijo de 200 milisegundos.

WAITFLAG

Sintaxis

WAITFLAG **nombreindicador, estado [, tiempodeespera [, GOTO etiqueta]]**
WAITFLAG **nombreindicador, estado [, tiempodeespera [, CALL**
nombreesubprograma]]
WAITFLAG **nombreindicador, estado [, tiempodeespera [, nombrefunción]]**

Argumentos

nombreindicador nombre de dispositivo indicador
estado constante predefinida. Los valores admitidos son:
 - **ON** estado indicador activado
 - **OFF** estado indicador desactivado
tiempodeespera constante o variable. Es el tiempo limite de espera
etiqueta etiqueta de salto (GOTO)
nombreesubprograma etiqueta de subprograma (CALL)
nombrefunción nombre de la función

Descripción

Espera hasta que el indicador **nombreindicador** no asume el estado indicado por el parámetro **estado** (ON/OFF).

Si de los argumentos opcionales está presente solo **tiempodeespera**, el error de ciclo "Indicador **nombreindicador** en espera de **estado**" es generado al termino del tiempo de espera.

Si la condición se verifica después que el tiempo de espera termine, es automáticamente cancelado el error de ciclo enviado en precedencia para esa tarea.

Cuando está presente **etiqueta** o **nombreesubprograma** o **nombrefunción**, al termino del tiempo de espera el programa salta a **etiqueta** o llama **nombreesubprograma** o **nombrefunción** sin generar ninguna visualización automática.

Atención

Para evitar una situación de espera de un indicador durante un ciclo de trabajo, se recomienda programar un tiempo de espera.

WAITINPUT

Sintaxis

WAITINPUT **nombrentrada, estado [, tiempodeespera [, GOTO etiqueta]]**
WAITINPUT **nombrentrada, estado [, tiempodeespera [, CALL**
nombreesubprograma]]
WAITINPUT **nombrentrada, estado [, tiempodeespera [, nombrefunción]]**

Argumentos

nombrentrada nombre entrada
estado constante predefinida. Los valores admitidos son:
 - **ON** estado entrada activado
 - **OFF** estado entrada desactivado
tiempodeespera constante o variable. Tiempo límite de espera
etiqueta etiqueta de salto (GOTO)
nombreesubprograma etiqueta de subprograma (CALL)
nombrefunción nombre de la función

Descripción

Espera hasta que la entrada **nombrentrada** no asume el estado indicado por el parámetro **estado** (ON/OFF).

Si no están especificados argumentos opcionales, después de 20 segundos desde el principio de la ejecución de la instrucción, es automáticamente generado el error de ciclo: "Entrada digital **nombrentrada** en espera de **estado**".

Si de los argumentos opcionales está presente solo **tiempodeespera**, la visualización sobredicha es generada al termino del mismo.

Si la condición se verifica después que el **tiempodeespera** se ha terminado, es automáticamente cancelado el error de ciclo enviado en precedencia para esa tarea.

Cuando está presente **etiqueta** o **nombresubprograma** o **nombrefunción**, al termino del tiempo de espera el programa salta a **etiqueta** o llama **nombresubprograma** o **nombrefunción** sin generar alguna visualización automática.

Atención

Para evitar una situación de espera de una señal de entrada durante un ciclo de trabajo, se recomienda programar un tiempo de espera inferior al tiempo predeterminado (20 segundos).

Ejemplo

[Rutina de Puesta a cero de un eje](#)

WAITPERSISTINPUT

Sintaxis

```
WAITPERSISTINPUT nombrentrada, estado, timepersist [, tiempodeespera [, GOTO
etiqueta]]
WAITPERSISTINPUT nombrentrada, estado, timepersist [, tiempodeespera [, CALL
nombresubprograma]]
WAITPERSISTINPUT nombrentrada, estado, timepersist [, tiempodeespera [,
nombrefunción]]
```

Argumentos

nombrentrada	nombre de dispositivo tipo entrada digital
estado	constante predefinida. Los valores admitidos son: - ON estado entrada activado - OFF estado entrada desactivado
timepersist	constante o variable
tiempodeespera	constante o variable. Tiempo límite de espera
etiqueta	etiqueta de salto (GOTO)
nombresubprograma	etiqueta de subprograma (CALL)
nombrefunción	nombre de la función

Descripción

Espera que la entrada **nombrentrada** esté en el estado indicado por el parámetro **estado** (ON/OFF) y quede en ese estado por el tiempo especificado en **timepersist**.

Si no están especificados argumentos opcionales, después de 20 segundos desde el principio de la ejecución de la instrucción, es automáticamente generado el error de ciclo: "Entrada digital **nombrentrada** en espera de **estado**".

Si de los argumentos opcionales está presente solo **tiempodeespera**, la visualización sobredicha es generada al termino del mismo.

Si la condición se verifica después que el **tiempodeespera** ha terminado, es automáticamente cancelado el error de ciclo enviado en precedencia para esa tarea.

Cuando está presente **etiqueta** o **nombresubprograma** o **nombrefunción**, al termino del tiempo de espera el programa salta a etiqueta o llama **nombresubprograma** o **nombrefunción** sin generar alguna visualización automática.

Atención

Para evitar una situación de espera de una señal de entrada durante un ciclo de trabajo, se recomienda programar un tiempo de espera inferior al tiempo predeterminado (20 segundos).

10.3.4 Ejes

CHAIN

Sintaxis

```
CHAIN eje_maestro, eje_esclavo1 [, ...eje_esclavo5]
```

Argumentos

eje_maestro	nombre del dispositivo tipo eje que funcionará como maestro
eje_esclavo1	nombre del dispositivo tipo eje que funcionará como esclavo
...	
eje_esclavo5	
]	

Descripción

Después de la ejecución de esta instrucción, los **ejes_esclavos** (1÷5) ejecutarán los movimientos ligados a los del eje máster por la relación de encadenamiento programado con la instrucción **RATIO**. Serán encadenados los movimientos punto-punto y también los movimientos interpolados.

El **eje_esclavo1** no es un parámetro opcional más debe de ser siempre definido.

El eje esclavo, para poder ser concadenado, no debe de estar ocupado en una interpolación y no puede ser maestro de otros esclavos.

El eje maestro no puede ser esclavo de otros ejes.

El enlace puede ser efectuado tanto con los ejes en cota como con los ejes en movimiento.

Para deshabilitar el enlace ejes es suficiente ejecutar la instrucción **NORMAL** en el eje maestro. Esta última operación puede ser efectuada tanto con los ejes en cota como también con los ejes en movimiento.

Cuando la concatenación es deshabilitada durante un movimiento de los ejes, el eje esclavo efectúa una rampa de deceleración y se detiene.

Es posible definir un máximo de 8 ejes maestro al mismo tiempo.

Esta instrucción es ejecutable también con ejes paso-a-paso (stepper), siempre que dichos ejes sean manejados por medio de TRS-AX.

Además, todos los ejes tienen que tener un codificador real y no simulado. De lo contrario, se generará el error de sistema "4101 - Gestión incoherente del eje númeroeje".

Véase también [RATIO](#).

Ejemplo

```
; concadeno el eje Y al eje X
CHAIN      X, Y
; muevo el eje X.
; Y replica el movimiento de X
MOVINC    X, 100
```

CIRCABS

Sintaxis

CIRCABS [etiqueta],eje1, cota1, eje2, cota2, sentido, ±radio [, ángulo]

Argumentos

etiqueta	constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento
eje1, eje2	nombres de dispositivos tipo eje
cota1, cota2	constante o variable. Representa la cota de desplazamiento absoluto
sentido	variable integer. especifica el tipo de rotación, son admitidos los valores: CW en el sentido de las agujas del reloj CCW en el sentido contrario a las agujas del reloj
radio	constante o variable. Representa el valor del radio del círculo
ángulo	constante o variable. Representa el ángulo de inicio

Descripción

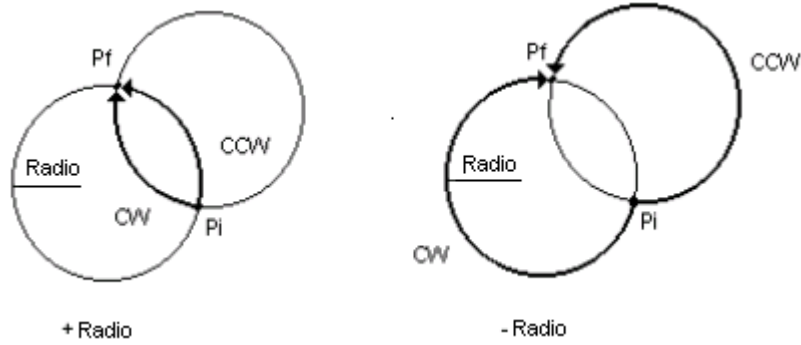
Interpolación circular en 2 ejes con desplazamiento absoluto basado en las cotas **cota1** y **cota2** programadas. El arco es determinado por el punto inicial (punto corriente), por el punto final, por el valor del **radio** y por el **sentido** de recorrido.

El signo aplicado al **radio** permite seleccionar el arco menor (+radio) o el arco mayor (-radio).

En el caso particular de cota inicial del eje 1 coincidente con la cota final **cota1** y la cota inicial del eje2 coincide con la cota final **cota2** se individua un círculo completo: en este caso es necesario indicar el argumento **ángulo**, con el mismo significado de la instrucción **CIRCLE** (a la cual hacer referencia).

El parámetro ángulo sirve para determinar de manera unívoca el centro del círculo, con el mismo significado de la instrucción **CIRCLE**. Es considerado solo cuando, antes de la ejecución de la instrucción, **cota1** y **cota2** corresponden a las cotas corrientes de los ejes. El parámetro opcional **etiqueta** es utilizado en asociación con la instrucción **SETLABELINTERP** para identificar de manera unívoca el bloqueo de desplazamiento.

Ejes paso-a-paso se pueden utilizar en esta instrucción si solo son gestionados por un remoto TRS-AX. En este caso hay que tomar en consideración que por el término interpolación se entiende el movimiento coordinado de más ejes, afectado por un error discreto debido al método de control del eje.



CIRCINC

Sintaxis

CIRCINC [etiqueta],eje1, cota1, eje2, cota2, sentido, ±radio [, ángulo]

Argumentos

etiqueta

constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento

eje1, eje2

nombres de dispositivos tipo eje

cota1, cota2

constante o variable. Representa la cota de desplazamiento incremental variable integer. especifica el tipo de rotación, son admitidos los valores:

sentido

CW en el sentido de las agujas del reloj

CCW en el sentido contrario a las agujas del reloj

radio

constante o variable. Representa el valor del radio del círculo

ángulo

constante o variable. Representa el ángulo de inicio

Descripción

Interpolación circular en 2 ejes con desplazamiento incremental basado en las cotas **cota1** y **cota2** programadas.

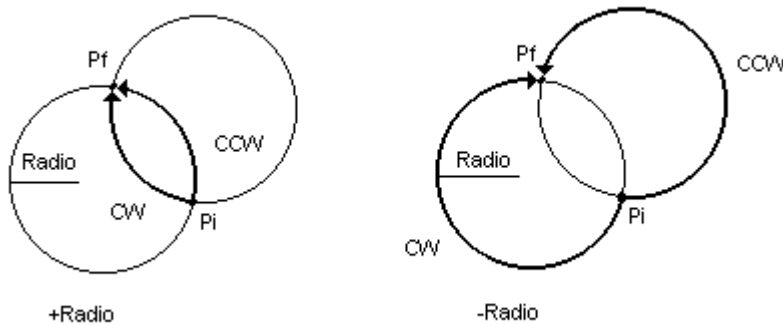
El arco es determinado por el punto inicial (punto corriente), por el punto final, por el valor del **radio** y por el **sentido** de recorrido.

El signo aplicado al **radio** permite seleccionar el arco menor (+radio) o el arco mayor (-radio).

En el caso particular de $cota1 = cota2 = 0$, se individua un círculo completo: en este caso es necesario indicar el argumento **ángulo**, con el mismo significado de la instrucción **CIRCLE** (a la cual hacer referencia).

El parámetro ángulo sirve para determinar de manera unívoca el centro del círculo, con el mismo significado de la instrucción **CIRCLE**. El parámetro opcional **etiqueta** es utilizado en asociación con la instrucción **SETLABELINTERP** para identificar de manera unívoca el bloqueo de desplazamiento.

Ejes paso-a-paso se pueden utilizar en esta instrucción si solo están gestionados por un remoto TRS-AX. En este caso hay que tomar en consideración que por el término interpolación se entiende el movimiento coordenado de más ejes, afectado por un error discreto debido al método de control del eje.



CIRCLE

Sintaxis

CIRCLE [etiqueta],eje1, eje2, sentido, radio, ángulo

Argumentos

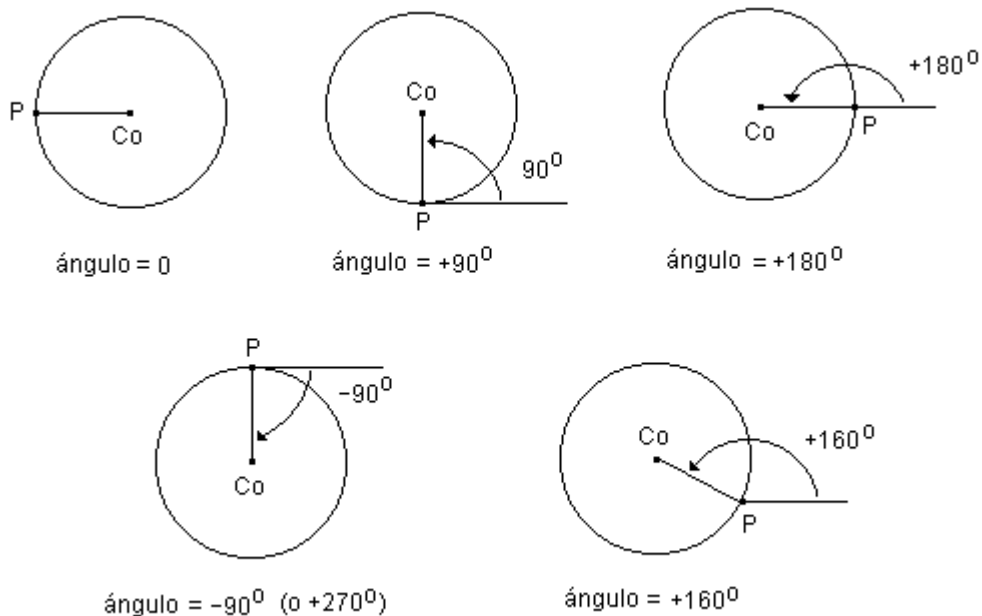
etiqueta

constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento

eje1, eje2 nombres de dispositivos tipo eje
sentido variable integer. especifica el tipo de rotación, son admitidos los valores:
CW en el sentido de las agujas del reloj
CCW en el sentido contrario a las agujas del reloj
radio constante o variable. Representa el valor del radio del círculo
ángulo constante o variable. Representa el ángulo de inicio

Descripción

Interpolación circular completa.
 Genera un círculo con **eje1** ed **eje2**, en el **sentido** indicado, con **radio** igual al valor asignado y en base al **ángulo** de inicio impuesto.
Radio puede asumir solo valores positivos.
 Hay que indicar el **ángulo** con convención trigonométrica positiva en sentido contrario a las agujas del reloj comenzando desde el eje X. De hecho queda determinada la posición del centro Co del círculo, especificando el ángulo formado por el radio pasante por el punto inicial P programado (punto corriente) y la dirección horizontal X+. El parámetro opcional **etiqueta** es utilizado en asociación con la instrucción [SETLABELINTERP](#) para identificar de manera unívoca el bloqueo de desplazamiento.
 Ejes paso-a-paso se pueden utilizar en esta instrucción si solo son gestionados por un remoto TRS-AX. En este caso hay que tomar en consideración que por el término interpolación se entiende el movimiento coordenado de más ejes, afectado por un error discreto debido al método de control del eje.



COORDIN

Sintaxis

COORDIN **matriz, valor deltaT, dirección, ini, fin, masc, eje1, n°_columna_eje1**
 [, (eje2, n°_columna_eje2) ÷ (eje32, n°_columna_eje32)]

Argumentos

matriz matriz de datos
valor deltaT constante o variable. Base tiempos
dirección constante predefinida. Dirección de lectura datos en matriz.
UP desde línea más baja a más alta
DOWN desde línea más alta a más baja.
ini variable global integer. Es el número de línea inicial
fin variable global integer. Es el número de línea final
masc máscara ejes por habilitar
eje1 [...eje32] nombres de dispositivos tipo eje
n°_columna_eje1[...n°_columna_eje32] número columna matriz referida al eje

Descripción

Esta instrucción permite efectuar movimientos sincronizados de los ejes **eje1**, **eje2**, etc., a través desplazamientos incrementales (microvectores) definidos por una **matriz** de datos. Los parámetros **eje1** y **n_columna_eje1** deben de ser siempre definidos. Los valores contenidos en la **matriz** indican las cotas absolutas a medida que son alcanzadas por los varios ejes.

Los relativos desplazamientos incrementales (diferencia cotas entre línea (n) y línea (n-1)) son ejecutados en un intervalo temporal igual a un **múltiplo** de la base tiempos (1 ms = tiempo real de refresco ejes), definido por el argumento **valor Δt**, que por lo tanto debe de ser expresado por un número entero. Prefijado dicho valor de tiempo, la entidad de los simples desplazamientos de cada eje determina la velocidad. Esta instrucción permite la movimentación coordinada de un máximo de 32 ejes a lo largo de cualquier recorrido curvilíneo en el espacio, como generados por técnicas SPLINE.

La instrucción es pasante y no es necesaria la instrucción STARTINTERP para partir. Sin embargo, hay que poner término a una instrucción WAITSTILL para esperar la correcta llegada en cota de los ejes. Hay que hacer eventuales modificaciones del feedrate mediante la instrucción SETFEEDI y gestionarlas mediante la instrucción SETFEEDCOORD.

El parámetro dirección permite determinar la **dirección** de recorrido de la matriz, permitiendo la ejecución de la trayectoria en los dos versos.

Las columnas de la matriz por escandir pueden ser de tipo float o de tipo doble más no ambas al mismo tiempo.

Además, del desplazamiento de los ejes a lo largo de una carrera completa (definida por el número de líneas de la matriz) es posible realizar un desplazamiento infinito utilizando:

- una matriz de una línea solamente. En esta modalidad de funcionamiento el control lee siempre la única línea de la matriz y aplica a los ejes las cotas que están escritas aquí. Para que los ejes se muevan será necesario modificar oportunamente la línea de la matriz, preferiblemente utilizando una tarea de tiempo real que garantice la sincronización de la actualización de las cotas con la frecuencia de refresco de los ejes. De esta manera se pueden realizar excéntricas electrónicas o enlaces con proporción diferente de 1:1. Para activar esta modalidad de funcionamiento habrá que programar **ini = 1**, **fin = 0** y **dirección = UP**. Cuando se utiliza esta modalidad de funcionamiento NO hay que utilizar la instrucción STOP.
- una matriz de más líneas. Es posible escandir la matriz con ciclos desde la primera hasta la última indefinitamente programando los valores **ini = 1**, **fin = 0** y **dirección = UP**. En el caso en que se desee ejecutar solamente una línea de una matriz multilíneas, es necesario programar los parámetros **ini**, **fin** y **dirección** de la siguiente manera: **ini** = número línea que se desea ejecutar, **fin** = número línea precedente a la que se desea ejecutar, **dirección** = UP. En otro caso se genera un error de sistema. Ejes paso-a-paso se pueden utilizar en esta instrucción si solo son gestionados por un remoto TRS-AX.

DISABLECORRECTION**Sintaxis**

DISABLECORRECTION **eje [, eje1, ..., eje6]**

Argumentos

eje nombre de dispositivo tipo eje
eje1, ..., eje6 nombre de dispositivo tipo eje

Descripción

Deshabilita la corrección lineal para el **eje** especificado.

El primer parámetro es el eje para el cual se desea desactivar la corrección, si es el único parámetro especificado serán desactivadas todas las correcciones presentes en configuración. Los siguientes parámetros permiten especificar cuales correcciones desactivar, si uno de estos coincide con el primer parámetro será desactivada la autocorrección.

Para una descripción más detallada, véase [ENABLECORRECTION](#).

Ejemplo

```
; deshabilita solo la autocorrección para el eje X
DISABLECORRECTION X, X

; deshabilita la corrección cruzada (hacia X y Y)
; para el eje Z, más no la autocorrección
DISABLECORRECTION Z, X, Y
```

EMERGENCYSTOP**Sintaxis**

EMERGENCYSTOP **eje, tiempo**

Argumentos

eje nombre de dispositivos tipo eje
tiempo constante o variable integer. Tiempo de rampa en ms

Descripción

Parada del eje especificado y, probablemente, con él, todos aquellos especificados en el movimiento interpolado. El movimiento se para con una rampa de deceleración en el tiempo indicado por la variable **[tiempo]**.

En los movimientos punto punto, si el tiempo configurado es mayor del tiempo de deceleración indicado en configuración, se utiliza este último.

En los movimientos interpolados, si el tiempo configurado es mayor del máximo de los tiempos de deceleración de todos los ejes incluidos, se utiliza el tiempo máximo programado en configuración.

El movimiento se podrá reanudar por una instrucción [START](#).

La instrucción no se puede utilizar si **[eje]** es un eje esclavo.

La instrucción puede generar los siguientes errores de sistema:

- "4101 – Gestión incoherente del eje" cuando **[eje]** está ejecutando un movimiento sincronizado o una interpolación multilínea o un movimiento ISO.
- "4105 – Instrucción no ejecutable en el eje" cuando **[eje]** es un eje de cuenta.
- "4399 – Parámetro fuera del intervalo" si el **[tiempo]** indicado es igual o menor de 0.

ENABLECORRECTION**Sintaxis**

ENABLECORRECTION **eje [, eje1, ..., eje6]**

Argumentos

eje nombre de dispositivo tipo eje
eje1, ..., eje6 nombre de dispositivo tipo eje

Descripción

Habilita la corrección lineal para el **eje** especificado. La corrección está compuesta por el autocorrección y por la corrección cruzada. El autocorrección es una corrección de la cota real de un eje en función de su propia posición, la corrección cruzada es una corrección de la cota real de un eje en función de la posición de otros ejes. Es posible definir un máximo de cinco correctores cruzados.

El primer parámetro es el eje para el cual se quiere activar la corrección, si es el único parámetro especificado serán activadas todas las correcciones presentes en configuración.

Los siguientes parámetros permiten especificar cuales correcciones activar, si uno de estos coincide con el primer parámetro será activada el autocorrección.

Véase también [DISABLECORRECTION](#).

ATENCIÓN: Para que la instrucción tenga efecto la corrección debe de ser habilitada también en configuración.

Ejemplo

```
;habilita todas las correcciones previstas en configuración
;para el eje X
ENABLECORRECTION      X

;habilita solo el autocorrección para el eje X
ENABLECORRECTION      X, X

;habilita el autocorrección y la corrección cruzada (hacia X e Y)
;para el eje Z
ENABLECORRECTION      Z, X, Y, Z
```

ENDMOV**Sintaxis**

ENDMOV **eje [, cota]**

Argumentos

eje nombre de dispositivos tipo eje
cota constante o variable

Descripción

Para el movimiento del eje especificado. Es diferente de una instrucción [STOP](#), puesto que el movimiento está terminado y ya no podrá ser reanudado por una eventual instrucción [START](#). Si especificado el parámetro **cota** permite programar la posición en la que el eje terminará el movimiento, sino el punto donde el eje se detiene depende de la velocidad corriente y de la última desaceleración programada. Si necesario, para alcanzar el punto de fin movimiento, el control efectúa una inversión de movimiento del eje.

Atención

El parámetro cota es utilizado solo si se trata de un movimiento punto a punto. En caso de movimiento interpolado el movimiento del eje se detiene sin tener en cuenta el valor de la cota.

Ejemplo

```
; detiene el movimiento corriente y lleva el eje a cota 0.0
ENDMOV X, 0.0
```

FASTREAD**Sintaxis**

FASTREAD eje1, estado, variable1 [,eje2, variable2],[..., eje8, variable8]

Argumentos

eje1...[...eje8] nombres de dispositivos tipo eje. eje1 es el eje maestro.
eje1...[...eje8] constante predefinida. Asume los siguientes valores:
ON frente de subida
OFF frente de bajada
variable1...[...variable8] variable o elemento de matriz/vector de tipo double. Cota memorizada.

Descripción

Las cotas de los ejes indicados son leídas y memorizadas en las variables en el instante en que la entrada veloz del eje1 (eje Maestro) conmuta en el estado configurado.

Si los ejes indicados son analógicos, deben pertenecer a la misma tarjeta (4 para TRS-AX).
 Si los ejes indicados son digitales una señal de entrada rápida está directamente en el accionamiento; por lo tanto en caso de FASTREAD multiple hay que conectar la señal en paralelo en diferentes dispositivos. Con bus EtherCAT, para cada eje que experimenta un cambio de codificador (véase la instrucción [SWITCHENC](#)), el número máximo de ejes permitido disminuye de uno. Cuando este límite es pasado, se genera el error de sistema "4400 Demasiados ejes activos en FASTREAD". Además, el codificador agregado tiene que ser conectado con una expansión TRS-AC-E sobre Trs-CAT. De lo contrario, se generará el error de sistema "4375 FASTREAD ejecutada sobre ejes de tarjetas diferentes".
 La instrucción termina cuando la entrada conmuta en el estado (ON/OFF) indicado.
 Si es ejecutada una instrucción STOP antes de la conmutación de la entrada veloz estas instrucciones quedan activas y vuelven a comenzar después de la instrucción START.
 Es posible activar más lecturas rápidas contemporáneamente en la misma tarjeta ejes.

Durante la ejecución de la instrucción no se puede ejecutar simultáneamente instrucciones [SETPFLY](#) y [FASTREAD](#) en el mismo eje, si eso está conectado a tarjetas con bus MECHATROLINK-II.

Atención

La entrada rápida para los ejes de tipo digital sobre tarjeta con bus MECHATROLINK-II está presente en la entrada EXTI2 y no es necesario configurarlo en Virtual-Físico. Las entradas rápidas de los ejes digitales MECHATROLINK-II deben de ser cortocircuitadas para que la memorización de la coordenada de un eje sea hecha solamente con referencia a la propia entrada rápida.

FREE**Sintaxis**

FREE eje [, tensión]

Argumentos

eje nombre de dispositivo tipo eje
tensión constante float o variable float. Tensión de referencia

Descripción

Pone el **eje** en la condición de "anillo abierto" (Libre), deshabilitando su control de posición. Si el **eje** es Esclavo en una concatenación con otros ejes, el vínculo se interrumpe y el movimiento del **eje** se para. Si es especificado el parámetro **tensión**, la tensión de referencia del eje es puesta igual al mismo. Esta instrucción puede ser usada en el caso de ejes de medida para detección cotas, o bien en ejes cuyo movimiento puede ser forzado por órganos mecánicos externos que pudieran alterar su posición. Durante el funcionamiento la cota del eje es detectada y actualizada, es posible por lo tanto posicionar el eje de manera absoluto después de la rehabilitación del control de posición (instrucción [NORMAL](#)).

HELICABS**Sintaxis**

HELICABS [etiqueta]eje1, cota1, eje2, cota2, eje3, cota3, sentido, ±radio
 [,ángulo [, númgiros [, eje4, cota4 [, ..., eje6, cota6]]]]

Argumentos

etiqueta	constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento
eje1...eje3[...eje6]	nombres de dispositivos tipo eje
cota1...cota3[...cota6]	constante o variable. Cota de desplazamiento absoluto
sentido	variable integer. Tipo de rotación en el sentido de las agujas del reloj/en el sentido contrario a las agujas del reloj (CW/CCW)
radio	constante o variable. radio de la hélice
ángulo	constante o variable. Ángulo de inicio
númgiros	constante o variable. Número de giros

Descripción

Interpolación helicoidal con desplazamiento absoluto par a las cotas **cota1**, **cota2** y **cota3** programadas. El movimiento está constituido por una interpolación circular asociada a los ejes **eje1** y **eje2** (con las mismas reglas sintácticas de [CIRCABS](#) /[CIRCINC](#), relativamente a los argumentos **sentido**, \pm **radio** y **ángulo**), y por una lineal asociada al **eje3** (eventualmente **eje4**, **eje5** y **eje6**). El movimiento helicoidal puede ser desarrollado en más giros indicados por el argumento **númgiros**. La cota relativa al eje con movimiento lineal (come las eventuales cotas **eje4**, **eje5**, **eje6**) se refiere al desplazamiento total (por lo tanto no al desplazamiento/giro). El parámetro opcional **etiqueta** es utilizado en asociación con la instrucción [SETLABELINTERP](#) para identificar de manera unívoca el bloqueo de desplazamiento. Ejes paso-a-paso se pueden utilizar en esta instrucción si solo están gestionados por un módulo remoto TRS-AX. En este caso hay que tomar en consideración que por el término interpolación se entiende el movimiento coordinado de más ejes, afectado por un error discreto debido al método de control del eje.

Atención

1. La condición de contorneado es evaluada solamente sobre los primeros tres ejes, los que forman el sistema de referencia. Agregando un eje ulterior y eventualmente modificandolo se obtiene una incorrecta gestión del perfil de velocidad. Para obtener un movimiento correcto, es necesario interponer una instrucción [WAITSTILL](#) entre una instrucción [HELICABS](#) y la otra.
2. si se programa un sistema de referencia local utilizando la instrucción [SETRIFLOC](#), los tres ejes que definen el nuevo sistema de referencia tienen que estar siempre indicados entre los parámetros de la instrucción [HELICABS](#) aunque si no ejecutan desplazamientos

HELICINC**Sintaxis**

HELICINC **[etiqueta], eje1, cota1, eje2, cota2, eje3, cota3, sentido, \pm radio [, ángulo [, númgiros [, eje4, cota4 [, ..., eje6, cota6]]]]]**

Argumentos

etiqueta	constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento
eje1...eje3[...eje6]	nombres de dispositivos tipo eje
cota1...cota3[...cota6]	constante o variable. Cota de desplazamiento incremental
sentido	variable integer. Tipo de rotación en el sentido de las agujas del reloj/en el sentido contrario a las agujas del reloj (CW/CCW)
radio	constante o variable. radio del hélice
ángulo	constante o variable. Ángulo de inicio
númgiros	constante o variable. Número de giros

Descripción

Interpolación helicoidal con desplazamiento incremental par a las cotas **cota1**, **cota2** y **cota3** programadas. El movimiento está constituido por una interpolación circular asociada a los ejes **eje1** y **eje2** (con las mismas reglas sintácticas de [CIRCABS](#) /[CIRCINC](#), relativamente a los argumentos **sentido**, \pm **radio** y **ángulo**), y por una lineal asociada al **eje3** (eventualmente **eje4**, **eje5** y **eje6**). El movimiento helicoidal puede ser desarrollado en más giros indicados por el argumento **númgiros**. La cota relativa al eje con movimiento lineal (come las eventuales cotas **eje4**, **eje5**, **eje6**) se refiere al desplazamiento total (por lo tanto no al desplazamiento/giro). El parámetro opcional **etiqueta** es utilizado en asociación con la instrucción [SETLABELINTERP](#) para identificar de manera unívoca el bloqueo de desplazamiento. Ejes paso-a-paso se pueden utilizar en esta instrucción si solo están gestionados por un módulo remoto TRS-AX. En este caso hay que tomar en consideración que por el término interpolación se entiende el movimiento coordinado de más ejes, afectado por un error discreto debido al método de control del eje.

Atención

1. La condición de contorneado es evaluada solamente sobre los primeros tres ejes, los que forman el sistema de referencia. Agregando un eje ulterior y eventualmente modificandolo se obtiene una incorrecta gestión del perfil de velocidad. Para obtener un movimiento correcto, es necesario interponer una instrucción [WAITSTILL](#) entre una instrucción [HELICINC](#) y la otra.

2. si se programa un sistema de referencia local utilizando la instrucción [SETRIFLOC](#), los tres ejes que definen el nuevo sistema de referencia tienen que estar siempre indicados entre los parámetros de la instrucción HELICINC aunque si no ejecutan desplazamientos.

JERKCONTROL

Sintaxis

JERKCONTROL **eje, estado**

Argumentos

eje nombres de dispositivos tipo eje
estado constante predefinida. Los valores admitidos son:
ON estado indicador activo
OFF estado indicador desactivo

Descripción

Habilita, si al parámetro estado es asignado valor ON, o inhabilita, si al parámetro estado está asignado valor OFF, el control del jerk en los movimientos en interpolación y punto-punto del eje. El control del jerk solo está habilitado con los ejes que han configurado una rampa de aceleración y deceleración en Ese o en doble Ese. Si el eje ha configurado una rampa Lineal el control del jerk no está efectuado.

JERKSMOOTH

Sintaxis

JERKSMOOTH **eje, valor**

Argumentos

eje nombres de dispositivos tipo eje
valor constante o variable de tipo float.

Descripción

Durante los movimientos interpolados clásicos, los ejes pueden moverse en condiciones de contorneado, o sea sin pegarse entre dos bloques seguidos de desplazamiento, cuando el ángulo entre las tangentes a la trayectoria está inferior al parámetro "Ángulo máximo de contorneado" (cuyo valor predeterminado es 15 grados y puede ser modificado mediante la instrucción [SETCONTORNATURE](#)).

En caso contrario los ejes son detenidos en el punto de esquina de los dos bloques con deceleración controlada y se vuelven a funcionar a lo largo del nuevo bloqueo con aceleraciones y velocidad controladas. Sin embargo, parada y puesta en marcha hacen el movimiento de la máquina poco eficiente. En los casos en que el ángulo de contorneado tome valores considerables, como, por ejemplo, un valor de discontinuidad de tangencia superior a 5 grados, se determinan saltos de velocidad no irrelevantes para los ejes en el movimiento de contorneado, con consiguientes valores infinitos de aceleración, jerk y notables sollicitaciones mecánicas, que pueden repercutir también en la calidad de la mecanización. La instrucción JERKSMOOTH permite en frente de un valor establecido por el usuario, de enlazar de manera continua, o sea con continuidad de aceleración y de velocidad, los perfiles de velocidad de los ejes durante los movimientos de contorneado. Cabe señalar que este enlace continuo introduce pequeñas variaciones en la trayectoria siguiente, con respecto de la trayectoria teórica, puesto que alrededor del punto de contorneado los ejes muestran un perfil de velocidad diferente del teórico.

La variable **valor**, que es expresa por medio de un valor porcentaje entre 0 y 100, define cuanto se desea enlazar de manera continua los perfiles de velocidad. Un valor igual a 0 mantiene el perfil teórico, creando discontinuidades en los aceleraciones y en los perfiles de velocidad. Un valor igual a 100 obtiene perfiles enlazados suavemente, un rendimiento mejor, sino también una máxima desviación de la trayectoria teórica, proporcional a la velocidad a lo largo de la trayectoria.

Atención

Se puede usar esta instrucción solo en los movimientos con interpolación clásica (instrucciones [LINEARABS](#), [LINEARINC](#), [CIRCABS](#), [CIRCINC](#), [HELICABS](#), [HELICINC](#)). No es aplicada en los movimientos de interpolación multieje (instrucciones [MULTIABS](#) y [MULTIINC](#)).

LINEARABS

Sintaxis

LINEARABS **[etiqueta], eje1, cota1, [eje2, cota2 [, eje3, cota3 [, ..., eje6, cota6]]]**

Argumentos

etiqueta constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento
eje1[...eje2[...eje6]] nombres de dispositivos tipo eje
cota1[...cota2[...cota6]] constante o variable. Cota de desplazamiento absoluto

Descripción

Interpolación lineal, con desplazamiento absoluto, a las cotas especificadas por **cota1**, **cota2**, etc. El parámetro opcional **etiqueta** es utilizado en asociación con la instrucción [SETLABELINTERP](#) para identificar de manera unívoca el bloqueo de desplazamiento. Ejes paso-a-paso se pueden utilizar en esta instrucción si solo son gestionados por un remoto TRS-AX. En este caso hay que tomar en consideración que por el término interpolación se entiende el movimiento coordinado de más ejes, afectado por un error discreto debido al método de control del eje.

Atención

1. La condición de contorneado es evaluada solamente sobre los primeros tres ejes, los que forman el sistema de referencia. Agregando un eje ulterior y eventualmente modificandolo se obtiene una incorrecta gestión del perfil de velocidad. Para obtener un movimiento correcto, es necesario interponer una instrucción [WAITSTILL](#) entre una instrucción LINEARABS y la otra.
2. si se programa un sistema de referencia local utilizando la instrucción [SETRIFLOC](#), los tres ejes que definen el nuevo sistema de referencia tienen que estar siempre indicados entre los tres primeros parámetros de la instrucción LINEARABS aunque si no ejecutan desplazamientos.

LINEARINC

Sintaxis

LINEARINC **[etiqueta], eje1, cota1, [eje2, cota2 [, eje3, cota3 [, ..., eje6, cota6]]]**

Argumentos

etiqueta	constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento
eje1[...eje2[...eje6]]	nombres de dispositivos tipo eje
cota1[...cota2[...cota6]]	constante o variable. Cota de desplazamiento incremental

Descripción

Interpolación lineal, con desplazamiento incremental, par a las cotas especificadas por **cota1**, **cota2**, etc. El parámetro opcional **etiqueta** se utiliza en asociación con la instrucción [SETLABELINTERP](#) para identificar de manera unívoca el bloqueo de desplazamiento. Ejes paso-a-paso se pueden utilizar en esta instrucción si solo son gestionados por un remoto TRS-AX. En este caso hay que tomar en consideración que por el término interpolación se entiende el movimiento coordinado de más ejes, afectado por un error discreto debido al método de control del eje.

Atención

1. La condición de contorneado es evaluada solamente sobre los primeros tres ejes, los que forman el sistema de referencia. Agregando un eje ulterior y eventualmente modificandolo se obtiene una incorrecta gestión del perfil de velocidad. Para obtener un movimiento correcto, es necesario interponer una instrucción [WAITSTILL](#) entre una instrucción LINEARINC y la otra.
2. si se programa un sistema de referencia local utilizando la instrucción [SETRIFLOC](#), los tres ejes que definen el nuevo sistema de referencia tienen que estar siempre indicados entre los parámetros de la instrucción LINEARINC aunque si no ejecutan desplazamientos.

MOVABS

Sintaxis

MOVABS **eje1, valor1 [, eje2, valor2 [, ..., eje6, valor6]]**

Argumentos

eje1...[...eje6]	nombres de dispositivos tipo eje
valor1...[...valor6]	constante o variable. Valor de desplazamiento absoluto

Descripción

Hace ejecutar a los ejes especificados un *movimiento absoluto* par a las cotas especificadas por **valor1** [... **valor6**].

Para ejecutar el movimiento el eje debe de ser ocupado en un movimiento interpolado y debe de estar en cota o en ventana. El movimiento del eje comienza apenas se ejecuta la instrucción. Si son ejecutadas más instrucciones de desplazamiento punto-punto en una misma tarea, estas son concadenadas. Si una segunda tarea trata de ejecutar unas instrucciones punto-punto sobre un eje que ya está ocupado en un movimiento, esta tarea permanecerá en espera que el movimiento comandado por la primera tarea termine.

Además, es posible modificar la velocidad entre una movimiento punto-punto y el siguiente con una instrucción [SETVEL](#). Los dos movimientos serán enlazados con una rampa de velocidad sin parar los ejes. Se no es utilizada la instrucción [SETVEL](#), la velocidad máxima posible es representada por el valor de velocidad manual definido en configuración.

Un movimiento punto-punto puede ser interrumpido con la instrucción [STOP](#) y luego reanudado con la instrucción [START](#). Durante la interrupción del movimiento el eje queda en estado "en régimen" aunque si físicamente no se mueve.

Un movimiento puede ser abortado con la instrucción [ENDMOV](#). En este caso ya no puede ser reanudado.

Atención

- 1) Antes los movimientos punto-punto:
 - no permitían la variación de la velocidad si no con el eje detenido. El comportamiento actual es similar al de los movimientos interpolados;
 - cuando eran interrumpidos por una STOP el eje correspondiente asumía el estado "en cota".
- 2) Les sugerimos que utilicen instrucciones de interpolación lineal en lugar de instrucciones de movimiento punto a punto, cuando el número de los bloques de desplazamiento supera 32 y los bloques constan de micro-tramos. Para más información detallada véase el documento "Limiti Firmware Movimiento Punto Punto.doc" que TPA pone a disposición.

Ejemplo 1

[Rutina de Puesta a cero en Interrupt](#)

Ejemplo 2

```
; cambio de velocidad
Function cambiovel
  setvel X, 20
  setvel X, 20
  movinc X, 100, Y, 200
  movinc X, 150, Y, 180
  setvel X, 5
  movinc X, 80, Y, 100
  waitstill X, Y
fret
```

MOVINC**Sintaxis**

MOVINC **eje1, valor1 [, eje2, valor2 [, ..., eje6, valor6]]**

Argumentos

eje1...[...eje6] nombres de dispositivos tipo eje
valor1...[...valor6] constante o variable. Valor de desplazamiento incremental

Descripción

Hace ejecutar a cada eje un *movimiento incremental* para al correspondiente **valor**. Para ejecutar el movimiento el eje debe de ser ocupado en un movimiento interpolado y debe de estar en cota o en ventana. El movimiento del eje comienza apenas es ejecutada la instrucción. Si son ejecutadas más instrucciones de desplazamiento punto-punto en la misma tarea, estas son concadenadas. Si una segunda tarea trata de ejecutar unas instrucciones punto-punto en un eje que ya está en movimiento, esta tarea quedará en espera que el movimiento comandado por las primeras tareas termine. Además, es posible modificar la velocidad entre un movimiento punto-punto y el siguiente con una instrucción [SETVEL](#). Los dos movimientos serán enlazados con una rampa de velocidad sin parar los ejes. Se no es utilizada la instrucción [SETVEL](#), la velocidad máxima posible es representada por el valor de velocidad manual definido en configuración. Un movimiento punto-punto puede ser interrumpido con la instrucción [STOP](#) y en seguida reanudado con la instrucción [START](#). Durante la interrupción del movimiento el eje queda en estado "régimen" aunque si físicamente no se mueve. Un movimiento puede ser abortado con la instrucción [ENDMOV](#). En este caso ya no puede ser reanudado.

Atención

- 1) Antes los movimientos punto-punto:
 - no permitían la variación de la velocidad si no con el eje detenido. El comportamiento actual es similar al de los movimientos interpolados;
 - cuando eran interrumpidos por una STOP el eje correspondiente asumía el estado "en cota".
- 2) Les sugerimos que utilicen instrucciones de interpolación lineal en lugar de instrucciones de movimiento punto a punto, cuando el número de los bloques de desplazamiento supera 32 y los bloques constan de micro-tramos. Para más información detallada véase el documento "Limiti Firmware Movimiento Punto Punto.doc" que T.P.A pone a disposición.

Ejemplo 1

[Rutina de Puesta a cero de un eje](#)

Ejemplo 2

```
; cambio de velocidad
Function cambiovel
  setvel X, 20
  setvel X, 20
```

```

movinc    X, 100, Y, 200
movinc    X, 150, Y, 180
setvel    X, 5
movinc    X, 80, Y, 100
waitstill X, Y
fret

```

MULTIABS

Sintaxis

MULTIABS **[etiqueta], eje1, valor1, [eje2, valor2 [, eje3, valor3 [, ..., eje16, valor 16]]]**

Argumentos

etiqueta constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento

eje1...[...eje16] nombres de dispositivos tipo eje

valor1...[...valor16] constante o variable. Valor de la cota teórica de fin bloqueo de desplazamiento de manera absoluta

Descripción

Interpolación multi-lineal de manera absoluta hasta 16 ejes. Este movimiento de interpolación permite el anticipación de los perfiles de velocidad programando sobre los ejes sus tolerancias correspondientes, a través de la instrucción [SETTOLERANCE](#) (por tolerancia de un eje se entiende la porción de recorrido en la cual es aceptada la no interpolación lineal con los demás ejes). El orden de inserción de los ejes en la instrucción MULTIAXES **tiene que** ser siempre el mismo y **todos** los ejes involucrados en el movimiento tienen que ser presentes. Los bloques de desplazamiento son puestos en cola en el normal lookahead y el movimiento comienza con la ejecución de una instrucción [WAITSTILL](#), [STARTINTERP](#) o al llenarse del lookahead mismo. Entre los ejes involucrados en el movimiento uno puede ser usado como collider a través de la instrucción [WAITCOLL](#). El parámetro opcional **etiqueta** es utilizado en asociación con la instrucción [SETLABELINTERP](#) para identificar de manera unívoca el bloqueo de desplazamiento. Ejes paso-a-paso se pueden utilizar en esta instrucción si solo son gestionados por un remoto TRS-AX. En este caso hay que tomar en consideración que por el término interpolación se entiende el movimiento coordenado de más ejes, afectado por un error discreto debido al método de control del eje.

Atención

Con este tipo de interpolación no es posible utilizar sistemas de referencia virtuales (instrucción [SETRIFLOC](#) y [RESRIFLOC](#)). Es posible ejecutar movimientos con ejes concadenados (en [CHAIN](#)). Los ejes involucrados en el movimiento interpolado multieje deben de ser declarados máster de otros ejes no involucrados en el movimiento. Es posible, además, aplicar el FeedRate Override. No se puede utilizar el array [SETINDEXINTERP](#).

Ejemplo

```

setquote  x, 0
setquote  y, 0
setquote  z, 0

; primer bloque
setveli   x, velx1
setveli   y, vely1
setveli   z, velz1
multiabs  x, cotax1,    y, cotay1,    z, cotaz1

; segundo bloque
settolerance x, tollx2,    y, tollly2,    z, tollz2
setveli     x, velx2
setveli     y, vely2
setveli     z, velz2
multiabs    x, cotax2,    y, cotay2,    z, cotaz2

; tercer bloque
settolerance x, tollx3,    y, tollly3,    z, tollz3
setveli     x, velx3
setveli     y, vely3
setveli     z, velz3
multiabs    x, cotax3,    y, cotay3,    z, cotaz3

; cuarto bloque
settolerance x, tollx4,    y, tollly4,    z, tollz4

```

```

setveli      x, ve|x4
setveli      y, ve|y4
setveli      z, ve|z4
multiabs     x,cotax4,    y,cotay4,    z,cotaz4
waitstill    x, y, z

```

MULTIINC

Sintaxis

MULTIINC [etiqueta], eje, valor1, [aeje2, valor2 [, eje3, valor3 [, ..., eje16, valor16]]]

Argumentos

etiqueta constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento

eje1...eje16 nombres de dispositivos tipo eje

valor1...[...valor16] constante o variable. Valor de incremento de la coordenada teórica de fin bloqueo de desplazamiento

Descripción

Interpolación multi-lineal incremental hasta 16 ejes. Este movimiento de interpolación permite el anticipación de los perfiles de velocidad programando de manera oportuna, a través de la instrucción [SETTOLERANCE](#), sobre los ejes las respectivas tolerancias (por tolerancia de un eje se entiende la porción de recorrido en la cual es aceptada la no interpolación lineal con los demás ejes). El orden de inserción de los ejes en la instrucción MULTIINC **tiene que** ser siempre el mismo y **todos** los ejes involucrados en el movimiento tienen que ser presentes. Los bloques de desplazamiento son puestos en cola en el normal lookahead y el movimiento comienza con la ejecución de una instrucción [WAITSTILL](#), [STARTINTERP](#) o al llenarse del lookahead mismo. Entre los ejes involucrados en el movimiento uno puede ser usado como collider a través de la instrucción [WAITCOLL](#). El parámetro opcional **etiqueta** es utilizado en asociación con la instrucción [SETLABELINTERP](#) para identificar de manera unívoca el bloqueo de desplazamiento. Ejes paso-a-paso se pueden utilizar en esta instrucción si solo son gestionados por un remoto TRS-AX. En este caso hay que tomar en consideración que por el término interpolación se entiende el movimiento coordenado de más ejes, afectado por un error discreto debido al método de control del eje.

Atención

Con este tipo de interpolación no es posible utilizar sistemas de referencia virtuales (instrucción [SETRIFLOC](#) y [RESRIFLOC](#)). Es posible ejecutar movimientos con ejes concatenados (en [CHAIN](#)). Los ejes involucrados en el movimiento interpolado multieje deben de ser declarados máster de otros ejes no involucrados en el movimiento. Es posible, además, aplicar el FeedRate Override. No se puede utilizar el array [SETINDEXINTERP](#).

NORMAL

Sintaxis

NORMAL eje

Argumentos

eje nombre de dispositivo tipo eje

Descripción

Habilita el control de posición sobre el **eje** y deshabilita la concatenación ejes. Cuando se inicia del sistema todos los ejes configurados son puestos en el estado free y pasan al estado normal cuando es ejecutada esta instrucción o bien cuando es efectuado el primer movimiento. De todas formas se recomienda ejecutar esta instrucción antes de efectuar el proceso de puesta a cero de los ejes, para restablecer eventuales condiciones de emergencia.

RESRIFLOC

Sintaxis

RESRIFLOC eje1, eje2, eje3

Argumentos

eje1...eje3 nombres de dispositivos tipo eje

Descripción

Restablece el sistema de referencia absoluto para los ejes X Y Z (**eje1, eje2, eje3**). Normalmente se utiliza esta instrucción después de la configuración de un sistema de referencia rototranslado con una instrucción [SETRIFLOC](#).

SETINDEXINTERP

Sintaxis

SETINDEXINTERP **eje, nombrevar**

Argumentos

eje nombre de dispositivo tipo eje
nombrevar nombre variable global de tipo integer

Descripción

Define un índice che contabiliza el número de bloques de interpolación ejecutados por un eje. Durante un movimiento en interpolación, en cada cambio bloque, la variable **nombrevar** es incrementada de 1.

Atención

La variable utilizada como índice debe de ser una variable global de grupo o global de máquina. Se puede utilizar la instrucción solo con interpolación clásica (instrucciones [LINEARABS](#), [LINEARINC](#), [CIRCABS](#), [CIRCINC](#), [HELICABS](#), [HELICINC](#)). No se puede utilizarla en los movimientos de interpolación multiaxial (instrucciones [MULTIABS](#) y [MULTIINC](#)).

SETLABELINTERP

Sintaxis

SETLABELINTERP **eje, valor**

Argumentos

eje nombre de dispositivo tipo eje
valor variable global de tipo entero

Descripción

En la variable **valor**, durante un movimiento en interpolación, en cada cambio bloque, se asigna un valor de etiqueta del nuevo bloque. La etiqueta es definida en las instrucciones de movimiento interpolado.

Atención

La variable **valor** tiene que ser una variable global de grupo o global de módulo.

SETPFLY

Sintaxis

SETPFLY **eje, estado, velocidad, cota,[error]**

Argumentos

eje nombre de dispositivo tipo eje
estado constante predefinida. Indica el estado del micro por verificar. Los valores configurables son:
ON
OFF
velocidad constante o variable float
cota constante o variable
error variable integer. Código de error

Descripción

Permite la puesta a cero de la cota eje "al vuelo". La puesta a cero es pilotada por un interruptor conectado a la entrada veloz del conector del eje (en tarjetas con bus MECHATROLINK-II se hace referencia a EXTI1).

Durante el movimiento del **eje**, ejecutado por medio de una instrucción MOVABS, la instrucción espera que el micro de puesta a cero relativo conmute en el **estado** indicado. Interceptada esta transición, la cota real del eje es puesta a cero sin aparar el movimiento y son automáticamente redefinidas dinámicamente la **cota** teórica de llegada y la **velocidad**. Por lo tanto, se reconstruye el movimiento correcto para alcanzar la cota de llegada y, si necesario, se ejecuta también una inversión de movimiento. Si se alcanza la cota configurada sin que sea detectada la conmutación de la entrada y no ha sido configurado el parámetro **error** es generado un error de sistema. Si ha sido configurado un parámetro **error** este tendrá que contener el código numérico del correspondiente error de sistema.

En este caso la puesta a cero no ha sido ejecutada y es necesario ejecutar la instrucción [SETQUOTE](#) para configurar de nuevo la búsqueda del micro.

Para interrumpir la ejecución de la puesta a cero "al vuelo" es suficiente ejecutar una NORMAL en el eje y bien terminar la tarea que ha pedido la ejecución de la puesta a cero.

Durante la ejecución de la instrucción no se puede ejecutar simultáneamente instrucciones [SETPFLY](#) y [FASTREAD](#) en el mismo eje, si eso está conectado a tarjetas con bus MECHATROLINK-II.

Ejemplo

[Rutina de Puesta a cero en Interrupt](#)

SETPFLYCHAINSTRAT**Sintaxis**

SETPFLYCHAINSTRAT **eje, tipo**

Argumentos

eje nombre de dispositivo tipo eje
tipo constante tipo integer. Los valores admitidos son:
 0 = solo el eje maestro pone la cota, el eje esclavo mantiene la cota precedente
 diferente de 0 = ambos el eje maestro y el eje esclavo ponen a cero la cota de manera síncrona.

Descripción

Esta instrucción permite programar manera con la cual el eje esclavo indicado se comportará en respuesta a una instrucción [SETPFLY](#) en el eje maestro.
 La instrucción tiene que ser ejecutada solamente en el eje esclavo. Si la variable tipo es omisa, es asumido un valor predeterminado igual a 0.

SETPZERO**Sintaxis**

SETPZERO **eje, cota [,error]**

Argumentos

eje nombre de dispositivo tipo eje
cota constante o variable. Es una cota incremental
error variable integer. Código de error

Descripción

Hace partir un movimiento incremental del **eje** a la **cota** especificada y espera que sea detectada la muesca de cero del encoder (antes de la llegada de la cota especificada).
 En el instante en que es detectada la muesca la cota real es puesta a cero y el eje es parado.
 Si se alcanza la cota configurada sin haber detectado la muesca de Cero y no ha sido configurado el parámetro **error** es generado un error de sistema. Si ha sido configurado un parámetro **error** este contendrá el código numérico del correspondiente error de sistema. En ese caso el set point no ha sido ejecutado y es necesario ejecutar la instrucción [SETQUOTE](#) para configurar nuevamente la búsqueda de la muesca.
 El movimiento de los ejes, generado por esta instrucción, puede ser interrumpido por una STOP y reactivado por una START.
 Si se ejecuta la instrucción con ejes S-CAN y con ejes EtherCAT es necesario ejecutar primero una instrucción FREE.

Durante la ejecución de la instrucción no se puede ejecutar simultáneamente instrucciones [SETPFLY](#) y [FASTREAD](#) en el mismo eje, si eso está conectado a tarjetas con bus MECHATROLINK-II.

Ejemplo

```
FREE                    x
SETPZERO              x, 100
```

SETPZEROCHAINSTRAT**Sintaxis**

SETPZEROCHAINSTRAT **eje, tipo**

Argumentos

eje nombre de dispositivo tipo eje
tipo constante tipo integer. Los valores admitidos son:
 0 = solo el eje maestro pone la cota, el eje esclavo mantiene la cota precedente
 diferente de 0 = ambos el eje maestro y el eje esclavo ponen a cero la cota de manera síncrona.

Descripción

Esta instrucción permite programar la manera con la cual el eje esclavo indicado se comportará en respuesta a una instrucción [SETPZERO](#) en el eje maestro. La instrucción tiene que ser ejecutada solamente en el eje esclavo.
 Si la variable **valor** está omisa, se asume un valor predeterminado igual a 0.

SETQUOTE

Sintaxis

SETQUOTE **eje, cota**

Argumentos

eje nombre del dispositivo tipo eje
cota constante o variable

Descripción

Esta instrucción fuerza, contestualmente, la cota teórica y la cota real de un eje al valor especificado por **cota**. Si el eje está en movimiento esto provoca una brusca parada del eje ya que, para el control, el eje se encuentra de repente en cota (cota real coincide con cota target). Está por lo tanto desaconsejado el uso de esta instrucción en un eje en movimiento, si no a velocidad muy baja.

Ejemplo

[Rutina de Puesta a cero de un eje](#)

SETQUOTECHAINSTRAT

Sintaxis

SETQUOTECHAINSTRAT **eje, tipo**

Argumentos

eje nombre de dispositivo tipo eje
tipo constante tipo integer. Los valores admitidos son:
0 = solo el eje maestro pone la cota, el eje esclavo mantiene la cota precedente
diferente de 0 = ambos el eje maestro y el eje esclavo ponen a cero la cota de manera síncrona.

Descripción

Esta instrucción permite programar la manera con la cual el eje esclavo indicado se comportará en respuesta a una instrucción [SETQUOTE](#) en el eje maestro.

La instrucción tiene que ser ejecutada solamente en el eje esclavo.

Si la variable **valor** está omisa, se asume un valor predeterminado igual a 0.

SETRIFLOC

Sintaxis

SETRIFLOC **cota1_ax1, cota2_ax1, cota3_ax1,**
cota1_ax2, cota2_ax2, cota3_ax2,
cota1_ax3, cota2_ax3, cota3_ax3,
eje1, eje2, eje3

Argumentos

cota1_ax1...cota3_ax3 cosenos directores de los tres ejes
eje1...eje3 nombres de dispositivos. Tipo eje

Descripción

Permite activar un sistema de referencia cartesiano X' Y' Z' rototranslado respecto al sistema de referencia absoluto de máquina X Y Z, representado por los ejes físicos **eje1, eje2 y eje3**.

Los nueve argumentos indican los Cosenos Directores de los tres ejes locales respecto a los absolutos,

$\cos\alpha_1$	$\cos\beta_1$	$\cos\gamma_1$
$\cos\alpha_2$	$\cos\beta_2$	$\cos\gamma_2$
$\cos\alpha_3$	$\cos\beta_3$	$\cos\gamma_3$

que constituyen la matriz de transformación de las coordenadas.

El origen del nuevo sistema de referencia es referida al punto corriente.

Todas las instrucciones de movimiento interpolado, que interesan los ejes X, Y y Z, son referidas a esto sistema de referencia, hasta la ejecución de la instrucción [RESRIFLOC](#).

SETTOLERANCE

Sintaxis

SETTOLERANCE **eje1, valor1, [eje2, valor2 [, eje3, valor3 [, ..., eje16, valor 16]]]**

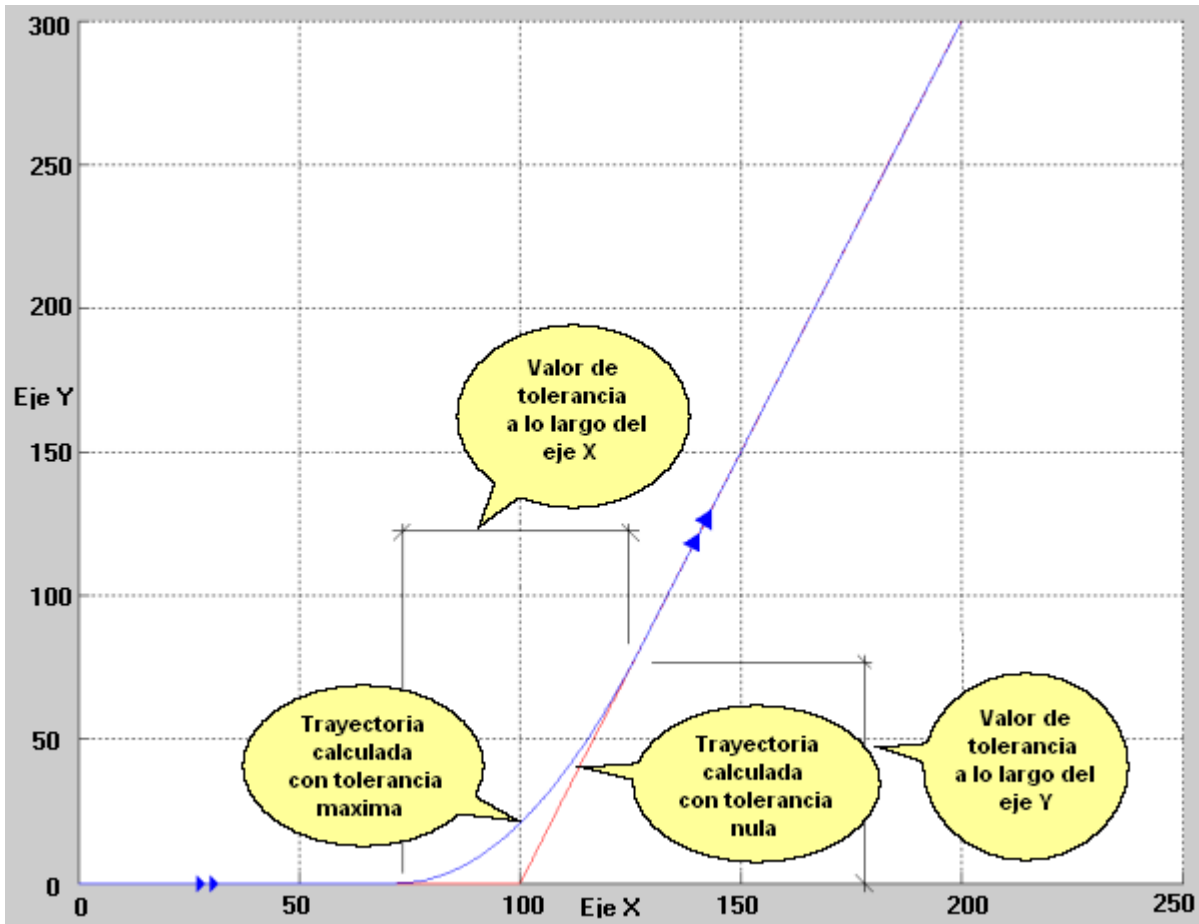
Argumentos

eje1...eje16 nombres de dispositivos tipo eje
valor1...[...valor16] constante o variable. Valor de la tolerancia máxima aplicable al eje

Descripción

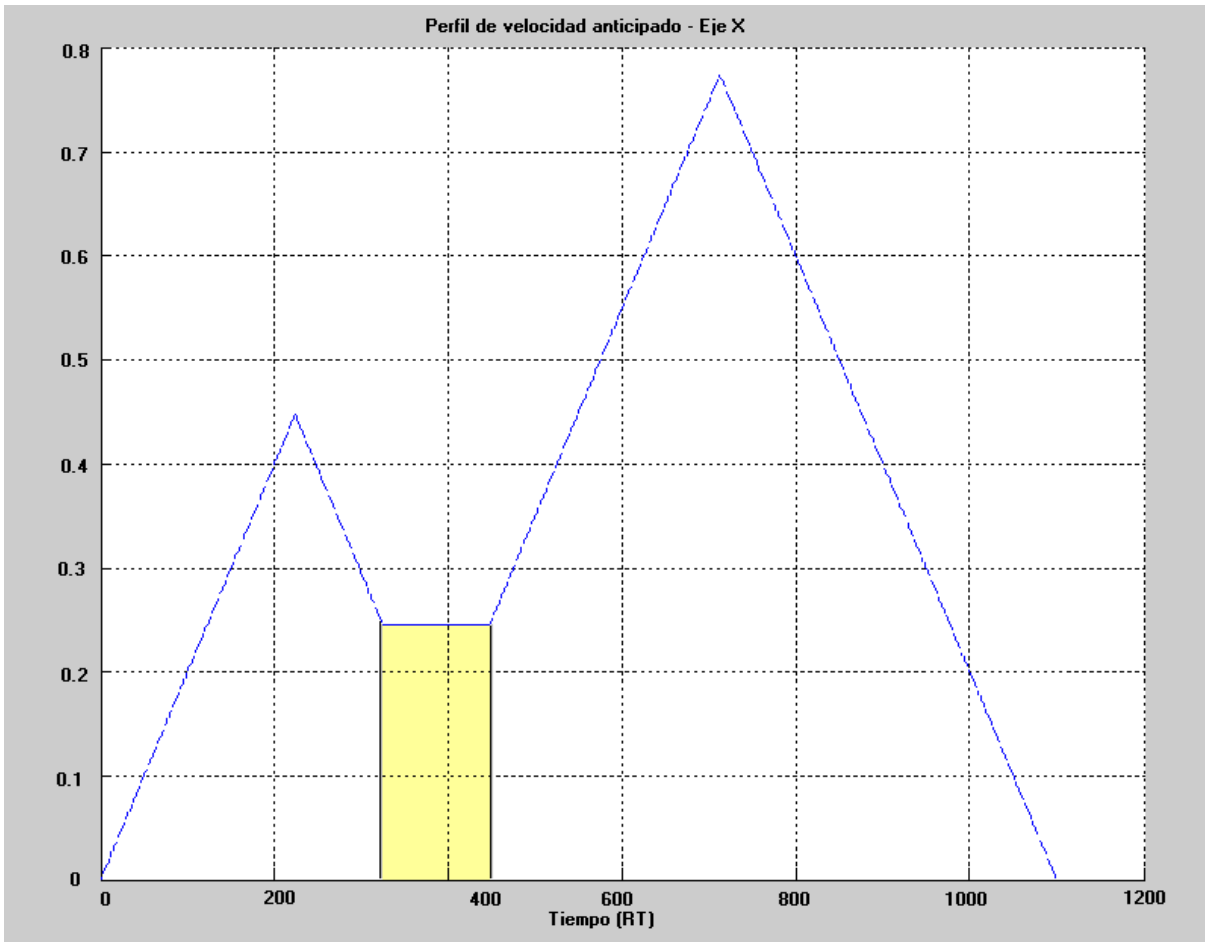
Programa para cada **eje** definido el **valor** de tolerancia da aplicar en los movimientos de interpolación multiteje. El valor de tolerancia es el valor de desplazamiento por el cual el eje se desplaza de la trayectoria original en una interpolación multiteje.

La tolerancia tiene que ser programada para cada **eje** involucrado en la interpolación, y el sistema proveerá a la anticipación de los perfiles de velocidad respetando las tolerancias sobre todos los ejes y no pasando por el espacio de rampa, que representa el límite máximo para el anticipación de los perfiles. La falta de asignación de la tolerancia antes de una instrucción multiteje conlleva el aplicación de la última tolerancia programada sobre el eje mismo. En el caso en que nunca ha sido asignado un valor de tolerancia a un determinado eje, el mismo es considerado a tolerancia nula. En este caso cada movimiento multiteje que involucrará ese eje no proveerá ninguna anticipación de rampa. Se recomienda por lo tanto el sistemático uso de esta instrucción para cada eje involucrado, antes de una instrucción de un movimiento de interpolación multiteje.



En la figura arriba es representada una clásica trayectoria multiteje, formada por dos bloques de movimiento, el primer de los cuales consiste en un desplazamiento de 100 del eje X, mientras que el segundo consiste en un movimiento del eje Y de 300 y uno siempre de 100 del eje X. La línea roja evidencia la trayectoria en caso de tolerancia nula, mientras que la azul es la trayectoria en caso de tolerancia máxima.

La tolerancia puede ser vista también como el área subtensa por el perfil de velocidad durante el intervalo de anticipación, como muestra la figura en seguida.



START

Sintaxis

START **eje**

Argumentos

eje nombre de dispositivo tipo eje

Descripción

Reactivación del movimiento del **eje** después del paro.

STARTINTERP

Sintaxis

STARTINTERP **eje**

Argumentos

eje nombre de dispositivo tipo eje

Descripción

Inicia una interpolación cuyo canal es identificado por **eje**. Normalmente el movimiento de los ejes asociados a un canal de interpolación comienza cuando se llena por completo el búfer del interpolador (512 instrucciones) o bien cuando se llega a una instrucción WAITSTILL, que de hecho termina el movimiento. Esto permite al algoritmo del interpolador determinar unos perfiles de velocidad optimales teniendo a disposición las informaciones relativas a un elevado número (o todos) de tramos del movimiento en interpolación. La instrucción STARTINTERP permite forzar el inicio del movimiento de los ejes aunque si no se verifican las condiciones apenas descritas.

STOP

Sintaxis

STOP **eje**

Argumentos

eje nombre de dispositivo tipo eje

Descripción

Paro del movimiento del **eje**. El eje ejecutará una rampa de deceleración cuya duración depende de la velocidad actual y de los parámetros de configuración.

Ejemplo

[Rutina de Puesta a cero de un eje](#)

SWITCHENC**Sintaxis**

SWITCHENC **eje1, [eje2, [dirección, cota]]**

Argumentos

eje1 nombre del dispositivo tipo eje
eje2 nombre del dispositivos tipo eje. indica un eje de conteo
dirección constante predefinida.
UP = intercambio encoder en la superación de la cota en dirección positiva
DOWN = intercambio encoder en la superación de la cota en dirección negativa
cota constante double o variable double

Descripción

Permite sustituir el encoder de **eje1** con el encoder de **eje2**. El encoder se intercambia cuando se supere en **dirección** positiva (UP) o negativa (DOWN) la cota indicada. Si se omiten los parámetros **dirección** y **cota**, en seguida se ejecuta el intercambio del encoder, a parte de la posición de los ejes.

Si se declara solo **eje1**, se restablece el funcionamiento con el encoder individual.

Eje1 no puede ser de tipo paso-a-paso, conteo y virtual, **Eje2** puede ser solamente un eje de conteo. Además, tanto el **eje1** como el **eje2** no pueden ser implicados en movimientos en cadena como eje esclavo.

La instrucción genera el error de sistema 4101 – Gestión incoherente del eje NombreEje, cuando el **eje1** o el **eje2** está declarado esclavo en un movimiento en cadena, o cuando **eje1** está ejecutando una instrucción FASTREAD o una instrucción SETPFLY. Además, se puede generar el error de sistema 4105 – Instrucción no ejecutable en el eje NombreEje, cuando el tipo de eje declarado no está incluido en los posibles.

WAITACC**Sintaxis**

WAITACC **eje1 [, ..., eje6]**

Argumentos

eje1 [...eje6] nombre del dispositivo tipo eje

Descripción

Espera del estado de aceleración o uno de los estados siguientes en todos los **ejes** (1÷6) especificados. La tarea en la cual es ejecutada la instrucción es puesta en espera hasta que el eje se encuentre en el estado de aceleración o en uno de los estados siguientes.

Los estados de los ejes son identificados por un número entero:

- aceleración = 1
- régimen = 2
- deceleración = 3
- cota = 4
- espera ventana grande = 5
- espera eje detenido = 6
- espera ventana pequeña = 7

WAITCOLL**Sintaxis**

WAITCOLL **eje, valor, timeout, delta**

Argumentos

eje nombre del dispositivo tipo eje

valor	constante o variable. Valor de coordenada absoluta
timeout	constante o variable. Es el tiempo de espera con eje parado
delta	constante o variable. Es el valor de ventana per obtener eje parado

Descripción

Durante el movimiento de un eje el alcance de una coordenada programada puede ser impedido por un obstaculo de natura mecánica, representado a veces también por la misma pieza en mecanización. El sistema en este caso genera un error de sistema de "servoerror" o "movimiento no finalizado". Con esta instrucción se define un **valor** de coordenada en correspondencia de la cual el sistema comienza a verificar la presencia de una colisión, el tiempo de espera (**timeout**) antes que el **eje**, después de la colisión, sea puesto en estado "coordenada" y el **delta** que define la tolerancia en el posicionamiento del eje. Cuando el eje supera la coordenada definida en **valor** es verificado si el eje todavía está en movimiento. Interceptado el obstáculo es individuada la situación crítica y, aún garantizando el arranque del motor, ya no está verificada la superación del límite del error de anillo. La dirección de moto sobre la cual es verificada la colisión es la misma dirección del último movimiento puesto en cola. El **timeout** está expresado en segundos, el valor **delta** tiene que ser mayor que 0.001 mm y menor que la diferencia entre la coordenada de llegada programada y la coordenada **valor**. La instrucción puede ser utilizada con el interpolador multiaxial, porque en tal interpolador se acepta la pérdida temporal vínculo de interpolación. La instrucción es aplicable también a ejes Máster de un movimiento en Chain.

Es generado un error de sistema cuando:

- el **eje** está ejecutando un movimiento interpolado clásico (Véase instrucciones LINNEARBS, LINEARINC, CIRCABS, CIRCINC, HELICABS, HELICINC) u en movimiento coordinado o movimientos interpolados ISO
- el **eje** es un eje Esclavo
- el **eje** es un eje de conteo o un eje paso-paso o un eje virtual
- el **valor** programado es superior a la coordenada de fin movimiento

Ejemplo

```
; programa la coordenada del eje x
SETQUOTE X, 0.0
; mueve el eje X a la coordenada absoluta 1000
MOVABS X, 1000.0
; espera la coordenada de colisión , espera 2 segundos antes de poner el
eje en estado "coordenada"
; después de haber intercectado una colisión con una presición de 0.01 mm
WAITCOLL X, 980.0,2.0,0.01
```

WAITDEC

Sintaxis

WAITDEC **eje1 [, ..., eje6]**

Argumentos

eje1 [...eje6] nombre del dispositivo tipo eje

Descripción

Espera del estado de deceleración o uno de los estado siguientes en todos los **ejes** (1÷6) especificados. La tarea en la cual es ejecutada la instrucción es puesta en espera hasta cuando el eje se encuentra en los estados de deceleración, cota, espera ventana grande, espera ventana pequeña o espera eje detenido.

Los estados de los ejes son identificados por un número entero:

- aceleración = 1
- régimen = 2
- deceleración = 3
- cota = 4
- espera ventana grande = 5
- espera eje detenido = 6
- espera ventana pequeña = 7

WAITREG

Sintaxis

WAITREG **eje1 [, ..., eje6]**

Argumentos

eje1 [...eje6] nombre del dispositivo tipo eje

Descripción

Espera del estado de régimen o uno de los estado siguientes en todos los **ejes** (1÷6) especificados.

La tarea en la cual es ejecutada la instrucción es puesta en espera hasta cuando el eje se encuentra en los estados de régimen, deceleración, cota, espera ventana grande, espera ventana pequeña o espera eje detenido.

Los estados de los ejes son identificados por un número entero:

- aceleración = 1
- régimen = 2
- deceleración = 3
- cota = 4
- espera ventana grande = 5
- espera eje detenido = 6
- espera ventana pequeña = 7

WAITSTILL

Sintaxis

WAITSTILL **eje1 [, ..., eje6]**

Argumentos

eje1 [...,eje6] nombre del dispositivo tipo eje

Descripción

Espera para todos los **ejes** (1÷6) especificados por el termino del movimiento (estado cota).

Ejemplo

[Rutina de Puesta a cero de un eje](#)

WAITTARGET

Sintaxis

WAITTARGET **eje1 [, ..., eje6]**

Argumentos

eje1 [...,eje6] nombre del dispositivo tipo eje

Descripción

Espera que para todos los **ejes** (1÷6) especificados la cota teórica corriente sea la misma de la cota teórica. La cota real será diferente de la cota teórica hasta que será puesto a cero el error de anillo.

WAITWIN

Sintaxis

WAITWIN **eje1 [, ..., eje6]**

Argumentos

eje1 [...,eje6] nombre del dispositivo tipo eje

Descripción

Espera del estado ventana o uno de los estado siguientes en todos los **ejes** (1÷6) especificados. La tarea en la cual es ejecutada la instrucción es puesta en espera hasta que el eje se encuentre en el estado de espera ventana grande, espera ventana pequeña o espera eje detenido.

Los estados de los ejes son identificados por un número entero:

- aceleración = 1
- régimen = 2
- deceleración = 3
- cota = 4
- espera ventana grande = 5
- espera eje detenido = 6
- espera ventana pequeña = 7

Parámetros Ejes

Lectura\Escritura

DEVICEID

Sintaxis

DEVICEID **dispositivo,variable**

Argumentos

dispositivo nombre del dispositivo o parámetro dispositivo
variable variable integer que recibe la dirección lógica

Descripción

Escribe en **variable** la dirección lógica asociada a **dispositivo**, de cualquier tipo. Por medio de esta instrucción se puede haber una "llave" univoca asociada al dispositivo, como indice o llave de búsqueda en estructuras de datos.

GETAXIS

Sintaxis

GETAXIS **eje, nombredato, nombrear**
GETAXIS **eje, nombredato1, nombredato2, [...,nombredato20,] matriz[línea]**

Argumentos

eje nombre del dispositivo tipo eje
nombredato constante predefinida (Véase lista abajo). Parámetro eje (1÷20)
nombrear variable o nombre de dispositivo
línea constante o variable integer. Número de línea de la matriz
matriz nombre de la matriz

Descripción

En la primera forma la instrucción lee uno de los datos (**nombredato**) de un eje y lo memoriza en una variable.

En la segunda forma esta instrucción lee en el mismo instante más datos de un eje (de 1 a 20) y los memoriza en el orden en que han sido solicitados en los elementos de la línea de matriz especificada.

En este caso el número de columnas de la matriz tiene que ser igual al número de datos solicitados. La lista, presentada en seguida, comprende todas las constantes predefinidas que pueden ser asignadas al parámetro **nombredato**.

La primera columna es el nombre de la constante.

La segunda columna es la descripción del tamaño del eje que está leído por la instrucción.

La tercera es el formato del dato que se retorna en la variable **nombrear** o en **matriz[línea]**, donde:

- **d** significa **doble**
- **f** significa **float**
- **i** significa **entero**
- **b** significa **char**

Si la declaración de la variable, en la cual será memorizado el dato, es diferente del valor retornado por la instrucción, el compilador ejecuta una transformación (cast) del dato en la tipología pedida por el utilizador. Esto a veces conlleva una pérdida de datos significativa. Por ejemplo, un valor doble igual a 12.345 transformado en un entero vuelve a ser 12. Por este hecho, en la declaración de las variables **nombrear** y **matriz[línea]** es mejor atenerse a la tipología solicitada.

La última columna describe o el valor de retorno o la unidad de medida del parámetro correspondiente.

Las constantes que comienzan con "_CFG" permiten leer los valores de configuración o sea los que se programan al arranque de la máquina.

constante	descripción	tipo	valor de retorno
_CFGTYPE	tipo eje	i	1=analógico,3=paso a paso,4=digital,5=cuenta, 6=no utilizado,7=virtual
_CFGUM	unidad de medida	i	0=milímetros,1=pulgadas,2=grados,3=r evoluciones
_CFGRIS	resolución	d	impulsos por _UM
_CFGVMAX	velocidad máxima	f	m/1' o pulgadas/1" o grados/1" o revoluciones/1'
_CFGVMAXD	velocidad máxima en manual	f	m/1' o pulgadas/1" o grados/1" o revoluciones/1'
_CFGVMAXI	velocidad máxima de interpolación	f	m/1' o pulgadas/1" o grados/1" o revoluciones/1'
_CFGPHINV	inversión fases encoder	b	0=no inversión, 1=inversión
_CFGRFINV	inversión referencia	b	0=no inversión, 1=inversión
_CFGZIND	habilitación retablecimiento cota muesca de cero	b	0=inhabilitado, 1=habilitado
_CFGSLOPE	tipo de rampa para aceleraciones/deceleraciones en	b	0=lineal, 1=a 'S', 2= en doble eje

constante	descripción	tipo	valor de retorno
	punto-a-punto		
_CFGKFFA	feed forward de aceleración	f	
_CFGKFFAI	feed forward de interpolación de aceleración	f	
_CFGSRPP	velocidad inicio rampa paso-a-paso	f	m/1' o pulgadas/1" o grados/1" o revoluciones/1'
_CFGACC	tiempo de aceleración de 0 a _CFGVMAX	i	mseg
_CFGDEC	tiempo de deceleración de _CFGVMAX a 0	i	mseg
_CFGACCI	tiempo de aceleración de 0 a CFGVMAXI	i	mseg
_CFGDECI	tiempo de deceleración de _CFGVMAX a 0	i	mseg
_CFGQLP	límite eje en positivo	d	cota
_CFGQLN	límite eje en negativo	d	cota
_CFGKP	coeficiente proporcional	f	
_CFGKI	coeficiente integrante	f	
_CFGKD	coeficiente derivativo	f	
_CFGKFF	feed forward	f	porcentaje
_CFGKPS	coeficiente proporcional eje esclavo	f	
_CFGKIS	coeficiente integrante eje esclavo	f	
_CFGKDS	coeficiente derivativo eje esclavo	f	
_CFGQEAP	error de anillo en positivo	d	cota
_CFGQEAN	error de anillo en negativo	d	cota
_CFGKPI	coeficiente proporcional de interpolación	f	
_CFGKII	coeficiente integrante de interpolación	f	
_CFGKDI	coeficiente derivativo de interpolación	f	
_CFGTMINP	tensión mínima positiva	f	volt
_CFGTMINN	tensión mínima negativa	f	volt
_CFGSTMINP	tensión de umbral positiva	f	volt
_CFGSTMINN	tensión de umbral negativa	f	volt
_CFGESC	tiempo de espera movimiento eje	i	mseg
_CFGDSE	habilitación servoerror dinámico	b	0=inhabilitado, 1=habilitado
_CGAEN	habilitación ajuste automático	b	0=inhabilitado, 1=habilitado
_CFGOFFSET	tensión de ajuste - desplazamiento inicial	f	volt
_CFGCEE	cota de conexión codificador incorrecta	d	cota
_CFGNOTCH	frecuencia filtro de notch	i	Hz
_CFGBUFI	tamaño búfer de cálculo integrante	i	[1, 200]
_CFGQAP	ventana llegada en cota positiva	d	
_CFGQAN	ventana llegada en cota negativa	d	
_CFGSLOPEI	tipo de rampa para aceleraciones/deceleraciones en interpolación	f	0=lineal, 1=a 'S', 2= a doble ese

constante	descripción	tipo	valor de retorno
_CFGKFFI	feed forward de interpolación	f	porcentaje
_CFGAAF	espera eje fijo	b	0=inhabilitado, 1=habilitado
_CFGENCTYPE	tipo de codificador	i	0=simulado o ausente, 1=real
_SRPP	velocidad inicio rampa paso-a-paso	f	m/1' o pulgadas/1" o grados/1" o revoluciones/1'
_ACC	tiempo de aceleración de 0 a _VMAX	i	mseg
_DEC	tiempo de deceleración de _VMAX a 0	i	mseg
_ACCI	tiempo de aceleración de 0 a _VMAXI en interpolación	i	mseg
_DECI	tiempo de deceleración de _VMAXI a 0 en interpolación	i	mseg
_QLP	límite eje en positivo	d	cota
_QLN	límite eje en negativo	d	cota
_KP	coeficiente proporcional	f	
_KI	coeficiente integrante	f	
_KD	coeficiente derivativo	f	
_KFF	feed forward	f	porcentaje
_KPS	coeficiente proporcional eje esclavo	f	
_KIS	coeficiente integrante eje esclavo	f	
_KDS	coeficiente derivativo eje esclavo	f	
_QEAP	error de anillo en positivo	d	cota
_QEAN	error de anillo en negativo	d	cota
_VEL	velocidad punto-a-punto	f	m/1' o pulgadas/1" o grados/1" o revoluciones/1'
_VELI	velocidad de interpolación	f	m/1' o pulgadas/1" o grados/1" o revoluciones/1'
_MODE	modo de funcionamiento eje	b	1=normal, 2=libre, 8=interpol., 10=coord.
_PHINV	inversión fases codificador	b	0=no inversión, 1=inversión
_RFINV	inversión referencia	b	0=no inversión, 1=inversión
_ZIND	habilitación restablecimiento cota muesca de cero	b	0=inhabilitado, 1=habilitado
_KPI	coeficiente proporcional de interpolación	f	
_KII	coeficiente integrante de interpolación	f	
_KDI	coeficiente derivativo de interpolación	f	
_KFFI	feed forward de interpolación	f	porcentaje
_KFFA	feed forward de aceleración	f	porcentaje
_KFFAI	feed forward de aceleración de interpolación	f	porcentaje
_ESC	tiempo de espera movimiento eje	i	mseg
_CEE	cota de conexión codificador incorrecta	d	cota
_NOTCH	frecuencia filtro de notch	i	Hz
_BUFI	tamaño búfer de cálculo integrante	i	[1,200]
_QAP	ventana llegada en cota positiva	d	

constante	descripción	tipo	valor de retorno
_QAN	ventana llegada en cota negativa	d	
_QEAPINV	límite error de anillo positivo en inversión	d	
_QEANINV	límite error de anillo negativo en inversión	d	
_OFSCoord	cota desplazamiento movimiento coordinado	d	
_MS	tipo eje maestro o esclavo	b	0=no en cadena,4=maestro,5=esclavo
_QENC	cota codificador	d	cota
_QR	cota real	d	cota
_RIS	resolución utilizada por el eje	d	
_ST	estado eje	b	1=acel., 2=régimen, 3=decel.,4=cota, 5=espera ventana grande, 6=espera eje detenido,7=espera ventana pequeña, 8=partida
_QT	cota teórica	d	cota
_EA	error de anillo	d	cota
_FF	feed forward	i	
_VC	velocidad corriente	f	
_P	corrector proporcional	i	
_I	corrector integrante	i	
_D	corrector derivativo	i	
_FLGS	indicadores ejes	b	
_VCR	velocidad corriente real	f	
_ADJUST	compensación desplazamiento eje	i	valor entero que representa la tensión de pasar al accionamiento, vista desde el lado tarjeta ejes del dac. La plena escala para el accionamiento es 10 volt y para el dac es de 32767
_DAC	DAC	i	valor entero que representa la tensión de pasar al accionamiento, vista desde el lado tarjeta ejes del dac. La plena escala para el accionamiento es 10 volt y para el dac es de 32767.
_ACCINST	valor de aceleración instantánea	f	
_FFA	feed forward de aceleración	i	
_GONETIME	tiempo transcurrido desde el inicio del movimiento	f	seg (0 para ejes esclavos y ejes paso-a-paso)
_RESTIME	tiempo residuo al final del movimiento. Los valores son relativos al movimiento asignado en el búfer al momento de la solicitud	f	seg (0 para ejes esclavos, ejes en movimiento interpolado y ejes paso-a-paso y movimiento ISO)
_TARGETTIME	tiempo dedicado a generar la cota teórica	f	microsegundos
_GONESPACE	espacio recorrido desde el inicio del movimiento. Los valores son relativos al movimiento asignado en el búfer al momento de la solicitud	f	porcentaje (100 para ejes esclavos, ejes en movimiento interpolado y ejes paso-a-paso)
_RESSPACE	espacio residuo al final del movimiento. Los valores son relativos al movimiento asignado en el búfer al momento de la solicitud	f	porcentaje (0 para ejes esclavos, ejes en movimiento coordinados y ejes paso-a-paso)

constante	descripción	tipo	valor de retorno
_AXESJERK	habilitación sobre el eje del control del jerk	b	1=control habilitado, 0=control no habilitado
_MOVEJERK	habilitación del control del jerk en el movimiento en el cual está empeñado el eje	b	1=control habilitado, 0=control no habilitado
_MOVETYPE	tipo de movimiento en el cual está empeñado el eje	b	1=movimiento interpolado clásico, 2=movimiento interpolado multieje, 3=movimiento coordinado, 4=movimiento punto- punto, 5=movimiento en cadena (solo eje esclavos)
_PARTYPESET	tipo de parámetros eje de uso durante el movimiento	i	1= interpolación, 0= punto a punto
_AXINRIFLOC	eje empleado en un sistema local de referencia	i	1= sí, 0=no
_QTARGETTOOL	cota teórica del eje. En caso de interpolación ISO cota teórica de la coordenada de la punta herramienta del eje	d	
_QREALTOOL	cota real del eje. En caso de interpolación ISO cota real de la coordenada de la punta herramienta del eje	d	
_BACKLASH	valor del juego mecánico definido para el eje	d	
_DISABLED	deshabilitación de un eje	b	1=eje inhabilitado,0=eje habilitado
_DYNLIMIT	habilitación del control dinámico de los límites eje	b	1=control habilitado, 0=control no habilitado
_AXESFEED	valor de velocidad de avance de alimentación aplicado actualmente al eje	f	
_CORRLIN	tipo de corrector de linealidad en uso	i	0=ningún corrector en uso, 1=auto corrector, 2=corrector cruzado, 3=auto corrector con corrector cruzado
_VELISO	velocidad de la broca de la herramienta por el movimiento 5 ejes	f	
_ISOSTOPS	número de paradas forzadas del movimiento interpolado por situaciones límites en la gestión del lookahead	i	
_CURRATIO	valor de la relación de encadenamiento actualmente en uso	d	
_DYNRATIO	retorna, si un cambio dinámico de relación de encadenamiento está en curso	i	0=no, 1 = sí
_RESBLOCK	número de bloqueos de desplazamiento puestos en cola	i	
_EXECBLOCK	número de bloqueos de desplazamiento ejecutados	i	
_TOTALBLOCK	número de bloqueos de desplazamiento en acuerdo total en el movimiento (valor actual)	i	
_SWITCHENC	controla si se está ejecutando el intercambio codificador	i	-1=el eje no utiliza la instrucción SWITCHENC, 0= ha sido ejecutada una instrucción SWITCHENC, pero el eje está ejecutando su codificador, 1= ha sido ejecutada una instrucción SWITCHENC y

constante	descripción	tipo	valor de retorno
			el eje está utilizando el codificador del eje de conteo
_QOFSENC	valor del desplazamiento codificador	d	
_LENSZETPZERO	distancia recorrida para alcanzar la muesca de cero	d	cota
_TORQUEINST	valor de par instantáneo	i	
_CURRSLOPE	devuelve el tipo de rampa actualmente en uso en los movimientos rápidos	i	0=lineal, 1=en 'S', 2=en doble 'S'
_CURRSLOPEI	devuelve el tipo de rampa actualmente en uso en los movimientos interpolados	i	0=lineal, 1=en 'S', 2= en doble 'S'
_REALSLOPEI	devuelve el tipo de rampa en uso en los movimientos interpolados	i	0=lineal, 1=en 'S', 2= en doble 'S'
_AXISPAR1...AXIS PAR8	parámetros de opciones adicionales para el eje EtherCAT	i	número
_ISOMOVETYPE	Tipo de movimiento ISO en ejecución	i	0 = movimiento ISO rápido, 1 = movimiento ISO interpolado, -1 = otro
_QMAINENC	Cota real del codificador principal cuando el secundario está en uso	d	

Movimiento punto a punto

SETACC

Sintaxis

SETACC **eje, [valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Tiempo de aceleración

Descripción

Asigna al **eje** el tiempo de aceleración identificado por **valor**. El tiempo de aceleración está expresado en milisegundos.

Si **valor** está omitido, se asume el parámetro de configuración. Si la instrucción es puesta entre dos instrucciones MOVABS o MOVINC, la primera instrucción de movimiento es ejecutada (el movimiento se detiene), utilizando los parámetros de aceleración y deceleración previamente programados. La segunda instrucción es ejecutada aplicando los nuevos parámetros de aceleración. La instrucción SETACC hace efecto solo en los movimientos sucesivos a su ejecución.

Si el **valor** especificado es más pequeño del valor de configuración es agarrado este último.

Véase también [SETDEC](#), [SETACCI](#) y [SETDECI](#).

SETDEC

Sintaxis

SETDEC **eje, [valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Tiempo de deceleración

Descripción

Asigna al **eje** el tiempo de deceleración identificado por **valor**. El tiempo de deceleración está expresado en milisegundos. Si **valor** está omitido, se asume el parámetro de configuración. Si la instrucción es puesta entre dos instrucciones MOVABS o MOVINC, la primera instrucción de movimiento es ejecutada (el movimiento se detiene), utilizando los parámetros de aceleración y deceleración previamente programados. La segunda instrucción es ejecutada aplicando los nuevos parámetros de deceleración. La instrucción SETACC hace efecto solo en los movimientos sucesivos a su ejecución.

Si el **valor** especificado es más pequeño del de configuración, se toma este último.

Véase también [SETACC](#), [SETACCI](#) y [SETDECI](#).

SETDERIV

Sintaxis

SETDERIV **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Coeficiente de acción derivativa. No son admitidas tipo char e integer

Descripción

Asigna al **eje** el **valor** *coeficiente de acción derivativa*.
Si **valor** es omitido, es usado el coeficiente de acción derivativa de configuración.
La instrucción no puede ser aplicada a un motor paso-paso.
Véase también la instrucción [SETDERIVI](#).

SETFEED

Sintaxis

SETFEED **eje, valor**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Representa la porcentual de feed rate override

Descripción

Modifica el **valor** porcentaje del feed rate override del **eje** relativamente a los *movimientos punto-punto*.
Véase también [SETFEEDI](#).

SETFEEDF

Sintaxis

SETFEEDF **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Porcentaje de feed forward

Descripción

Asigna al **eje** el **valor** *porcentaje de feed forward*.
Se **valor** está omitido, se usa el coeficiente de feed forward de configuración.
Si la instrucción es aplicada a un motor paso-paso es generado un error de sistema. Lo mismo pasa si a la variable **valor** es asignado un **valor** no comprendido entre 0 y 100.
Véase también las instrucciones [SETFEEDFI](#), [SETFEEDFA](#), [SETFEEDFAI](#).

SETFEEDFA

Sintaxis

SETFEEDFA **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Porcentaje de feed forward

Descripción

Asigna al **eje** el **valor** *porcentual de feed forward* de aceleración para los movimientos punto-punto.
Se **valor** está omitido, se usa el coeficiente de feed forward de configuración.
Si la instrucción es aplicada a un motor paso-paso es generado un error de sistema. Lo mismo pasa si a la variable **valor** está asignado un valor no comprendido entre 0 y 100.
Véase también las instrucciones [SETFEEDF](#), [SETFEEDFI](#), [SETFEEDFAI](#).

SETINTEG

Sintaxis

SETINTEG **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Coeficiente de acción integral. No son admitidos tipos char e integer.

Descripción

Asigna al **eje** el **valor** *coeficiente de acción integral*.
 Se **valor** está omitido, se usa el coeficiente de acción integral de configuración.
 La instrucción no puede ser aplicada a un motor paso-paso.
 Véase también la instrucción [SETINTEGI](#).

SETMULTIFEED**Sintaxis**

SETMULTIFEED **eje1, valor1, eje2, valor2 [, eje3, valor3 [, ..., eje16, valor 16]]]**

Argumentos

eje1...eje16 nombres de dispositivos tipo eje
valor1...[...valor16] constante o variable. Representa el porcentaje de feed rate override

Descripción

Modifica el **valor** porcentual de feed rate override de los **ejes** indicados, relativamente a los *movimientos punto-punto*. Para cada eje puede ser programado un valor diferente.

SETPROP**Sintaxis**

SETPROP **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Coeficiente de acción proporcional. No so admitidos tipos char e integer

Descripción

Asigna al **eje** el **valor** *coeficiente de acción proporcional*.
 Si **valor** está omitido, se usa el coeficiente de acción proporcional de configuración.
 La instrucción no puede ser aplicada a un motor paso-paso.
 Véase también la instrucción [SETPROPI](#).

SETSLOPE**Sintaxis**

SETSLOPE **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable entera. Tipo de rampa.

Descripción

Configura el tipo de rampa que se debe utilizar para el movimiento rápido:

- 0 rampa lineal
- 1 rampa en ese
- 2 rampa en doble ese

En caso de omisión de **valor**, se restablece la rampa de configuración.
 Se puede cambiar el tipo de rampa solo cuando el eje permanece parado en el estado COTA. En caso contrario se produce el error de sistema 4101 – Gestión incoherente del eje NombreEje.
 En asociación con esta instrucción y por medio de la instrucción, se puede comprobar el tipo de rampa actualmente utilizado por el eje [GETAXIS](#) con ayuda del parámetro `_CURRSLOPE`.

Véase también la instrucción [SETSLOPEI](#).

SETVEL**Sintaxis**

SETVEL **eje [, velocidad]**

Argumentos

eje nombre del dispositivo tipo eje
velocidad constante float o variable float

Descripción

Programa la **velocidad** máxima del **eje** para los movimientos punto-punto.
 La **velocidad** es expresa en la unidad de medida del eje, especificada en configuración.
 Si la **velocidad** programada está superior a la máxima velocidad de configuración, se usa esta última.

Si el argumento **velocidad** está omitido, se asume la de configuración. Son admitidos solo valores positivos de **velocidad**. Véase la instrucción [SETVELI](#).

Ejemplo

[Rutina de Puesta a cero de un eje](#)

Movimiento interpolado

LOOKAHEAD

Sintaxis

LOOKAHEAD [valor]

Argumentos

valor constante o variable. Valor de antebúsqueda

Descripción

Programa el valor de antebúsqueda del interpolador. El antebúsqueda es el número de bloques de interpolación que son procesados antes del inicio del movimiento de los ejes. El antebúsqueda permite generar unos perfiles de movimiento optimales, en particular cuando se utilizan rampas en "S". Si no es especificado el parámetro **valor**, el sistema utiliza un antebúsqueda de 512 bloques (predeterminados).

El valor máximo admitido es de **4096/númerocanales** donde **númerocanales** es el número de canales de interpolación definidos en configuración de módulo. El mínimo valor admitido es 256.

Atención

Por bloqueo de interpolación se entiende el conjunto de informaciones asociadas a una cualquiera instrucción de movimiento interpolado (por ej. LINEARABS).

Ejemplo

LOOKAHEAD 1024

SETACCI

Sintaxis

SETACCI eje1 [, ..., eje6] [, valor]

Argumentos

eje1,[...eje6] nombre del dispositivo tipo eje
valor constante o variable. Tiempo de aceleración

Descripción

Asigna a los ejes **eje1**, **eje2** el tiempo de aceleración para movimientos interpolados identificado por **valor**. Hay que expresar el tiempo en milisegundos. Si valor está omitido, se usa el parámetro de configuración.

Véase también [SETACC](#), [SETDEC](#), y [SETDECI](#).

SETACCLIMIT

Sintaxis

SETACCLIMIT eje,[valor]

Argumentos

eje nombre del dispositivo tipo eje
valor constante de tiempo del accionamiento

Descripción

Habilita y deshabilita el cálculo automático de la velocidad de régimen de interpolación en función de las aceleraciones toleradas por los ejes. El parámetro **valor** es una constante de tiempo utilizada para determinar el grado de velocidad tolerado por el **eje**, en milisegundos. Este parámetro es opcional. Si el parámetro está omiso, la instrucción tendrá el efecto de deshabilitar el cálculo automático. Un valor estándar para este parámetro es 30 milisegundos. Bajando esto tiempo se tendrá el efecto bajar la velocidad del perfil, haciendo más blando el movimiento. Aumentandolo se obtendrá el efecto contrario. Esta instrucción no puede ser aplicada a las interpolaciones elicoidales.

SETACCSTRATEGY

Sintaxis

SETACCSTRATEGY eje, [valor]

Argumentos

eje nombre del dispositivo tipo eje
[valor] variable o constante entera

Descripción

Permite seleccionar el tipo de aceleración que se desea para los siguientes movimientos de interpolación. La instrucción debe de ser ejecutada para todos los ejes ocupados en la interpolación.

Los valores admitidos para el parámetro **valor** son 0, 1 y 2. En el caso en que sea pasado el valor 0, se adoptará la consuetud estrategia de aceleración (será elegida como aceleración de perfil la menor entre todos los ejes ocupados en la interpolación). En caso de valor igual a 2 e interpolación lineal, se tomará la aceleración máxima que cada eje puede soportar (teniendo en cuenta los componentes individuales de cada eje, lineal y/o rotativo), la gestión de las aceleraciones en caso de interpolación circular sigue sin cambios. El caso de valor 1 llama una gestión obsoleta que se mantiene por compatibilidad.

SETAXPARTYPE

Sintaxis

SETAXPARTYPE eje, [valor]

Argumentos

eje nombre de dispositivo tipo eje
[valor] variable o constante integer.

Descripción

Durante la ejecución de una interpolación multilineal esta instrucción permite cambiar el conjunto de parámetros de uso, pasando de los típicos de la interpolación (**valor** = 1) a los empleados para el movimiento punto a punto (**valor** = 0). Si la variable **valor** está omitida, los parámetros usados son los de la interpolación.

El cambio del conjunto de parámetros se puede hacer solo si el eje está detenido en estado COTA, si no la instrucción genere el error de sistema 4101 – Gestión incoherente del eje NombreEje.

SETCONTORNATURE

Sintaxis

SETCONTORNATURE [valor1[,valor2]]

Argumentos

valor1 constante o variable. Máximo ángulo de contorneado
valor2 constante o variable. Máximo ángulo de desaceleración

Descripción

Establece el ángulo mínimo entre las tangentes de dos trayectorias ejecutadas en interpolación más allá del cual la máquina no ejecutará un contorneado, o sea los ejes serán detenidos al termino de la primera trayectoria y, por lo tanto, volverán a partir a lo largo de la segunda. Por lo tanto, se define un *ángulo máximo de contorneado* el **valor1**, que representa el angulo máximo entre dos tramos de desplazamiento, abajo del cual el movimiento no se detiene. Si el ángulo entre dos bloques de desplazamiento es mayor que el ángulo máximo de contorneado, el movimiento se detiene. Para evitar la parada, el **valor2** puede establecerse como valor de ángulo máximo de desaceleración. Si el ángulo entre dos bloques de desplazamiento está entre el *ángulo máximo de contorneado* y el *ángulo máximo de desaceleración*, el movimiento no se detiene, sino solo desacelera. Por lo tanto, el *ángulo máximo de desaceleración* representa el ángulo más allá del cual el movimiento tiene necesariamente que detenerse. En ángulos menores que el angulo máximo de contorneado, el movimiento **no desacelera**, en ángulos entre el angulo máximo de contorneado y el angulo máximo de desaceleración el movimiento **desacelera**, en ángulos mayores que el angulo máximo de contorneado el movimiento **se para**.

Valor1 y **valor2** son parámetros opcionales; si ambos no están establecidos, se toma el valor de 15° como valor predeterminado. Si solo se establece el primer parámetro, se supone que el *ángulo máximo de desaceleración* es igual al *ángulo máximo de contorneado*. La función de desaceleración se deshabilita, cuando el *ángulo máximo de desaceleración* es menor o igual al *ángulo máximo de contorneado*. El *ángulo máximo de desaceleración* es 180°. Si se establece un valor superior, se produce el error de sistema 4399: "Parámetro fuera del intervalo".

La función de desaceleración se habilita solo cuando la instrucción [JERKSMOOTH](#) está activa, mientras el contorneado está siempre activo.

Atención

El uso de esta instrucción se relaciona con el uso de las instrucciones [JERKSMOOTH](#) y [SETSLOWPARAM](#) y produce sus efectos solo en los movimientos con interpolación clásica (instrucciones [LINEARABS](#), [LINEARINC](#), [CIRCABS](#), [CIRCINC](#), [HELICABS](#), [HELICINC](#)).

SETDECI

Sintaxis

SETDECI **eje1 [, ..., eje6] [, valor]**

Argumentos

eje1,[...eje6] nombre del dispositivo tipo eje
valor constante o variable. Tiempo de deceleración

Descripción

Asigna a los ejes **eje1**, **eje2** el tiempo de deceleración para movimientos interpolados identificados por valor. Hay que expresar el tiempo en milisegundos. Si **valor** está omitido, está utilizado el parámetro de configuración.

Véase también [SETACC](#), [SETDEC](#), y [SETACCI](#).

SETDERIVI

Sintaxis

SETDERIVI **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Coeficiente de acción derivativa. Non son admitidos tipo char e integer

Descripción

Asigna al **eje** el **valor** *coeficiente de acción derivativa* utilizado durante los movimientos de interpolación de los ejes.

Si **valor** está omitido, se usa el coeficiente de acción derivativa de configuración.

La instrucción no puede ser aplicada a un motor paso-paso.

Véase también la instrucción [SETDERIV](#).

SETFEEDFAI

Sintaxis

SETFEEDFAI **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Porcentaje de feed forward

Descripción

Asigna al **eje** el **valor** *porcentual de feed forward* de aceleración para los movimientos interpolados.

Se **valor** está omitido, se usa el coeficiente de feed forward de configuración.

Si la instrucción es aplicada a un motor paso-paso es generado un error de sistema. Lo mismo pasa si a la variable **valor** está asignado un valor no comprendido entre 0 y 100.

Véase también las instrucciones [SETFEEDF](#), [SETFEEDFI](#), [SETFEEDFA](#).

SETFEEDI

Sintaxis

SETFEEDI **eje, valor**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable. Representa el porcentaje de feed rate override

Descripción

Modifica el **valor** porcentual de feed rate override del **eje** relativamente a los *movimientos en interpolación*. Véase también la instrucción [SETFEED](#).

SETFEEDFI**Sintaxis****SETFEEDFI** **eje [, valor]****Argumentos****eje** nombre del dispositivo tipo eje
valor constante o variable. Porcentual de feed forward**Descripción**

Asigna al **eje** el **valor porcentual de feed forward** para los movimientos en interpolación. Si el argumento **valor** está omitido, el sistema asume como porcentaje de feed forward el que está definido en los parámetros de configuración del dispositivo eje interesado. La instrucción no puede ser aplicada ad un motor paso-paso. Para la variable **valor** se admiten valores comprendidos entre 0 y 100. Véase también la instrucción [SETFEEDF](#), [SETFEEDFA](#), [SETFEEDFAI](#).

SETINTEGI**Sintaxis****SETINTEGI** **eje [, valor]****Argumentos****eje** nombre del dispositivo tipo eje
valor constante o variable. Coeficiente de acción integral. No son admitidos tipos char e integer**Descripción**

Asigna al **eje** el **valor coeficiente de acción integral** utilizado durante los movimientos de interpolación de los ejes. Si **valor** está omitido, se usa el coeficiente de acción integral de configuración. La instrucción no puede ser aplicada a un motor paso-paso. Véase también la instrucción [SETINTEG](#).

SETPROPI**Sintaxis****SETPROPI** **eje [, valor]****Argumentos****eje** nombre del dispositivo tipo eje
valor constante o variable. Coeficiente de acción proporcional. No son admitidos tipos char e integer**Descripción**

Asigna al **eje** el **valor coeficiente de acción proporcional** usado durante los movimientos de interpolación de los ejes. Si **valor** está omitido, se usa el coeficiente de acción proporcional de configuración. La instrucción no puede ser aplicada a un motor paso-paso. Véase también la instrucción [SETPROP](#).

SETSLOPEI**Sintaxis****SETSLOPEI** **eje [, valor]****Argumentos****eje** nombre del dispositivo tipo eje
valor constante o variable integer. Tipo de rampa.**Descripción**

Configura el tipo de rampa.que se debe utilizar para el movimiento de interpolación (cuando lo permita):

- 0 rampa lineal
- 1 rampa en ese
- 2 rampa en doble ese

En caso de omisión de **valor**, se restablece la rampa de configuración.

Se puede cambiar el tipo de rampa solo cuando el eje no está captado todavía en un canal de interpolación. En caso contrario se produce el error de sistema 4101 – Gestión incoherente del eje NombreEje.

En asociación con esta instrucción y por medio de la instrucción, [GETAXIS](#), se puede comprobar el tipo de rampa actualmente utilizado por el eje con ayuda del parámetro `_CURRSLOPEI`; mediante `_REALSLOPEI` se puede ver el tipo de rampa utilizado (la del canal donde se capta el eje) por el eje.

Véase también la instrucción [SETSLOPE](#).

SETSLOWPARAM

Sintaxis

SETSLOWPARAM **eje, [,valor1, [valor2]]**

Argumentos

eje	nombre de dispositivo tipo eje
valor1	constante o variable doble. Factor de reducción general
valor2	constante o variable doble. Factor de reducción inversión

Descripción

Esta instrucción modifica los parámetros necesarios para calcular la velocidad de desaceleración, si la desaceleración de contorneado ha sido habilitada (véase la instrucción [SETCONTORNATURE](#)). La velocidad de desaceleración se calcula inicialmente de manera teórica eje por eje. En caso de inversión de movimiento, puede ser reducida mediante el uso del **valor2** en el cálculo. Luego, entre cada velocidad calculada, se considera la velocidad mínima de manera que se respete la dinámica del eje que limita más. Por último, se puede reducir aún más la velocidad de desaceleración de un factor que depende del **valor1**.

Si se omiten **valor1** o **valor2**, se asumen valores predeterminados cuyo valor genera una acción nula. El parámetro **valor1** representa el valor porcentual de reducción de la velocidad de desaceleración teórica. La velocidad de desaceleración aplicada es $(100 - \text{valor1})/100$ veces la velocidad de desaceleración teórica. El valor máximo de reducción es 100. En este caso la velocidad resultante es igual al 1% de la velocidad teórica. Viceversa, cuando el valor es cero o ha sido omitido, se considera el valor predeterminado, o sea se considera la velocidad teórica entera.

El parámetro **valor2** representa el porcentaje de reducción. entre 1 y 10 veces, de la velocidad de desaceleración teórica en caso de que un eje haya una inversión de movimiento. En particular, se supone por lo tanto que, cuando **valor2** es 100, la velocidad es reducida de 10 veces. Viceversa, cuando es cero o se ha omitido, la velocidad no disminuye.

Esta instrucción genera un error de sistema 4399: "Parámetro fuera de intervalo", cuando el valor establecido es menor de 0 o mayor de 100. Es importante registrar que si se omite el parámetro **valor1**, hay que omitir el parámetro **valor2** también.

Atención

El uso de esta instrucción se relaciona con el uso de las instrucciones [JERKSMOOTH](#) y [SETSLOWPARAM](#) y produce sus efectos solo en los movimientos con interpolación clásica (instrucciones [LINEARABS](#), [LINEARINC](#), [CIRCABS](#), [CIRCINC](#), [HELICABS](#), [HELICINC](#)).

SETVELI

Sintaxis

SETVELI **eje1 [, ..., eje6] [, velocidad]**

Argumentos

eje1 [...eje6]	nombre del dispositivo tipo eje que interpolar
velocidad	constante float o variable float

Descripción

Programa la **velocidad** máxima de los ejes **eje1**, **eje2**, para los movimientos interpolados. La velocidad es expresada en la unidad de medida de los ejes, especificada por el parámetro de configuración. Si el argumento **velocidad** está omitido, se asume la velocidad máxima de configuración. Ejes paso-a-paso se pueden usar en esta instrucción si solo son gestionados por un remoto TRS-AX. Véase la instrucción [SETVEL](#).

SETVELILIMIT

Sintaxis

SETVELILIMIT **eje [, velocidad]**

Argumentos

eje	nombre del dispositivo tipo eje
velocidad	constante float o variable float

Descripción

Programa cada componente de **velocidad** del **eje especificado**, para los movimientos interpolados. La velocidad es expresada en la unidad de medida del eje.

Movimiento coordinado

SETFEEDCOORD

Sintaxis

SETFEEDCOORD **eje, valor1, valor2**

Argumentos

eje	nombre del dispositivo tipo eje
valor1	constante o variable double. Representa el porcentaje máximo de variación instantánea del feedrate
valor2	constante o variable integer. Representa el número de tiempo reales en el cual se aplica la variación de los feedrate

Descripción

Modifica el **valor1** porcentual de variación máxima instantánea del feed rate del **eje**. El feed rate ya no es variado para el intervalo de tiempo, expresado en tiempo real, definido en la variable **valor2**. En otros términos, después de haber aplicado una variación de feedrate override, a los más igual a **valor1**, por un número de tiempos reales de **valor2**, ninguna nueva variación de feedrate puede aplicarse. La combinación de estos dos parámetros define un tipo de aceleración/deceleración que el eje puede soportar. Modulando estos dos parámetros se pueden obtener unas rampas "por grados" de la inclinación deseada.

Atención

Para cada eje involucrado en el movimiento coordinado tienen que ser programados el valor de feedrate y el intervalo de tiempo, si no son asumidos los valores predeterminados que son **valor1** = 100 y **valor2** = 1. Durante la ejecución del movimiento coordinado (instrucción [COORDIN](#)) el sistema vuelve a calcular los parámetros **valor1** y **valor2** que aplicar al movimiento, en base a los parámetros de todos los ejes involucrados. Son excluidos del control los ejes parados. Los dos parámetros son calculados de manera que:
valor1: valor mínimo programado sobre los ejes que se están desplazando
valor2: valor obtenido dividiendo **valor1** por la mínima relación **valor1/valor2**.

Ejemplo

Funcion MovimientoCoordenado

```
Setquote      X,0
Setquote      Y,0
Setquote      Z,0

setFeedCoord  X, 20, 80
setFeedCoord  Y, 10, 1
setFeedCoord  Z, 3, 3

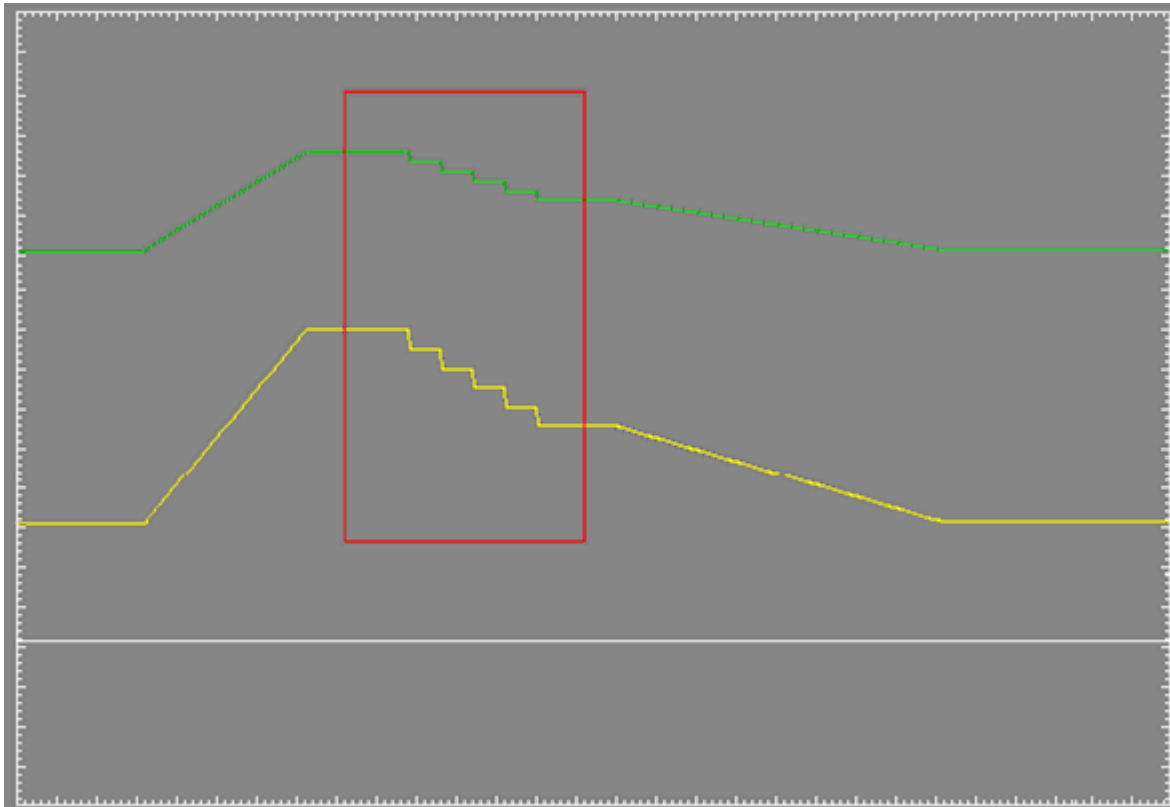
coordin       matrix, deltaT, UP, rigaInit, rigaEnd,mask,_X,columnX,
              Y,columnY, Z,columnZ
waitstill     X,y,z
```

fret

Supongamos que en un determinado pasaje del movimiento coordinado el eje Z no se desplace. Los parámetros programados resultan ser:

```
Variación_Max = 10
Delta_T       = 10/0.25 = 40
```

Tenemos por lo tanto el siguiente trazo de osciloscopio con en Verde el perfil de velocidad del eje X y en amarillo el perfil del eje Y



SETOFFSET

Sintaxis

SETOFFSET **eje, cota**

Argumentos

eje nombre del dispositivo tipo eje
cota constante o variable. Desplazamiento para movimiento coordinado

Descripción

Permite aplicar un desplazamiento a las cotas de un movimiento coordinado. El desplazamiento especificado por el parámetro **cota** será usado en los movimientos coordinados sucesivos, sumando la cota indicada a todas las cotas presentes en la tabla. Véase también la instrucción [COORDIN](#).

Movimiento concatenado

RATIO

Sintaxis

RATIO **eje, [valor]**

Argumentos

eje nombre de dispositivos tipo eje.
valor constante o variable. Relación de reducción.

Descripción

Programa la relación de encadenamiento de un eje esclavo con respecto del propio eje maestro. Los movimientos del eje esclavo resultarán escalado con respecto de los del eje maestro de la relación de encadenamiento programada. Si no está presente el parámetro **valor**, la relación es restablecida a 1.0 (movimientos idénticos). La instrucción genera error de sistema si ejecutada cuando el eje no está en estado esclavo y el eje maestro correspondiente no está en estado cota. Véase la instrucción [CHAIN](#).

Ejemplo

```
CHAIN                      X, Y
RATIO                      Y, 0.5 ; relación de reducción 1/2
```

MOVABS X, 100 ; el eje Y irá a cota 50
WAITSTILL X

SETDYNRATIO

Sintaxis

SETDYNRATIO eje, valor

Argumentos

eje nombre de dispositivo tipo eje
valor constante o variable double

Descripción

Esta instrucción permite variar de manera dinámica, durante el movimiento del eje máster, la relación de encadenamiento. Si puede aplicar un nuevo valor de la relación de encadenamiento también si la variación anterior no ha terminado. El **eje** declarado tiene que ser un eje esclavo.

En caso de que la instrucción sea ejecutada con eje máster en el estado COTA, el nuevo valor de relación de encadenamiento **valor** se aplica en seguida.

La variación de la relación de encadenamiento se realiza por medio de una rampa de aceleración o (deceleración) linear. El valor de aceleración utilizado está dado por la aceleración del eje máster, empleado en la actualidad para el movimiento punto-a-punto. Esto significa que se puede también modificar tal rampa por medio de un nuevo valor de aceleración, utilizando la instrucción [SETACC](#).

La instrucción puede generar el error de sistema siguiente:

- "4101: Gestión incoherente del eje" en caso de que el **eje** declarado no sea un eje esclavo.

Parámetros genéricos

DYNLIMIT

Sintaxis

DYNLIMIT eje, estado

Argumentos

eje nombre de dispositivo tipo eje
estado constante predefinida. Estado de verificar:
ON habilitación control dinámico límites eje
OFF inhabilitación control dinámico límites eje

Descripción

Habilita o inhabilita el test dinámico sobre la superación de los límites eje.

El test dinámico de superación de los límites eje es diferente del test estático de superación de los límites eje ya que verifica en cada tiempo real que el eje supere sus límites, en base a su velocidad actual y a su deceleración máxima. El test de tipo estático a cambio verifica instante tras instante que la cota de llegada actual de cada eje se encuentre al interior de los límites eje positivo o negativo programados. Además, antes del inicio del movimiento el test de tipo estático verifica si las cotas pasadas a través de las instrucciones de movimiento superan los límites programados.

Antes de una instrucción DYNLIMIT hay que programar las instrucciones [SETLIMPOS](#) y [SETLIMNEG](#), para que definen los nuevos límites.

Ejemplo

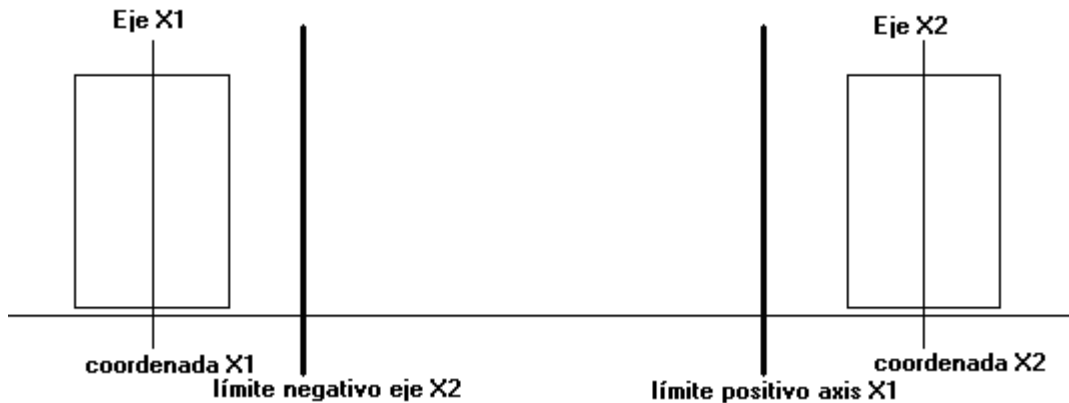
Verificación de los límites ejes según dos tipologías de test estático y dinámico, con ejes sobre la misma directora de movimiento.

Test estático

En un genérico movimiento el **Eje X1** no puede ser superior al límite positivo inicial dado por la cota del **Eje X2**. La verificación de los límites ejes genera un error de sistema n.4108 "Eje X1: Cota final más allá de los límites software".

Test dinámico

En un genérico movimiento se verifica que la cota instantánea cota X1 esté dentro de los límites eje disminuidos, con el signo oportuno según la dirección de movimiento del eje, del espacio mínimo de paro del eje mismo. El espacio mínimo de paro es calculado en función de la velocidad instantánea y de la deceleración programada en configuración para el movimiento punto-punto. Además, es inhabilitado el control inicial de superación de los límites programados de parte de las cotas pasadas a través de las instrucciones de movimiento.



ENABLESTARTCONTROL

Sintaxis

ENABLESTARTCONTROL eje, [timeout]

Argumentos

eje nombre del dispositivo tipo eje
timeout variable o constante entera. Es el tiempo límite de espera expresado en tiempo real.

Descripción

Esta instrucción permite habilitar y programar el **timeout** para el control de la partida fallida o repentino paro del eje.

Si el eje no se mueve de por lo menos 2 pasos en 200 tiempo reales en respuesta a un pedido de movimiento se genera el error de sistema no. 3 "Servoerror".

Si el parámetro **timeout** es programado a cero el control es desactivado. La instrucción no tiene efecto si la velocidad teorica es menor de dos pasos en 200 tiempo reales o si el movimiento termina en menos de 200 tiempo reales.

Ejemplo

```
; timeout de partida ejes de 10 tiempo reales
ENABLESTARTCONTROL X, 10
```

NOTCHFILTER

Sintaxis

NOTCHFILTER eje, [valor]

Argumentos

eje nombre de dispositivos tipo eje
valor constante o variable. Valor de frecuencia [Hz]. Permitidos los valores comprendidos entre **0** y **500**.

Descripción

Programa la frecuencia de corte del filtro notch para el eje especificado. Si **valor** es igual a 0 el filtro es deshabilitado. En caso en que el parámetro **valor** no esté presente será usado el valor programado en configuración.

Ejemplo

```
; corte de la frecuencia 97 Hz
NOTCHFILTER X, 97
```

RESLIMNEG

Sintaxis

RESLIMNEG eje

Argumentos

eje nombre del dispositivo tipo eje

Descripción

Deshabilita el test en límite negativo del **eje** especificado.

Estas instrucciones son utilizadas de s3lito en las rutinas de puesta a cero para la b3squeda de los interruptores de puesta a cero, permitiendo de esta manera a los ejes superar los valores de configuraci3n programados.

V3ase tambi3n las instrucciones [SETLIMNEG](#), [SETLIMPOS](#), [RESLIMPOS](#).

Ejemplo

[Rutina de Puesta a cero de un eje](#)

RESLIMPOS

Sintaxis

RESLIMPOS **eje**

Argumentos

eje nombre del dispositivo tipo eje

Descripci3n

Deshabilita el test en l3mite positivo del **eje** especificado.

Estas instrucciones son utilizadas de s3lito en las rutinas de puesta a cero para la b3squeda de los interruptores de puesta a cero, permitiendo de esta manera a los ejes superar los valores de configuraci3n programados.

V3ase tambi3n las instrucciones [RESLIMNEG](#), [SETLIMPOS](#), [SETLIMNEG](#).

Ejemplo

[Rutina de Puesta a cero de un eje](#)

SETADJUST

Sintaxis

SETADJUST **eje, estado, [valor]**

Argumentos

eje nombre del dispositivo tipo eje
estado constante predefinida. Puede asumir los siguientes valores:
ON activa
OFF desactiva
[valor] variable o constante flotante. Tensi3n [Volt]

Descripci3n

Activa o desactiva, en el **eje**, indicado el c3lculo de recuperaci3n desplazamiento autom3tico, o sea el ajuste.

El ajuste permite compensar unos ligeros desplazamiento de posici3n al final del movimiento del eje. Normalmente el ajuste est3 habilitado.

Puede ser conveniente deshabilitar el ajuste para ejes movidos por motores que presentan una elevada hist3resis de posici3n que por lo tanto no obtendr3an ning3n beneficio del uso de esta funci3n del control.

Cuando es reactivado el ajuste despu3s de una deshabilitaci3n, el control no toma en cuenta el valor calculado en precedencia. La instrucci3n puede por lo tanto ser utilizada para poner a cero el ajuste acumulado por un eje sin tener que inicializar de nuevo el control.

Cuando est3 presente el tercer par3metro el desplazamiento est3 configurado al **valor** indicado independientemente de la activaci3n o desactivaci3n del ajuste autom3tico. Este tipo de uso de la instrucci3n permite compensar a trav3s del software un eventual desplazamiento de la referencia de velocidad en vez de compensarlo en el accionamiento, aunque si la compensaci3n en el accionamiento es preferible a la compensaci3n a trav3s del software.

La instrucci3n puede aplicarse solo en ejes controlados en anal3gico.

SETBACKLASH

Sintaxis

SETBACKLASH **eje, valor**

Argumentos

eje nombre del dispositivo tipo eje
valor variable o constante float. Valor del juego

Descripci3n

Esta instrucci3n permite disminuir o eliminar los efectos de los juegos mecanicos sobre la trayectoria del **eje**. El **valor** del juego que puede ser programado tiene que ser incluido entre 0.0 y 3.0. Este valor es independiente del unidad de medida seleccionada.

Situaciones particulares pasan en los siguientes casos:

con eje deshabilitado la función de recupero juegos, aunque si requerida, no es aplicada.

- con eje vertical, dada la particular configuración, no se manifiesta ningun juego.
- con eje con cargo de grande inercia, se puede verificar una parcial o a veces total compensación del cargo. Es posible de hecho que, dada la masa del cargo, el movimiento de los ejes se pare con retraso con respecto al paro del motor. El posicionamiento resultante de los dientes del engranaje reductor respecto al posicionamiento de los dientes del engranaje motor puede ser por lo tanto tal de reducir o hasta anular el juego.
- la visualización de las coordenadas real y encoder del **eje**, campionadas con el osciloscopio, en los puntos en que es activado el recupero juegos (inversión del movimiento), presenta un pico igual al valor del juego mismo.

La instrucción genera error de sistema si utilizada:

- sobre ejes paso-paso, no gestionados por remotos TRS-AX, de conteo, virtuales.
- en ejes paso-a-paso gestionados de remotos TRS-AX con codificadores simulados.

Ejemplo

; Función con recupero juegos deshabilitado

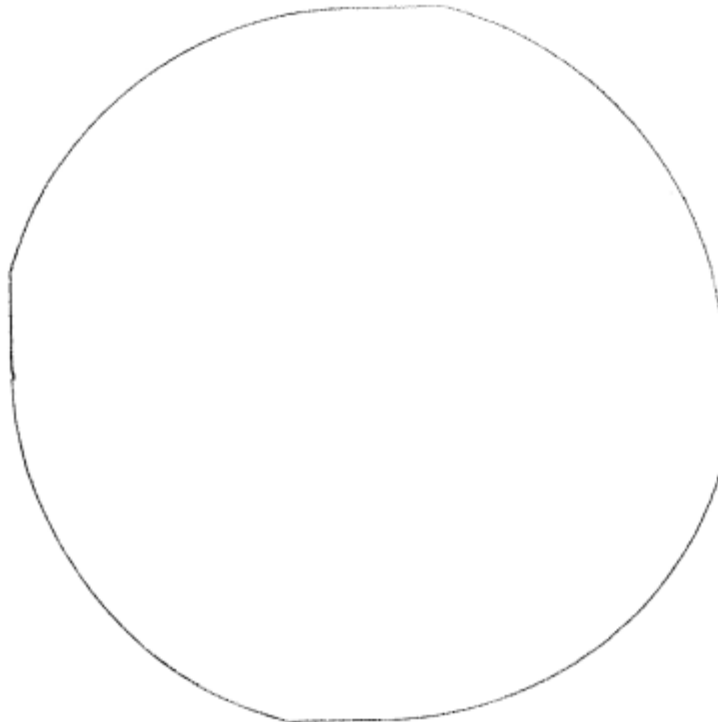
```
SETQUOTE X, 0
SETQUOTE y, 0
SETVELI X, 1.0
CIRCLE X,Y,cw,100,90
WAITSTILL X,Y
```

; Función con recupero juegos habilitado

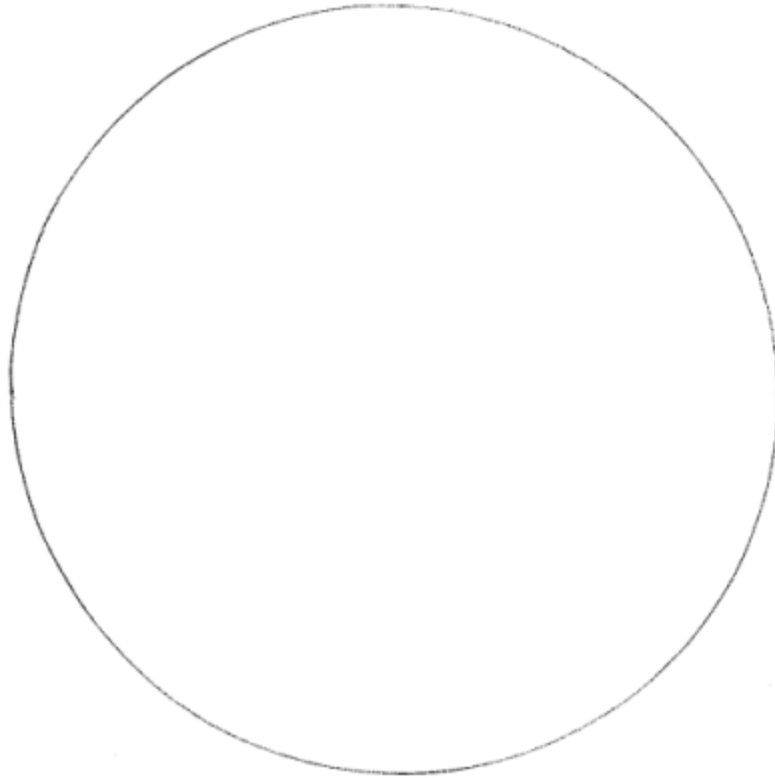
```
SETQUOTE X, 0
SETQUOTE y, 0
SETVELI X, 1.0
SETBACKLASH X, 1.9
SETBACKLASH y, 1.8
CIRCLE X,Y,cw,100,90
WAITSTILL X,Y
```

La ejecución de las dos funciones genera dos trazas diferentes.

La primera figura muestra la interpolación de dos ejes que presentan un juego en la copia motor-reductor.



La segunda figura representa la misma interpolación, pero con el uso de la instrucción de recupero juegos.



SETBIGWINFACTOR

Sintaxis

SETBIGWINFACTOR **eje, valor**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable double. Factor de multiplicación para el cálculo de la ventana grande.

Descripción

Esta instrucción permite modificar el factor de multiplicación para el cálculo de la ventana grande sobre el **eje** seleccionado. Para el cálculo de la ventana grande se multiplica la variable **valor** para el parámetro definido en configuración de los ejes de Ventana de Llegada en cota. El **valor** que puede ser programado debe de ser incluido entre 1 y 257 extremos excluidos. El predeterminado es 4.0.

SETDEADBAND

Sintaxis

SETDEADBAND **eje,VMinPos,VMinNeg,VUmbraIPos,VUmbraINeg**

Argumentos

eje nombre de dispositivo tipo eje
VMinPos variable o constante float. Tensión mínima positiva [Voltios]
VMinNeg variable o constante float. Tensión mínima negativa [Voltios]
VUmbraIPos variable o constante float. Umbral positivo [Voltios]
VUmbraINeg variable o constante float. Umbral negativo [Voltios]

Descripción

Programa los parámetros de tensión mínima para el eje indicado. Los valores de tensión mínima (positiva/negativa) son sumados a la tensión de referencia teórica (positiva/negativa) si el valor de esta supera el valor de umbral (positivo/negativo) programado. Si la tensión de referencia teórica se encuentra al interno de los valores de umbral, la tensión de referencia efectiva es forzada a cero. Es posible deshabilitar la gestión de la tensión mínima programando todos los valores a cero. Los valores de umbral deben ser siempre menores o iguales a los respectivos valores de tensión mínima. En la inicialización del sistema la gestión de la tensión mínima está deshabilitada.

SETENCLIMIT

Sintaxis

SETENCLIMIT **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante o variable doble

Descripción

Modifica el límite de enlace encoder incorrecto. El parámetro es expresado en la unidad de medida del eje. Los valores admitidos tienen que ser incluidos en un intervalo equivalente a 128 – 16384 pasos encoder. Si el parámetro no está presente tiene que ser restablecido el valor predeterminado que corresponde a 1024 pasos.

Por ejemplo, para un eje con resolución de 1000 impulsos/mm los valores admitidos serán incluidos entre 0,128 y 16,384 mm.

Si el parámetro **valor** es programado a cero es desactivado el control del límite de enlace encoder incorrecto.

Ejemplo

; programado un límite de enlace encoder de 3.5

SETENCLIMIT X, 3.5

SETINDEXEN

Sintaxis

SETINDEXEN **eje, estado**

Argumentos

eje nombre de dispositivo tipo eje
estado constante predefinida. Los valores admitidos son:
ON estado muesca de cero activo
OFF estado muesca de cero desactivo

Descripción

Activa o desactiva en el **eje** indicado la puesta a cero de la cuota en correspondencia de la muesca de cero. Para poder ejecutar esta instrucción el eje tiene que ser de tipo cuenta.

SETINTEGTIME

Sintaxis

SETINTEGTIME **eje [, valor]**

Argumentos

eje nombre del dispositivo tipo eje
valor constante integer o variable

Descripción

Programa el número de muestreos del error de anillo utilizados para el cálculo de la componente integral. Los valores válidos son entre 1 y 200. La variación de dicho parámetro al vuelo es posible pero puede causar unas discontinuidades en la referencia de velocidad del eje. Por lo tanto se recomienda utilizar este parámetro con ejes detenidos y deshabilitados o preferiblemente en free.

SETIRMPP

Sintaxis

SETIRMPP **eje, velocidad**

Argumentos

eje nombre del dispositivo tipo eje
velocidad constante float o variable float. Velocidad de inicio rampa

Descripción

Asigna al **eje** el valor de **velocidad de inicio rampa**. Es la velocidad mínima del motor paso-paso. Esta instrucción es utilizada para ejes movimentados con motores paso-paso.

SETLIMNEG**Sintaxis****SETLIMNEG** **eje [, cota]****Argumentos****eje** nombre del dispositivo tipo eje
cota constante o variable. Límite negativo**Descripción**Asigna al **eje** la **cota** límite negativo.Si el parámetro **cota** está omitido, se habilita el límite negativo de configuración.

Estas instrucciones son utilizadas de sólo en la rutinas de puesta a cero para la búsqueda de los interruptores de puesta a cero, permitiendo de esta manera a los ejes superar los valores de configuración configurados.

Véase también las instrucciones [RESLIMNEG](#), [SETLIMPOS](#), [RESLIMPOS](#).**Ejemplo**[Rutina de puesta a cero de un eje](#)**SETLIMPOS****Sintaxis****SETLIMPOS** **eje [, cota]****Argumentos****eje** nombre del dispositivo tipo eje
cota constante o variable. Límite positivo**Descripción**Asigna al **eje** la **cota** límite positivo.Si el parámetro **cota** está omitido, se habilita el límite positivo de configuración.

Estas instrucciones son utilizadas de sólo en la rutinas de puesta a cero para la búsqueda de los interruptores de puesta a cero, permitiendo de esta manera a los ejes superar los valores de configuración configurados.

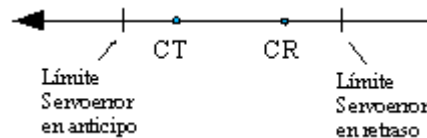
Véase también las instrucciones [RESLIMNEG](#), [RESLIMPOS](#), [SETLIMNEG](#).**Ejemplo**[Rutina de puesta a cero de un eje](#)**SETMAXER****Sintaxis****SETMAXER** **eje, valor [, dirección]****Argumentos****eje** nombre del dispositivo tipo eje
valor constante o variable. Máximo error de persecución
dirección constante predefinida. Dirección eje. Puede asumir los siguientes valores:
POSITIVE, NEGATIVE**Descripción**Asigna al **eje**:

- el **valor** máximo de persecución tolerado en el caso de test estático del servoerror
- el **valor** que sumado al error teórico proporcional a la velocidad determina el valor máximo de persecución tolerado en el caso de test dinámico del servoerror.

Si la **dirección** está omitida, el valor máximo de persecución se configura para ambas direcciones.**SETMAXERNEG****Sintaxis****SETMAXERNEG** **eje, retraso, anticipo****Argumentos****eje** nombre del dispositivo tipo eje
retraso constante o variable. Máximo error de persecución
anticipo constante o variable. Máximo error de anticipo**Descripción**Asigna al **eje** los valores máximos de error de persecución (**retraso**) y de anticipo (**anticipo**) admitidos por el control en la sola dirección negativa antes de generar la señalización de "servoerror".

- **retraso**: es el valor máximo de persecución tolerado en caso de test estático del servoerror o en el caso de test dinámico del servoerror; es el valor que sumado al error teórico proporcional a la velocidad determina el valor máximo de persecución tolerado.
- **anticipo**: es el valor máximo de persecución tolerado durante la inversión de movimiento desde movimiento negativo a movimiento positivo.

El error de persecución es la diferencia entre la cota teórica (en la cual el eje debería de encontrarse) y la cota real. Cuando el eje se mueve en dirección negativa un error de persecución con signo negativo indica una condición de retraso del eje, mientras un error de persecución con signo positivo indica una condición de anticipo. Si no se utiliza esta instrucción el control utiliza los valores máximos de error de persecución presentes en configuración eje, en tal caso el límite de anticipo es de 1/4 del límite de retraso.



Ejemplo

SETMAXERNEG Ejes.X, 10, 5

; El retraso máximo del eje es de 10mm, el anticipo máximo de 5mm

SETMAXERPOS

Sintaxis

SETMAXERPOS eje, retraso, anticipo

Argumentos

eje	nombre del dispositivo tipo eje
retraso	constante o variable. Máximo error de persecución
anticipo	constante o variable. Máximo error de anticipo

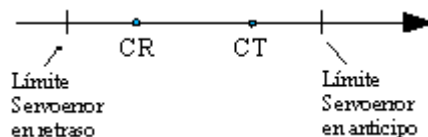
Descripción

Asigna al **eje** los valores máximos de error de persecución (**retraso**) y de anticipo (**anticipo**) admitidos por el control en la sola dirección positiva antes de generar la señalización de "servoerror".

- **retraso**: es el valor máximo de persecución tolerado en caso de test estático del servoerror o en el caso de test dinámico del servoerror; es el valor que sumado al error teórico proporcional a la velocidad determina el valor máximo de persecución tolerado.
- **anticipo**: es el valor máximo de persecución tolerado durante la inversión de movimiento desde movimiento negativo a movimiento positivo.

El error de persecución es la diferencia entre la cota teórica (en la cual el eje debería de encontrarse) y la cota real. Cuando el eje se mueve en dirección positiva un error de persecución con signo positivo indica una condición de retraso del eje, mientras un error de persecución con signo negativo indica una condición de anticipo.

Si no se utiliza esta instrucción el control utiliza los valores máximos de error de persecución presentes en configuración eje, en tal caso el límite de anticipo es de 1/4 del límite de retraso.



Ejemplo

SETMAXERPOS Ejes.X, 10, 5

; el retraso máximo del eje es de 10mm, el anticipo máximo de 5mm

SETMAXERTYPE

Sintaxis

SETMAXERTYPE eje, tipo

Argumentos

eje nombre de dispositivo tipo eje
tipo constante tipo integer. Los valores admitidos son:
 0 = programa servoerror de umbral (valor predeterminado)
 1 = programa servoerror dinámico

Descripción

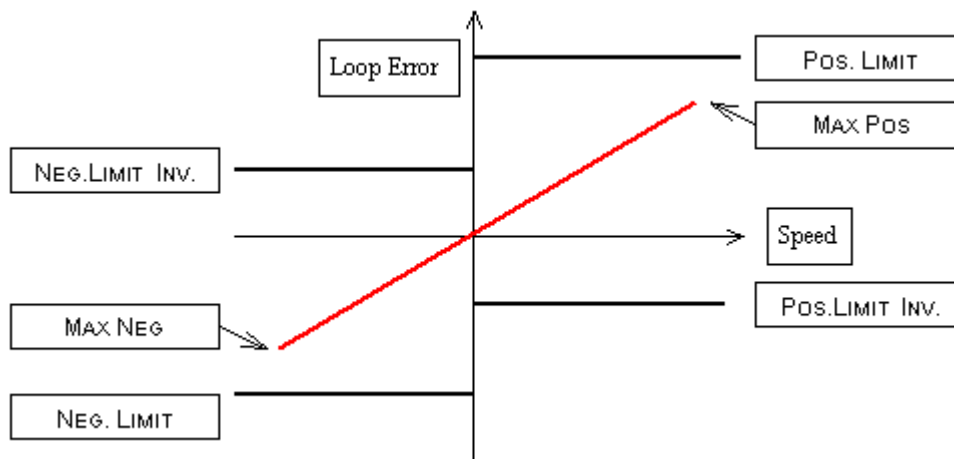
Esta instrucción permite programar el **tipo** de test en el servoerror. La gestión tradicional del servoerror prevé una pareja de límites (positivo y negativo) constantes al variar de la velocidad del eje. Este tipo de gestión lleva a dimensionar los límites en función de la velocidad máxima del eje, o sea se programa un límite de manera de no dejar activar el error en condiciones de normal funcionamiento. Más a baja velocidad el error de anillo tiene valores generalmente mucho más bajos respecto al limite programado, esto comporta un retardo en la identificación de una condición de error.

La gestión del servoerror en la ventadas se basa sobre cálculo del error de anillo teórico. Los límites de servoerror positivo y negativo son calculados en función de este sumando y restandoles un valor de umbral. Si el error de anillo real sale de este umbral se genera el error de servoerror.

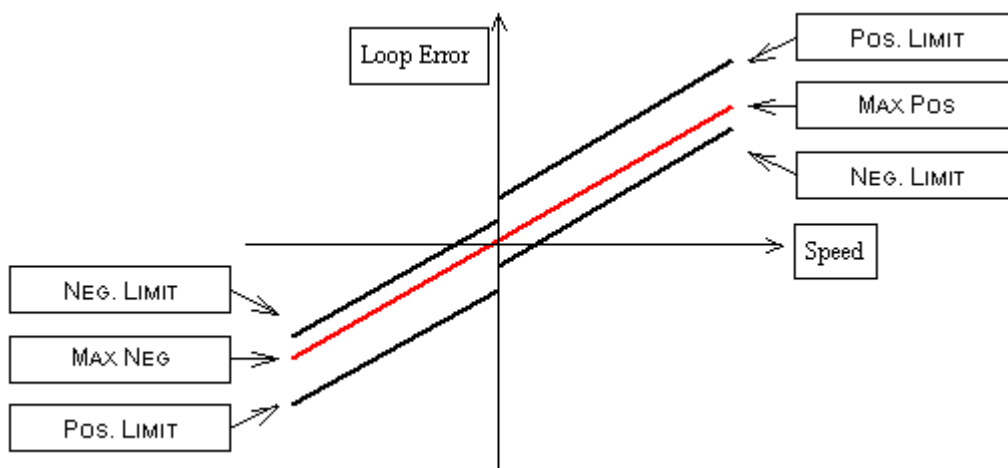
Atención

Si se programa el test en el servoerror dinámico es generalmente necesario modificar los valores de Límite servoerror positivo y Límite servoerror negativo programados en Configuración ejes para el servoerror en umbral. Esto porque dichos valores son utilizados como valores iniciales para el cálculo del error de anillo.

Límite ServoError a umbral:



Límite ServoError dinámico:



SETPHASESINV

Sintaxis

SETPHASESINV eje, estado

Argumentos

eje nombre de dispositivo tipo eje
estado constante predefinida. Los valores admitidos son:
ON estado inversión fases activo
OFF estado inversión fases desactivo

Descripción

Activa o desactiva en el **eje** indicado la inversión fases. Permite compensar vía software una eventual inversión de cableo de las fases encoder. Si utilizado junto con la inversión referencia permite invertir el verso del eje (si el cableo está correcto).
 Para poder ejecutar esta instrucción el eje debe de estar en el estado FREE.

SETREFINV**Sintaxis**

SETREFINV **eje, estado**

Argumentos

eje nombre de dispositivo tipo eje
estado constante predefinida. Los valores admitidos son:
ON estado inversión referencia de velocidad activa
OFF estado inversión referencia de velocidad desactiva

Descripción

Activa o desactiva en el eje indicado la inversión de la referencia de velocidad. Si utilizado junto con la inversión fases, permite invertir el verso del eje (si el cableo está correcto).
 Para poder ejecutar esta instrucción el eje debe de estar en el estado FREE.
 Véase también [SETPHASESINV](#).

SETRESOLUTION**Sintaxis**

SETRESOLUTION **eje [, valor]**

Argumentos

eje nombre de dispositivo tipo eje
valor constante o variable doble

Descripción

Cambia la resolución del eje especificado. Si **valor** está omitido, se usa el valor de resolución establecido en configuración. El valor de resolución se puede modificar solo si el eje está parado (estado eje = coordenada), por lo demás se genera el error de sistema n. 4101 "Gestión incoherente del eje".

10.3.5 Contadores**DECOUNTER****Sintaxis**

DECOUNTER **nombrecounter [, valor]**

Argumentos

nombrecounter nombre de dispositivo tipo contador
valor constante o variable o dispositivo tipo contador

Descripción

Incrementa el contador **nombrecounter** del **valor** especificado. Si **valor** está omitido, se asume el valor 1. Véase también las instrucciones [SETCOUNTER](#) y [INCOUNTER](#).

INCOUNTER**Sintaxis**

INCOUNTER **nombrecontador [, valor]**

Argumentos

nombrecontador nombre de dispositivo tipo contador
valor constante o variable o dispositivo tipo contador

Descripción

Incrementa el contador **nombrecontador** del **valor** especificado. Si **valor** está omitido, se asume el valor 1. Véase también las instrucciones [SETCOUNTER](#) y [DECOUNTER](#).

SETCOUNTER

Sintaxis

SETCOUNTER **nombrecontador, valor**

Argumentos

nombrecontador nombre di dispositivo tipo contador
valor constante o variable o dispositivo tipo contador

Descripción

Programa el contador **nombrecounter** al **valor** especificado.
 Véase también [INCOUNTER](#) y [DECOUNTER](#).

10.3.6 Temporizadores

HOLDTIMER

Sintaxis

HOLDTIMER **nombretemporizador**

Argumentos

nombretemporizador nombre de dispositivo tipo temporizador

Descripción

Bloquea la actualización del temporizador **nombretemporizador**.
 Véase también [STARTTIMER](#) y [SETTIMER](#).

SETTIMER

Sintaxis

SETTIMER **nombretemporizador, tiempo**

Argumentos

nombretemporizador nombre de dispositivo tipo temporizador
tiempo constante o variable o dispositivo tipo temporizador

Descripción

Programa **nombretemporizador** al **tiempo** especificado (en segundos).
 Son admitidos solo valores mayores de cero. La precisión máxima de los temporizador es de 4 ms.
 Véase también [STARTTIMER](#) y [HOLDTIMER](#).

Ejemplo

```
; La función programa un temporizador
; Programo el temporizador
; al valor 20 segundos
STARTTIMER T_espera,20
; inicia el temporizador en modalidad decreciente. Cuando llega a 0 se
para
```

```
STARTTIMER T_espera,DOWN
```

STARTTIMER

Sintaxis

STARTTIMER **nombretemporizador [, dirección]**

Argumentos

nombretemporizador nombre de dispositivo tipo temporizador
dirección constante predefinida. Los valores configurables son:
UP creciente
DOWN decreciente

Descripción

Hace partir el temporizador **nombretemporizador** con modalidad eventualmente especificada por **dirección**.
 Si el parámetro **dirección** está omitido, se asume la modalidad **DOWN**.
 Cuando un temporizador (iniciado con modalidad decreciente) llega a cero se detiene automáticamente.
 Véase también [HOLDTIMER](#) y [SETTIMER](#).

10.3.7 Variables, Vectores y Matrices

CLEAR

Sintaxis

CLEAR **nombrevar o vector o matriz[líneamatriz]**

Argumentos

nombrevar nombre variable
vector nombre vector
matriz nombre matriz
líneamatriz constante o variable o contador. Línea de la matriz

Descripción

Inicializa al valor 0 el espacio de memoria reservado a variables (**nombrevar**), vectores (**vector**), matrices (**matriz**) o a los elementos de una línea de matriz.

FIND

Sintaxis

FIND **matriz, columna, límite_min, límite_max, valor, variable**
FIND **vector, límite_min, límite_max, valor, variable**

Argumentos

matriz nombre de la matriz. Matriz en la cual ejecutar la búsqueda
vector nombre del vector. Vector en el cual ejecutar la búsqueda
columna constante o variable entera o nombre contador. Número de columna de la matriz en la cual ejecutar la búsqueda
límite_min constante o variable. Índice mínimo del vector o de la matriz desde la cual empezar la búsqueda
límite_max constante o variable. Índice máximo del vector o de la matriz donde terminar la búsqueda
valor constante o variable. Valor por buscar
variable variable. Resultado de la búsqueda

Descripción

Ejecuta una búsqueda secuencial de un valor al interno de un **vector** o de una **columna** de la **matriz** y pone el índice del elemento en **variable**.
 Si el valor no ha sido encontrado, **variable** contendrá el valor -1.

FINDB

Sintaxis

FINDB **matriz, columna, límite_min, límite_max, valor, variable**
FINDB **vector, límite_min, límite_max, valor, variable**

Argumentos

matriz nombre de la matriz. Matriz en la cual ejecutar la búsqueda
vector nombre del vector. Vector en el cual ejecutar la búsqueda
columna constante o variable entera o nombre contador. Número de columna de la matriz en la cual ejecutar la búsqueda
límite_min constante o variable. Índice mínimo del vector o de la matriz desde la cual comenzar la búsqueda
límite_max constante o variable. Índice máximo del vector o de la matriz donde terminar la búsqueda
valor constante o variable. Valor por buscar
variable variable. Resultado de la búsqueda

Descripción

Ejecuta una búsqueda rápida de un valor al interno de un **vector** o de una **columna** de la **matriz** y pone el índice del elemento en **variable**. Para que la búsqueda tenga éxito es necesario que el **vector** o la **columna** de la **matriz** hayan sido ordenados precedentemente con la instrucción SORT según un ordenamiento creciente.
 Si el valor no ha sido encontrado, **variable** contendrá el valor -1.

LASTELEM

Sintaxis

LASTELEM **vector, elementosvet**
LASTELEM **matriz, líneasmat**

Argumentos

matriz	nombre de matriz
vector	nombre de vector
elementosvet	variable. Número de elementos del vector
líneasmat	variable. Número de líneas de la matriz

Descripción

Escribe en la variable **elementosvet** el número de elementos del **vector** o en la variable **líneasmat** el número de líneas de la **matriz**.

LOCAL**Sintaxis**

LOCAL	nombrevar AS tipo
LOCAL	vector[no. elementos] AS tipo
LOCAL	matriz[no. líneas] AS tipo, tipo, tipo, etc.
LOCAL	matriz[no. líneas] AS tipo:nombrecol1, tipo:nombrecol2, tipo:nombrecol3, etc.

Argumentos

nombrevar	nombre variable
[no. elementos]	variable o constante (argumento obligatorio). Número de elementos del vector
[no. líneas]	constante o variable (argumento obligatorio). Número de líneas de la matriz
tipo	char, integer (32 bit), float (32 bit), double (64 bit), string, temporizador
nombrecol1...nombrecolN	nombre columna. Etiqueta

Descripción

Declaración de una variable local. Antes de esta instrucción puede presentarse solo la instrucción PARAM que define los parámetros de la función. Para más informaciones sobre las variables locales, véase [Las variables locales](#).

MOVEMAT**Sintaxis**

MOVEMAT	nombrematfuente, nombrematdest
MOVEMAT	nombrematfuente[líneafuente], nombrematdest[líneadest]
MOVEMAT	nomematsorg[líneafuente], nomematdest[líneadest], numlíneas

Argumentos

nombrematfuente	nombre matriz fuente
líneafuente	número de línea de salida para el par de la matriz fuente (argumento obligatorio)
nombrematdest	nombre matriz destinataria
líneadest	número de línea de salida para el par en la matriz destinataria (argumento obligatorio)
numlíneas	número de líneas a copiar

Descripción

Copia el contenido de una entera matriz **nombrematfuente** en otra matriz **nombrematdest** o una o más líneas **numlíneas** de la línea de la matriz **nombrematfuente[líneafuente]** en la línea de la matriz **nombrematdest[líneadest]**. Si no se especifica el parámetro **numlíneas** se copia una sola línea. Las dos matrices deben de haber la misma tipología de estructura (número de columnas y tipo de dato de cada columna) y en el caso de matrices enteras también el mismo número de líneas. Se puede cambiar de sitio líneas de datos al interno de la misma matriz.

Ejemplo

```
Movemat    Mx1, Mx2           ; copia la matriz Mx1 en Mx2
; copia la fila 10 de la matriz Mx1 en la fila 3 de Mx2
Movemat    Mx1[10], Mx2[3]

; copia la fila 1 de la matriz Mx1 en la fila 7 de Mx1
Movemat    Mx1[1], Mx1[7]
; copia 6 filas a partir de la fila 2 de la matriz Mx1 en la matriz
; Mx2 a partir de la fila 8
Movemat    Mx1[2], Mx2[8],6
```

```

; copia 4 filas a partir de la fila 2
; de la matriz Mx1 en la misma matriz
; Mx1 a partir de la fila 10
Movemat      Mx1[2], Mx1[10],4

```

PARAM

Sintaxis

[PARAM]	nombrevar AS tipo
[PARAM]	vector[n° elementos] AS tipo
[PARAM]	matriz[n° líneas] AS tipo, tipo, etc.
[PARAM]	matriz[n° líneas] AS tipo:alias, tipo:alias, etc.

Argumentos

nombrevar	nombre variable
[n° elementos]	constante (argumento obligatorio)
[n° líneas]	constante (argumento obligatorio)
tipo	char, integer (32 bit), float (32 bit), double (64 bit), string

Descripción

Los parámetros se comportan como las variables locales (Véase [LOCAL](#)), más son inicializados por quien llama la función. La sintaxis para la declaración de los parámetros es la misma usada para las variables locales.

Los parámetros son pasados por valor o por referencia según el tipo. Véase "[Las funciones](#)".

Hay que declararlos antes de cualquier otra instrucción.

Para más informaciones, véase [Las variables locales](#).

SETVAL

Sintaxis

SETVAL	valor, nombrevar
---------------	-------------------------

Argumentos

valor	constante o variable o nombredispositivo
nombrevar	variable o nombredevice

Descripción

Asigna el **valor** especificado a la variable **nombrevar** o al i-ésimo elemento de un vector o de una matriz.

SORT

Sintaxis

SORT	matriz, columna [, orden], límite_min, límite_max
SORT	vector [,orden], límite_min, límite_max

Argumentos

matriz	nombre de la matriz
vector	nombre del vector
columna	constante o variable. Número de columna de la matriz
orden	constante predefinida. Indica la modalidad de ordenamiento
	Los valores admitidos son:
	UP ordenamiento creciente
	DOWN ordenamiento decreciente
límite_min	constante o variable. Índice mínimo del vector o de la matriz desde la cual iniciar el ordenamiento
límite_max	constante o variable. Índice máximo del vector o de la matriz en donde terminar el ordenamiento

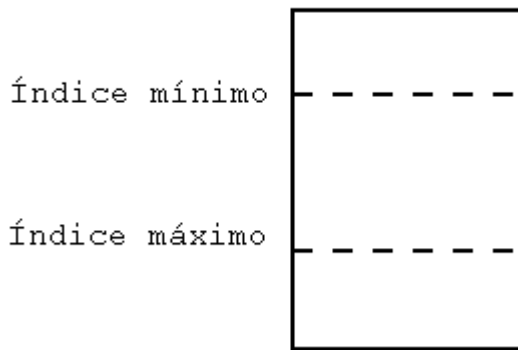
Descripción

Efectúa un ordenamiento de los valores presentes al interno de un **vector** o de una **matriz**, con sentido definido por la constante **orden**.

En el caso de una matriz el ordenamiento de las líneas es dictado por la disposición creciente (UP) o decreciente (DOWN) de los valores contenidos en la **columna** seleccionada.

Si el argumento **orden** está omitido, se asume automáticamente la modalidad UP.

Matriz



10.3.8 Cadenas

ADDSTRING

Sintaxis

ADDSTRING **nombrecadena1, nombrecadena2, nombrecadena3**

Argumentos

nombrecadena1 constante cadena o variable cadena. Cadena fuente
nombrecadena2 constante cadena o variable cadena. Cadena de agregar
nombrecadena3 variable cadena. Cadena resultado

Descripción

Concatenación de dos cadenas.
 Agrega a la cadena identificada por **nombrecadena1** la cadena identificada por **nombrecadena2** y pone el resultado en la cadena identificada por **nombrecadena3**.
 La máxima dimensión de una cadena es de 255 caracteres + el terminador, por lo tanto el resultado del enlace de las primeras dos cadenas no tendrá que superar este límite.

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#)

CONTROLCHAR

Sintaxis

CONTROLCHAR **valor, nombrecadena**

Argumentos

valor constante char o integer o variable char o integer. Valor que convertir
nombrecadena variable cadena. Cadena resultado

Descripción

Convierte en carácter ASCII el valor identificado por **valor** y pone el resultado en la cadena **nombrecadena** (en practica en el primer byte).
 El precedente contenido de la cadena se pierde. Esta instrucción es útil si hay que insertar en una cadena unos caracteres de control o que no pueden ser imprimidos (por ejemplo, el carácter NULL = 0x00).
 Acepta cadenas de por lo menos 2 caracteres: 1 carácter + el terminador. Si la cadena es de solo un carácter array[1] as char es señalado un error de sistema "Argumento macro incorrecto".

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#)

LEFT

Sintaxis

LEFT **nombrecadenasorg, numcaracteres, nombrecadenadest**

Argumentos

nombrecadenasorg constante cadena o variable cadena. Cadena fuente
numcaracteres constante o variable. Número de caracteres que copiar
nombrecadenadest variable cadena. Cadena destinación

Descripción

Copia los primeros **numcaracteres** de la cadena **nombrecadenasorg** en la cadena **nombrecadenadest**.

En practica está cogida la parte a la izquierda de la cadena fuente. Véase también las instrucciones [MID](#) y [RIGHT](#).

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#)

LEN

Sintaxis

LEN **nombrecadena, variable**

Argumentos

nombrecadena variable cadena. Cadena
variable variable

Descripción

Calcula el número de caracteres contenidos en la cadena **nombrecadena** (a exclusión del terminador) y lo pone en **variable**.

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#)

MID

Sintaxis

MID **nombrecadenasorg, primercar [, numcaracteres], nombrecadenadest**

Argumentos

nombrecadenasorg constante cadena o variable cadena. Cadena fuente
numcaracteres constante o variable. Número de caracteres por copiar
nombrecadenadest variable cadena. Cadena destinación
primercar constante o variable. Posición carácter partencia copia

Descripción

Extrae de la cadena identificada por **nombrecadenasorg** un número de caracteres identificados por **numcaracteres**, a partir de la posición **primercar**.

La subcadena extraída es puesta en la cadena identificada por **nombrecadenadest**.

Si **numcaracteres** es omitido, es copiada **cadenasorg** a partir de la posición **primercar** hasta el final de la misma. En practica cogida la parte central de la cadena fuente. Véase también las instrucciones [LEFT](#) y [RIGHT](#).

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#)

RIGHT

Sintaxis

RIGHT **nombrecadenasorg, numcaracteres, nombrecadenadest**

Argumentos

nombrecadenasorg constante cadena o variable cadena. Cadena fuente
numcaracteres constante o variable. Número de caracteres por copiar
nombrecadenadest variable cadena. Cadena destinación

Descripción

Copia los últimos **numcaracteres** desde la cadena **nombrecadenasorg** en la cadena **nombrecadenadest**.

En practica es prelevada la parte derecha de la cadena fuente. Véase también las instrucciones [LEFT](#) y [MID](#).

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#)

SEARCH

Sintaxis

SEARCH **nombrecadena, carácter, variable**

Argumentos

nombrecadena variable cadena

carácter constante char o constante cadena o variable cadena. Carácter o cadena de buscar

variable variable

Descripción

Busca la posición del carácter ASCII identificado por **carácter** (que puede ser también una cadena) al interno de la cadena **nombrecadena** y pone el índice del resultado en la **variable**. Si **carácter** no ha sido encontrado, **variable** contendrá el valor -1.

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#).

SETSTRING**Sintaxis**

SETSTRING "valor", nombrecadena

Argumentos

valor constante cadena o variable cadena (encerrada entre comillas)

nombrecadena cadena destinación

Descripción

Copia de una cadena. Ejecuta una copia de los caracteres ASCII presentes en la cadena identificada por "**valor**" en la cadena identificada por **nombrecadena**. Para insertar caracteres no imprimibles en una cadena, véase la instrucción [CONTROLCHAR](#).

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#).

STR**Sintaxis**

STR valor, nombrecadena

Argumentos

valor constante o variable. Valor fuente de convertir

nombrecadena variable cadena. Cadena destinataria

Descripción

Convierte en caracteres ASCII el **valor** y pone el resultado en la cadena **nombrecadena**. Puede ser utilizada para transformar una variable de tipo integer en una cadena. Por ejemplo, el número 10 es transformado en la cadena "10".

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#).

VAL**Sintaxis**

VAL nombrecadena, resultado

Argumentos

nombrecadena variable cadena. Cadena de convertir

resultado variable. Cadena transformada

Descripción

Transforma el contenido de la cadena **nombrecadena** en un número decimal y pone el resultado en la **variable**. Por ejemplo, la cadena "123" es transformada en el valor 123.

Ejemplo

Véase el ejemplo [Operaciones en las cadenas](#).

10.3.9 Comunicaciones**CLEARRECEIVE****Sintaxis**

CLEARRECEIVE

Argumentos

Ningún argumento

Descripción

Vacía la lista de RECEIVE ejecutadas más aún no satisfechas.

COMCLEARRXBUFFER**Sintaxis**

COMCLEARRXBUFFER **númeroCOM**

Argumentos

númeroCOM

constante predefinida. Número de la puerta serial. Los valores que pueden ser asignados son: desde **COM1** a **COM8**.

Descripción

La instrucción vacía el búfer de recepción de la serial **númeroCOM**. Todos los datos, eventualmente presentes, son borrados.

COMCLOSE**Sintaxis**

COMCLOSE **númeroCOM**

Argumentos

númeroCOM

constante predefinida. Número de la puerta serial. Los valores que pueden ser asignados son: desde **COM1** hasta **COM8**.

Descripción

Cierra la línea serial **NúmeroCOM** abierta por una **COMOPEN**. Es necesario cerrar la línea serial también cuando la tarea que ha abierto una puerta serial es terminada por cualquier motivo.

COMGETERROR**Sintaxis**

COMGETERROR **númeroCOM, variable**

Argumentos

númeroCOM

constante predefinida. Número de la puerta serial. Los valores que pueden ser asignados son: desde **COM1** hasta **COM8**.

variable

variable integer. Resultado de la última operación ejecutada en la serial

Descripción

La instrucción lee el código de retorno de la última instrucción de comunicación serial llamada en la puerta **númeroCOM**. Con esta instrucción es posible saber si una operación de lectura o de escritura tubo éxito e, en caso negativo, también el código de error retornado.

En seguida están enumerados los códigos de error:

Retorno normal	0
Búfer de transmisión lleno	2
Dispositivo ya abierto	3
Puerta no valida o no configurada	6
Habilitación puerta E/S fallada	7
Imposible conectersas al interrupt	8
Puerta serial (com) todavía no abierta	9
El dispositivo serial (com) está ocupado	12
No es posible la conexión a RTX	14

COMGETRXCOUNT**Sintaxis**

COMGETRXCOUNT **númeroCOM, númchar**

Argumentos

númeroCOM

constante predefinida. Número de la puerta serial. Los valores que pueden ser asignados son: desde **COM1** a **COM8**.

númchar

número de caracteres presentes en el búfer

Descripción

La instrucción retorna el número de caracteres presentes en el búfer de recepción. Permite saber si han sido recibidos caracteres desde la puerta serial.

COMOPEN

Sintaxis

COMOPEN **númeroCOM, baudrate, dimensiónpalabra, stopbits, paridad**

Argumentos

númeroCOM constante predefinida. Número de la puerta serial. Los valores que pueden ser asignados son: desde **COM1** hasta **COM8**.

baudrate baudrate (velocidad en baudios) de comunicación. Los valores asignables son: 2400, 4800, 9600, 19200, 38400, 57600, 115200

dimensiónpalabra dimensión de la palabra de datos. Los valores asignables son: 5, 6, 7, 8

stopbits bit de paro. Los valores asignables son: 1, 2

paridad constante predefinida. Paridad. Los valores que pueden ser asignados son: **NOPARITY, ODDPARITY** y **EVENPARITY**

Descripción

Abre una línea serial. Hay que ejecutarla antes de cualquier otra instrucción que gestione la línea serial. En el caso en que sea ejecutada otra instrucción relativa a la misma línea serial antes de COMOPEN, será generado un error de sistema. Los parámetros pasados deben de ser valores comprendidos entre los que están indicados arriba.
 El canal de comunicación con línea serial está asociado a la tarea que ha ejecutado la instrucción COMOPEN. Si la tarea termina el canal de comunicación es cerrado de manera automática.
 Véase también [COMCLOSE](#), [COMREAD](#), [COMWRITE](#), [COMREADSTRING](#), [COMWRITESTRING](#).

Atención

El número de líneas de serie disponibles no depende del entorno hardware del control numérico (véase la documentación). En el entorno RTX solo COM1 y COM2 son disponibles.

COMREAD

Sintaxis

COMREAD **númeroCOM, búfer, númcharqueleer, númcharleídos [,tiempodeespera]**

Argumentos

númeroCOM constante predefinida. Número de la puerta serial. Los valores que pueden ser asignados son: desde **COM1** hasta **COM8**

búfer vector de char. Es el vector donde se guardan los caracteres leídos

númcharqueleer número de caracteres que se desea leer desde la línea serial

númcharleídos número de caracteres efectivamente leídos

tiempodeespera tiempo de espera (en segundos)

Descripción

La instrucción lee caracteres desde la serial **NúmeroCOM**. Los caracteres leídos son memorizados en la variable **búfer**. El campo **númcharqueleer** indica el número de caracteres que la instrucción tendrá que leer. Si en el búfer de recepción de la serial hay menos caracteres y el parámetro **tiempodeespera** no está especificado, la instrucción terminará de inmediato especificando en el parámetro **númcharleídos** el número de caracteres efectivamente leídos. Si el parámetro **tiempodeespera** está especificado, entonces la instrucción tendrá que esperar al máximo el número de segundos indicado en la variable en espera de la llegada de otros caracteres. Si el **tiempodeespera** termina, entonces la instrucción saldrá, siempre especificando en **númcharleídos** el número de caracteres efectivamente copiados en **búfer**.

COMREADSTRING

Sintaxis

COMREADSTRING **númeroCOM, búfer, númcharleídos [,terminador [,tiempodeespera]]**

Argumentos

númeroCOM constante predefinida. Número de la puerta serial. Los valores que pueden ser asignados son: desde **COM1** a **COM8**.

búfer vector de char. Es el vector en donde son depositados los datos.

númcharleídos número de caracteres efectivamente leídos

terminador carácter de terminacion transmisión

tiempodeespera tiempo de espera de espera (en segundos)

Descripción

La instrucción lee unos caracteres desde la serial **NúmeroCOM**. Se diferencia con la [COMREAD](#) porque lee desde la serial hasta que no encuentra el carácter terminador. Los caracteres leídos son memorizados en la variable **búfer**. Esta variable debe de ser de tipo vector de char. El campo **númcharleídos** indica el número de caracteres que la instrucción ha efectivamente leído desde la línea serial y por lo tanto copiado en el **búfer**. El parámetro **terminador** indica el carácter que servirá de terminador de la transmisión. En

practica la instrucción tendrá que leer caracteres desde la serial hasta que no encontrará un carácter igual al que está especificado en este parámetro. Este parámetro es opcional. Si no está se entiende que el carácter de terminación es cero. El carácter cero no será copiado en el búfer pasado por parámetro, mientras un eventual carácter de terminación especificado en la instrucción será copiado en el búfer. El **tiempodeespera** es otro parámetro que indica cuantos segundos la instrucción debe de esperar que lleguen nuevos caracteres si ha vaciado el búfer de recepción sin haber encontrado el carácter de terminación. Si el parámetro **tiempodeespera** no está especificado, entonces la instrucción terminará apenas vaciado el búfer de recepción.

COMWRITE

Sintaxis

COMWRITE **númeroCOM, búfer, queescribir**

Argumentos

númeroCOM constante predefinida. Número de la puerta serial. Los valores que pueden ser asignados son: desde **COM1** hasta **COM8**.

búfer vector de char. Es el vector que contiene los datos que escribir

queescribir número de caracteres que escribir

Descripción

La instrucción escribe los caracteres presentes en la variable **búfer** en la línea serial **númeroCOM**. En el parámetro **queescribir** será especificado el número de caracteres que efectivamente hay que escribir.

COMWRITESTRING

Sintaxis

COMWRITESTRING **númeroCOM, búfer [,terminador]**

Argumentos

númeroCOM constante predefinida. Número de la puerta serial. Los valores que pueden ser asignados son: desde **COM1** a **COM8**.

búfer vector de char. Es el vector que contiene los datos que escribir

terminador carácter de terminación transmisión

Descripción

La instrucción escribe los caracteres presentes en la variable búfer en la línea serial **númeroCOM**. Se distingue de la [COMWRITE](#) porque escribe en la serial hasta que no encuentra el carácter **terminador**. El parámetro **terminador** es opcional. Si no está especificado, la instrucción transmitirá hasta que encuentre un carácter cero. El cero no será transmitido, mientras un eventual carácter de control especificado sí.

RECEIVE

Sintaxis

RECEIVE **[fuente,] identificador, indicadores [, contenedor]**

Argumentos

fuentes constante de tipo cadena

identificador constante de tipo cadena

indicadores constante de tipo integer

contenedor nombre de dispositivo o variable (numérica o cadena)

Descripción

Esta instrucción, junto con la [SEND](#) está utilizada para intercambiar informaciones entre los módulos de la instalación y el ordenador supervisor. La [SEND](#) es utilizada para enviar informaciones, la [RECEIVE](#) para requerirlas. Las informaciones pueden ser requeridas a Albatros o a un programa externo (Servidor OLE Automation). En el segundo caso el pedido será de todas formas recibido por Albatros que se ocupará de enviarlo al programa externo.

El parámetro **fuentes** es una cadena que permite especificar a quien es dirigido el pedido de información. Hay tres clases de destinatarios:

- fuentes que comienzan con el carácter "@" (Véase la lista más adelante). La fuente es en realidad Albatros o más bien dicho una de sus funciones.
- fuentes que no comienzan con el carácter "@". Están considerados como Servidores OLE, cuando reciben el primer pedido de información dirigida a ellos, Albatros intentará la manera de mandarlos en ejecución y, por lo tanto, de pasar a estos el pedido de información recibido por el módulo.
- fuente no especificada (el parámetro de hecho es opcional). En este caso la información es leída en una tabla conservada por Albatros. Si la información no está presente en la tabla el pedido queda pendiente y será cumplido apenas la información será disponible (proveído por otro módulo o un programa externo).

El parámetro **identificador** es el nombre de la información pedida, no puede ser omitido. Asume diferentes expresados según la fuente:

- si la fuente es Albatros será un comando relacionado a la función a la cual se está accediendo.
- si la fuente es un Servidor OLE será una propiedad del objeto OLE instanciado.
- si la fuente no está especificada será la etiqueta que identifica la información al interior de la tabla conservada por Albatros.

El parámetro **indicadores** permite especificar como la información pedida se debe tratar por Albatros. Lo valores admitidos y los efectos que comportan son los siguientes:

valor	comando	descripción
\$0008H	CancelAfter	La información será cancelada después de haber sido leída
\$0800H	UpdateFlags	Modifica el estado de la información (leída / de leer) sin modificar los datos
\$8000H	Delete	Cancela la información

El parámetro **contenedor** es, al final, la variable (o el dispositivo) donde será almacenada la información que se está solicitando. Está puede ser omitida, en tal caso se está solicitando la notifica de un evento (puede ser utilizada para sincronizar la ejecución del código GPL en módulos diferentes).

Lista de las **fuentes** gestionadas por Albatros y de los comandos relativos:

"@List"

Permite la gestión de los comandos Simulado y Setpoint.

Se admiten los siguientes comandos (parámetro **identificador**):

- Sim,0,contenedor: requiere el estado del botón Simulado, que es escrito en el flagSwitch Simulado. La variable de retorno **contenedor** vale 1 si no han sido descubiertos errores, si no vale 0.
- Setp,0,contenedor: requiere el estado del botón Setpoint, que es escrito en el flagSwitch CmdSetP. La variable de retorno **contenedor** vale 1 si no han sido descubiertos errores, si no vale 0.
- Esc,0,requiere el estado del pulsador Setpoint, que es escrito en el flagSwitch Escluso.. La variable de retorno **contenedor** vale 1 si no han sido descubiertos errores, si no vale 0.

"@Environ"

Permite obtener informaciones sobre el estado del sistema: [nivel de acceso](#) del usuario, módulos conectados al supervisor etc. La información solicitada se almacena en le parámetro **contenedor**. Los valores admitidos para el parámetro **identificador** y las correspondientes respuestas son:

- "AccessLev nivel de acceso al sistema 0 = usuario, 1 = asistencia, 2 = constructor, 3 = tpa el"
- "MaskConf máscara de los módulos configurados Modules"
- "MaskActiv máscara de los módulos conectados eModules"
- "CurrentMo módulo desde el cual proviene el pedido dule"
- "mod:Nam nombre del ordenador correspondiente al módulo "mod". (mod comprendido entre 0 y ePC" 15)
- "LocalDate Time"

El parámetro **contenedor** recibirá la fecha y la hora del PC en un formato que está relacionado con su tipo:

- char: número del día de la semana
- integer: número de segundos desde el 1/1/1970
- float: número de días y fracciones de un día desde el 1/1/1900
- Doble: número de días y fracciones de un día desde el 1/1/1900
- string: texto "AAAA/MM/DD hh:mm:ss"

Las máscaras de los módulos conectados y configurados son máscaras de bit. El bit de menor peso corresponde al módulo 0. El bit de cada módulo será en 1 si el módulo está conectado o configurado. En el caso de "NamePC" el número de módulo es facultativo, si omitido se asume el número del módulo desde el cual proviene el solicitado.

"@Syn"

Comunicación entre GPL y visualizador de los cuadros sinópticos. Permite la apertura y cierre de cuadros sinópticos mandados por GPL y el pedido de informaciones de una casilla de un sinóptico. Se admiten los siguientes comandos (parámetro **identificador**):

- "Open:*no* apertura del sinóptico *nombreachivo.xsyn mbreachivo o*"
- "Close:*no* cierre del sinóptico *nombreachivo.xsyn mbreachivo o*"
- "*nombrecasilla* casilla desde la cual es leída la información solicitada *silla*"

Es posible tener informaciones sobre la ventana de desplazamiento de los ejes según las especificaciones definidas también para el parámetro **fuentes** "[@Devices](#)" indicado a continuación.

"@FileName"

almacena una asociación entre una cadena constante y un nombre de archivo que se pueden componer con variables cadena. Desde Albatros recibe la comunicación de la asociación, reemplaza todos los siguientes nombres de archivo con el nombre recibido por medio de esta instrucción. El parámetro **identificador** es el nombre del archivo. El nombre del archivo es una variable cadena. Si en el parámetro **identificador** no se muestra la ruta de acceso completa en que archivar el archivo, Albatros considera ruta la que está definida en tpa.ini en la sección [tpa] bajo dirreport. En tpa.ini, en la sección [GPLFileName] bajo Log se almacena el valor del parámetro **identificador**, para que pueda utilizarse también en las ejecuciones de Albatros que siguen. Para eliminar la asociación hay que establecer como parámetro **identificador** una cadena vacía. La asociación así definida vale para todos los módulos.

"@FileDelete"

Cancelación de un archivo. El parámetro **identificador** es el nombre del archivo que se cancelará (ruta completa). Si en el parámetro **identificador** no se muestra la ruta de acceso completa en que archivar el archivo, Albatros considera ruta la que está definida en tpa.ini en la sección [tpa] bajo dirreport.. El nombre del archivo se puede definir según las reglas descritas en correspondencia del parámetro **fuentes** [@FileRead](#). El parámetro **contenedor** contendrá el valor:

- 1 si el archivo fue cancelado
- 0 en caso contrario

"@FileRead"

Lee el contenido de un archivo. El parámetro **identificador** es el nombre del archivo que será leído (ruta completa). Si en el parámetro **identificador** no se muestra la ruta de acceso completa en que archivar el archivo, Albatros considera ruta la que está definida en tpa.ini en la sección [tpa] bajo dirreport. Si el **identificador** comienza y termina con un carácter de %, la cadena en su interior es buscada en tpa.ini, en la sección [tpa] y usada como nombre del archivo. Al interno del nombre pueden ser insertados caracteres que serán sustituidos durante la ejecución de la instrucción:

- %n número del módulo que ejecuta la instrucción RECEIVE
- %h hora actual (formato 00-23)
- %d día actual (formato 01-31)
- %m mes actual (formato 01-12)
- %y año actual (formato a cuatro cifras)

Si el parámetro **contenedor** está definido como variable char contendrá un bit leído desde el archivo, si está definido como cadena contendrá una línea entera del archivo texto, si está definido como integer contendrá el número de bytes que faltan al final del archivo (0 = final del archivo).

Para posicionar el cursor al principio del archivo el parámetro **contenedor** tiene que estar omitido.

"@FileExist"

Verifica la existencia de un archivo. El parámetro **identificador** es el nombre del archivo que será leído (ruta completa). Si en el parámetro **identificador** no se muestra la ruta de acceso completa en que archivar el archivo, Albatros considera ruta la que está definida en tpa.ini en la sección [tpa] bajo dirreport. El nombre del archivo se puede definir según las reglas descritas en correspondencia del parámetro **fuentes** [@FileRead](#). El parámetro **contenedor** contendrá el valor:

- diferente de 0 si el archivo existe
- 0 si el archivo no existe

"@FileLastWrite"

Obtiene la fecha de la última modificación hecha en un archivo. El parámetro **identificador** es el nombre del archivo (ruta completa). Si en el parámetro **identificador** no se muestra la ruta completa donde poner el archivo, Albatros considera la ruta definida en tpa.ini en la sección [tpa] bajo dirreport. El nombre del archivo se puede definir según las reglas descritas en correspondencia del parámetro **fuentes** [@FileRead](#). El parámetro **contenedor** tendrá la fecha de la última modificación del archivo en un formato que está relacionado al tipo de parámetro:

- char: número del día de la semana
- integer: número de segundos desde el 1 enero 1970
- float: número de días y fracciones del día desde el 1 enero 1900
- double: número de días y fracciones del día desde el 1 enero 1900
- string: texto en el formato "AAAA/MM/GG hh:mm:ss"

"@FileInfo"

Lee unas informaciones de un archivo. El parámetro **identificador** tiene que ser expresado en el formato "propiedad:nombrearchivo", donde **propiedad** indica el nombre de la propiedad que leer y **nombrearchivo** es el nombre del archivo. El nombre del archivo puede ser configurado a través de "Name". El parámetro **contenedor** contendrá el dato leído desde el archivo.

Lista de las propiedades:

- "version": devuelve en contenedor un dato de tipo entero. Los cuatros números que identifican la versión se encuentran en los 4 bytes de la variable contenedor. Si ocurre un error, el valor de la variable contenedor es 0.
- "size": devuelve en contenedor un dato de tipo entero o flotante o doble. El dato es el tamaño del archivo. Si ocurre un error, el valor de la variable contenedor es -1.

"@Devices"

Solicitud de apertura o cierre de la ventana Diagnóstico relativa al módulo che envía la información. El parámetro **identificador** puede asumir los siguientes valores:

- "Open" apertura del Diagnóstico
- "Close" cierre del Diagnóstico

El parámetro **identificador**, cuando se quiere trabajar con la ventana de desplazamiento eje, puede asumir los siguientes valores:

"MoveAX#nombre_eje#HasFocus"	el parámetro contenedor contendrá 1 si la ventana de desplazamiento eje especificada está activa, si no contendrá 0.
"MoveAX#nombre_eje#Jog"	el parámetro contenedor contendrá 1 si está programado el movimiento para los desplazamientos gestionados en tiempo real por el operador, si no contendrá 0.
"MoveAX#nombre_eje#Step"	el parámetro contenedor contendrá 1 si está programado el movimiento para los desplazamientos de paso predefinido, si no contendrá 0.
"MoveAX#nombre_eje#Absolute"	el parámetro contenedor contendrá 1 si está programado el movimiento con desplazamiento a cota predefinida, si no contendrá 0.

donde "nombre_eje" representa el nombre del eje visualizado en la ventana. Por ejemplo, si se desea verificar si la ventana de desplazamiento eje X esté activa el parámetro **identificador** será "[@MoveAX#X#HasFocus](#)". El nombre del eje se puede presentar en una de las siguientes formas:

1. Nombre_Grupo.Nombre_Subgrupo.Nombre_Eje o Nombre_Grupo.Nombre_Eje: se ofrece la ruta completa del eje.
2. Nombre_Eje: para identificar el eje correcto son hechas en orden las verificaciones:
 - si la tarea desde la cual llega el comando es una función de subgrupo, el eje es buscado en ese subgrupo.
 - si la tarea desde la cual llega el comando es una función del subgrupo principal, el eje es buscado en todo el grupo. Si está presente más de un eje con ese nombre la búsqueda falla.
 - si las verificaciones precedentes han fallado, el eje es buscado en todos los grupos del módulo. Si está presente más de un eje con nombre Nombre_eje la búsqueda no tiene éxito positivo.

"@Vars"

Solicita la actualización de una variable global GPL. Permite efectuar una actualización de los datos de las Paramétricas tecnológicas y herramientas. Los datos de paramétrica se envían, normalmente, al GPL durante la inicialización de la máquina. El parámetro **identificador** toma el expresado del nombre de la variable global (de máquina o de grupo) por la cual se requiere la actualización. El parámetro **contenedor** tendrá el valor:

- 1 si la variable ha sido actualizada correctamente
- 0 en caso contrario

"@Application"

Interacción con Albatros. Permite visualizar unos "message box" en la pantalla y cerrar Albatros. Los valores admitidos por el parámetro **identificador** son:

"Quit"	cierra Albatros
"IsLocked"	verifica si la salida de Albatros está bloqueada. El parámetro contenedor tendrá 1 si la interfaz está bloqueada, 0 si es posible salir de Albatros.
"MsgBox"	lee la respuesta de un mensaje box abierto antes por una SEND

El parámetro **contenedor** permite saber, en el caso de un mensaje box, cual botón fue seleccionado por el operador:

- 1 botón "Aceptar"
- 2 botón "Cancelar"
- 4 botón "Reintentar"
- 6 botón "Sí"
- 7 botón "No"

En el caso del comando "Quit" el parámetro **contenedor** contendrá el valor:

- 1 si Albatros ha sido cerrado correctamente
- 0 en caso contrario

"@Param"

Permite conocer el número progresivo de memorización de los archivos de paramétrica Partec.xpar y Partool.xpar. La información pedida se memoriza en el parámetro **contenedor**. Los valores posibles para el parámetro **identificador** son:

- "partec" requiere el progresivo de memorización de partec.xpar
- "partool" requiere el progresivo de memorización de partool.xpar

"@Ini"

Lee una combinación de teclas=valor en el archivo tpa.ini. El parámetro **identificador** es el nombre de la llave que se debe leer en el archivo tpa.ini en la sección [Tpa]. Para leer desde una sección específica, al nombre de las teclas hay que agregar el nombre de la sección encerrado entre corchetes ("[Sección]Teclas").

"@ShellExecute"

solicita al sistema operativo que abra un archivo por medio de un programa asociado a la extensión del archivo. Es posible también iniciar un ejecutable. El parámetro **identificador** es el nombre del archivo que hay de abrir o el nombre del programa que hay de iniciar. Se puede declarar el nombre del archivo con un nombre de ruta completa, de lo contrario se busca en el directorio actual de Albatros. El nombre del archivo se busca también en los definidos por medio de "@FileName". El parámetro **contenedor** tendrá el valor 0 si no hay errores en la abertura del archivo, en caso contrario tendrá el código de error.

"@StartProg"

Ejecuta el programa definido en el parámetro **identificador**. No se puede pasar los argumentos al programa que hay de iniciar. El nombre del programa tiene que contener la ruta entera, de lo contrario se busca en el directorio actual de Albatros. El nombre del programa se busca también entre aquellos definidos por "@FileName". El parámetro **contenedor** tendrá el valor 0 si el inicio del programa tuvo éxito, en caso contrario tendrá el código de error. Si el programa ya fue iniciado, el código de error es 1056.

"@TermProg"

Acaba el programa definido en el parámetro **identificador** y iniciado por "@StartProg". El nombre del programa tiene que contener la ruta entera, de lo contrario se busca en el directorio actual de Albatros. El nombre del programa se busca también entre aquellos definidos por "@FileName". El parámetro **contenedor** tendrá el valor 0 si el inicio del programa tuvo éxito, en caso contrario tendrá el código de error. Si el programa ya fue iniciado, el código de error es 1056.

"@ProgRunning"

Verifica si el programa iniciado con "@StartProg" sigue ejecutándose. El nombre programa está definido en el parámetro **identificador**. El nombre del programa tiene que contener la ruta entera, de lo contrario se busca en el directorio actual de Albatros. El nombre del programa se busca también entre aquellos definidos por "@FileName". El parámetro **contenedor** tendrá valor 1 si el programa todavía está en ejecución, en caso contrario tendrá valor 0.

"@DialogFile"

Abre la ventana de diálogo de Archivo Abrir o de Archivo Guardar para escoger un nombre del archivo. Para abrir la ventana de "Abrir Archivo" configurar el parámetro **identificador** = "Open", para abrir la ventana de "Guardar archivo" configurar el parámetro **identificador** = "Save". El nombre del archivo elegido se memoriza en el parámetro **contenedor**.

"@AxisCorrectors"

Substituye la tabla de los correctores de linealidad de un eje con una nueva, cargada desde archivo, que de toda manera debe tener el mismo número de correctores y los mismos ejes para los correctores cruzados.

El parámetro **identificador** es el nombre del archivo, que típicamente tiene como extensión .csv y se encuentra en la carpeta \Mod.n\Config (el nombre del archivo es "normalizado", como ocurre, por ejemplo, con "@FileExist").

El parámetro **contenedor** es definido como variable entera y va a incluir 1, si los nuevos correctores han sido enviados, 0 de lo contrario.

"@Language"

Recibe el texto traducible correspondiente con un mensaje de grupo o biblioteca o módulo, asociado con una instrucción MESSAGE o una instrucción ERROR. Los valores admitidos para el parámetro **identificador** son:

- "DEFMSG:número", donde número es una secuencia de dígitos. Albatros escribe en **contenedor** el texto del mensaje de módulo número "número".
- "DEFMSG:nombre", donde "nombre" es el nombre, grupo y biblioteca incluidos, de una DEFMSG. Albatros escribe en **contenedor** el texto del mensaje indicado. Si falta el nombre del grupo o de la biblioteca, se usa el nombre de la tarea que ha enviado la RECEIVE.

- "DEFMSG:*", Albatros escribe en **contenedor** el texto del mensaje de grupo o de módulo previamente indicado con la instrucción SEND.

Ejemplo

```

; en GPL
RECEIVE "@Param", "partec", 0, prog
RECEIVE "@Param", "partool", 0, prog

; en GPL
; lee el valor de la llave Radix en la sección [Albatros] desde archivo
tpa.ini
RECEIVE "@INI", "[Albatros]Radix", 0, valor

; abre la ventana de "Abrir archivo" y memoriza el nombre del archivo en
la variable NombreArchivo
RECEIVE "@DialogFile", "Open", 0, NombreArchivo

; lectura completa de un archivo
Function ReadProperties
PARAM file AS STRING
LOCAL version AS INTEGER
LOCAL size AS DOUBLE

SEND "@FileName" "theFile" 0 file
WAITRECEIVE "@FileInfo", "version:theFile", 0, version
WAITRECEIVE "@FileInfo", "size:theFile", 0, size

```

SEND**Sintaxis**

SEND [destinatario,] **identificador**, **indicadores**, [**información**]

Argumentos

destinatario	constante de tipo cadena
identificador	constante de tipo cadena
indicadores	constante de tipo integer
información	nombre de dispositivo o constante o variable (numérica o cadena)

Descripción

Esta instrucción, junto con la RECEIVE, es utilizada para intercambiar informaciones entre los módulos de la instalación y el PC supervisor. La SEND es utilizada para enviar informaciones, la RECEIVE para requerirlas. Las informaciones pueden ser requeridas a Albatros o a un programa externo (Servidor OLE Automation). En el segundo caso el pedido será de todas formas recibido por Albatros que se ocupará de enviarlo al programa externo.

El parámetro **destinatario** es una cadena que permite especificar a quien es dirigida la información. Hay tres clases de destinatarios:

- destinatarios que comienzan con el carácter "@" (Véase la lista más adelante). El destinatario es en realidad Albatros o más bien dicho una de sus funciones.
- destinatarios que no comienzan con el carácter "@". Son considerados como Servidores OLE, en la recepción del primer pedido de información dirigida a ellos, Albatros tratará la manera de mandarlos en ejecución y por lo tanto de pasarles el pedido de información recibido por el módulo.
- destinatario no especificado (el parámetro de hecho es opcional). En este caso la información es conservada por Albatros en una tabla y queda disponible para quien la requiere (otro módulo o un programa externo).

El parámetro **identificador** es el nombre de la información, no puede ser omitido. Asume diversos significados en función del destinatario:

- si el destinatario es Albatros será un comando ligado a la función a la cual se está teniendo acceso.
- si el destinatario es un Servidor OLE será una propiedad del objeto OLE definido.
- si el destinatario no está especificado será la etiqueta que identifica la información al interno de la tabla conservada por Albatros.

El parámetro **indicadores** permite especificar como Albatros debe tratar la información. Lo valores admitidos y los efectos que comportan son los siguientes:

valor	comando	descripción
\$0001H	Broadcast	Normal envío de la información

\$0008H	CancelAfter	La información será cancelada después de haber sido leída
\$0020H	ReadOnly	La información puede ser cancelada solo por el remitente
\$1000H	UpdateFlags	Modifica el estado de la información (leída / de leer) sin modificar los datos
\$8000H	Delete	Cancela la información

El parámetro **información** es en fin la información que se está enviando. Esta puede ser omitida y en tal caso el envío de una información vacía asume el significado de una notifica de un evento (puede ser utilizada para sincronizar la ejecución del código GPL en módulos diferentes). Son admitidos, como parámetro información, todos los dispositivos (excluidos los ejes), las variables simples del GPL y las cadenas.

Lista de los **destinatarios** gestionados por Albatros y de los relativos comandos:

"@List"

Permite gestionar los comandos Simulación y Punto de consigna

Son admitidos los siguientes comandos (parámetro **identificador**):

- Sim: notifica el cambio de estado del flag switch Simulado. En base al estado del flag es visualizado seleccionado o soltado el pulsador que lo identifica en la barra de las Herramientas.(1=seleccionado, 0=deseleccionado)
- Setp: notifica el cambio de estado del flag switch CmdSetp. En base al estado del flag es visualizado seleccionado o soltado el pulsador que lo identifica en la barra de las Herramientas.(1=seleccionado, 0=deseleccionado)
- Escp: notifica el cambio de estado del flag switch Escluso. En base al estado del flag es visualizado seleccionado o soltado el pulsador que lo identifica en la barra de las Herramientas.(1=seleccionado, 0=deseleccionado)
- End: termina la ejecución de la lista. Este comando baja los pulsadores de Start y Stop y deshabilita las opciones desde menú Start y Stop.
- Hold:baja el pulsador de Stop y habilita la opción desde menú Stop. Levanta el pulsador de Start y deshabilita la opción desde menú Start.

"@Syn"

Comunicación entre GPL y visualizador de los cuadros sinópticos. Permite la apertura y cierre de cuadros sinópticos mandados por GPL y el envío de informaciones en una casilla de un sinóptico. Son admitidos los siguientes comandos (parámetro **identificador**):

- "Open:*nombreachivo*" apertura del sinóptico *nombreachivo.xsyn*
- "Close:*nombreachivo*" cierre del sinóptico *nombreachivo.xsyn*
- "Open" apertura de un sinóptico. El nombre del archivo es leído de la variable **información**
- "Close" cierre de un sinóptico. El nombre del archivo es leído de la variable **información**
- "*nombrecasilla*" casilla en la que está visualizada la información enviada

Es posible interactuar con la ventana de desplazamiento de los ejes según las especificaciones definidas también para el parámetro **destinatario** "[@Devices](#)" presentado abajo.

"@File"

Escritura en archivo. Permite crear archivos de log personalizados en los cuales almacenar las operaciones ejecutadas por una máquina. Son archivos de texto (ASCII). El parámetro **identificador** es el nombre del archivo en el cual será efectuada la escritura.

Si en el parámetro identificador no se muestra la ruta de acceso completa en que archivar el fichero, Albatros considera ruta la que está definida en tpa.ini en la sección [tpa] en la opción dirreport.

Si el identificador comienza i termina con un carácter de % la cadena en su interior es buscada en tpa.ini en la sección [tpa] y usada como nombre del archivo. Al interno del nombre pueden ser insertados caracteres que serán sustituidos durante la ejecución de la instrucción:

- %n número del módulo que ejecuta la instrucción SEND
- %h hora actual (formato 00-23)
- %d día actual (formato 01-31)
- %m mes actual (formato 01-12)
- %y año actual (formato en cuatro cifras)

Véase el ejemplo.

Las operaciones de escritura se ejecutan en modalidad append (los datos son adicionados al fondo del archivo). A un archivo pueden ser enviados datos numéricos (que son transformados automáticamente en ASCII) o cadenas. Es posible escribir cadenas en formato fecha y hora utilizando los caracteres de formato %d para la fecha y %t para la hora. Para la hora se usa el formato "HH:mm:ss" (o sea: horas, minutos y segundos separados por ":") y para la fecha un formato según las configuraciones nacionales. Es posible usar otro formato programando en tpa.ini en la sección [Albatros] la entrada

"LogNoLocale=1" (LogNoLocale=0 predeterminado, o sea uso del formato actual). Es posible también establecer el formato para data y hora, independientemente del formato programado en Windows, definido siempre en tpa.ini en la sección [Albatros] las entradas "LogDateFormat=" y "LogTimeFormat=" y asignando una cadena de caracteres según la tabla abajo. Si estas entradas no están presentes o están vacías se usan los formatos establecidos en Windows.

Formato de la hora

h	hora en el formato 12 horas sin ceros a la izquierda
hh	hora en el formato 12 horas con ceros a la izquierda
H	hora en el formato 24 horas sin ceros a la izquierda
HH	hora en el formato 24 horas con ceros a la izquierda
m	minutos sin ceros a la izquierda
mm	minutos con ceros a la izquierda
s	segundos sin ceros a la izquierda
ss	segundos con ceros a la izquierda
t	un solo carácter para indicar el marcador temporal, por ejemplo A o P
tt	más caracteres para indicar el marcador temporal, por ejemplo AM o PM

Nota: los formatos "t" y "tt" emplean el marcador temporal, que está en el panel de control del usuario actual. No es necesariamente "AM" y "PM".

Ejemplo: si son la 11:29 de la noche y la cadena es la siguiente "hh':mm':ss tt", se escribirá "11:29:40 PM".

Formato del día

d	día del mes sin ceros a la izquierda, en números
dd	día del mes con ceros a la izquierda, en números
ddd	día de la semana, en caracteres y abreviado a tres letras
dddd	día de la semana, representado en caracteres con nombre completo
M	mes sin ceros a la izquierda, en números
MM	mes con ceros a la izquierda, en números
MMM	mes, en caracteres y abreviado a tres letras
MMMM	mes, en caracteres con nombre completo
y	año con dos dígitos sin ceros a la izquierda para años menores de 10.
yy	año con dos dígitos con ceros a la izquierda para años menores de 10.
yyyy	año con cuatro o cinco dígitos según el calendario regional
yyyyy	año con cuatro o cinco dígitos según el calendario en uso

Ejemplo: si es miércoles 31 de Agosto 1994 y la cadena es la siguiente "ddd,' MMM dd yy", se escribirá "Mié, Agosto 31 94"

Si la información es omitida es adicionado al archivo un "punto y a parte".

"@FileName"

almacena una asociación entre una cadena constante y un nombre de archivo que se pueden componer con variables cadena. Desde Albatros recibe la comunicación de la asociación, reemplaza todos los siguientes nombres de archivo con el nombre recibido por medio de esta instrucción. El parámetro **identificador** es el nombre del archivo en el cual será efectuada la escritura. El nombre del archivo es una variable cadena. Si en el parámetro identificador no se muestra la ruta de acceso completa en que archivar el archivo, Albatros considera ruta la que está definida en tpa.ini en la sección [tpa] bajo dirreport. En tpa.ini, en la sección [GPLFileName] bajo Log se almacena el valor del parámetro identificador, para que pueda utilizarse también en las ejecuciones de Albatros que siguen. Para eliminar la asociación hay que establecer como parámetro identificador una cadena vacía. La asociación así definida vale para todos los módulos.

"@FileDelete"

Cancelación de un archivo. El parámetro **identificador** es el nombre del archivo que será cancelado (ruta completa). El nombre del archivo puede ser definido según las reglas descritas en correspondencia del parámetro **destinatario @File**.

"@FileRead"

Posiciona el puntador al archivo a inicio archivo. El parámetro **identificador** es el nombre del archivo (ruta completa). Si en el parámetro **identificador** no se muestra la ruta de acceso completa en que archivar el fichero, Albatros considera ruta la que está definida en tpa.ini en la sección [tpa] en la voz de Informe. El nombre del archivo puede ser definido según las reglas descritas en correspondencia del parámetro **destinatario** [@FileRead](#).

"@Axis"

Interactúa con la ventana de movimiento manual de los ejes según las especificaciones definidas también para el parámetro **destinatario** "@Devices" presentado abajo. Si una ventana que controla el movimiento del eje indicado está abierta ya, el comando opera en esta ventana si está abierta tanto en un cuadro Sinóptico, como en Diagnóstico. Si la ventana está cerrada, el comando intenta abrirla en Diagnóstico o en uno de los cuadros Sinópticos abiertos ya y que contiene el eje mismo.

"@Devices"

Pedido de apertura o cierre de la ventana Diagnóstico relativa al módulo che envía la información. Ejecución de comandos al interior de la ventana de diagnóstico. El parámetro **identificador** puede asumir los siguientes valores:

- "Open" apertura del Diagnóstico
- "Close" cierre del Diagnóstico

El parámetro **identificador**, cuando se desea interactuar con la ventana de desplazamiento eje, puede asumir los siguientes valores:

"MoveAX#nombre_eje#Open"	apertura de la ventana de desplazamiento eje.
"MoveAX#nombre_eje#Close"	cierre de la ventana de desplazamiento eje.
"MoveAX#nombre_eje#Plus"	presión del botón de desplazamiento eje en dirección positiva
"MoveAX#nombre_eje#Minus"	presión del botón de desplazamiento eje en dirección negativa
"MoveAX#nombre_eje#Stop"	presión del botón de paro del desplazamiento
"MoveAX#nombre_eje#Jog"	programa la modalidad de desplazamiento continuo del eje
"MoveAX#nombre_eje#Step"	programa la modalidad de desplazamiento a pasos discretos del eje
"MoveAX#nombre_eje#Absolute"	programa la modalidad de desplazamiento a coordenada establecida del eje

donde nombre_eje representa el nombre del eje visualizado en la ventana. Por ejemplo, si se desea abrir la ventana de desplazamiento eje X. El parámetro **identificador** será "[@MoveAX#X#Open](#)". El nombre del eje se puede encontrar en una de las siguientes formas:

1. Nombre_Grupo.Nombre_Subgrupo.Nombre_eje o Nombre_Grupo.Nombre_eje: se da la ruta completa del eje.
2. Nombre_eje: para identificar el eje correcto son hechas en orden las verificaciones:
 - si la tarea desde la cual llega el comando es una función de subgrupo, el eje es buscado en ese subgrupo.
 - si la tarea desde la cual llega el comando es una función del subgrupo principal, el eje es buscado en todo el grupo. Si está presente más de un eje con ese nombre la búsqueda falla.
 - si las verificaciones precedentes han fallado, el eje es buscado en todos los grupos del módulo. Si está presente más de un eje con nombre Nombre_eje la búsqueda no tiene éxito positivo.

Es posible impedir al usuario de actuar sobre las teclas de desplazamiento eje de todas las ventanas de desplazamiento eje del módulo en diagnóstico, programando el parámetro **identificador** de la siguiente manera:

- "MoveAX##UIENABLE" si el parámetro **información** es programado a 0, el desplazamiento ejes desde Albatros es deshabilitado; si es programado a 1 es habilitado el desplazamiento ejes desde Albatros.

La deshabilitación del desplazamiento ejes desde Albatros es recomendable cuando se mueven los ejes desde pulsadores de máquina.

"@Vars"

Requiere guardar el contenido de una variable global GPL en el archivo de los parámetros tecnológicos o herramientas. El parámetro **identificador** es el nombre de la variable global (de máquina, de grupo o de librería) del cual se requiere la actualización.

"@Application"

Interacción con Albatros. Permite quitar Albatros o ver unos "cuadros de mensajes" en la pantalla para informar el usuario o solicitar el consentimiento para actividades posteriores. Los valores admitidos para el parámetro **identificador** son:

"Quit"	cierra Albatros
"Lock"	impide el cierre de Albatros a través de Archivo->Salir , de la combinación de teclas [ALT+F4] o del botón de cierre.

"UnLock" restablece la posibilidad de cerrar Albatros

"MsgBox:flags" abre un cuadro de mensaje.

El comportamiento de los cuadros de mensajes está controlado por la parte "indicadores" de la cadena **identificador**. Esta puede ser una secuencia de los siguientes caracteres (pueden ser indistintamente mayúsculos o minúsculos):

"O" botón "Aceptar"

"C" botón "Cancelar"

"Y" botón "Sí"

"N" botón "No"

"R" botón "Reintentar"

"S" icono del señal de Stop

"?" icono de información, compuesto por una letra 'i' minúscula dentro de un círculo

"!" icono con signo de exclamación

"*" icono informativo

"1" el primer botón es el predeterminado

"2" el segundo botón es el predeterminado

"3" el tercer botón es el predeterminado

"4" el cuarto botón es el predeterminado

Si no está indicado, el botón predeterminado es el primero.

Por ejemplo, "MsgBox:?YN2" define un cuadro de mensaje con el icono informativo, dos botones "Sí" y "No" de los cuales el segundo es el predeterminado. El parámetro **información** puede ser una cadena, que contiene el texto por visualizar, o bien un número entero que es interpretado como código de un mensaje de módulo gestionado por TpaLangs.exe o bien una etiqueta de un mensaje de grupo definido con la instrucción [DEFMSG](#).

En cuanto al texto, si dentro se encuentra el carácter de avance de línea, "\u000A", el texto será dividido en dos porciones, y la primera será mostrada como texto del cuadro de mensaje, mientras la segunda porción será mostrada como explicación, o detalle, del texto.

El idioma en el que se muestran los botones es el de Windows.

"@Help"

Apertura de un archivo de ayuda. Permite comandar la visualización de un archivo de ayuda especificando el argumento por visualizar. Los valores admitidos para el parámetro **identificador** son:

- "Open:*nombrearchivo*" apertura de un archivo de ayuda
- "Close:*nombrearchivo*" cierre de un archivo de ayuda

La parte "*nombrearchivo*" de la cadena especifica el nombre del archivo de ayuda por abrir.

El parámetro **información** puede ser una cadena o un número y asume respectivamente el significado de clave o de número de contexto (sirve para identificar la página o argumento de la ayuda que se quiere visualizar).

"@Report"

Agrega señalizaciones al archivo de informe de Albatros (MONTH(n. mes).TER). El parámetro

identificador es:

- "Add"

El parámetro **información** puede ser:

- una variable cadena o una constante cadena: en este caso es memorizado el texto contenido en la cadena
- una variable integer o un valor numérico integer: en este caso es memorizado el texto definido en la instrucción [DEFMSG](#).

"@Ini"

Escribe una combinación de teclas=valor en el archivo tpa.ini. El parámetro **identificador** es el nombre de las teclas que se deben agregar en el archivo tpa.ini en la sección [Tpa]. Para escribir en una sección específica, al nombre de las teclas hay que agregar el nombre de la sección encerrado entre corchetes ("[Sección]Tecla").

El parámetro **información** puede ser una variable cadena o numérica, una constante cadena o numérica.

"@ShellExecute"

Solicita al sistema operativo que abra un archivo por medio de un programa asociado a la extensión del archivo. También es posible iniciar un ejecutable. El parámetro **identificador** es el nombre del archivo que hay de abrir o el nombre del programa que hay de iniciar. Se puede declarar el nombre del archivo

con un nombre de ruta completa; de lo contrario se busca en el directorio actual de Albatros. El nombre del archivo se busca también en los que se definen por medio de "@FileName".

"@StartProg"

Ejecuta al programa definido en el parámetro **identificador**. No se puede pasar argumentos al programa que hay de iniciar. El nombre del programa tiene que contener la ruta entera; de lo contrario se busca en el presente directorio de Albatros. El nombre del programa también se carga en los que se definen por medio de "@FileName".

"@TermProg"

Finaliza el programa definido en el parámetro **identificador** y iniciado por medio de "@StartProg". El nombre del programa tiene que contener la ruta entera; de lo contrario se busca en el presente directorio de Albatros. El nombre del programa también se carga en los que se definen por medio de "@FileName".

"@DialogFile"

Permite programar algunos parámetros relativos a una ventana de dialogo de Archivo Abrir o de Archivo Guardar.

Los valores admitidos por el parámetro **identificador** son:

"Extension"	si el usuario non inserta una extensión, se utiliza la extensión definida en al parámetro información (variable o constante cadena)
"Filter"	configura el filtro de los tipos de archivo a utilizar. El parámetro información puede ser una variable cadena o una constante cadena y en este caso se utiliza como filtre el texto en la cadena, una variable integer o un valor numérico integer; en este caso se utiliza como filtro el texto definido en la instrucción DEFMSG .
"Flags"	configura las marcas de inicialización. Para la lista de los valores a programar en el campo información (variable o constante integer) véase la documentación oficial Microsoft sobre el miembro Flags de la estructura OPENFILENAME.
"InitalDir"	configura la carpeta inicial, definida en el campo información (variable o constante cadena).
"Title"	configura el título de la ventana. El parámetro información puede ser una variable cadena o una constante cadena y en este caso se utiliza como filtre el texto en la cadena, una variable integer o un valor numérico integer; en este caso se utiliza como título el texto definido en la instrucción DEFMSG .

"@Language"

Configura el número de mensaje de grupo o módulo o biblioteca que será usado en la siguiente RECEIVE con el mismo identificador. El valor admitido para el parámetro **identificador** es "DEFMSG:*". El parámetro **información** puede ser una variable o constante entera, y en este caso define el número del mensaje de grupo que mostrar. Puede ser de tipo cadena o constante cadena, y en este caso define el nombre de la DEFMSG.

Ejemplo

```
; Ejemplo de instrucción envío file con nombre construido en ejecución.
; Supongamos que la fecha en que es ejecutada la instrucción sea
; 31-01-2000
```

```
; en GPL
SEND "@File", "%Log%", 0, "Inicio ejecución"
; agrega un "retorno de carro"
; en tpa.ini a la sección [TPA]
SEND "@File", "%Log%", 0
Log=c:\Albatros\report\%y\Rep%m%d.txt
```

```
; El nombre del archivo resultante es:
c:\Albatros\report\2000\Rep0131.txt
```

```
; Ejemplo de instrucción send Vars
; se define una variable Var_SendVars as double en el archivo de
; las variables globales
; en Paramétrica Tecnológica se define un campo que tiene como Nombre
; Matriz Var_SendVars
; en GPL
SETVAL 100.0,Var_SendVars
; envía el valor 100.0 al parámetro de la paramétrica tecnológica
; asociado a la variable Var_SendVars
SEND "@Vars", "Var_SendVars", 0
```

```

; Ejemplo de instrucción send INI
; se escribe en tpa.ini la clave Radix en la sección [Albatros] para
; establecer una base numérica para la visualización decimal
SEND "@INI", "[Albatros]Radix", 0;1

; Ejemplo de como se establece una asociación de cadena constante GPL
; y nombre de un archivo.

; declaración de una variable cadena
nombreambrivo as string
; composición del nombre del archivo
setstring "C:\albatros\report\LogFile.txt",nombreambrivo
; asociación
SEND "@FileName", "LOG",0,nombreambrivo
; de aquí todas las operaciones de escritura
; se ejecutan en el archivo definido por la variable nombreambrivo
SEND "@File", "LOG",0,"Escribir en el archivo LOG"

```

SENDIPC

Sintaxis

```

SENDIPC nombreIPC, espera [, nombrevar1 [, nombrevarN, ...]]
SENDIPC nombreIPC, espera, matriz[línea]
SENDIPC nombreIPC, espera, vector
SENDIPC nombreIPC, espera, matriz

```

Argumentos

nombreIPC	constante cadena. Nombre de la IPC
espera	constante predefinida. Modo de espera de lectura del comando Los valores admitidos son: WAIT espera la lectura del comando NOWAIT no espera la lectura del comando
nombrevar1[... nombreVarN]	constante o variable. Nombres variables 1÷N
matriz[línea]	constante o variable integer. Número de línea de la matriz
vector	Nombre de vector
matriz	Nombre de matriz

Descripción

Envía un comando IPC a la memoria compartida "**nombreIPC**". La primera vez que es ejecutada una instrucción de SENDIPC es asignada la memoria compartida, cuya dimensión es calculada en base a la dimensión de los datos que son enviados. La máxima dimensión de la memoria tamponada es 64 Kb. Pueden ser definidas como máximo 48 memorias compartidas, identificadas por 48 nombres unívocos.

A la memoria compartida es asociado un semáforo que permite sincronizar la ejecución de las tareas que acceden a ella. La tarea que escribe los datos activa el semáforo al termino de la escritura, la tarea que lee los datos lo desactiva al termino de la lectura.

Si como parámetro **espera** ha sido indicado WAIT, la tarea que ha escrito los datos esperará que estos sean leídos (semáforo desactivado) antes de seguir con la ejecución.

Una SENDIPC sin datos se reduce a un sincronismo entre tareas. En este caso no es asignada la memoria compartida.

IPC intermódulo

Dos módulos remotos pueden intercambiar datos a través de IPC. Estos IPC se llaman IPC intermódulo.

Para definir un PC intermódulo hay que escribir el **nombreIPC** según el formalismo siguiente: número del módulo fuente, "->", número del módulo destinatario, ":", y luego otros caracteres del nombre del IPC.

Por ejemplo, "0->1:Parámetros Básicos".

Véase también [WAITIPC](#) y [TESTIPC](#).

WAITIPC

Sintaxis

```

WAITIPC nombreIPC [, nombrevar1 [, nombrevarN, ...]]
WAITIPC nombreIPC, matriz[línea]

```

WAITIPC
WAITIPC

nombreIPC, vector
nombreIPC, matriz

Argumentos

nombreIPC	constante cadena. Nombre de la IPC
nombrev1[... nombreVarN]	constante o variable. Nombres variables 1÷N
matriz[línea]	constante o variable integer. Número de línea de la matriz
vector	nombre de vector
matriz	nombre de matriz

Descripción

Recibe un comando IPC desde la memoria compartida "**nombreIPC**". La primera vez que es ejecutada una instrucción de WAITIPC es asignada la memoria compartida, cuya dimensión es calculada en base a la dimensión de los datos que son enviados. La máxima dimensión de la memoria tamponada es 64 Kb. Pueden ser definidas como máximo 48 memorias compartidas, identificadas por 48 nombres unívocos. A la memoria compartida es asociado un semáforo que permite sincronizar la ejecución de las tareas que acceden a ella. La tarea que lee los datos espera que el semáforo sea activado por la tarea que escribe los datos, lee los datos y desactiva el semáforo. Una WAITIPC sin datos se reduce a un sincronismo entre tareas. En este caso no es asignada la memoria compartida. Véase también [SENDIPC](#) y [TESTIPC](#).

WAITRECEIVE

Sintaxis

WAITRECEIVE **[fuente,] identificador, indicadores, [contenedor]**

Argumentos

fuente	constante de tipo cadena
identificador	constante de tipo cadena
indicadores	constante de tipo entero
contenedor	nombre de dispositivo o variable (numérica o cadena)

Descripción

Espera que la información pedido (especificada por **identificador**) llegue antes de continuar la ejecución del programa GPL. Para el uso hacer referencia a la documentación de la instrucción [RECEIVE](#).

10.3.10 Matemática

ABS

Sintaxis

ABS **operando, resultado**

Argumentos

operando	constante o variable o nombre de dispositivo
resultado	variable o nombre de dispositivo

Descripción

Extrae el valor absoluto de **operando** y lo pone en **resultado**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal  -10,op      ; asigna -10 a la variable op
Abs     op,var      ; El valor puesto en la variable var será 10
```

ADD

Sintaxis

ADD **operando1, operando2, resultado**

Argumentos

operando1	constante o variable o nombre de dispositivo
operando2	constante o variable o nombre de dispositivo

resultado variable o nombre de dispositivo

Descripción

Ejecuta una operación de suma entre **operando1** y **operando2** y pone el resultado en **resultado**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal 5,op1 ; asigna 5 a la variable op1
Add op1,3,var
;El valor puesto en la variable var será 8
```

AND

Sintaxis

AND **operando1, operando2, resultado**

Argumentos

operando1 constante o variable o nombre de dispositivo
operando2 constante o variable o nombre de dispositivo
resultado variable o nombre de dispositivo

Descripción

Ejecuta una operación de AND binario (*entre dos bit, resulta 1 solo si ambos valen 1*) entre **operando1** y **operando2** y pone el resultado en **resultado**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
;El valor puesto en la variable var será 1
;(Notación binaria: 5 = 0101, 3 = 0011, 1 = 0001)
And 5,3,var
```

ARCCOS

Sintaxis

ARCCOS **operando, resultado**

Argumentos

operando constante o variable o nombre de dispositivo
resultado variable o nombre de dispositivo

Descripción

Ejecuta una operación de arcocoseno en **operando** y pone el valor en **resultado**, en grados. El valor del resultado está dado en el intervalo $0^\circ \div 180^\circ$. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

ARCSIN

Sintaxis

ARCSIN **operando, resultado**

Argumentos

operando constante o variable o nombre de dispositivo
resultado variable o nombre de dispositivo

Descripción

Ejecuta una operación de arcoseno en **operando** y pone el valor en **resultado**, en grados. El valor del resultado está dado en el intervalo $-90^\circ \div +90^\circ$. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

ARCTAN

Sintaxis

ARCTAN **operando1, [operando2], resultado**

Argumentos

operando1...[operando2] constante o variable o nombre de dispositivo

resultado variable o nombre de dispositivo

Descripción

Si **operando2** es omitido, ejecuta una operación de arcotangente en **operando1** y pone el valor en **resultado**, en grados. Si **operando2** está presente, el ángulo considerado será aquel cuyo seno es dado por **operando1** y cuyo coseno es dado por **operando2**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

COS

Sintaxis

COS **operando, resultado**

Argumentos

operando constante o variable o nombre de dispositivo
resultado variable o nombre de dispositivo

Descripción

Ejecuta una operación de coseno en **operando** y pone el valor en **resultado**. El argumento **operando** es expresado en grados con eventual parte fraccionaria centesimal (ejemplo: 30° 15" = 30,25). Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal 60,op ; asigna 60 a la variable op
Cos op,var
;El valor puesto en la variable var será 0.5
```

DIV

Sintaxis

DIV **operando1, operando2, resultado**

Argumentos

operando1 constante o variable o nombre de dispositivo
operando2 constante o variable o nombre de dispositivo
resultado variable o nombre de dispositivo

Descripción

Ejecuta una operación de división entre **operando1** y **operando2** y pone el resultado en **resultado**. La instrucción puede generar un error de sistema cuando **operando2** es igual a 0. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal 10,op1 ; asigna 10 a la variable op1
SetVal 5,op2 ; asigna 5 a la variable op2
Div op1,op2,var
;El valor puesto en la variable var será 2
```

EXP

Sintaxis

EXP **operando, resultado**

Argumentos

operando constante o variable o nombre de dispositivo
resultado variable o nombre de dispositivo

Descripción

Calcula el exponencial de **operando** y pone el valor en **resultado**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal 2.302,op ; asigna 2.302585093 a la variable op
Exp op,var
```

;El valor puesto en la variable var será 10

EXPR

Sintaxis

EXPR **variable = expresión**

Argumentos

variable nombre de dispositivo o variable
expresión conjunto de operadores

Descripción

Esta instrucción permite calcular expresiones matemáticas. Los operadores pueden ser constantes, nombres de dispositivos o variables. La sintaxis de la instrucción requiere que entre cada operador y cada operando tiene que haber un espacio.

Si todos los operandos no son del mismo tipo, se realizará una conversión automática y el tipo del resultado de la operación será igual al mayor de los dos según la siguiente regla:

- char < integer
- float < double
- char o integer < float o double.

Resuelta la **expresión**, el resultado se convertirá al tipo **variable**.

Los operadores admitidos, agrupados según niveles de prioridad son los siguientes:

()	paréntesis
-	operador de cambio signo
ABS	valor absoluto del operando
ROUND	redondeo a la unidad
TRUNC	truncamiento a entero
LOG	logaritmo natural
LOGDEC	logaritmo en base 10
EXP	exponencial
SRQ	raíz cuadrada
SIN	operación de seno. El operando es expresado en grados con eventual parte fraccionaria centesimal (ej.: 30° 15" = 30,25)
COS	operación de coseno. El operando es expresado en grados con eventual parte fraccionaria centesimal (ej.: 30° 15" = 30,25)
TAN	operación de tangente. El argumento es expresado en grados
ARCSIN	operación de arcoseno. El valor del resultado expresado en grados y está dado en el intervalo -90°÷+90°
ARCCOS	operación de arcocoseno. El valor del resultado expresado en grados y está dado en el intervalo -90°÷+90°
ARCTAN	operación de arcotangente. Véase ARCTAN
^	operador de potencia
*	multiplicación
/	división
%	resto de la división (módulo)
+	suma
-	sustracción

Esta instrucción permite simplificar la escritura del código GPL en los casos en que se tengan que efectuar cálculos matemáticos, substituyendo cada instrucción GPL correspondiente a los operadores enumerados en la tabla. Dichas instrucciones quedan disponibles por compatibilidad.

Ejemplo

; cálculo de la distancia entre dos puntos

EXPR dist = SQR ((Xb - Xa) ^ 2 + (Yb - Ya) ^ 2)


```

; conversión de datos en una EXPR
local op1 as double
local op2 as integer
local op3 as float
local resultado as integer
setval 900,op1
setval 100,op2
setval 500.0,op3
EXPR resultado = ( op1 + op2 ) / op3
; primero se resuelve la expresión en doble
; después se realiza una conversión en entero
; como el tipo de dato resultado

```

LOG

Sintaxis

LOG **operando, resultado**

Argumentos

operando	constante o variable o nombre de dispositivo
resultado	variable o nombre de dispositivo

Descripción

Calcula el logaritmo natural de **operando** y pone el valor en **resultado**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```

SetVal 10,op      ; asigna 10 a la variable op
log      op,var
;El valor puesto en la variable var será 2.302585093

```

LOGDEC

Sintaxis

LOGDEC **operando, resultado**

Argumentos

operando	constante o variable o nombre de dispositivo
resultado	variable o nombre de dispositivo

Descripción

Calcula el logaritmo en base 10 de **operando** y pone el valor en **resultado**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```

SetVal 10,op      ; asigna 10 a la variable op
Logdec op,var
;El valor puesto en la variable var será 1

```

MOD

Sintaxis

MOD **operando1, operando2, resultado**

Argumentos

operando1	constante o variable integer o nombre de dispositivo
operando2	constante o variable integer o nombre de dispositivo
resultado	variable integer o nombre de dispositivo

Descripción

Ejecuta una operación de módulo entre **operando1** y **operando2** y pone el resultado en **resultado**. El módulo es el resto resultante de la división entre el primer y el segundo operando. La instrucción puede generar error de sistema cuando **operando2** es igual a 0. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo


```
;El valor puesto en la variable var será 7
;(Notación binaria: 5 = 0101, 3 = 0011, 7 = 0111 )
```

```
Or      5,3,var
```

RANDOM

Sintaxis

```
RANDOM      min, max, resultado
```

Argumentos

```
min          constante o variable
max          constante o variable
resultado    variable o nombre de dispositivo
```

Descripción

Retorna en **resultado** un número pseudocasual incluido entre **min** y **max** (extremos incluidos). Ejecutando la instrucción numerosas veces se obtiene una secuencia de números pseudocasuales. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal 2,op1      ; asigna 2 a la variable op1
SetVal 100,op2    ; asigna 100 a la variable op2
Random op1,op2,var ; El valor puesto en la variable var
                  ; será un número casual
                  ; comprendido entre 2 y 100
```

RESETBIT

Sintaxis

```
RESETBIT    máscara, nbit
```

Argumentos

```
máscara     constante o variable entera o nombrecontador o nombrepuerto. Representa
             el valor por modificar (max 32 bit)
nbit        constante o variable entera o nombrecontador. Número del bit por modificar
```

Descripción

Programa a 0 un solo bit, especificado por **nbit**, de la **máscara** de bit pasada. El argumento **máscara** debe poder corresponder a un valor entero con un máximo de 32 bit. El número de bit, **nbit**, va desde 1 hasta 32.

Ejemplo

Estado del puerto antes de la ejecución del código



Estado del puerto después de la ejecución del código



```
-----
; Ejemplo para deshabilitar una línea de un puerto indicador:
-----
```

```
SetVal 2,nbit
Resetbit PuertoIndicador,nbit

; desactiva la línea 2 del puerto indicador
```

ROUND

Sintaxis

```
ROUND      operando, resultado
```

Argumentos

operando constante o variable o nombre de dispositivo
resultado variable o nombre de dispositivo

Descripción

Ejecuta una operación de redondeo a la unidad en **operando** y pone el valor en **resultado**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal 5.7,op ; asigna 5.7 a la variable op
Round op,var
```

; El valor puesto en la variable var será 6

```
SetVal 5.2,op ; asigna 5.2 a la variable op
Round op,var
```

; El valor puesto en la variable var será 5

SETBIT**Sintaxis**

SETBIT máscara, nbit

Argumentos

máscara constante o variable entera o nombrecontador o nombrepuerto. Valor por modificar (max 32 bit)
nbit constante o variable entera o nombrecontador. Número del bit por modificar (1÷32)

Descripción

Pone a 1 un solo bit, especificado por **nbit**, de la **máscara** de bit pasada. Hay que el argumento **máscara** se pueda corresponderse con un valor entero con un máximo de 32 bit. El número de bit, **nbit**, va desde 1 hasta 32.

Ejemplo

estado del puerto antes de la ejecución del código



estado del puerto después de la ejecución del código



```
; Ejemplo para habilitar una línea de un puerto indicador:
```

```
SetVal 2,nbit
Setbit PuertoIndicador,nbit
```

; activa la línea 2 del puerto indicador

SHIFTL**Sintaxis**

SHIFTL operando1, [operando2]

Argumentos

operando1 variable (integer o char) o nombre de dispositivo
operando2 variable (integer o char) o nombre de dispositivo

Descripción

Si **operando2** no está especificado, esta instrucción realiza una operación de desplazamiento a la izquierda de los bits que componen el **operando1**. Si está especificado el segundo operando también, esta instrucción realiza una rotación entre los bits de **operando2** y los bit de **operando1**. Al final de la operación, operando2 contendrá el agregado, es decir el bit alto de **operando1**.

Descripción

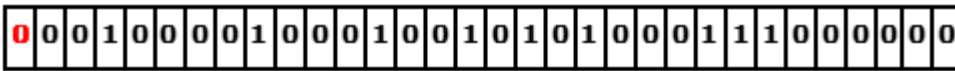
Si **operando2** no está especificado, esta instrucción realiza una operación de desplazamiento a la derecha de los bits que componen el **operando1**. Si **operando1** está definido de tipo char, el bit alto de entrada es siempre 0. Si **operando1** está definido de tipo integer, el bit 32 de entrada es el bit de marca. Si está especificado el segundo operando también, esta instrucción realiza una rotación entre **operando2**, entendido como valor 0 o diferente de 0 y los bit de **operando1**. Al final de la operación, **operando2** contendrá el agregado de la operación y el bit de mayor peso de operando1 se volverá 0 o bien 1, según el valor inicial de **operando2** (0 o 1).

Ejemplo

Rotación de operandos de tipo integer (desplazamiento a la derecha con agregado)
Antes de la rotación

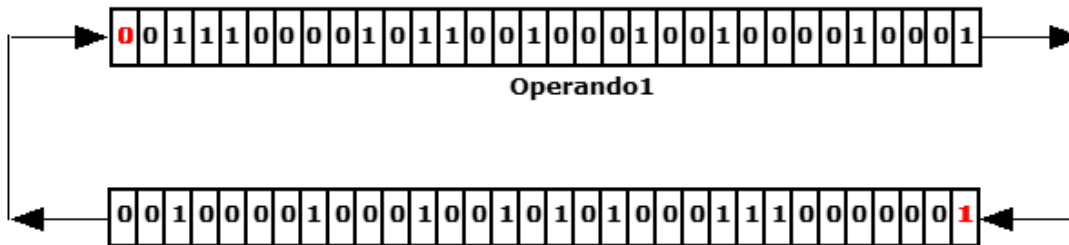


Operando1



Operando2

Después de la rotación



Desplazamiento a la derecha de un char (desplazamiento a la derecha sin agregado)

Descripción

Ejecuta una operación de raíz cuadrada en **operando** y pone el valor en **resultado**. Para el parámetro **operando** son admitidos solo valores positivos. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal 81,op ; asigna 81 a la variable op
Sqr    op,var

; El valor puesto en la variable var será 9
```

SUB**Sintaxis**

SUB **operando1, operando2, resultado**

Argumentos

operando1	constante o variable o nombre de dispositivo
operando2	constante o variable o nombre de dispositivo
resultado	variable o nombre de dispositivo

Descripción

Ejecuta una operación de sustracción entre **operando1** y **operando2** y pone el resultado en **resultado**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal 10,op1 ; asigna 10 a la variable op1
SetVal 4,op2 ; asigna 4 a la variable op2
Sub    op1,op2,var

;El valor puesto en la variable var será 6
```

TAN**Sintaxis**

TAN **operando, resultado**

Argumentos

operando	constante o variable o nombre de dispositivo
resultado	variable o nombre de dispositivo

Descripción

Ejecuta una operación de tangente en **operando** y pone el valor en **resultado**. El argumento **operando** es expresado en grados. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversion de datos](#).

Ejemplo

```
SetVal 45,op ; asigna 45 a la variable op
Tan    op,var

;El valor puesto en la variable var será 1
```

TRUNC**Sintaxis**

TRUNC **operando, resultado**

Argumentos

operando	constante o variable o nombre de dispositivo
resultado	variable o nombre de dispositivo

Descripción

Trunca a entero el valor expreso por **operando** y pone el valor en **resultado**. (La parte decimal se pierde). Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
SetVal 5.7,op ; asigna 5.7 a la variable op
Trunc op,var
;El valor puesto en la variable var será 5
```

XOR**Sintaxis**

```
XOR operando1, operando2, resultado
```

Argumentos

operando1	constante o variable o nombre de dispositivo
operando2	constante o variable o nombre de dispositivo
resultado	variable o nombre de dispositivo

Descripción

Ejecuta una operación de XOR binario (entre dos bit, resulta 1 si uno solo de los dos vale 1) entre **operando1** y **operando2** y pone el resultado en **resultado**. Para convertir los datos, en base al tipo de dato declarado, hacer referencia al capítulo [Conversión de datos](#).

Ejemplo

```
Xor 5,3,var
;El valor puesto en la variable var será 6
;(Notación binaria: 5 = 0101, 3 = 0011, 6 = 0110)
```

10.3.11 Multitasking**ENDMAIL****Sintaxis**

```
ENDMAIL mail
```

Argumentos

mail	constante o variable de tipo integer. Número del buzón de correos (1÷256)
-------------	---

Descripción

Señala la fin de la ejecución de un comando asociado a un mensaje tomado desde la casilla de correos mail. La tarea que había enviado el mensaje (con la instrucción [SENDMAIL](#)) y había quedado en espera de la ejecución del comando (argumento de espera WAITACK) puede ahora seguir con su propia ejecución. Esta instrucción **tiene efecto solo si ejecutada por la tarea** que ha precedentemente recibido el mensaje (con la instrucción WAITMAIL o TESTMAIL).

Véase también las instrucciones [SENDMAIL](#), [WAITMAIL](#) y [TESTMAIL](#).

Ejemplo

[Servidor de desplazamiento ejes](#)

ENDREALTIMETASK**Sintaxis**

```
ENDREALTIMETASK nombrefunción
```

Argumentos

nombrefunción	nombre función
----------------------	----------------

Descripción

Termina la ejecución de una [tarea de tiempo real](#). Véase también [STARTREALTIMETASK](#).

ENDTASK**Sintaxis**

```
ENDTASK [nombretarea]
```

Argumentos

nombretarea nombre tarea

Descripción

Termina la ejecución de una tarea y de todas las tarea que esta ha activado (tarea hijas). Esta instrucción suspende además el movimiento de los ejes, anula las RECEIVE pendientes y cierra las eventuales conexiones a puertos seriales. Si la variable **nombretarea** es omitida, dejar terminar la ejecución de la tarea corriente.

GETPRIORITYLEVEL

Sintaxis

GETPRIORITYLEVEL **nivel**, [**nombrefunción**]

Argumentos

nivel variable. Nivel de prioridad de ejecución.
nombrefunción nombre función

Descripción

Devuelve en la variable **nivel** el valor de prioridad de la tarea definida por **nombrefunción**. Dicho valor es un número comprendido entre 1 y 255, donde 1 representa el nivel más alto de prioridad y 255 el más bajo. Si **nombrefunción** no está definido, es retornado el valor de prioridad de la tarea corriente, o sea de la función en la cual la instrucción GETPRIORITYLEVEL es ejecutada. Véase también [SETPRIORITYLEVEL](#).

GETREALTIME

Sintaxis

GETREALTIME **nombrevr**

Argumentos

nombrevr variable integer

Descripción

Retorna en la variable **nombrevr** el tiempo pasado desde el inicio del último tiempo real de gestión de los ejes. El tiempo retornado está expresado en microsegundos. Véase también [GETREALTIMECOUNT](#).

GETREALTIMECOUNT

Sintaxis

GETREALTIMECOUNT **nombrevr**

Argumentos

nombrevr variable integer

Descripción

Retorna en la variable **nombrevr** el número de tiempo real de gestión de los ejes ejecutados desde la última inicialización del control numérico. Véase también [GETREALTIME](#).

HOLDTASK

Sintaxis

HOLDTASK [**nombretarea**]

Argumentos

nombretarea nombre tarea

Descripción

Suspende la ejecución de la tarea definida en **nombretarea**. Esta instrucción no suspende el movimiento de los ejes, que hay que parar con instrucciones de STOP. Si **nombretarea** es omitido, suspende la ejecución de la tarea corriente.

RESUMETASK

Sintaxis

RESUMETASK [**nombretarea**]

Argumentos

nombretarea nombre tarea

Descripción

Reanuda la ejecución de las tareas definidas en **nombretarea**. Si **nombretarea** es omitido, reanuda la ejecución de la tarea corriente. Si la tarea había sido suspendida con la instrucción **STOPTASK** son reanudados también los eventuales movimientos de los ejes.

SENDMAIL

Sintaxis

SENDMAIL **mail, espera [, nombrevar1 [...nombrevarN]]**
SENDMAIL **mail, espera, matriz[filas]**

Argumentos

mail constante o variable integer. Número del mailbox (1÷256)
espera constante predefinida. Modo de espera de lectura o de ejecución del comando.
 Los valores que pueden ser asignados a la constante **espera** son:
 - **WAIT** espera la lectura del comando
 - **NOWAIT** no espera la lectura del comando
 - **WAITACK** espera la ejecución del comando
nombrevar1[...nombrevarN] constante o variable integer. Nombres variables 1÷20
matriz[filas] constante o variable integer. Número de fila de la matriz

Descripción

Envía un mensaje (o comando) a la casilla **mail**. Los mensajes pueden ser utilizados para sincronizar y cambiar informaciones entre dos o más tareas.

Si la casilla **mail** no existe, o sea no ha sido aún ejecutada una instrucción [WAITMAIL](#) o [TESTMAIL](#), la instrucción es ignorada.

Si la tarea receptor no está en espera de un mensaje (instrucción [WAITMAIL](#)) o bien está ocupada, los datos (**nombrevar** (1÷20) o la fila de matriz indicada por **matriz[filas]**) pasados por la instrucción son memorizados en una cola. En este caso:

1. si el argumento de espera es **NOWAIT** la ejecución continua con la instrucción sucesiva;
2. si el argumento de espera es **WAIT** la ejecución espera que el mensaje sea leído por la tarea receptor;
3. si el argumento de espera es **WAITACK** la ejecución espera que el mensaje sea leído y que sea confirmada la ejecución del comando por la tarea receptor (a través la instrucción [ENDMAIL](#) o una nueva [WAITMAIL](#)).

Es muy importante que el número de las variables pasadas y sus tipos coincidan con los que son utilizados para crear la casilla de correos con la instrucción [WAITMAIL](#). El control no consiente el uso de tipos diferentes y no efectúa conversiones de tipo automáticos (cast) como ocurre solitamente.

Una SENDMAIL sin los parámetros opcionales (datos) se reduce a un simple mecanismo de sincronización entre tareas.

Ejemplo

[Servidor de desplazamiento ejes](#)

SETPRIORITYLEVEL

Sintaxis

SETPRIORITYLEVEL **nivel [, nombrefunción]**

Argumentos

nivel constante o variable. nivel de prioridad de ejecución.
nombrefunción nombre función

Descripción

Asigna a la tarea definida en **nombrefunción** el valor de prioridad contenido en la variable **nivel**. Dicho valor es un número comprendido entre 0 y 255, donde 0 representa el nivel más alto de prioridad y 255 el más bajo. Si no está definido el nombre de la tarea en la variable **nombrefunción** es modificado el valor de prioridad de la tarea corriente, o sea el nivel de ejecución de la función en la cual la instrucción es ejecutada.

Véase también [GETPRIORITYLEVEL](#).

STARTREALTIMETASK

Sintaxis

STARTREALTIMETASK **nombrefunción**

Argumentos

nombrefunción nombre función

Descripción

Activa la ejecución de una [tarea de tiempo real](#). Dicha tarea es ejecutada con la misma frecuencia del tiempo real de control de los ejes. A diferencia de las normales tareas GPL es ejecutado por completo en cada tiempo real, desde la primera instrucción de la función hasta la primera instrucción FRET. Véase también [ENDREALTIMETASK](#).

Atención:

Las variables locales declaradas en la tarea de tiempo real son inicializadas sólo al iniciar de la tarea y en seguida mantienen el valor de la última ejecución.

STARTTASK**Sintaxis**

STARTTASK **nombrtarea [, parámetros]**

Argumentos

nombrtarea nombre tarea
parámetros eventuales parámetros necesarios durante la ejecución de la tarea

Descripción

Activa la ejecución de la tarea definida en la variable **nombrtarea**.

A la tarea pueden ser pasados eventuales **parámetros** utilizados durante la ejecución. El número y el tipo de los parámetros pasados debe de corresponder a los que son declarados por la función que implementa la tarea. Si la tarea ya está en ejecución la instrucción no tiene ningún efecto.

Ejemplo

[Ejecución Secuencial / Paralela](#)

STOPTASK**Sintaxis**

STOPTASK **nombrtarea**

Argumentos

nombrtarea nombre función

Descripción

Detiene la ejecución de una [tarea](#) y de todas las demás tareas que esta ha activado (tareas hijas), deteniendo el movimiento de los ejes (en el caso en que esté en acto dicha acción).

Si **nombrtarea** es omitida, detiene la ejecución de la tarea corriente. La ejecución de la tarea y el movimiento de los ejes puede ser reactivado con la instrucción [RESUMETASK](#).

WAITMAIL**Sintaxis**

WAITMAIL **mail [, nombrevar1 [,...nombrevarN]]**
WAITMAIL **mail, matriz[filas]**

Argumentos

mail constante o variable integer. Número del buzón de correo (1÷256)
nombrevar1[...nombrevarN] constante o variable integer. Nombres variables 1÷20
matriz[filas] constante o variable integer. Número de filas de la matriz

Descripción

Recibe un mensaje desde el buzón de correos **mail**. Al mensaje pueden ser asociados unos datos.

Los datos recibidos juntos con el mensaje son memorizados en las variables **nombrevar** (1÷20) indicadas en la fila de la matriz indicada por **matriz[filas]**.

Si en el momento en que es ejecutada la instrucción WAITMAIL no están ya presentes unos mensajes en espera de lectura, la tarea es puesta en estado de HOLD, del cual sale solo cuando otra tarea envía un mensaje a la casilla con la instrucción [SENDMAIL](#).

La congruencia entre los datos pasados y los esperados por la instrucción, es controlada durante la ejecución de la instrucción.

Una WAITMAIL sin parámetros opcionales se reduce a un simple mecanismo de sincronización entre tareas. Véase también las instrucciones [SENDMAIL](#), [ENDMAIL](#) y [TESTMAIL](#).

Ejemplo

[Servidor de desplazamiento ejes](#)

WAITTASK

Sintaxis

WAITTASK **nombretarea**

Argumentos

nombretarea nombre tarea

Descripción

Espera que la tarea **nombretarea** haya terminado la ejecución.

Ejemplo

[Ejecución Secuencial / Paralela](#)

10.3.12 Gestión de Flujo

CALL

Sintaxis

CALL **nombresubprograma**

Argumentos

nombresubprograma nombre subprograma, etiqueta

Descripción

Ejecuta el subprograma definido por la etiqueta **nombresubprograma**.

Cada subprograma, para poder regresar a la instrucción sucesiva a la CALL, debe de terminar, en el punto de salida, con instrucción [RET](#).

Atención

Esta instrucción, junto con la RET, es una típica fuente de errores de programación. Se recomienda poner atención a su uso, en particular se recomienda posicionar los subprocedimientos al termino del cuerpo de una función (después de la instrucción FRET) de manera tal de evitar la ejecución accidental del código del subprocedimiento como si fuera parte integrante del código principal. El resultado de esta situación es en la mejor de las hipótesis un error de sistema, en otros casos se obtienen comportamientos anómalos de la máquina cuya causa es de difícil individuación.

DELONFLAG

Sintaxis

DELONFLAG **nombreindicador**

Argumentos

nombreindicador nombre de dispositivo indicador

Descripción

Deshabilita la gestión de la interrupción software en el estado de un bit indicador o de un interruptor indicador que había sido precedentemente habilitada con [ONFLAG](#).

DELONINPUT

Sintaxis

DELONINPUT **nombrentrada**

Argumentos

nombrentrada nombre entrada

Descripción

Deshabilita la gestión de la interrupción software en el estado de una entrada precedentemente habilitata con la instrucción [ONINPUT](#).

FCALL

Sintaxis

[FCALL] **nombrefunción [, parámetros]**
nombrefunción **[parámetros]**

Argumentos

nombrefunción nombre de la función de volver a llamar
parámetros eventuales parámetros pasados a la función

Descripción

Efectúa una llamada a función, o sea es ejecutada la función **nombrefunción**.

A la función pueden ser pasados eventuales **parámetros**. Estos deben de corresponder en número y tipo a los declarados en la función llamada.

La ejecución de la función llamante (en la cual es ejecutada la FCALL) vuelve a partir al término de la ejecución de la función llamada (la que está especificada por el parámetro **nombrefunción**).

Note la diferencia con la instrucción **STARTTASK** que pone en ejecución otra función en paralelo a la llamante (utilizada para tener más tareas en ejecución contemporánea).

Ejemplo

[Ejecución Secuencial / Paralela](#)

FOR/NEXT**Sintaxis**

```
FOR           índice, inicio, fin [, step]
  instrucción
  instrucción
  ...
NEXT
```

Argumentos

índice	variable o nombrecontador
inicio	constante o variable o nombrecontador. Valor inicial
fin	constante o variable o nombrecontador. Valor final
step	constante o variable o nombrecontador. Paso de incremento o decremento

Descripción

Repite ciclicamente la ejecución de las instrucciones comprendidas entre la instrucción FOR y la instrucción NEXT.

Durante el primer ciclo la variable **índice** es inicializada al valor de la variable **inicio**. Al segundo ciclo la variable **índice** tendrá valor igual a (**inicio + step**), y continúa así hasta que la variable **índice** sea mayor, (o menor en el caso de valor negativo de la variable **step**), de la variable **fin**. Si la variable **step** está omitida, se asume un valor predeterminado igual a +1.

Las instrucciones comprendidas entre FOR y NEXT pueden modificar el número de las repeticiones modificando **índice**.

Cuando las repeticiones son terminadas es ejecutada la instrucción sucesiva a NEXT.

Ejemplo

```
Function Loop
  local i As integer
  local vector[10] as integer

  For i,1,10
    SetVal i, vector[i]           ; llena los elementos del vector
                                ; con los números 1,2, .... 10
  Next
Fret

Function loop2
  local j As integer
  local vector[10] as integer

  For j,1,10,2
    SetVal 27, vector[j]         ; inserta el valor 27 en los
                                ; siguientes elementos del vector: 1,3,5,7,9
  Next
Fret
```

FRET**Sintaxis**

```
FRET
```

Argumentos

ningún argumento

Descripción

Retorno desde una función. Causa la terminación de la ejecución de la función y la suelta de la memoria asignada a las variables locales. Si la función había sido puesta en ejecución con una FCALL la ejecución de la función llamante se reanudada con la instrucción sucesiva.

Si en precedencia habían sido ejecutadas unas WAITTASK con la función corriente (en la cual es ejecutada la FRET) como argumento, las tareas en espera son desbloqueadas.

GOTO

Sintaxis

GOTO **etiqueta**

Argumentos

etiqueta etiqueta

Descripción

Efectúa un salto incondicionado a la etiqueta especificada por el parámetro **etiqueta**.

Una etiqueta es definida con una palabra clave seguida inmediatamente por el carácter ":".

La etiqueta debe de encontrarse al interno del cuerpo de la función en la cual es ejecutada la instrucción GOTO.

Atención

El cuerpo de una función es comprendido entre la instrucción FUNCTION que declara el nombre de la función y la análoga instrucción que define la función sucesiva (o bien desde la fin del archivo). De esto se deduce la posibilidad de ejecutar saltos desde el cuerpo principal de la función a eventuales subprocedimientos (Véase instrucciones [CALL](#) y [RET](#)). Se desaconseja vivamente este estilo de programación, porque es fuente de numerosos errores de difícil individuación.

Ejemplo

```
; Function que hace parpadear un indicador
; (por ej. una espia de alarma en un cuadro sinoptico)
```

Function Loop

```
loop:
  setflag   alarma   ; activa el indicador
  delay    1
  resetflag alarma   ; desactiva el indicador
  delay    1
  goto     loop
  fret
```

IF/IFVALUE/IF-THEN-ELSE

Sintaxis

IF **nombrevar, operador de comparación, valor, GOTO etiqueta**

IF **nombrevar, operador de comparación, valor, CALL**

IF **nombresubprograma**

IF **nombrevar, operador de comparación, valor, nombrefunción**

IF **nombrevar, operador de comparación, valor THEN**

instrucción

instrucción

...

ENDIF

IF **nombrevar, operador de comparación, valor THEN**

instrucción

instrucción

...

ELSE

instrucción

instrucción

...

ENDIF

Argumentos

- nombrevar** constante o variable o nombredispositivo
- operador de comparación** los símbolos que pueden ser utilizados para ejecutar la comparación son:
 - < (menor) = (igual)
 - > (mayor) =< (menor igual)
 - >= (mayor igual) <> (diverso)
- valor** constante o variable o nombredispositivo
- etiqueta** nombre de etiqueta a la cual saltar
- nombresubprograma** nombre del subprograma
- nombrefunción** nombre de la función

Descripción

Las instrucciones IF y IFVALUE son sinónimos. Se recomienda el uso de la forma breve. La instrucción permite efectuar un comparación entre **nombrevar** y **valor** y en base al resultado de la comparación ejecutar una acción. En las primeras tres formas si la comparación es positiva es efectuado un salto a etiqueta (GOTO) o bien es efectuada una llamada a subprograma (CALL) o una llamada a función (nombrefunción). Al termino de la ejecución de la función o del subprograma llamado la ejecución se reanudada desde la línea sucesiva. Si a cambio la comparación es negativa la ejecución del programa sigue. La construcción IF...THEN permite ejecutar una o más instrucciones de manera condicionada. Las instrucciones comprendidas entre las palabras clave THEN y ENDIF son ejecutadas si la comparación entre **nombrevar** y **valor** es positiva. La construcción IF...THEN...ELSE permite definir dos bloques de instrucciones, uno solo de los cuales será ejecutado. Si la comparación entre **nombrevar** y **valor** es positiva, son ejecutadas las instrucciones comprendidas entre las palabras clave THEN y ELSE, si es negativa son ejecutadas las instrucciones comprendidas entre las palabras clave ELSE y ENDIF. En ambos casos la ejecución sigue con la instrucción sucesiva a ENDIF.

Atención

IFVALUE es mantenido por compatibilidad con las versiones precedentes del GPL.

IFACC

Sintaxis

IFACC	eje, GOTO etiqueta
IFACC	eje, CALL nombresubprograma
IFACC	eje, nombrefunción

Argumentos

eje	nombre del dispositivo tipo eje
etiqueta	nombre de la etiqueta a la cual saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Comprueba si el eje definido por la variable **eje** está en el estado de aceleración. Si la condición es verificada, salta a **etiqueta** o llama **nombresubprograma** o **nombrefunción**.

IFAND

Sintaxis

IFAND	operando1, operando2, valorverificación, GOTO etiqueta
IFAND	operando1, operando2, valorverificación, CALL nombresubprograma
IFAND	operando1, operando2, valorverificación, nombrefunción

IFAND	operando1, operando2, valorverificación THEN
instrucción	
instrucción	
...	
ENDIF	

IFAND	operando1, operando2, valorverificación THEN
instrucción	
instrucción	
...	
ELSE	


```

    instrucción
    instrucción
    ...
ENDIF

```

Argumentos

operando1	constante o variable o nombredispositivo
operando2	constante o variable o nombredispositivo
valorverificación	constante. valor utilizado para la verificación del resultado de la operación. Puede asumir los siguientes valores: TRUE 1, FALSE 0
etiqueta	nombre de la etiqueta a la cual saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Son efectuadas dos comparaciones, la primera entre **operando1** y **operando2**, la segunda entre el resultado de la primera comparación y **valorverificación**.
 La primera comparación consiste en un AND binario entre **operando1** y **operando2**. Los dos operandos son considerados como máscaras de bit. Si el resultado del AND binario tiene por lo menos un bit diverso de 0, el resultado de la primera comparación es TRUE. Esto a su vez es comparado con **valorverificación**. Si los dos valores coinciden es efectuado el salto a etiqueta o la llamada a función o subprograma. Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFBIT**Sintaxis**

```

IFBIT máscara, nbit, estado, GOTO etiqueta
IFBIT máscara, nbit, estado, CALL nombresubprograma
IFBIT máscara, nbit, estado, nombrefunción

```

```

IFBIT máscara, nbit, estado THEN

```

```

    instrucción
    instrucción
    ...
ENDIF

```

```

IFBIT máscara, nbit, estado THEN

```

```

    instrucción
    instrucción
    ...

```

```

ELSE
    instrucción
    instrucción
    ...

```

```

ENDIF

```

Argumentos

máscara	constante o variable entera o nombrecontador o nombrepuerto. Valor de verificar
nbit	constante o variable entera o nombrecontador. Número del bit (1÷32)
estado	constante predefinida. Estado por verificar en máscara. Los valores admitidos son: ON bit elegido a 1 OFF bit elegido a 0
etiqueta	etiqueta de salto (GOTO)
nombresubprograma	llamada subprograma (CALL)
nombrefunción	nombre de la función

Descripción

Test sobre un solo bit de la **máscara** de bit pasada. Hay que el argumento **máscara** pueda corresponder a un valor entero con un máximo de 32 bit. El número de asignar a la variable **nbit** para identificar el bit por verificar va de 1 a 32. Si la condición indicada en estado es verificada, salta a **etiqueta** o llama **nombresubprograma** o **nombrefunción**. Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFBLACKBOX**Sintaxis**

```

IFBLACKBOX GOTO etiqueta
IFBLACKBOX CALL nombresubprograma
IFBLACKBOX nombrefunción

```

Argumentos

etiqueta nombre de la etiqueta donde saltar
nombresubprograma nombre del subprograma
nombrefunción nombre de la función

Descripción

Si el registro está activo, saltar a **etiqueta** o llamar **nombresubprograma** o **nombrefunción**. Véase también [STARTBLACKBOX](#), [PAUSEBLACKBOX](#) y [ENDBLACKBOX](#).

IFCHANGEVEL

Sintaxis

IFCHANGEVEL eje [, estado], **GOTO** etiqueta
IFCHANGEVEL eje [, estado], **CALL** nombresubprograma
IFCHANGEVEL eje [, estado], nombrefunción

Argumentos

eje nombre del dispositivo tipo eje
estado tipo de variación. Los valores admitidos son: **POSITIVE**, **NEGATIVE**
etiqueta nombre de la etiqueta a la cual saltar
nombresubprograma nombre del subprograma
nombrefunción nombre de la función

Descripción

Comprueba si se ha verificado una variación de velocidad de un eje.
 Si el eje definido por la variable **eje** está efectuando un cambio de velocidad durante un desplazamiento, es efectuado un salto a **etiqueta** o una llamada a **nombresubprograma** o **nombrefunción**.
 El parámetro **estado** especifica si la velocidad ha aumentado (**POSITIVE**) o disminuido (**NEGATIVE**).

IFCOUNTER

Sintaxis

IFCOUNTER nombrecontador, operador de comparación, valor, **GOTO** etiqueta
IFCOUNTER nombrecontador, operador de comparación, valor, **CALL** nombresubprograma
IFCOUNTER nombrecontador, operador de comparación, valor, nombrefunción

IFCOUNTER nombrecontador, operador de comparación, valor **THEN**
 instrucción
 instrucción
 ...
ENDIF

IFCOUNTER nombrecontador, operador de comparación, valor **THEN**
 instrucción
 instrucción
 ...
ELSE
 instrucción
 instrucción
 ...
ENDIF

Argumentos

nombrecontador nombre del contador
operador de comparación los símbolos que pueden ser utilizados para ejecutar la comparación son:
 < (menor) = (igual)
 > (mayor) =< (menor igual)
 >= (mayor igual) <> (diverso)
valor constante o variable o nombrecontador
etiqueta nombre de la etiqueta a la cual saltar
nombresubprograma nombre del subprograma
nombrefunción nombre de la función

Descripción

La instrucción ejecuta el test de un contador.

Si el contenido del contador definido en la variable **nombrecontador** verifica la condición especificada por el **operador de comparación** con el valor expreso por la variable **valor**, se efectua un salto a la etiqueta especificada por **etiqueta** o una llamada al subprograma definido por **nombresubprograma** o a la función definida por **nombrefunción**.

Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFDEC

Sintaxis

IFDEC	eje, GOTO etiqueta
IFDEC	eje, CALL nombresubprograma
IFDEC	eje, nombrefunción

Argumentos

eje	nombre del dispositivo tipo eje
etiqueta	nombre de la etiqueta a la cual saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Comprueba si el eje definido por la variable **eje** está en el estado de deceleración.

Si la condición es verificada, salta a **etiqueta** o llama **nombresubprograma** o **nombrefunción**.

IFDIR

Sintaxis

IFDIR	eje, dirección, GOTO etiqueta
IFDIR	eje, dirección, CALL nombresubprograma
IFDIR	eje, dirección, nombrefunción

IFDIR	eje, dirección THEN
-------	---------------------

```
instrucción
instrucción
```

```
...
```

```
ENDIF
```

IFDIR	eje, dirección THEN
-------	---------------------

```
instrucción
instrucción
```

```
...
```

```
ELSE
```

```
instrucción
instrucción
```

```
...
```

```
ENDIF
```

Argumentos

eje	nombre de dispositivo tipo eje
dirección	dirección eje. Los valores admitidos son: POSITIVE dirección eje positiva NEGATIVE dirección eje negativa
etiqueta	nombre de la etiqueta a la cual saltar
nombresubprograma	nombre de subprograma
nombrefunción	nombre de función

Descripción

Test sobre la dirección corriente de un eje.

Si el **eje** se mueve en la dirección especificada por la variable **dirección**, es efectuado un salto a **etiqueta** o una llamada a **nombresubprograma** o **nombrefunción**.

Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFERRAN

Sintaxis

IFERRAN	eje, operador de comparación, valor, GOTO etiqueta
IFERRAN	eje, operador de comparación, valor, CALL nombresubprograma
IFERRAN	eje, operador de comparación, valor, nombrefunción

IFERRAN	eje, operador de comparación, valor THEN
---------	--

```
instrucción
instrucción
```

```

...
ENDIF

IFERRAN eje, operador de comparación, valor THEN
    instrucción
    instrucción
...
ELSE
    instrucción
    instrucción
...
ENDIF

```

Argumentos

eje	nombre del dispositivo tipo eje
operador de comparación	los símbolos que pueden ser utilizados para ejecutar la comparación son: < (menor) = (igual) > (mayor) =< (menor igual) >= (mayor igual) <> (diverso)
valor	constante o variable o nombrecontador
etiqueta	nombre de la etiqueta a la cual saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Test sobre el valor del error de persecución (error de anillo) del eje definido en la variable **eje**. Si el error de anillo del **eje** verifica la condición expresada por el **operador de comparación** con el valor expreso por **valor**, se efectúa un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**. Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFERROR**Sintaxis**

```

IFERROR número, IDposic, GOTO etiqueta
IFERROR número, IDposic, CALL etiqueta
IFERROR número, IDposic, nombrefunción
IFERROR nombredispositivo, estado, IDposic, GOTO etiqueta
IFERROR nombredispositivo, estado, IDposic, CALL etiqueta
IFERROR nombredispositivo, estado, IDposic, nombrefunción

```

Argumentos

número	DEFMSG o constante o variable
nombredispositivo	nombredispositivo
estado	constante predefinida. Puede asumir los valores siguientes: ON, OFF
IDposic	constante o variable. valor numérico usado en los sinópticos.
etiqueta	nombre de la etiqueta a la que saltar
nombrefunción	nombre función

Descripción

Test sobre el error de ciclo activo. Si el error de ciclo, identificado por **número** y **IDposic** o por **nombredispositivo**, **estado** y **IDposic** está activo, se efectúa un salto a **etiqueta** o una llamada a la función **nombrefunción**. El parámetro **número** puede identificar un error de ciclo de módulo (por lo tanto un valor numérico entero) o de grupo (en este caso se usa una DEFMSG). El parámetro **nombredispositivo** es el nombre de un dispositivo y el parámetro **estado** representa el estado ON/OFF en el que debe encontrarse el dispositivo en el momento en que genera el error. El parámetro **IDposic** es un parámetro opcional que especifica el valor numérico utilizado en los sinópticos para clasificar errores de ciclo en casillas diferentes. Debe corresponder al valor especificado en el constructor sinópticos para aquella casilla de visualización en concreto. Si no es necesario indicar una casilla específica se asigna la constante predefinida NOPLACE. El rango de valores que se puede configurar oscila de 0 (NOPLACE) a 1023. Si se utiliza la instrucción sin que la gestión de las alarmas de estado se haya activado, se produce un error de sistema. Véase también la instrucción [ERROR](#).

IFFLAG

Sintaxis

IFFLAG nombreindicador, estado, GOTO etiqueta
IFFLAG nombreindicador, estado, CALL nombresubprograma
IFFLAG nombreindicador, estado, nombrefunción

IFFLAG nombreindicador, estado THEN
instrucción
instrucción
...
ENDIF

IFFLAG nombreindicador, estado THEN
instrucción
instrucción
...
ELSE instrucción
instrucción
...
ENDIF

Argumentos

nombreindicador nombre de dispositivo indicador
estado constante predefinida. Estado de verificar. Los valores admitidos son:
ON activo
OFF desactivo
etiqueta nombre de la etiqueta a la cual saltar
nombresubprograma nombre del subprograma
nombrefunción nombre de la función

Descripción

Test del estado lógico de un indicador.

Si el indicador definido por la variable **nombreindicador** está en el **estado** indicado, es efectuado un salto a etiqueta o una llamada a **nombresubprograma** o a **nombrefunción**.

Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFINPUT

Sintaxis

IFINPUT nombreentrada, estado, GOTO etiqueta
IFINPUT nombreentrada, estado, CALL nombresubprograma
IFINPUT nombreentrada, estado, nombrefunción

IFINPUT nombreentrada, estado THEN
instrucción
instrucción
...
ENDIF

IFINPUT nombreentrada, estado THEN
instrucción
instrucción
...
ELSE instrucción
instrucción
...
ENDIF

Argumentos

nombreentrada nombre entrada
estado constante predefinida. Estado que verificar
Los valores admitidos son:
- **ON** activo
- **OFF** desactivo
etiqueta nombre de la etiqueta a la cual saltar
nombresubprograma nombre del subprograma
nombrefunción nombre de la función

Descripción

Test del estado lógico de una entrada.

Si la entrada definida en la variable **nombreentrada** se encuentra en el **estado** indicado, es efectuado un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.

Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFMESSAGE**Sintaxis**

IFMESSAGE **número, IDposic, GOTO etiqueta**
IFMESSAGE **número, IDposic, CALL etiqueta**
IFMESSAGE **número, IDposic, nombrefunción**

Argumentos

número DEFMSG o constante o variable
IDposic constante o variable. Valor numérico usado en los sinópticos
etiqueta nombre de la etiqueta a la que saltar
nombrefunción nombre función

Descripción

Test sobre el mensaje activo.

Si el mensaje, identificado por **número** e **IDposic** es activo se efectúa un salto a **etiqueta** o una llamada a la función **nombrefunción**.

El parámetro **IDposic** es un parámetro opcional que especifica el valor numérico utilizado en los sinópticos para clasificar errores de ciclo en casillas diferentes. Debe corresponder al valor especificado en el constructor sinópticos para aquella casilla de visualización en concreto. Si no es necesario indicar una casilla específica se asigna la constante predefinida NOPLACE. El rango de valores que se puede configurar oscila de 0 (NOPLACE) a 1023.

Si se utiliza la instrucción sin que la gestión de las alarmas de estado se haya activado, se produce un error de sistema.

Véase también la instrucción [MESSAGE](#).

IFOR**Sintaxis**

IFOR **operando1, operando2, valorverificación, GOTO etiqueta**
IFOR **operando1, operando2, valorverificación, CALL nombresubprograma**
IFOR **operando1, operando2, valorverificación, nombrefunción**

IFOR **operando1, operando2, valorverificación THEN**

instrucción
 instrucción

...

ENDIF

IFOR **operando1, operando2, valorverificación THEN**

instrucción
 instrucción

...

ELSE

instrucción
 instrucción

...

ENDIF

Argumentos

operando1 constante o variable o nombredispositivo
operando2 constante o variable o nombredispositivo
valorverificación constante. Es el valor utilizado para la verificación del resultado de la operación
Puede asumir los siguientes valores:
TRUE 1
FALSE 0
etiqueta nombre de la etiqueta a la que saltar
nombresubprograma nombre del subprograma
nombrefunción nombre de la función

Descripción

Son efectuadas dos comparaciones, la primera entre **operando1** y **operando2**, la segunda entre el resultado de la primera comparación y **valorverificación**.

operador de comparación	los símbolos que pueden ser usados para ejecutar la comparación son: < (menor) = (igual) > (mayor) =< (menor igual) >= (mayor igual) <> (diferente)
valor	constante o variable o nombrecontador
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Test sobre cota real expresa por la variable **eje**.

Si el valor de la variable **eje** verifica la condición expresa por el **operador de comparación** con el valor expreso por **valor**, se efectua un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.

Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFQUOTET**Sintaxis**

IFQUOTET **eje, operador de comparación, valor, GOTO etiqueta**
IFQUOTET **eje, operador de comparación, valor, CALL nombresubprograma**
IFQUOTET **eje, operador de comparación, valor, nombrefunción**

IFQUOTET **eje, operador de comparación, valor THEN**
 instrucción
 instrucción
 ...
ENDIF

IFQUOTET **eje, operador de comparación, valor THEN**
 instrucción
 instrucción
 ...
ELSE
 instrucción
 instrucción
 ...
ENDIF

Argumentos

eje	nombre del dispositivo tipo eje
operador de comparación	los símbolos que pueden ser usados para ejecutar la comparación son: < (menor) = (igual) > (mayor) =< (menor igual) >= (mayor igual) <> (diverso)
valor	constante o variable o nombrecontador
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Test sobre cota teórica expresa por la variable **eje**.

Si el valor de la variable **eje** verifica la condición expresa por el **operador de comparación** con el valor expreso por **valor**, es efectuado un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.

Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFRECEIVED**Sintaxis**

IFRECEIVED **[fuente,] identificador, GOTO etiqueta**
IFRECEIVED **[fuente,] identificador, CALL nombresubprograma**
IFRECEIVED **[fuente,] identificador, nombrefunción**

Argumentos

fuentes	constante cadena
identificador	constante cadena
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Verifica si una instrucción [RECEIVE](#) ha sido satisfecha.

Si una particular RECEIVE precedente ha sido satisfecha, es efectuado un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.

Véase también instrucciones [RECEIVE](#), [WAITRECEIVE](#), [SEND](#).

IFREG**Sintaxis**

IFREG	eje, GOTO etiqueta
IFREG	eje, CALL nombresubprograma
IFREG	eje, nombrefunción

Argumentos

eje	nombre del dispositivo tipo eje
etiqueta	nombre de la etiqueta a la cual saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Comprueba si el eje definido por la variable **eje** se encuentra en el estado de régimen.

Si la condición es verificada, salta a **etiqueta** o llama **nombresubprograma** o **nombrefunción**.

IFSAME**Sintaxis**

IFSAME	operando1, operando2, GOTO etiqueta
IFSAME	operando1, operando2, CALL nombresubprograma
IFSAME	operando1, operando2, nombrefunción

Argumentos

operando1	variable o nombredispositivo
operando2	variable o nombredispositivo
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Test entre dos operandos.

Verifica que el valor definido en **operando1** y **operando2** hagan referencia o al mismo dispositivo o a la misma área de memoria.

Si el test entre los dos operandos es verificado, se efectua un salto a **etiqueta** o llama **nombresubprograma** o **nombrefunción**.

IFSTILL**Sintaxis**

IFSTILL	eje, GOTO etiqueta
IFSTILL	eje, CALL nombresubprograma
IFSTILL	eje, nombrefunción

Argumentos

eje	nombre del dispositivo tipo eje
etiqueta	nombre de la etiqueta a la cual saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Test si el eje definido por la variable **eje** está parado, es decir, en "cota".

Si la condición es verificada, salta a **etiqueta** o llama **nombresubprograma** o **nombrefunción**.

Véase también [IFTARGET](#) y [IFWIN](#).

IFSTR**Sintaxis**

IFSTR	cadena1, operador de comparación, cadena2, GOTO etiqueta
IFSTR	cadena1, operador de comparación, cadena2, CALL nombresubprograma
IFSTR	cadena1, operador de comparación, cadena2, nombrefunción
IFSTR	cadena1, operador de comparación, cadena2 THEN

```

    instrucción
    instrucción
    ...
ENDIF

IFSTR          cadena1, operador de comparación, cadena2 THEN
    instrucción
    instrucción
    ...
ELSE
    instrucción
    instrucción
    ...
ENDIF

```

Argumentos

cadena1	variable string. Es la primera cadena ASCII
operador de comparación	los símbolos que pueden ser usados para ejecutar la comparación entre cadenas son: < (menor) = (igual) > (mayor) =< (menor igual) >= (mayor igual) <> (diferente)
cadena2	variable string. Es la segunda cadena ASCII
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Test sobre cadenas ASCII.

Si la cadena definida en **cadena1** verifica la condición expresa por el **operador de comparación** con la cadena contenida en **cadena2**, se efectúa un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.

Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFTARGET**Sintaxis**

```

IFTARGET      eje, GOTO etiqueta
IFTARGET      eje, CALL nombresubprograma
IFTARGET      eje, nombrefunción

```

Argumentos

eje	nombre del dispositivo tipo eje
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Comprueba si el eje definido por la variable **eje** ha alcanzado la cota final programada. El eje, aunque si ha alcanzado la cota teórica final, no está necesariamente parado, en general tendrá que recuperar el error de anillo. Si la condición es verificada, salta a **etiqueta** o llama **nombresubprograma** o **nombrefunción**.

Véase también [IFSTILL](#) y [IFWIN](#).

IFTASKHOLD**Sintaxis**

```

IFTASKHOLD   nombretarea, GOTO etiqueta
IFTASKHOLD   nombretarea, CALL etiqueta
IFTASKHOLD   nombretarea, nombrefunción

```

Argumentos

nombretarea	nombre tarea paralela
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Verifica si la tarea está suspendida (estado hold).

Si la tarea **nombretarea** está suspendida, se efectúa un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.

IFTASKRUN

Sintaxis

IFTASKRUN	nombretarea, GOTO etiqueta
IFTASKRUN	nombretarea, CALL nombresubprograma
IFTASKRUN	nombretarea, nombrefunción

Argumentos

nombretarea	nombre tarea paralela
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Verifica si la tarea está en ejecución.

Si la tarea definida en **nombretarea** está en ejecución, se efectua un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.

IFTIMER

Sintaxis

IFTIMER	nombretemporizador, operador de comparación, valor, GOTO etiqueta
---------	--

IFTIMER	nombretemporizador, operador de comparación, valor, CALL nombresubprograma
---------	---

IFTIMER	nombretemporizador, operador de comparación, valor, nombrefunción
---------	--

IFTIMER	nombretemporizador, operador de comparación, valor THEN
---------	--

```

instrucción
instrucción
...

```

ENDIF

IFTIMER	nombretemporizador, operador de comparación, valor THEN
---------	--

```

instrucción
instrucción
...

```

ELSE

```

instrucción
instrucción
...

```

ENDIF

Argumentos

nombretimer	nombre de dispositivo temporizador
operador de comparación	los símbolos que pueden ser usados para ejecutar la comparación son:

valor	constante o variable o nombre temporizador. Es el valor en el cual ejecutar la comparación
--------------	--

etiqueta	nombre de la etiqueta a la que saltar
-----------------	---------------------------------------

nombresubprograma	nombre del subprograma
--------------------------	------------------------

nombrefunción	nombre de la función
----------------------	----------------------

Descripción

Test de un temporizador.

Si el contenido del temporizador **nombretemporizador** averigua la condición expresa por el **operador de comparación** con el valor expreso por **valor**, se efectua un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.

Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFVEL

Sintaxis

IFVEL	eje, operador de comparación, valor, GOTO etiqueta
-------	---

IFVEL	eje, operador de comparación, valor, CALL nombresubprograma
-------	--

IFVEL	eje, operador de comparación, valor, nombrefunción
-------	---

IFVEL	eje, operador de comparación, valor THEN
-------	---

```

    instrucción
    instrucción
    ...
ENDIF

IFVEL           eje, operador de comparación, valor THEN
    instrucción
    instrucción
    ...
ELSE
    instrucción
    instrucción
    ...
ENDIF

```

Argumentos

eje nombre del dispositivo tipo eje
operador de comparación los símbolos que pueden ser usados para ejecutar la comparación son:
 < (menor) = (igual)
 > (mayor) =< (menor igual)
 >= (mayor igual) <> (diverso)
etiqueta nombre de la etiqueta a la que saltar
nombresubprograma nombre del subprograma
nombrefunción nombre de la función

Descripción

Test sobre la velocidad corriente de un eje.
 Si la velocidad del eje verifica la condición expresa por el **operador de comparación** con el valor expreso por **valor**, se efectua un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.
 Para mayores detalles, véase la construcción [IF-THEN-ELSE](#).

IFWIN

Sintaxis

```

IFWIN           eje, GOTO etiqueta
IFWIN           eje, CALL nombresubprograma
IFWIN           eje, nombrefunción

```

Argumentos

eje nombre del dispositivo tipo eje
etiqueta nombre de la etiqueta a la cual saltar
nombresubprograma nombre del subprograma
nombrefunción nombre de la función

Descripción

Comprueba si el eje definido por la variable **eje** entró al interno de la ventana de llegada en cota (Véase [Convenciones y terminología](#)).
 Si la condición es verificada, salta a **etiqueta** o llama **nombresubprograma** o **nombrefunción**.
 Véase también [IFTARGET](#) y [IFSTILL](#).

IFXOR

Sintaxis

```

IFXOR           operando1, operando2, valorverificación, GOTO etiqueta
IFXOR           operando1, operando2, valorverificación, CALL nombresubprograma
IFXOR           operando1, operando2, valorverificación, nombrefunción

```

```

IFXOR           operando1, operando2, valorverificación THEN
    instrucción
    instrucción
    ...
ENDIF

```

```

IFXOR           operando1, operando2, valorverificación THEN
    instrucción
    instrucción
    ...
ELSE
    instrucción
    instrucción
    ...

```


Habilita la interrupción software, asociada al estado del indicador especificado, de la tarea en la cual es ejecutado. Cuando el indicador conmuta en el estado indicado (interrupt), la ejecución de la tarea es interrumpida y es puesta en ejecución la función especificada por **nombrefunción**. Al termino de esta la ejecución de la tarea es reanudada desde donde había sido interrumpida. La función ejecutada al verificarse del interrupt es sujeta a unas limitaciones. En particular no todas las instrucciones GPL pueden aparecer en el cuerpo de la función. Esta limitación tiene el fin de evitar situaciones de bloqueo critico del código GPL o de largas esperas. Véase [la lista de las instrucciones no ejecutables en interrupt](#).

Si el argumento **estado** está omitido, la función es llamada en cada cambio de estado del indicador.

El test sobre el estado del indicador es ejecutado cada 5 ms, por lo tanto desde la variación del indicador a la ejecución de la función se puede haber un tiempo máximo de latencia de 5 ms.

No se pueden definir más ONFLAG en el mismo indicador.

Como argumentos de la función definida en **nombrefunción** no pueden ser utilizados vectores o matrices locales.

Véase también las instrucciones [DELONFLAG](#), [ONINPUT](#), [DELONINPUT](#).

ONINPUT

Sintaxis

ONINPUT **nombrentrada**, [**estado**,] **nombrefunción** [,**argumentos**]

Argumentos

nombrentrada	nombre entrada
estado	constante predefinida. Estado que verificar Los valores admitidos son: ON activo OFF desactivo
nombrefunción	nombre función
argumentos	eventuales argumentos de la función

Descripción

Habilita la interrupción software, asociada al estado de la entrada especificada, de la tarea en la cual es ejecutado. Cuando la entrada conmuta en el estado indicado (interrupt), la ejecución de la tarea es interrumpida y es puesta en ejecución la función especificada por **nombrefunción**. Al termino de esta la ejecución de la tarea se reanudada desde donde había sido interrumpida. La función ejecutada al verificarse del interrupt es sujeta a unas limitaciones. En particular no todas las instrucciones GPL pueden aparecer en el cuerpo de la función. Esta limitación tiene el fin de evitar situaciones de bloqueo critico del código GPL o de largas esperas. Véase [la lista de las instrucciones no ejecutables en interrupt](#).

Si el argumento **estado** está omitido, se llama la función para cualquier cambio de estado de la entrada.

El test sobre el estado de la entrada es ejecutado cada 5 ms a los cuales se suman 4 ms de filtro antirebote sobre la gestión entradas, se puede por lo tanto tener un tiempo de latencia de 9 ms antes del inicio de la función.

No se pueden definir más ONINPUT en la misma entrada.

Véase también las instrucciones [DELONINPUT](#), [ONFLAG](#) y [DELONFLAG](#).

REPEAT/ENDREP

Sintaxis

```
REPEAT valor
  instrucción
  instrucción
  ...
ENDREP
```

Argumentos

valor constante o variable o nombrecontador. Número de repeticiones

Descripción

Repite la ejecución de las instrucciones comprendidas entre la instrucción REPEAT y la instrucción ENDREP por un número de veces indicado por la variable **valor**.

Cuando el programa encuentra la instrucción ENDREP el contador del número de repeticiones es decrementado y si no es menor o igual a cero el bloqueo de instrucciones es vuelto a ejecutar comenzando por la instrucción en la línea sucesiva al REPEAT. Las instrucciones son por lo tanto ejecutadas por lo menos una vez (aunque si el parámetro valor es en principio nulo o negativo).

Cuando las repeticiones son terminadas es ejecutada la instrucción sucesiva a ENDREP.

Véase también la instrucción [FOR/NEXT](#).

Ejemplo

```
; Ejemplo de ciclo che movimienta un eje
; entre dos cotas por 10 veces
Function Ciclo
```

```

Repeat 10
  MovAbs     eje,100
  waitinput  interruptor,ON
  Movabs     eje,-100
  waitinput  interruptor, OFF
EndRep
Fret

```

RET

Sintaxis

RET

Argumentos

ningún argumento

Descripción

Termina la ejecución del subprograma con retorno a la instrucción inmediatamente sucesiva a la CALL de llamada.

Véase también la instrucción [CALL](#).

Atención

Esta instrucción, junto con la CALL, es una típica fuente de errores de programación. Se recomienda poner atención a su uso, en particular se recomienda posicionar los subprocedimientos al termino del cuerpo de una función (después de la instrucción FRET) de manera tal de evitar la ejecución accidental del código del subprocedimiento como si fuera parte integrante del código principal. El resultado de esta situación es en la mejor de las hipótesis un error de sistema, en otros casos se obtienen comportamientos anómalos de la máquina cuya causa es de difícil individuación.

SELECT

Sintaxis

```

SELECT nombrevar
  CASE valor
    instrucción
  CASE valor1 TO valor2
    instrucción
  CASE IS <=> valor
    instrucción
  CASE ELSE
    instrucción
ENDSELECT

```

Se necesita reemplazar **instrucción** por uno de los valores siguientes:

```

GOTO etiqueta
CALL nombresubprograma
[FCALL] nombrefunción [parámetro1,...parámetroN]
[EXPR] variable = expresión
[EXPR] dispositivo = expresión

```

Argumentos

nombrevar	constante o variable entera o nombrecontador
valor, valor1, valor2	constantes enteras
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función
parámetro1...parámetroN	parámetro pasado a la función llamada
variable	nombre de variable
dispositivo	nombre de dispositivo
expresión	conjunto de operadores

Descripción

Selección múltiple según el **valor** de la variable **nombrevar**. Se ejecuta el código que se encuentra en el CASE de la condición comprobada. El ramo CASE-ELSE es ejecutado si ningún CASE precedente ha sido satisfecho. Por cada CASE (opcional) puede haber una sola instrucción [GOTO](#), [CALL](#) o [FCALL](#) o [EXPR](#). Se necesita por lo menos un CASE entre SELECT y ENDSELECT. Esta última señala el termino de la instrucción SELECT.

Después de cada CALL o FCALL o EXPR la ejecución de la función pasa a la instrucción sucesiva a ENDSELECT.

Ejemplo[Servidor de desplazamiento ejes](#)**TESTIPC****Sintaxis**

TESTIPC	nombreIPC, [, nombrevar1 [, nombrevarN, ...]], GOTO etiqueta
TESTIPC	nombreIPC, [, nombrevar1 [, nombrevarN, ...]], CALL nombresubprograma
TESTIPC	nombreIPC, [, nombrevar1 [, nombrevarN, ...]], nombrefunción
TESTIPC	nombreIPC, matriz[línea], GOTO etiqueta
TESTIPC	nombreIPC, matriz[línea], CALL nombresubprograma
TESTIPC	nombreIPC, matriz[línea], nombrefunción
TESTIPC	nombreIPC, vector, GOTO etiqueta
TESTIPC	nombreIPC, vector, CALL nombresubprograma
TESTIPC	nombreIPC, vector, nombrefunción

Argumentos

nombreIPC	constante cadena. Nombre de la IPC
nombrevar1[... nombreVarN]	constante o variable. Nombres variables 1÷N
matriz[línea]	constante o variable integer. Número de línea de la matriz
vector	nombre del vector
matriz	nombre de la matriz
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Test y recepción de una instrucción IPC.

La primera vez que es ejecutada una instrucción de TESTIPC es asignada la memoria compartida, cuya dimensión es calculada en base a la dimensión de los datos que son enviados. La máxima dimensión de la memoria tamponada es 64 Kb.

A la memoria compartida es asociado un semáforo que permite sincronizar la ejecución de las tareas que acceden a ella. La tarea que accede verifica la presencia de un semáforo activo, lee los datos desde la memoria compartida y desactiva el semáforo. En seguida es ejecutada la instrucción de salto a etiqueta, o la función o el subprograma descritos como último parámetro della instrucción TESTIPC.

Véase también [SENDIPC](#) y [WAITIPC](#).

TESTMAIL**Sintaxis**

TESTMAIL	mail, [nombrevar1 [,...nombrevarN]], GOTO etiqueta
TESTMAIL	mail, [nombrevar1 [,...nombrevarN]], CALL etiqueta
TESTMAIL	mail, [nombrevar1 [,...nombrevarN]], nombrefunción
TESTMAIL	mail, matriz[línea], GOTO etiqueta
TESTMAIL	mail, matriz[línea], CALL nombresubprograma
TESTMAIL	mail, matriz[línea], nombrefunción

Argumentos

mail	constante o variable integer (1÷256). Número del mailbox
[nombrevar1[,... nombrevarN]]	variable integer. Nombres variables 1÷20
matriz[línea]	constante o variable integer. Número de línea de la matriz
etiqueta	nombre de la etiqueta a la que saltar
nombresubprograma	nombre del subprograma
nombrefunción	nombre de la función

Descripción

Test y recepción de un mensaje.

La primera TESTMAIL en el mailbox **mail** crea el mailbox.

Si el mensaje está presente en la casilla de correos **mail**, los datos enviados junto con el mensaje son memorizados en las variables **nombrevar** (1÷20), solo si estas son indicadas, o en la línea de la matriz indicada por **matriz[línea]**; es, además, efectuado un salto a **etiqueta** o una llamada a **nombresubprograma** o a **nombrefunción**.

Durante la ejecución es controlada la congruencia entre los datos pasados y los que son esperados por la instrucción.

Véase también las instrucciones [SENDMAIL](#), [WAITMAIL](#) y [ENDMAIL](#).

10.3.13 Varias

CLEARERRORS

Sintaxis

CLEARERRORS [IDposic]

Argumentos

IDposic constante o variable. Es un valor numérico utilizado en los sinópticos

Descripción

Hace ejecutar al PC supervisor la cancelación de todos los errores de ciclo, relativos al módulo que ejecuta la instrucción, enviados precedentemente con la instrucción ERROR. El parámetro **IDposic** es un parámetro opcional que especifica el valor numérico utilizado en los sinópticos para clasificar errores de ciclo en casillas diferentes. Debe de corresponder al valor especificado en el constructor sinópticos para la particular casilla de visualización. Albatros utiliza este identificativo para gestionar los errores de ciclo en colas separadas. Para cada IDposic es creada una cola. Si el parámetro IDposic no está especificado el error de ciclo es puesto en la cola predeterminada que corresponde a un valor IDposic=0. El rango de valores que se puede configurar oscila de -1 (NOPLACE) a 1023. El rango de valores que se puede configurar oscila de -1 (NOPLACE) a 1023. Si el parámetro **IDposic** no está indicado, se cancelan todos los errores de ciclo tanto en la cola predeterminada como en las eventuales otras colas).

Véase también las instrucciones [ERROR](#) y [DELEERROR](#).

CLEARMESSAGES

Sintaxis

CLEARMESSAGES [IDposic]

Argumentos

IDposic constante o variable. Es un valor numérico utilizado en los sinópticos

Descripción

Hace ejecutar a la PC supervisor la cancelación de todos los mensajes, relativos al módulo que ejecuta la instrucción, enviados precedentemente con la instrucción MESSAGE. El parámetro **IDposic** es un parámetro opcional que especifica el valor numérico utilizado en los sinópticos para distribuir mensajes en casillas diferentes. Debe de corresponder al valor especificado en el constructor sinópticos para la particular casilla de visualización. Albatros utiliza este identificativo para gestionar los mensajes en colas separadas. Para cada IDposic se crea una cola. El rango de valores que se puede configurar oscila de -1 (NOPLACE) a 1023. Si el parámetro **IDposic** no está indicado, se cancelan todos los mensajes tanto en la cola predeterminada como en las eventuales otras colas.

Véase también las instrucciones [MESSAGE](#) y [DELMESSAGE](#).

DEFMSG

Sintaxis

DEFMSG etiqueta [, prefijoidioma1], "cadenamensaje" , ... , [, prefijoidiomaN, "cadenamensaje"]

Argumentos

etiqueta nombre mnemónico del mensaje que visualizar
prefijoidioma constante predefinida. Idioma en el cual está escrito el mensaje
cadenamensaje mensaje que visualizar. Hay que ser escrito entre comillas ("")

Descripción

Asigna una **etiqueta** a un mensaje. La instrucción DEFMSG debe de ser declarada antes de la implementación de las funciones. La definición del mensaje puede ser utilizada solo al interno del archivo (o grupo) en el cual está declarada. Es posible insertar mensajes en idiomas diferentes utilizando la constante predefinida **prefijoidioma** (véase el capítulo "[Importación mensajes](#)" para la lista de los prefijos de idioma). En este caso una instrucción MESSAGE visualiza el mensaje en el idioma correspondiente al que está corrientemente en uso para Albatros. Un mensaje al cual no está asociado un prefijo es utilizado cuando la lengua corrientemente en uso no corresponde a ninguno de los prefijos existentes.

Las etiquetas de los diferentes idiomas tienen que ser escritas o todas en la misma línea o bien en más líneas. Para cambiar línea se usa el carácter "_" precedido por un espacio.

La instrucción DEFMSG puede ser pasada como parámetro a una función. De esta manera la función que la recibe puede utilizarla como uno de los tres argumentos de ERROR y MESSAGE. (Véase ejemplo 2).

Véase también las instrucciones [MESSAGE](#), [DELMESSAGE](#), [ERROR](#), [DELEERROR](#).

Ejemplo 1

```

;asignación a una etiqueta de una cadena de mensaje
;sin selección de idioma
DEFMSG MSG_GRU_1 "mensaje grupo 1"

;empleo de la definición
MESSAGE MSG_GRU_1 ; muestra: "mensaje grupo 1"

;asignación a una etiqueta de una cadena de mensaje
;con selección de idioma
DEFMSG MSG_GRU_1 ESP "Mensaje grupo 1"
ENG "Message group 1"

;empleo de la definición cuando el idioma
;de Albatros es ENG
MESSAGE MSG_GRU_1 ; muestra: "Message group 1"

```

Ejemplo 2:

```

; En un grupo:
DEFMSG MSG_TEST "Error de ejecución"

FUNCTION LlamaTest
  Test MSG_TEST
FRET

; En una biblioteca:
DEFMSG MSG_BASE "Advertencia de error: $1"
...
FUNCTION Test Public
  PARAM código AS integer
  ERROR MSG_BASE NOPLACE NOSTORE código
FRET
; El error de ciclo visualizado es: Advertencia de error: Error de
ejecución

```

DELAY**Sintaxis**

DELAY **valor**

Argumentos

valor constante o variable. Tiempo de retraso expreso en segundos

Descripción

Espera que pase el tiempo especificado por **valor**. Al terminar el intervalo de tiempo es ejecutada la instrucción sucesiva. El valor mínimo programable es 4 mseg (0,004 segundos).

DELEERROR**Sintaxis**

DELEERROR **nombredispositivo** [,estado [, IDposic [log]]
DELEERROR **número** [, IDposic [log]]

Argumentos

nombredispositivo nombredispositivo
número DEFMSG o constante o variable
IDposic constante o variable. Es un valor numérico usado en los sinópticos
estado constante predefinida. Puede asumir los siguientes valores: **ON**, **OFF**
log constante numérica predefinida o variable integer. Puede asumir los valores siguientes: **STORE** error guardado en archivo, **NOSTORE** error no guardado en archivo

Descripción

Hace ejecutar al PC supervisor la cancelación de un error de ciclo enviado anteriormente con la instrucción ERROR. Si un nombre de dispositivo se especifica, en lugar del número, se da al PC el tipo y la dirección lógica del dispositivo.

Si es especificado un nombre de dispositivo entonces en lugar del número es pasado a la PC el tipo y la dirección lógica del dispositivo. Para que la cancelación tenga efecto, todos los valores fijados en los

es generado el error, con el valor o el nombre del dispositivo o de la variable pasada como parámetro. Los marcadores a añadir en la cadena son:

- \$1, ... \$2 substituidos por el **nombre** del dispositivo o de la variable (\$1 corresponde a arg1 etc.)
- \$(1), ..., \$(3) substituidos por el **valor** del dispositivo o de la variable

Los tipos de datos permitidos para los parámetros arg1, ..., arg3 son:

- CHAR
- INTEGER
- FLOAT
- DOUBLE (que es transformado automáticamente en FLOAT)
- número de mensaje (o etiqueta de DEFMSG)
- dispositivo
- variable global o local
- parámetro de función. Es posible utilizar como parámetro de función también la etiqueta definida con una instrucción [DEFMSG](#)

No pueden ser usados como parámetros las cadenas, las matrices y los vectores (es permitido también solo un elemento de vector o matriz). En caso de variables locales, se puede descodificar tan solo el valor y no el nombre.

Al final de la cancelación de un mensaje con la instrucción DELERROR los parámetros arg1, ...arg3 son ignorados.

Son definidas dos modalidades de gestión de los errores de ciclo establecidas por el constructor de la máquina:

Alarmas gestionadas como advertencias: son enviados todos los errores de ciclo. Albatros mantiene una cola de los últimos 100 Errors de la coda especificada y de los últimos 100 errores de la cola predeterminada.

Alarmas gestionadas como estados: el error es considerado activo o no activo. Si activo cada ulterior envío del mismo error de ciclo (a través de instrucción ERROR) es ignorada.

Véase también las instrucciones [DELERROR](#), [CLEARERRORS](#).

Ejemplo 1

```
DEFMSG ERR_TOOL      "Herramienta no presente"
DEFMSG ERR_TOOL_P    "Cargar la herramienta $(1) en posición $(2)"

; tag para los sinópticos
CONST  TOOLCHANGE = 5

; error visualizado en la barra de los errores o en las casillas
; de sinóptico no marcadas.
ERROR  ERR_TOOL

; error guardado en los Informes y visualizado en las casillas
; de sinóptico marcadas con el código 5
ERROR  ERR_TOOL, TOOLCHANGE, STORE

; error guardado en los Informes pero no asociado a las casillas
; de sinóptico
ERROR  ERR_TOOL, NOPLACE, STORE

; error con parámetros
ERROR  ERR_TOOL_P, NOPLACE, NOSTORE, MxHerramientas[3].Cód, 5
```

Ejemplo 2

```
; definida en un grupo
DEFMSG MSG_ERR_CARGA "Carga herramienta no ejecutada"

Function MuestraMensaje
  MsgTool MSG_ERR_CARGA MxHerramientas[3].Cód
fret

Function MuestraError
  ErrTool STORE MSG_ERR_CARGA MxHerramientas[3].Cód
Fret

; definida en una biblioteca
DEFMSG MSG_ERR_TOOL "Error herramienta: $1 $(2)"
```

```

Function MsgTool public
PARAM parameter1 as integer
PARAM parameter2 as integer

    MESSAGE MSG_ERR_TOOL NOPLACE parameter1 parameter2
fret

Function ErrTool public
PARAM log as integer
PARAM argument1 as integer
PARAM argument2 as integer

    ERROR MSG_ERR_TOOL NOPLACE log argument1 argument2
fret

```

IFDEF/ELSEDEF/ENDIF

Sintaxis

IFDEF instrucción ... ENDIF	constante
IFDEF instrucción ... ENDIF	constante, operador de comparación, valor
IFDEF instrucción ... ENDIF	EXIST, nombregrupo
IFDEF instrucción ... ENDIF	LINKED, nombredispositivo
IFDEF instrucción ... ENDIF	UNLINKED, nombredispositivo
IFDEF instrucción ... ELSEDEF instrucción ... ENDIF	constante, operador de comparación, valor

Argumentos

constante	constante de tipo entero, char, doble, cadena
operador de comparación	los símbolos que pueden ser usados para ejecutar la comparación son: < (menor) = (igual) > (mayor) =< (menor igual) >= (mayor igual) <> (diferente)
valor	constante o nombre de dispositivo
nombregrupo	constante, cadena o nombre de grupo
nombredispositivo	constante, cadena o nombre de grupo

Descripción

La compilación condicional permite controlar cuales partes de un archivo de función GPL deben ser compiladas y por lo tanto ejecutadas. El compilador verifica que la condición requerida como argumento de la instrucción IFDEF sea verificada. En este caso el código comprendido entre la instrucción IFDEF y la instrucción ENDEF o ELSEDEF está compilado. Si hay la instrucción ELSEDEF y la condición no está verificada, será compilado el código comprendido entre la instrucción ELSEDEF y la instrucción ENDEF. La condición de compilación se puede expresar en diferentes modalidades:

- después de la instrucción IFDEF está especificado el nombre de una [constante](#). En este caso la condición es verificada si existe una constante global o del grupo actual con el nombre especificado.
- después de la instrucción IFDEF está especificada una relación entre dos operadores y un operando. El primer operando debe ser una constante. En este caso la condición está verificada si la relación es verdadera (por ej. MAX_TOOLS = 100).
- después de la instrucción IFDEF está especificada la palabra llave EXIST o NOTEXIST seguida por el nombre de un grupo de la máquina o de una cadena que contiene el nombre de un grupo de la máquina o el nombre de una librería. En este caso la condición es verificada si en configuración de máquina está presente o no está presente un grupo con el mismo nombre.
- después de la instrucción IFDEF está especificada la palabra llave LINKED o UNLINKED seguida por el nombre de un dispositivo. En este caso la condición está verificada si el dispositivo está conectado (LINKLED) o no está conectado (UNLINKLED) en Virtual-Físico. El nombre del dispositivo se puede expresar en la forma: Nombre_Grupo.Nombre_Subgrupo.Nombre_Dispositivo o Nombre_Grupo.Nombre_Dispositivo o Nombre_Subgrupo_NombreDispositivo o Nombre_Dispositivo. Si el dispositivo no existe en configuración, se considera por no conectado.

Se puede injertar más instrucciones IFDEF, teniendo presente que a cada instrucción IFDEF hay que corresponda una instrucción ENDEF.

Ejemplo 1

```

; la ejecución del código GPL cambia según
; de la presencia en máquina del grupo FRESADORA
Const GrupoFresadora = "Fresadora"
IFDEF Exist GrupoFresadora
  instrucción
  instrucción
ELSEDEF
  instrucción
  instrucción
ENDEF

```

Ejemplo 2

```

; la ejecución del código GPL cambia
; según el módulo
IFDEF _ID_MODULE = 1      ; compila las instrucciones para el módulo 1
  instrucción
  instrucción
ELSEDEF                  ; compila las instrucciones para otros módulos
  instrucción
  instrucción
ENDEF

```

```

; compila el código para la
; versión 3.2.0 de Albatros

```

```

IFDEF _VER_MAJOR = 3
  IFDEF _VER_MINOR = 2
    IFDEF _VER_REVISION = 0
      instrucción
      instrucción
    ENDEF
  ENDEF
ENDEF

```

```

; compila el código para la versión
; de service pack 10 Albatros

```

```

IFDEF _VER_SP = "Service Pack 10"
  instrucción
ENDEF

```

```

; compila el código solo si el sistema
; está configurado para un módulo remoto

```

```

IFDEF _REMOTE_MODULE = 1 ; 1 = módulo remoto, de otra manera 0 = módulo
                          local
  instrucción
ENDEF

```

```

; compila el código para la versión
; 2.4 service pack 10 Albatros

```

```
IFDEF _VER_FULL = $0002040AH
instrucción
ENDIF
```

Ejemplo 3

```
; la ejecución del código GPL cambia
; si el dispositivo está conectado en Virtual-Físico.
IFDEF LINKED out1 ; si out1 está conectado, el código se ejecuta
instrucción
instrucción
instrucción
ENDIF
```

MESSAGE**Sintaxis**

```
MESSAGE número [, IDposic [, arg1, ..., arg3]]
```

Argumentos

número	DEFMSG o constante o variable
IDposic	constante o variable. Valor numérico usado en los sinópticos
arg1, ..., arg3	constante o dispositivo o variable

Descripción

Genera un mensaje para el operador. El mensaje es identificado por el parámetro **número**. El parámetro **número** puede identificar un mensaje de módulo (por lo tanto un valor numérico entero) o de grupo (en este caso se usa una DEFMSG). Excepcionalmente puede ser pasado un argumento, representado por **IDposic**. Indica en cual ventana de sinóptico visualizar el mensaje. Hay que corresponder al valor especificado en el constructor sinópticos para la particular casilla de visualización. Albatros utiliza este identificador para manejar los mensajes en colas separadas. Para cada IDposic es creada una cola. Si no está especificado IDposic, el mensaje va en la cola predeterminada (IDposic=0). El rango de valores que se puede configurar oscila de 0 (NOPLACE) a 1023.

Los parámetros opcionales **arg1, ..., arg3** permiten definir unos mensajes paramétricos. En la cadena de definición del mensaje de error serán añadidos unos marcadores que serán substituidos, cuando es generado el mensaje, con el valor o el nombre del dispositivo o de la variable pasada como parámetro. Los marcadores a añadir en la cadena son:

- \$1, ... \$2 substitutos por el **nombre** del dispositivo o de la variable (\$1 corresponde a arg1 etc.)
- \$(1), ..., \$(3) substitutos por el **valor** del dispositivo o de la variable

Los tipos de datos permitidos para los parámetros arg1, ..., arg3 son:

- CHAR
- INTEGER
- FLOAT
- DOUBLE (que es transformado automáticamente en FLOAT)
- número de mensaje (o etiqueta de DEFMSG)
- dispositivo
- variable global o local
- parámetro de función. Es posible utilizar como parámetro de función también la etiqueta definida con una instrucción [DEFMSG](#)

Son definidas dos modalidades de gestión de los mensajes establecidos por el constructor de la máquina:

Mensajes gestionados como advertencias: son enviados todos los mensajes. Albatros mantiene una cola de los últimos 100 mensajes de la cola especificada y de los últimos 100 mensajes de la cola predeterminada. Cuando la cola de los mensajes está llena es sobrescrito el mensaje más viejo. Si el mensaje precedente de la cola es igual al que se está enviando, el mensaje no es enviado (tarea igual, número igual, argumento igual).

Mensajes gestionados como estados: el mensaje es considerado activo o no activo. Si activo, cada ulterior envío del mismo mensaje (a través de la instrucción MESSAGE) es ignorada.

No pueden ser utilizados como parámetros las cadenas, las matrices y los vectores (es permitido también solo un elemento de vector o matriz). En caso de variables locales, se puede descodificar tan solo el valor y no el nombre.

Al final de la cancelación de un mensaje con la instrucción DELERROR los parámetros arg1, ...arg3 son ignorados.

Véase también las instrucciones [DELMESSAGE](#) y [CLEARMESSAGES](#).

Ejemplo 1

```
DEFMSG MSG_TOOL "Efectuar cambio de herramienta"
DEFMSG MSG_TOOL_P "Cargada herramienta n° $(1)"
```

```

; tag para los sinópticos
CONST    TOOLCHANGE = 7

;mensaje visualizado en la barra de los errores o
;en las casillas de sinóptico no marcadas.
MESSAGE  MSG_TOOL

;mensaje visualizado en la barra de los errores y
;en las casillas de sinóptico marcadas por el código 7
MESSAGE  MSG_TOOL, TOOLCHANGE

; mensaje con parámetros
MESSAGE  MSG_TOOL_P, NOPLACE, MxHerramientas[3].Cod

```

Ejemplo 2

```

; definida en un grupo
DEFMSG   MSG_CARGA   "Carga ejecutada"

Function MuestraMensaje
    MsgTool MSG_CARGA MxHerramientas[3].Cód
fret

; definida en una biblioteca
DEFMSG   MSG_TOOL    "Herramienta: $(1) $2"

Function MsgTool public
PARAM parameter1 as integer
PARAM parameter2 as integer

    MESSAGE MSG_TOOL NOPLACE parameter1 parameter2
fret

```

SYSFAULT**Sintaxis****SYSFAULT****Argumentos**

ningún argumento

Descripción

Desactiva la señal de SYSOK.

Esta señal es desactivada para indicar que la máquina ya no está en seguridad (ej.: ya no están en ejecución las tareas GPL que manejan las emergencias).

Véase también la instrucción [SYSOK](#).**SYSOK****Sintaxis****SYSOK** [nombresalida1 [, ... nombresalida8]]**Argumentos****nombresalida1** [...] nombre dispositivo de tipo salida digital
nombresalida8**Descripción**

Indica al control numérico cuales son las salidas conectadas circuito de seguridad de la máquina (puede ser una salida conectada a un relé de seguridad, que controla la alimentación de potencia de la máquina). Las salidas son activadas cuando el control numérico ha completado la inicialización de la máquina y ha activado todas las tareas de gestión de las emergencias. La máquina puede ser, por lo tanto, considerada en condición de seguridad. Pueden ser definidos hasta un total de 8 salidas digitales. En cada remoto se puede habilitar solo una salida. La lista de salidas declaradas en el primer uso de la instrucción **SYSOK** **no** se puede modificar durante eventuales sucesivas llamadas al **sysok**, si no después de la inicialización del control.

Si se ejecuta la instrucción sin parámetros se restaura la señal de SYSOK.

Véase también la instrucción [SYSFAULT](#).

Atención:

La instrucción SYSOK puede ser habilitada:

- en todos los módulos remotos GreenBus v3.0 con salidas digitales.
- en los módulos remotos GreenBus v4.0 de tipo TRS-IO.
- en los módulos remotos TRS-CAT, solo en la base y no en las expansiones, que tienen una versión firmware 1.2 o mayor (rev 1.00 del módulo remoto).

TYPEOF

Sintaxis

TYPEOF nombre, resultado

Argumentos

nombre nombre de dispositivo, constante, nombrefunción, variable, vector, matriz o línea de matriz
resultado variable integer. Tipo del primer argumento

Descripción

Retorna en la variable **resultado** el tipo del argumento **nombre**.

WATCHDOG

Sintaxis

WATCHDOG estado

Argumentos

estado constante predefinida. Puede asumir los siguientes valores: **ON**, **OFF**

Descripción

La instrucción habilita el uso del guardián conectado al hardware TMSWD. Permite detectar situaciones de error que se producen durante la ejecución del código GPL.

La primera vez que se realiza con parámetro **estado** a ON, la instrucción permite el uso del guardián. Todas las veces siguientes hay que asignar ON al parámetro **estado** para realizar la actualización del contador de la tarjeta. Si la actualización falla, el guardián se activa y el módulo TMSWD desactiva la salida de seguridad de la máquina.

Para terminar el uso del guardián hay que asignar OFF al parámetro **estado**.

Esta instrucción solo se puede utilizar con las tarjetas TMSbus+, TMSCan+ y TMSCombo+ con versión FPGA 2.0 o superior y con montado el módulo de hardware TMSWD TMSWD.

Ejemplo

Function TestWatchDog autorun

watchdog ON ; habilita la gestión del guardián

Loop:

watchdog ON ; ejecuta la actualización del contador de la tarjeta

goto Loop

fret

10.3.14 MECHATROLINK-II

MECCOMMAND

Sintaxis

MECCOMMAND eje,comando,parametros,respuesta,error

Argumentos

eje nombre de dispositivo de tipo eje digital.
comando constante integer.
parámetros vector de integer.
respuesta vector d integer.
error variable integer. Código de error

Descripción

Envía al accionamiento del **eje** indicado un **comando** y espera la respuesta. Los datos necesarios a la ejecución del comando son insertados en el vector **parámetros**, mientras los datos que retornan de la ejecución de la instrucción son guardados en el vector **respuesta**. Hay que los vectores **parámetros** y **respuesta** tengan el mismo tamaño y un número de elementos máximo de 14. El valor considerado es el byte más bajo de cada entero. El parámetro error contiene los códigos de los eventuales errores generados durante la operación. Los códigos de error tienen que ser gestionados por Gpl como errores de ciclo. Los códigos de **error** devueltos son:

Código Error Mensaje

-40	Comando no permitido en las actuales condiciones de funcionamiento
-41	Error de timeout durante la ejecución de un comando MECHATROLINK-II
-44	Error de timeout durante la ejecución de un subcomando MECHATROLINK-II
-45	Error de conexión del accionamiento

Para los valores de asignar a los parámetros **comando**, **parámetros**, **respuesta** y **error**, véase la documentación oficial Yaskawa MECHATROLINK-II. En ella los valores de asignar per el comando son descritos a partir del índice 2 hasta el índice 15. Los valores de programar para los subcomandos son descritos a partir del índice 18 hasta el índice 32.

Los valores de **comando** pueden variar como en seguida respecto a la documentación oficial.

Los comandos pueden ser distinguidos en:

- comando. Tienen un código incluso entre 0x00 y 0xFF. Por razones de seguridad son ejecutados solamente con servo eje deshabilitado.
- subcomando. Los comandos considerados como subcomandos deben de sumar al valor documentado el código 0x100. Por ejemplo, el comando NOP tiene código documentado 0x00, considerado como subcomando es 0x100.
- procedimiento. Los comandos considerados como procesos tienen código con valor a partir de 0x200. Actualmente son previstos estos procesos:
 - \$201H proceso de toma en carga parámetros offline (de usar con eje deshabilitado)

Esta instrucción es utilizable solamente con tarjetas AlbMech, DualMech y DualMech Mono. Para más informaciones sobre el uso de esta instrucción póngase en contacto con TPA.

Atención

Esta instrucción actúa sobre el comportamiento de ejes digitales y por lo tanto tiene que ser utilizada en un contexto controlado.

MECGETPARAM

Sintaxis

MECGETPARAM eje,parámetro,dimensión,dato,error

Argumentos

eje	nombre de dispositivo de tipo eje digital.
parámetro	constante o variable integer.
dimensión	constante o variable integer.
dato	variable integer.
error	variable integer. Código de error

Descripción

Lee un parámetro del accionamiento del **eje** indicado y lo guarda en la variable **dato**. El parámetro **error** contiene los códigos de los eventuales errores generados durante la operación. Los códigos de error tienen que ser gestionados desde Gpl como errores de ciclo. Los códigos de error devueltos son:

Código Error Mensaje

-40	Comando no permitido en las actuales condiciones de funcionamiento
-41	Error de timeout durante la ejecución de un comando MECHATROLINK-II
-44	Error de timeout durante la ejecución de un subcomando MECHATROLINK-II
-45	Error de conexión del accionamiento

Para los valores de asignar a las variables **parámetros** y **dimensión**, véase la documentación oficial Yaskawa MECHATROLINK-II.

Esta instrucción es usable solamente con tarjetas AlbMech, DualMech y DualMech Mono. Para más informaciones sobre el uso de esta instrucción póngase en contacto con TPA.

MECGETSTATUS

Sintaxis

MECGETSTATUS **eje,estado,inout,error**

Argumentos

eje nombre de dispositivo de tipo eje digital.
estado constante o variable integer.
inout constante o variable integer.
error variable integer. Código de error

Descripción

Lee y guarda en la variable **estado** el valor de STATUS y de ALARM y en la variable **inout** el valor de IO_MON relativos al **eje** especificado. Para los valores de STATUS, ALARM, IO_MON, véase la documentación oficial Yaskawa MECHATROLINK-II.

El parámetro **error** contiene los códigos de los eventuales errores generados durante la operación. Los códigos de error tienen que ser gestionados desde Gpl como errores de ciclo.

Los códigos de **error** devueltos son:

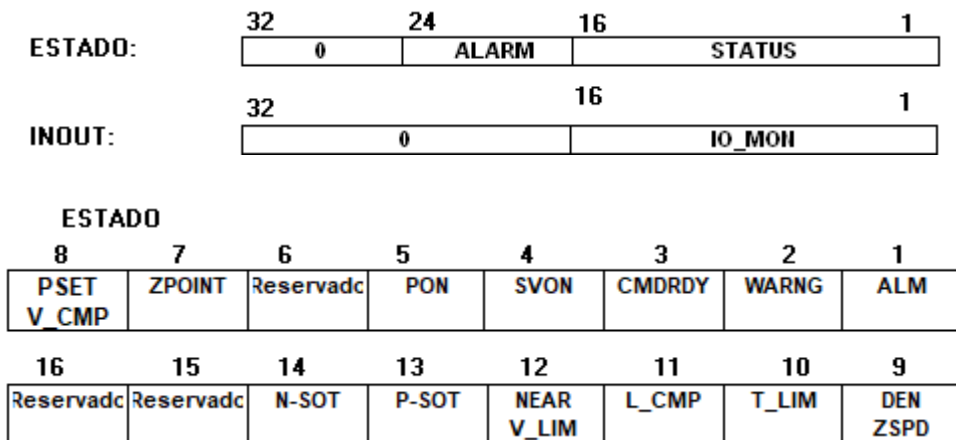
Código Error Mensaje

- 40 Comando no permitido en las actuales condiciones de funcionamiento
- 41 Error de timeout durante la ejecución de un comando MECHATROLINK-II
- 44 Error de timeout durante la ejecución de un subcomando MECHATROLINK-II
- 45 Error de conexión del accionamiento

Se define una serie de categorías de errores. Las categorías representan el valor del nibble más alto de ALARM.

Para salir de alarmas que son parte de una de las siguientes categorías 0x30,0x70,0xD0,0xF0 debe de ser enviado un comando de CLEAR (0x06). Alarmas que son parte de una de las siguientes categorías 0x00,0x10,0x40,0xB0 no pueden ser eliminadas con ningún comando. Es necesario resolver el problema que genera el alarma, apagar el servo-accionamiento y volver a encenderlo

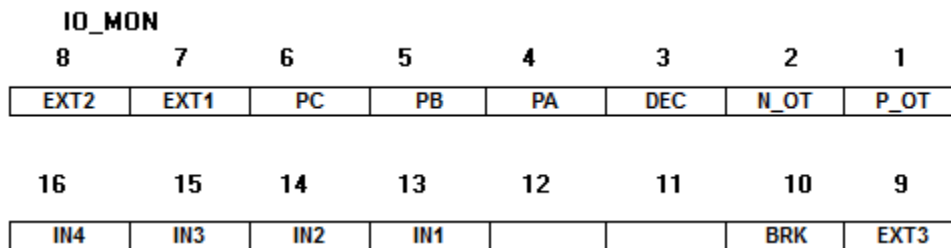
La estructura de las variables **estado** e **inout** es una máscara de bit estructurada como en la representación siguiente:



Significado de los bits de ESTADO

Bit	Comando	Pin físicos que se pueden conectar en Virtual-Físicos
1	ALM (Alarm)	Entrada digital
2	WARNG (Warning)	Entrada digital
3	CMDRDY (Command Ready)	
4	SVON (Servo ON)	Salida digital
5	PON (Main Power ON)	Entrada digital
6	Reservado	
7	ZPOINT (Zero Point)	
8	PSET (Position Complete) V_CMP (Velocity Agreement)	

9	DEN (Command Distribution Completed Flag) ZSPD (Zero Velocity)	
10	T_LIM (Torque Limit)	
11	L_CMP (Latch Completed)	
12	NEAR (Position Proximity) V_LIM (Velocity Limit)	
13	P-SOT (Forward-direction Software Limit)	
14	N-SOT (Reverse-direction Software Limit)	
15	Reservado	
16	Reservado	

**Significado de los bits de IO_MON**

Bit	Comando	Pin físicos que se pueden conectar en Virtual-Físico
1	P_OT (Forward Over Travel)	
2	N_OT (Reverse Over Travel)	
3	DEC (Deceleration Limit Switch)	
4	PA (Phase A)	
5	PB (Phase B)	
6	PC (Phase C)	Entrada digital
7	EXT1 (First external latch input)	Entrada digital
8	EXT2 (Second external latch input)	Entrada digital
9	EXT3 (Third external latch input)	
10	BRK (Brake output)	
11		
12		
13	IN1 (General-purpose input 1)	
14	IN2 (General-purpose input 2)	
15	IN3 (General-purpose input 3)	
16	IN4 (General-purpose input 4)	

Esta instrucción es utilizable solamente con tarjetas AlbMech, DualMech y DualMech Mono. Para más informaciones sobre el uso de esta instrucción póngase en contacto con TPA.

MECSETPARAM**Sintaxis****MECSETPARAM****eje, parámetro, dimensión, dato, error****Argumentos**

eje
parámetro
dimensión
dato
error

nombre de dispositivo de tipo eje digital.
 constante o variable integer.
 constante o variable integer.
 variable integer.
 variable integer. Código de error

13	ms=Manufacturer specific (Específica del fabricante)	CW13
14	ms=Manufacturer specific (Específica del fabricante)	CW14
15	ms=Manufacturer specific (Específica del fabricante)	CW15
16	ms=Manufacturer specific (Específica del fabricante)	CW16

Tabla de definición de los valores para S-CAN

CONTROLWORD						
8	7	5	4	3	2	1
fr	oms		eo	qs	ev	so

Bit	Significado	Nombre en Virtual-Físico
1	Ten_cmd=comando activación torque. 1: torque 0: eje libre	SVON
2	Ten_cmd=comando activación desplazamientos 1: desplazamiento activado 0: eje parado	ENMOVE
3	Stp_cmd=comando stop. 1: comando de stop activo 0: comando de parada activo	STOP
4	Alm_rst=estado alarmas 1: comando restablecimiento alarmas	RESALM
5	Ltc_rst:reset bit 5 de StatusWord	CW5
6	oms=específico del modo seleccionado	CW6
7	oms=específico del modo seleccionado	CW7
8	oms=específico del modo seleccionado	CW8

AXSTATUS

Sintaxis

AXSTATUS eje,valor

Argumentos

eje nombre de dispositivo tipo eje
valor variable integer.

Descripción

Restituye el valor contenido en StatusWord de conformidad con "CiA 402 CANopen device profile".

Tabla de definición de los valores para EtherCAT

STATUSWORD						
8	7	5	4	3	2	1
fr	oms		f	oe	so	rtso

16	12	11	10	9
ms		r	oms	h

Bit	Significado	Nombre en Virtual-Físico
1	rtso=Ready to switch on (Listo para encender)	RTSO
2	so=Switched ON (Encendido ON)	SW2
3	oe=Operation enabled (Operación habilitada)	OE
4	f=Fault (Fallo)	ALM

5	ve=Voltage enable (Habilitar voltaje)	VE
6	qs=Quick stop (Parada rápida)	QS
7	sod=Switch on disabled (Desactivado conmutador ON)	SOD
8	w=Warning (Aviso)	WARNG
9	ms=Manufacturer specific (Específica del fabricante)	SW9
10	rm=Remote (Remoto)	SW10
11	tr=Target reached or reserved (Objetivo conseguido o reservado)	SW11
12	ila=Internal limit active (Limitación interna activa)	SW12
13	oms=Operation mode specific (Modo de operación específica)	SW13
14	oms=Operation mode specific (Modo de operación específica)	SW14
15	ms=Manufacturer specific (Específica del fabricante)	SW15
16	ms=Manufacturer specific (Específica del fabricante)	SW16

Tabla de definición de los valores para S-CAN

CONTROLWORD						
8	7	5	4	3	2	1
fr	oms	eo	qs	ev	so	

Bit	Significado	Nombre en Virtual-Físico
1	Ten_st=estado habilitación pareja 1: torque 0: eje libre	SW1
2	Ien_st=estado habilitación movimientos 1: desplazamiento activado 0: eje parado	SW2
3	Stp_st=estado de parada 1: rampa de parada en curso 0: parada no activada o rampa concluida	SW3
4	Alm_rst=estado alarmas 1: máquina en alarma 0: no se detecta ninguna alarma	ALM
5	Ltc_st=estado de bloqueo posición 1: bloqueo de posición ejecutado, registro listo para la lectura 0: ningún bloqueo de posición señalado	SW5
6	oms=específico del modo seleccionado	SW6
7	oms=específico del modo seleccionado	SW7
8	oms=específico del modo seleccionado	SW8

CNBYDEVICE

Sintaxis

CNBYDEVICE**dispositivo,tarjeta,cn**

Argumentos

dispositivo
tarjeta
cnnombre de dispositivo.
variable integer. Nombre de tarjeta restituido
variable integer. Nombre de nodo restituido

Descripción

Restituye número de tarjeta y número de nodo del dispositivo definido en el parámetro **dispositivo**. Esta instrucción sirve para poder utilizar las instrucciones que no tienen referencias directas a dispositivos, como, por ejemplo, las instrucciones [READDICTIONARY](#) y [WRITEDICTIONARY](#).

La instrucción puede ser utilizada para dispositivos configurados sobre bus CAN, S-CAN, GreenBUS o EtherCAT.

Si el número nodo restituido es un valor negativo, significa que el nodo está deshabilitado.

READDICTIONARY

Sintaxis

READDICTIONARY **tarjeta, cn, índice, subíndice, dimdato, dato, error**

Argumentos

tarjeta constante o variable integer. Número tarjeta
cn constante o variable integer. Número nodo
índice constante o variable integer. Índice objeto en el diccionario
subíndice constante o variable integer. Subíndice objeto en el diccionario
dimdato variable integer. Tamaño del dato leído.
dato variable char, integer, float, doble, vector de char, cadena. Variable que recibe el dato
error variable integer. Código de error restituido por nodo

Descripción

Lee el contenido de un objeto del diccionario de objetos contenidos en el nodo. La instrucción permite leer por medio del protocolo SDO todos los objetos definidos según "CiA 402 CANopen device profile", además de todos los objetos puestos a disposición por el constructor del nodo. Para el significado de los parámetros **índice**, **subíndice** y **dimdato** hacer referencia a "CiA 402 CANopen device profile" o a la especificación del fabricante del nodo. Para los dispositivos S-CAN el parámetro subíndice hay que ponerse siempre a cero.

WRITEDICTIONARY

Sintaxis

WRITEDICTIONARY **tarjeta, cn, índice, subíndice, dimdato, dato, error**

Argumentos

tarjeta constante o variable integer. Número tarjeta
cn constante o variable integer. Número nodo
índice constante o variable integer. Índice objeto en el diccionario
subíndice constante o variable integer. Subíndice objeto en el diccionario
dimdato constante o variable integer. Tamaño del dato que escribir
dato variable char, integer, float, doble, vector de char, cadena. Variable que contiene el dato
error variable integer. Código de error restituido por nodo

Descripción

Escribe el contenido de un objeto del diccionario de objetos contenidos en el nodo. La instrucción permite escribir por medio del protocolo SDO todos los objetos según "CiA 402 CANopen device profile", además de todos los objetos puestos a disposición por el constructor del nodo. Para el significado de los parámetros **índice**, **subíndice** y **dimdato** que escribir, hacer referencia a "CiA 402 CANopen device profile" o a la especificación del constructor del nodo. Para los dispositivos S-CAN el parámetro subíndice debe ponerse siempre a cero.

10.3.16 EtherCAT

ACTIVATEMODE

Sintaxis

ACTIVATEMODE **eje,dato, err**

Argumentos

eje nombre de dispositivo tipo eje
dato variable o constante integer. Modalidad operativa
err variable integer. Código de error restituido del servocontrol

Descripción

Programa la modalidad operativa definida en la variable **dato**, según "CiA 402 CANopen device profile". La modalidad operativa del eje de partida corresponde al valor de **dato**=9, es decir "Modalidad de velocidad sincrónica cíclica". La instrucción restituye valor **err**=0, si el comando ha sido un éxito, de lo contrario restituye un código de error.

En seguida la tabla de los valores que asignar a **dato** para elegir la modalidad operativa.

Valor	Definición
+6	Modalidad Homing
+9	Modalidad de velocidad sincrónica cíclica

ECATGETREGISTER

Sintaxis

ECATGETREGISTER **nodo,dirección,dim,dato,error**

Argumentos

nodo	constante o variable integer. Posición ocupada por el esclavo en la cadena EtherCAT (desde 1 en adelante)
dirección	constante o variable integer. Dirección del registro del ESC (EtherCAT Slave Controller) que leer (de 0 en adelante)
dim	constante o variable integer. Número de bytes que leer (1, 2 o 4)
dato	variable integer. Contenedor del dato leído
error	variable integer. Código de error.

Descripción

Restituye **[dato]** el contenido de un registro del ESC (EtherCAT Slave Controller) del nodo EtherCAT indicado. El parámetro **error** contendrá el código numérico del error, 0 si no se han averiguado errores.

ECATSETREGISTER

Sintaxis

ECATSETREGISTER **nodo,dirección,dim,dato,error**

Argumentos

nodo	constante o variable integer. Posición ocupada por el esclavo en la cadena EtherCAT (desde 1 en adelante)
dirección	constante o variable integer. Dirección del registro del ESC (EtherCAT Slave Controller) que leer (de 0 en adelante)
dim	constante o variable integer. Número de bytes que leer (1, 2 o 4)
dato	constante o variable integer. Dato que hay que esté escrito
error	variable integer. Código de error

Descripción

La instrucción escribe el contenido **[dato]** de un registro del ESC (EtherCAT Slave Controller) del nodo EtherCAT indicado. El parámetro **error** contendrá el código numérico del error, 0 si no se han averiguado errores.

GETPDO

Sintaxis

GETPDO **tarjeta,nodo,nPDO,nObj,dato,[error]**

Argumentos

tarjeta	constante o variable integer. Número tarjeta
nodo	constante o variable integer. Posición ocupada por el esclavo en la cadena EtherCAT (desde 1 en adelante)
nPDO	constante o variable integer. Identificativo del PDO (es. \$1600h) o posición del mismo en la lista de PDOs definidos en configuración hardware para el nodo en consideración (desde 1 hasta 16)
nObj	constante o variable integer. Identificativo del objeto (es. \$6040h) o posición del mismo dentro de la lista de objetos configurados en el PDO (desde 1 hasta 32)
dato	variable integer. Recibe el valor.
error	variable integer. Código de error.

Descripción

Restituye en **[dato]** el contenido de un objeto intercambiado a través de los PDOs configurados para el nodo EtherCAT. Si los argumentos pasados son incorrectos y no está configurado el parámetro **error**, se crea un error de sistema. Si fue configurado un parámetro **error**, esto contendrá el código numérico del correspondiente error de sistema.

SETEOE

Sintaxis

SETEOE **estado,[error]**

Argumentos

estado	constante predefinida. Los valores admitidos son: - ON activa la gestión del protocolo EoE - OFF desactiva la gestión del protocolo EoE
error	variable integer. Código de error.

Descripción

Restituye el estado del protocolo NMT por el nodo máster de la **tarjeta** indicada. Para mayores informaciones sobre el significado de estos parámetros, hacer referencia directamente a la documentación de cada dispositivo.

RECEIVEPDO**Sintaxis**

RECEIVEPDO **tarjeta, nodo, númeroPDO**

Argumentos

Tarjeta constante o variable integer. Número de la tarjeta (desde 1 a 4)
Tarjeta constante o variable integer. Número del nodo
NúmeroPDO constante o variable integer. Número del PDO

Descripción

Lee el contenido del PDO especificado por **númeroPDO** para el nodo especificado. Esta instrucción se usa para leer PDOs asíncronicos (es decir, los PDOs que en configuración hardware tienen la opción **Asíncronico** activada). El dato leído es copiado en los dispositivos conectados en Virtual-Físico. Solo se puede utilizar esta instrucción con tarjetas TMSCan y TMSCan+.

SENDPDO**Sintaxis**

SENDPDO **tarjeta, nodo, númeroPDO**

Argumentos

Tarjeta constante o variable integer. Número de la tarjeta
Tarjeta constante o variable integer. Número del nodo
NúmeroPDO constante o variable integer. Número del PDO

Descripción

Escribe el contenido del PDO especificado por **númeroPDO** para el nodo especificado. Esta instrucción se utiliza para escribir PDOs asíncronicos (o sea, los PDOs que en configuración hardware tienen la opción **Asíncronico** activada). Solo se puede utilizar esta instrucción con tarjetas TMSCan y TMSCan+.

SETNMTSTATE**Sintaxis**

SETNMTSTATE **tarjeta, nodo, estado**

Argumentos

tarjeta constante o variable integer. Número de la tarjeta (de 1 a 4)
nodo constante o variable integer. Número del nodo
estado constante o variable integer.

Descripción

Configura el estado del protocolo NMT para el **nodo** de la **tarjeta** indicado. Si el valor del nodo es igual a 0 (cero) o superior a 126, la configuración se aplica a todos los nodos configurados y presentes en el canal. Para más informaciones sobre el significado de estos parámetros, consulte directamente los documentos relativos a cada dispositivo.

Valor	Estado del protocolo
1	Operational
128	Pre-Operational

10.3.18 Simulación**DISABLE****Sintaxis**

DISABLE **eje1,[...eje6]**

Argumentos

eje1...[...eje6] nombres de dispositivos tipo eje

Descripción

Deshabilita los ejes especificados. La deshabilitación permite efectuar unas simulaciones de la cíclica de máquina sin movimentar realmente los ejes. Un eje deshabilitado no lee las informaciones que llegan desde el encoder más simula un error de anillo proporcional con la velocidad corriente. La deshabilitación del eje no desactiva sin embargo la referencia de velocidad, se habrá por lo tanto una tensión diferente de cero en el conector del eje también durante los movimientos simulados. Por lo tanto es necesario desalimentar los accionamientos o desconectarlos de la tarjeta ejes durante los movimientos simulados o sea a ejes deshabilitados. Véase también [ENABLE](#).

Atención

Ejes paso-a-paso se pueden estar usados en esta instrucción si solo son gestionados por un remoto TRS-AX.

DISABLEFORCEDINPUT**Sintaxis**

DISABLEFORCEDINPUT

Argumentos

ningún argumento

Descripción

Deshabilita la posibilidad de utilizar las funciones para el forzamiento de las entradas. Si en precedencia habían sido forzados unas entradas la ejecución de esta instrucción los devuelve al estado real. Véase también [ENABLEFORCEDINPUT](#), [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

ENABLE**Sintaxis**

ENABLE **eje1,[...eje6]**

Argumentos

eje1...[...eje6] nombres de dispositivos tipo eje

Descripción

Habilita los ejes especificados. Los ejes en fase de inicialización son siempre habilitados. Hay que llamar esta instrucción solo si los ejes han sido precedentemente deshabilitados por una instrucción [DISABLE](#).

Atención

Ejes paso-a-paso se pueden estar usados en esta instrucción si solo son gestionados por un remoto TRS-AX.

ENABLEFORCEDINPUT**Sintaxis**

ENABLEFORCEDINPUT

Argumentos

ningún argumento

Descripción

Habilita el forzamiento de las entradas. Antes de utilizar las instrucciones para habilitar o deshabilitar los dispositivos de entrada de manera forzada, es necesario ejecutar esta instrucción. En caso contrario las instrucciones de forzamiento de las entradas no producen ningún efecto. Véase también [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

RESETFORCEDINPUT**Sintaxis**

RESETFORCEDINPUT **nombreentrada**

Argumentos

nombreentrada nombre entrada digital

Descripción

Fuerza en OFF la entrada especificada por **nombreentrada**.

Para poder utilizar esta instrucción es necesario haber ya habilitado el forzamiento de las entradas, con la instrucción [ENABLEFORCEDINPUT](#).

Véase también [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

SETFORCEDANALOG

Sintaxis

SETFORCEDANALOG **entradaanalogica, valor**

Argumentos

**entradaanalogica
variable** nombre de dispositivo entrada analógica
constante o variable integer o float o double

Descripción

Fuerza el valor en la entrada analógica especificada por **entradaanalogica**. Para poder utilizar esta instrucción es necesario haber ya habilitado el forzamiento de las entradas, con la instrucción [ENABLEFORCEDINPUT](#). Véase también [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#).

SETFORCEDINPUT

Sintaxis

SETFORCEDINPUT **nombreentrada**

Argumentos

nombreentrada nombre entrada digital

Descripción

Fuerza en ON la entrada especificada por **nombreentrada**. Para poder utilizar esta instrucción es necesario haber ya habilitado el forzamiento de las entradas, con la instrucción [ENABLEFORCEDINPUT](#). Véase también [DISABLEFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDPORT](#), [SETFORCEDANALOG](#).

SETFORCEDPORT

Sintaxis

SETFORCEDPORT **nombrepuerto, valor**

Argumentos

**nombrepuerto
variable** nombre de dispositivo puerto de entradas
constante o variable integer o char

Descripción

Fuerza el **valor** en el puerto de entradas expreso por **nombrepuerto**. El puerto de entradas es visto como una máscara de bit. Si un bit vale 1 la correspondiente entrada es forzada en "ON". Para poder utilizar esta instrucción es necesario haber ya habilitado el forzamiento de las entradas, con la instrucción [ENABLEFORCEDINPUT](#). Véase también [DISABLEFORCEDINPUT](#), [SETFORCEDINPUT](#), [RESETFORCEDINPUT](#), [SETFORCEDANALOG](#).

10.3.19 BlackBox

El fin de la funcionalidad "BlackBox" es lo de registrar, en una base de datos, toda la actividad de una máquina, es decir un módulo local o remoto. Para "actividad de máquina" se entiende la variación, en el tiempo, de un subconjunto de todos los dispositivos lógicos que se pueden emplear en GPL. Así se puede analizar el comportamiento de la máquina en un segundo momento, relacionando los estados de los dispositivos almacenados. La base de datos tiene una tabla que contiene una información temporal o el estado de todos los dispositivos en aquel momento, uno por cada columna. En el lenguaje GPL se introdujeron nuevas instrucciones para iniciar, terminar y consultar la actividad de registro; a continuación, hay la descripción.

Cada archivo BlackBox es una base de datos SQLite y contiene informaciones cerca de un solo módulo. El nombre del archivo incluye el número del módulo y la fecha y la hora de cuando se inició la actividad de registro.

La añadidura de registro en la base de datos se hace de manera transaccional. Cada transacción contiene los registros generados en 1 segundo. En caso de blackout está garantida la consistencia del archivo y se puede perder la última transacción. La duración límite de la transacción se puede modificar con una voz en Tpa.ini (para informaciones póngase en contacto con TPA).

Se insertó el límite de 12 horas a la duración del registro. Por este hecho, cada base de datos tendrá siempre y solo las últimas 12 horas de registro. Durante el registro se eliminan los registros más viejos desde la base de datos. La duración límite de la historia registrada se puede modificar con una voz en Tpa.ini (para informaciones póngase en contacto con TPA).

Esta funcionalidad está disponible para dispositivos físicos sobre bus GreenBUS, EtherCAT, CAN, S-CAN, MECHATROLINK-II, que son conectados a través de TMSbus, TMSbus+, TMScan, TMScan+, DualMech, DualMech Mono, AlbMech.

ENDBLACKBOX

Sintaxis
ENDBLACKBOX

Descripción
Termina la funcionalidad de registro en archivo de toda la actividad de un módulo local o remoto. Véase también [STARTBLACKBOX](#) y [PAUSEBLACKBOX](#).

PAUSEBLACKBOX

Sintaxis
PAUSEBLACKBOX

Descripción
Termina la funcionalidad de registro en archivo de toda la actividad de un módulo local o remoto. Para seguir con el registro, hay que ejecutar la instrucción [STARTBLACKBOX](#) sin argumentos. Véase también [ENDBLACKBOX](#).

STARTBLACKBOX

Sintaxis
STARTBLACKBOX **[valor][,error]**

Argumentos
valor constante o variable integer. Período de registro
error variable integer. Código de error

Descripción
Activa la funcionalidad de registro sobre archivo de toda la actividad de un módulo local o remoto. Para actividad de un módulo se entiende la variación en el tiempo del estado de los dispositivos lógicos, excluidos los dispositivos de tipo interruptor indicador. El período de registro (**valor**) está expresado en mili-segundos. No puede ser menor de 10 y hay que ser un múltiplo del período de tiempo real. Si así no es, se genera el error de sistema n 4399 (Parámetro fuera del intervalo). Si la instrucción inicia un registro y el **valor** está omitido, se considera como valor predeterminado 20. Si la instrucción retoma un registro interrumpido previamente, no se considera algún **valor** configurado. Si no fue posible iniciar el registro, **error** contiene un valor diferente de 0, de otro modo contiene 0:

Código Error	Descripción
0	Ningún error
1	Hay diferencias entre la configuración de los dispositivos en el control numérico y en Albatros
2	El número de los dispositivos, que se deben registrar, excede el número máximo establecido para el sistema
3	En configuración no hay dispositivos
4	El software de comunicación en el módulo remoto no suporta la funcionalidad de BlackBox (solo módulos remotos)
5	El control numérico impide el inicio del registro
6	Error en la carga de la librería de gestión de la base de datos
7	El número de columnas por la tabla excede el número máximo de columnas gestionables por la base de datos
8	No fue posible abrir la base de datos sobre disco
9	No fue posible crear la tabla en la base de datos de los registros
10	Error en la dirección IP para la comunicación con el módulo remoto (solo módulos remotos)
11	No fue posible crear el socket de comunicación para recibir los datos (solo módulos remotos)

12	No fue posible asociar una dirección local al socket de comunicación (solo módulos remotos)
13	No fue posible asociarse al socket remoto (solo módulos remotos)
14	No fue posible acceder a la área de memoria compartida con el control numérico
15	La configuración hardware impide el uso de la funcionalidad "BlackBox"
16	La funcionalidad fue deshabilitada en tpa.ini

Véase también [PAUSEBLACKBOX](#) y [ENDBLACKBOX](#).

10.3.20 ISO

ISOG0

Sintaxis

ISOG0

etiqueta, eje1, cota1, eje2, cota2, eje3, cota3, eje4, cota4, eje5, cota5, [valor]

Argumentos

etiqueta

constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento. N en el estándar ISO

eje1

nombre de dispositivo tipo eje. (X en el estándar ISO)

cota1

constante o variable Cota espacio operacional del eje1

eje2

nombre de dispositivo tipo eje. (Y en el estándar ISO)

cota2

constante o variable Cota espacio operacional del eje2

eje3

nombre de dispositivo tipo eje. (Z en el estándar ISO)

cota3

constante o variable Cota espacio operacional del eje3

eje4

nombre de dispositivo tipo eje. (C en el estándar ISO)

cota4

constante o variable Cota espacio operacional del eje4

eje5

nombre de dispositivo tipo eje. (B en el estándar ISO)

cota5

constante o variable Cota espacio operacional del eje5

valor

constante o variable double. Representa el porcentaje de feed rate. (F en el estándar ISO)

Descripción

Programa el movimiento rápido. Los tramos de movimiento rápido son gestionados en sincronización. Los puntos definidos por el usuario son extremos del tramo individual de desplazamiento, que es recorrido, de manera que los ejes sean sincronizados entre ellos. Esto significa que los ejes físicos se mueven de modo independiente pero manteniendo arranque y llegada contemporáneos, en semejanza de lo que pasa en las instrucciones [MULTIABS](#) y [MULTIINC](#). La punta de la herramienta no recorre una línea en el espacio operacional y su trayectoria no resulta controlada. El parámetro **etiqueta** es usado en asociación con la instrucción [SETLABELINTERP](#) para identificar de manera unívoca el bloqueo de desplazamiento. Las primeras tres **cotas** identifican la posición de la punta en el espacio operacional, mientras las dos siguientes definen el valor de los ejes rotatorios en el espacio entre las juntas. El **valor** de FeedRate define el porcentaje de reducción con respecto a la velocidad máxima posible (En ISO: F0 velocidad máxima, F100 FeedRate nulo y por lo tanto ejes parados).

La instrucción genera error de sistema (4105 - Instrucción no ejecutable en el eje NombreEje), si usada en ejes paso-a-paso.

No es posible emplear la instrucción [WAITCOLL](#), en cuanto desde el momento de la colisión se perdería el enlace de interpolación con los demás ejes, que concurren al movimiento y generan un perfil diferente del esperado. Si empleada, es generado el error de sistema "4101 - Gestión incoherente del eje NombreEje".

ISOG1

Sintaxis

ISOG1

etiqueta, eje1, cota1, eje2, cota2, eje3, cota3, eje4, cota4, eje5, cota5, [valor]

Argumentos

etiqueta

constante o variable integer. Etiqueta que identifica un bloqueo de desplazamiento. (N en el estándar ISO)

eje1

nombre de dispositivo tipo eje. (X en el estándar ISO)

cota1

constante o variable Cota espacio operacional del eje1

eje2

nombre de dispositivo tipo eje. (Y en el estándar ISO)

cota2

constante o variable Cota espacio operacional del eje2

eje3

nombre de dispositivo tipo eje. (Z en el estándar ISO)

cota3

constante o variable Cota espacio operacional del eje3

eje4

nombre de dispositivo tipo eje. (C en el estándar ISO)

cota4

constante o variable Cota espacio operacional del eje4

Programa la interpretación de las cotas como inverso del tiempo de ejecución. El parámetro **eje** individua el canal de interpolación 5 Ejes, que a partir de esta instrucción interpretará el valor pasado a través de los parámetros F de la instrucción [ISOG1](#) como inverso del tiempo de ejecución expreso en minutos. Como resultado, el control puede determinar la velocidad que la punta de la herramienta hay que mantener en los bloqueos de interpolación.

ISOG94

Sintaxis

ISOG94 **eje**

Argumentos

eje nombre de dispositivo tipo eje

Descripción

Programa la interpretación de las velocidades como unidad de medida por minuto. El parámetro **eje** individua el canal de interpolación 5 Ejes, que a partir de esta instrucción interpretará las velocidades como unidad de medida pro minuto (condición estandar).

ISOG216

Sintaxis

ISOG216 **nombreMatrizRotatorios, nombreMatrizPortaherramientas, nombreMatrizHerramientas, máscaraHabilitación, eje1, eje2, eje3, eje4, eje5**

Argumentos

nombreMatrizRotatorios	nombre de la matriz. Contiene los datos relativos a los ejes rotatorios
nombreMatrizPortaherramientas	nombre de la matriz. Contiene los datos relativos a las portaherramientas
nombreMatrizHerramientas	nombre de la matriz. Contiene los datos relativos a las herramientas
máscaraHabilitación	variable o constante integer, máscara de habilitación de los ejes C y B
eje1	nombre de dispositivo tipo eje (X en el estándar ISO)
eje2	nombre de dispositivo tipo eje (Y en el estándar ISO)
eje3	nombre de dispositivo tipo eje (Z en el estándar ISO)
eje4	nombre de dispositivo tipo eje (C en el estándar ISO)
eje5	nombre de dispositivo tipo eje (B en el estándar ISO)

Descripción

Identifica las tres matrices para la parametrización de la máquina y los cinco dispositivos de tipo eje que forman la misma. El parámetro **máscaraHabilitación** define cuales ejes rotatorios (C y/o B) habilitar. Para establecer los valores hacer referencia a la tabla siguiente:

máscaraHabilitación	Descripción
31	Habilitacion ejes C y B
23	Habilitacion del solo eje B
15	Habilitacion del solo eje C
7	Deshabilitacion ejes C y B

Atención

La unidad de medida en que se expresan, en configuración, los valores de los ejes rotatorios, hay que ser grados.

El vínculo entre los ejes físicos y los ejes ISO virtuales programados a través de esta instrucción se disuelve cuando se ha completado la tarea, donde la instrucción es definida o a través de la instrucción [ISOM2](#). Por lo tanto los ejes pueden ser utilizados para movimientos de tipo clásico.

ISOG217

Sintaxis

ISOG217 **eje1, eje2, eje3, eje4, eje5, ejeVirtual1, ejeVirtual2, ejeVirtual3, ejeVirtual4, ejeVirtual5.**

Desplazamiento del eje X	Offset en X entre el eje de rotación y el de oscilación (cuando la posición del eje C = 0)
Desplazamiento del eje Y	Offset en Y entre el eje de rotación y el de oscilación (cuando la posición del eje C = 0)
Desplazamiento del eje Z	Distancia Nariz-Fulcro
Ángulo δ	Ángulo alrededor de Z para la correcta colocación de la cabeza con respecto al cero máquina
Ángulo γ	Ángulo entre el plano de rotación y el de oscilación.

Campos Matriz	Matriz Portaherramienta
DesplazamientoPU X	Offset en X entre el punto de enganche al motor de la portaherramienta y el punto de enganche de la herramienta a la portaherramienta (cuando la posición del eje C = 0 y motor vertical)
DesplazamientoPU Y	Offset en Y entre el punto de enganche al motor de la portaherramienta y el punto de enganche de la herramienta a la portaherramienta (cuando la posición del eje C = 0 y motor vertical)
DesplazamientoPU Z	Offset en Z entre el punto de enganche al motor de la portaherramienta y el punto de enganche de la herramienta a la portaherramienta (cuando la posición del eje C = 0 y motor vertical)
Ángulo α	Ángulo de desfase entre el eje del motor y el de la herramienta (con respecto a Z)
Ángulo β	Ángulo de desfase entre el eje del motor y el de la herramienta (en respecto a Y)

Campos Matriz	Matriz Herramientas
Largura herramienta	Largura de la herramienta

ISOSETPARAM

Sintaxis

ISOSETPARAM

IdentificadorParámetro, valor

Argumentos

IdentificadorParámetro
valor

constante o variable entera. Es el número que identifica un parámetro
constante o variable float. Es el valor que programar.

Descripción

Programa unos parámetros que caracterizan la fluidez del movimiento interpolado ISO. La tabla que sigue explica el expresado de cada **IdentificadorParámetro**, los valores entre los cuales la variable **valor** y los valores predeterminados hay que estar.

IdentificadorParámetro	Rango	Predeterminado	Significado
0	0.0-100.0	50.0	Porcentaje de desaceleración en los ejes lineales en caso de punto anguloso. (0 = sin desaceleración, 100 = desaceleración máxima permitida por el interpolador)
1	0.0-100.0	50.0	Porcentaje de desaceleración en los ejes giratorios en caso de punto anguloso. (0 = sin desaceleración, 100 = desaceleración máxima permitida por el interpolador)
2	0.5-1.0	0.9	Factor de reducción de la velocidad en abscisa

			curvilínea en caso de punto anguloso. (1 = sin reducción, 0.5 = reducción máxima permitida)
3	0.0-100.0	60.0	Porcentaje de desaceleración en caso de discontinuidades cercanas. (0 = sin desaceleración, 100 = desaceleración máxima permitida por el interpolador)
4	0.0-100.0	10.0	Porcentaje de suave de la trayectoria
5	(Mayor que 0.0)-1.0	0.2	Dimensión mínima del tramo que recorrer sólo con ejes lineales. El valor está expresado en milímetros.
6	(Mayor que 0.0)-1.0	0.1	Dimensión mínima del tramo que recorrer sólo con ejes giratorios. El valor está expresado en grados.
7	0.0-100.0	100.0	Límite inferior del filtro de suavización
8	1.0-100.0	1.0	Factor multiplicativo aplicado a las aceleraciones y a las desaceleraciones definidas en configuración. Aumenta la aceleración y la desaceleración máximas solo de los ejes lineales. Valores externos del intervalo crean el error de sistema 4399 – Parámetro fuera del intervalo.
9	1.0-100.0	1.0	Factor multiplicativo aplicado a las aceleraciones y a las desaceleraciones definidas en configuración. Aumenta la aceleración y la desaceleración máximas solo de los ejes giratorios. Valores externos del intervalo crean el error de sistema 4399 – Parámetro fuera del intervalo.
10	0.0-1.0	0.0	Marca para habilitar (0.0) o deshabilitar (1.0) la reducción de velocidad entre bloques consecutivos en el caso de punto anguloso. La desactivación se puede obtener también utilizando la instrucción con los siguientes parámetros: ISOSETPARAM 0 0.0 ISOSETPARAM 1 0.0 ISOSETPARAM 2 1.0

KINEMATICEXPR*Sintaxis***KINEMATICEXPR****eje = expresión**

Argumentos

eje nombre de dispositivo tipo eje físico o virtual
expresión conjunto de operadores

Descripción

Permite la definición de expresiones individuales de cinemática directa e inversa. Antes de ejecutar esta instrucción se debe llamar la instrucción [ISOG217](#) que describe los ejes físicos y los ejes virtuales que constituyen la máquina. Para cada eje definido en la [ISOG217](#) se debe llamar la instrucción KINEMATICEXPR. La expresión de cinemática de un eje en el espacio articular (cinemática inversa) puede ser función de variables, de constantes y de las cotas de los ejes en el espacio operacional. La expresión de cinemática de un eje en el espacio operacional (cinemática directa) puede ser función de variables, de constantes y de las cotas de los ejes en el espacio articular.

La sintaxis de la **expresión** es la misma descrita en la instrucción [EXPR](#), con la diferencia de que no se pueden utilizar variables locales. Además, no se pueden utilizar ejes que tienen el mismo tipo que el eje declarado en **eje** y no declarados en la instrucción [ISOG217](#). Por ejemplo, definiendo la cinemática de un eje virtual, ya declarado en la instrucción [ISOG217](#); en la expresión solo se puede utilizar los cinco ejes físicos declarados en la [ISOG217](#).

Ejemplos

```
ut as double ; nombre herramienta
offsety as double ; offset Y nariz fulcro
offsetz as double ; offset Z nariz fulcro
```

Function IS05Ax

```
setval 100,ut
setval 120.0,offsety
setval 60.0,offsetz
; CINEMÁTICA EXPLÍCITA
ISOG217 Rx Ry Rz Rc Rb X Y Z C B

; DEFINICIÓN EXPRESIONES DE CINEMÁTICA
; CINEMÁTICA EXPLÍCITA INVERSA EJE físico Rx
KinematicExpr Rx = X - 135 + ut * sin ( B ) * cos ( C )

; CINEMÁTICA EXPLÍCITA INVERSA EJE físico Ry
KinematicExpr Ry = Y + offsety + ut * sin ( B ) * sin ( C )

; CINEMÁTICA EXPLÍCITA INVERSA EJE Rz
KinematicExpr Rz = Z + offsetz + ut * cos ( B )

; CINEMÁTICA EXPLÍCITA INVERSA EJE Rc
KinematicExpr Rc = C

; CINEMÁTICA EXPLÍCITA INVERSA EJE Rb
KinematicExpr Rb = B

; CINEMÁTICA EXPLÍCITA DIRECTA EJE X
KinematicExpr X = Rx + 135 - ut * sin ( Rb ) * cos ( C )

; CINEMÁTICA EXPLÍCITA DIRECTA EJE Y
KinematicExpr Y = Ry - offsety - ut * sin ( Rb ) * sin ( C )

; CINEMÁTICA EXPLÍCITA DIRECTA EJE Z
KinematicExpr Z = Rz - offsetz - ut * cos ( Rb )

; CINEMÁTICA EXPLÍCITA DIRECTA EJE C
KinematicExpr C = Rc

; CINEMÁTICA EXPLÍCITA DIRECTA EJE B
KinematicExpr B = Rb

; MOVIMIENTO
ISOG0 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,0.0
ISOG1 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,1000.0
ISOG1 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,1000.0
ISOG1 1003,X 996.0,Y 600.0,Z 0.0,C 90.0,B 45.0,1000.0
```

WAITSTILL X
ISOM2 X
FRet

10.3.21 Instrucciones no más disponibles

INPBCD	lee una serie de cuartetos digitales en formato BCD
OUTBCD	modifica una serie de cuartetos digitales en formato BCD
SETFORCEBCD	fuerza algunos cuartetos en formato BCD
CANOPENDRIVER	abre un canal de comunicación CANopen
CANCLOSEDRIVER	cierra un canal de comunicación CANopen
CANRESETBOARD	ejecuta la re-activación de una tarjeta CANopen
CANSETOBJECT	escribe un objeto CANopen
CANGETOBJECT	lee un objeto CANopen
SLMCOMMAND	ejecuta un comando SLM
SLMEEPROMDISABLE	ejecuta un comando de inhabilitación a la escritura EEPROM
SLMEEPROMENABLE	ejecuta un comando de habilitación a la escritura EEPROM
SLMGETEEPROM	lee una locación de memoria EEPROM
SLMGETPARAM	lee un parámetro SLM
SLMGETREGISTER	lee un registro SLM
SLMGETSTATUS	lee un tamaño del accionamiento
SLMSETEEPROM	escribe una locación de memoria EEPROM
SLMSETPARAM	programa un parámetro SLM
SLMSETREGISTER	programa un registro SLM
HOMING	ejecuta la "búsqueda del cero"
SYNCRROOPEN	abre un canal de desplazamiento sincronizado
SYNCRROCLOSE	cierra el canal de desplazamiento sincronizado
SYNCRROMOVE	asigna un punto de desplazamiento sincronizado
SYNCRROSETACC	programa la aceleración para los desplazamientos sincronizados
SYNCRROSETDEC	programa la deceleración para los desplazamientos sincronizados
SYNCRROSETVEL	programa la velocidad para los desplazamientos sincronizados
SYNCRROSETFEED	escala la velocidad de los ejes en un desplazamiento sincronizado
SYNCRROSTARTMOVE	inicia la elaboración de un desplazamiento sincronizado
GETVF	lee el valor del convertidor voltaje/frecuencia

10.3.22 Instrucciones no utilizables en evento

Las siguientes instrucciones no se pueden usar en las funciones que se llaman desde las instrucciones [ONFLAG](#), [ONINPUT](#) y [ONERRSYS](#). Su uso está prohibido también en [tareas en tiempo real](#).

Instrucciones que a su vez efectúan una llamada a función en interrupt:

- ONFLAG
- ONINPUT
- ONERRSYS

Instrucciones que conllevan espera:

- WAITINPUT
- WAITFLAG
- WAITACC
- WAITCOLL
- WAITDEC
- WAITREG
- WAITTARGET
- WAITWIN
- WAITSTILL
- WAITTASK
- WAITRECEIVE
- WAITPERSISTINPUT
- MULTIWAITFLAG
- MULTIWAITINPUT

Instrucciones de comunicación:

- SEND
- RECEIVE
- CLEARRECEIVE
- COMOPEN

- COMCLOSE
- COMREAD
- COMREADSTRING
- COMWRITE
- COMWRITESTRING
- COMGETERROR
- COMCLEARRXbúfer
- COMGETRXCOUNT

Las siguientes instrucciones son relacionadas al movimiento de los ejes:

- MOVINC
- MOVABS
- LINEARINC
- LINEARABS
- CIRCLE
- CIRCINC
- CIRCABS
- HELICINC
- HELICABS
- COORDIN
- MULTIABS
- MULTIINC
- SETRIFLOC
- SETTOLERANCE
- RESRIFLOC
- SETPFY
- SETPZERO
- SETINDEXINTERP
- STARTINTERP
- FASTREAD
- ENABLE
- DISABLE
- ENDMOV

Instrucciones ISO:

- ISOG0
- ISOG1
- ISOG9
- ISOG90
- ISOG9
- ISOG93
- ISOG94
- ISOG216
- ISOG217
- ISOM2
- ISOM6
- ISOSETPARAM
- KINEMATICEXPR

Instrucciones BlackBox:

- ENDBLACKBOX
- PAUSEBLACKBOX
- STARTBLACKBOX

Las siguientes instrucciones son relacionadas a la gestión de EtherCAT:

- READDICTIONARY
- WRITEDICTIONARY

Instrucciones relacionadas al multitasking:

- SENDMAIL
- WAITMAIL
- ENDMAIL
- SENDIPC
- WAITIPC
- TESTMAIL
- TESTIPC

Instrucciones que conllevan elevado tiempo de elaboración:

- SORT

- FIND
- FINDB
- MOVEMAT

10.4 Ejemplos

10.4.1 Puesta a cero en Interrupt

```

;-----
; Ejemplo de rutina de puesta a cero al vuelo
; La función ejecuta las siguientes operaciones:
; 1) Programa el eje deshabilitando los límites software
;     y pone a cero la cota
; 2) Verifica que el tastador no esté ya en estado ON.
;     Si está en estado ON mueve el eje y espera que regrese
;     en estado OFF. Si esto no pasa en 30 segundos
;     genera un mensaje de error.
; 3) Programa la velocidad de búsqueda tastador
; 4) Lanza un movimiento del eje y habilita la puesta a cero
;     al vuelo para dicho eje. Al terminar el interrupt
;     la cota del eje es puesta a cero y parte en automático
;     un movimiento a una cota de desempeño.
; 5) Espera la llegada del eje a la cota de desempeño
; 6) Restablece los límites eje
;
; © TPA.
;-----
Function Fast_Puesta_a_cero

  ResLimPos   eje           ; Inicialización eje
  ResLimNeg   eje
  SetQuote    eje,0

  IfInput     FastInput,OFF,Goto Continue ; Test Sensor ocupado
  SetVel      eje,5         ; Programa velocidad de
                               ; desempeño
  MovAbs      eje,30        ; mueve el eje
  WaitInput   FastInput,OFF,30,Call Error ; Controla desempeño micro,
                               ; error después TimeOut=30

  EndMov      eje           ; Paro eje
  WaitStill   eje           ; Espera eje parado

  Continue:
  SetVel      eje,10        ; velocidad de búsqueda
                               ; sensor de puesta a cero
  MovAbs      eje,-1000     ; movimiento en Negativo de
                               ; búsqueda sensor
  SetPFly     eje,ON,10,0   ; Enganche Interrupt
                               ; y configuración cota y
                               ; velocidad de
                               ; desempeño
  WaitStill   eje           ; Espera eje Parado

  SetLimPos   eje           ; Restablece límites eje
  SetLimNeg   eje

  Fret

; sub-procesamiento de envío mensaje de error
Error:
  Error       ERR_SETP     ;advertencia de error
                               ;imposible seguir
  Ret

```


10.4.2 Servidor de desplazamiento ejes

```

-----
; Ejemplo de un servidor de offset ejes:
; El servidor mueve los ejes de la máquina
; por cuenta de otras tareas.
; Las tareas cliente envían los comandos bajo la forma de
; mensajes (correo) y una casilla de correo.
; El servidor toma los comandos de la casilla y los ejecuta.
; Los pedidos son puestos en cola en la casilla, por lo tanto si un
; pedido llega mientras el servidor ya está ocupado,
; no es perdido, más será ejecutado apenas posible.
; El servidos es la única tarea que mueve los ejes. De esta
; forma se evitan conflictos.
; El servidor es implementado por la función Máster_ejes.
; Un ejemplo de cliente es implementado por la función Check_flag.
; Esta función verifica periódicamente el estado de
; un indicador y cuando lo encuentra en estado ON envía al servidor
; el comando de ejecución puesta a cero ejes.
; El indicador será probablemente puesto en ON manualmente
; por el operador, usando, por ejemplo, un cuadro sinóptico.
-----

-----
; -- CONSTANTES GLOBALES DE MAQUINA --
-----

Const MBOX = 101 ; identifica la casilla de correos
                  ; comandos
Const SETP = 10  ; puesta a cero ejes
Const CHG = 11   ; cambio herramienta
Const FORO = 12  ; ejecución de un perforado

;-----
; --- GRUPO EJES ---
;-----

; definición de los mensajes de error
Defmsg ERR_CMD "Comando grupo ejes desconocido"

; --- Servidor ---
Function Máster_ejes autorun

    local cmd as integer          ; comando
    local cota_X as double       ; cota X foro
    local cota_Y as double       ; cota Y foro

loop:
    waitmail MBOX,cmd,cota_X,cota_Y ; espera comando

    ; Recibido el comando lo identificamos
    ; y ejecutamos la acción oportuna
    select cmd

    case SETP
        fcall Puesta_a_cero_ejes          ; Puesta a cero de los ejes
    case CHG
        fcall Cambio_herramienta        ; ejecuta un cambio herramienta
    case FORO

```

```

    fcall Foro cota_X,cota_Y          ; perforación a las cotas
                                      ; especificadas

case else
    call error
endselect

endmail MBOX                          ; notifica ejecución comando
goto loop                              ; regresa en espera de un
                                      ; nuevo comando

fret

; subproceso de envío mensaje de error
error:
    error ERR_CMD
    ret

;-----
; --- GENÉRICO GRUPO ---
;-----

; --- Cliente ---
Function Check_flag

loop:

    ifflag Setp_ejes,OFF, goto loop ; test estado indicador

    ; OK el indicador está en ON, envío el comando
    sendmail MBOX,WAITTACK,SETP,0.0,0.0

    resetflag Setp_ejes              ; reseteo el indicador

    goto loop                        ; regreso en espera

fret

; NOTE QUE:
; - después del comando "SETP", hay que especificar
;   los dos parámetros "cota_X" y "cota_Y" aunque si esto
;   no tiene sentido para la operación de Puesta a cero.
;   El servidor sin embargo no puede saber desde el principio
;   cual comando le será enviado, por lo tanto tenemos que
;   especificar dos valores del mismo tipo de los que el
;   servidor espera recibir, en este caso dos DOBLE.
;   Los valores pasados son "0.0" y "0.0".
; - El parámetro "WAITACK" trata la manera que el cliente
;   quede en espera de la avenida ejecución del comando
;   de parte del servidor.
;   El cliente proseguirá la propia ejecución solo cuando
;   el servidor habrá ejecutado una ENDMAIL o habrá
;   iniciado a procesar un nuevo comando (WAITMAIL).

```

10.4.3 Ciclo de Main con gestión errores

```

;-----
; Hipotética función principal
; inicializa la máquina y ejecuta un ciclo de control
;-----
Function Main
    OnErrSys    GestErrSys    ; Habilita gestión errores

    StartTask   Emergencias   ; inicializa
    StartTask   Elaborador

    HabilitaEjes

```

```

loop:
  IfFlag      Flag,OFF, RestablEmerg
  .....
  Goto       loop
Fret

;-----
; función de gestión de los errores
;-----
Function GestErrSys
  Param nerror as integer
  Param tarea as function
  Param tipodevice as device

  EndTasElaborador      ; Termina tarea
  k                      ; Elaborador
  If nerror,>,5,got      ; Los primeros 5 errores
  onoerraxis             ; son relativos
                       ; a los ejes

  ResetFlag  Indicador
  DeshabilitaEjes

noerraxis:
Fret

```

10.4.4 Operaciones con las cadenas

```

;-----
; Ejemplo de manipulación de las cadenas
;-----
Function Ejemplo
  Local cadena1 as string
  Local cadena2 as string
  Local cadena3 as string
  Local longitud as integer
  Local posición as integer

  SetString      "cadena de",cadena1      ; cadena1 contiene
                                           ; ahora "cadena de"

  SetString      " prueba",cadena2

  AddString      cadena1,cadena2,cadena3  ; cadena3 contiene
                                           ; "cadena de prueba"

  Search         cadena3,'d',posición     ; posición vale 2
  Search         cadena3,'z',posición     ; posición vale -1

  Left          cadena3,6,cadena1        ; cadena1
                                           ; contiene "Cadena"

  Right         cadena3,2,cadena2        ; cadena2
                                           ; contiene "ba"

  Mid           cadena3,8,2,cadena3      ; cadena3
                                           ; contiene "de"

  ControlChar    65,cadena1              ; cadena1
                                           ; contiene "A"

  Len           cadena3,longitud         ; longitud vale 2

  Str           longitud,cadena3         ; cadena3
                                           ; contiene "2"

  Val          posición,cadena1         ; cadena1

```

```

; contiene "-1"
AddString "El resultado es ",cadena1,cadena2
; cadena2 contiene "El resultado es -1"

```

Fret

10.4.5 Ejecución Secuencial/Paralela

```

;-----
; Ejemplo de rutina que controla la puesta a cero
; de una máquina a 3 ejes evitando eventuales
; interferencias mecánicas.
;
; Las puestas a cero de cada eje son implementadas
; por unas funciones cuyo texto es omitido.
; Véase el ejemplo "Rutina de puesta a cero".
;
; Es efectuado primero solo la puesta a cero
; del eje Z (que suponemos no pueda ser
; efectuado al mismo tiempo de los demás),
; terminado esta son ejecutados
; al mismo tiempo las puestas a cero de los ejes
; X y Y.
;-----
; mensaje para el operador (traducido en lengua)
DefMsg MSG_SETP ESP "puesta a cero ejes en curso..."
ENG "Homing in progress..."

Function PuestaACeroEjes

Message MSG_SETP ; informa al operador

Fcall PuestaACeroEjeZ ; puesta a cero del eje Z

; OK la puesta a cero del eje Z se ha terminado

StartTask PuestaACeroEjeX ; lanza la puesta a cero X y Y
StartTask PuestaACeroEjeY

WaitTask PuestaACeroEjeX ; espera la terminación
WaitTask PuestaACeroEjeY

DelMessage MSG_SETP ; cancela el mensaje
; para el operador

```

Fret

10.4.6 Rutina de Puesta a cero

```

;-----
; Ejemplo de rutina de puesta a cero de un eje
;
; La funcion ejecuta las siguientes operaciones:
; 1) deshabilita los límites software del eje
; 2) programa la velocidad de búsqueda interruptor
; 3) mueve el eje a una cota incremental que
; garantiza el alcance del interruptor
; 4) espera que el eje haga saltar el interruptor
; 5) para el eje y espera la fin del movimiento
; 6) programa la velocidad (baja) de desempeño interruptor
; 7) hace retroceder el eje cuanto basta
; para desempeñar el interruptor
; 8) espera el desempeño del interruptor
; 9) asigna la nueva cota al eje

```

```

; 10) restablece la velocidad predeterminada y los límites software
;
;   © TPA.
;-----

```

Function Puesta_a_cero

```

ResLi eje ; deshabilita los límites software
mPos
ResLi eje
mNeg

SetVe eje,1; programa la velocidad
l 0

MovIn eje,1; mueve el eje
c 0000

waitI Switc; espera interruptor
nput h,ON

EndMo eje ; para el eje
v
waits eje ; espera eje parado
till
SetVe eje, ; programa velocidad desempeño
l 0.1

MovIn eje,-; mueve el eje
c 100

waitI Switc; espera desempeño interruptor
nput h,OFF

SetQu eje,0; asigna la nueva cota
ote

SetVe eje ; restablece velocidad
l
SetLi eje ; restablece límites software
mpos
SetLi eje
mneg

```

Fret

10.4.7 Movimientos ISO

```

;-----
; Ejemplo de movimiento ISO
;
; Es generado un perfil mediante las instrucciones ISOG0 e ISOG1
;
;   © TPA.
;-----*

; Declaración de las matrices ISO
; Matriz ejes Rotatorios
MzRot[5] as double:off_X double:off_Y double:off_Z double:dis_X _
double:dis_Y double:dis_Z double:delta double:gamma
; Matriz portaherramientas
MzPorta[1] as double:off_X double:off_Y double:off_Z _double:alpha
double:beta
; Matriz herramientas
MzHerramientas[10] as double:ut double

```

Function InterpolaciónISO

```

; programación de los valores estándar de parametrización máquina
setval 90.0 MzRot[5].gamma
setval 260.3 MzHerramientas[10].ut
setval MxHerramientas[10].ut ut

; programación de los parámetros del algoritmo
IsoSetParam 0 50
IsoSetParam 1 50
IsoSetParam 2 0.9
IsoSetParam 3 60
IsoSetParam 4 30

; programación de la máquina; declara las tres matrices
; para la parametrización de la máquina y los ejes físicos utilizados
; en los movimientos ISO
isoG216 MzRot MzPorta MzHerramientas 31 X Y Z C B ; CINEMÁTICA IMPLÍCITA

; programación del grupo de parámetros que describen la cinemática
; de la máquina.
isoM6 X 5 1 10 ; CINEMÁTICA IMPLÍCITA

; programación de los valores iniciales
setquote x 500
setquote y 300
setquote z 0
setquote c 0
setquote b 0
setvel x
setvel y
setvel z
setvel c
setvel b
setveli x y z c b

; ejecución del perfil
isoG0 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,50.0
isoG1 1001,X 998.0,Y 600.0,Z 0.0,C 90.0,B 45.0,10000.0
isoG1 1003,X 996.0,Y 600.0,Z 0.0,C 90.0,B 45.0,10000.0
isoG1 1002,X 600.0,Y 600.0,Z 0.0,C 90.0,B 45.0,10000.0
isoG1 1004,X 599.131759111665,Y 599.924038765061,Z 0,C 100,B 45.0,10000.0
isoG1 1006,X 598.289899283372,Y 599.69846310393,Z 0,C 110,B 45.0,10000.0
isoG1 1005,X 597.5,Y 599.330127018922,Z 0,C 120,B 45.0,10000.0
isoG1 1003,X 596.786061951567,Y 598.830222215595,Z 0,C 130,B 45.0,10000.0
isoG1 1002,X 596.169777784405,Y 598.213938048433,Z 0,C 140,B 45.0,10000.0
isoG1 1012,X 595.669872981078,Y 597.5,Z 0,C 150,B 45.0,10000.0
isoG1 1011,X 595.301536896071,Y 596.710100716628,Z 0,C 160,B 45.0,10000.0
isoG1 1031,X 595.075961234939,Y 595.868240888335,Z 0,C 170,B 45.0,10000.0
isoG1 1102,X 595.0,Y 0.0,Z 0.0,C 180.0,B 45.0,10000.0
waitstill X Y Z C B
fret

```

Tecnologie e Prodotti per l'Automazione srl

Via Carducci 221
I - 20099 Sesto S.Giovanni
(MI)
Tel. +39 02.36527550
Fax. +39 02.2481008
www.tpaspa.com