

Concept 2.6

Biblioteca de módulos IEC

Parte: IEC

01/2007

Tabla de materias



	Información de seguridad	13
	Acerca de este libro	15
Parte I	Generalidades acerca de la biblioteca de bloques IEC.	17
	Vista general	17
Capítulo 1	Parametrización de funciones y bloques de función	19
	Parametrización de funciones y bloques de función	20
Parte II	Descripciones EFB.	23
	Vista general	23
Capítulo 2	ABS_***: Formación del valor absoluto	27
	Vista general	27
	Descripción breve	28
	Representación	28
	Error de tiempo de ejecución	28
Capítulo 3	ACOS_REAL: Arco coseno en la medida de arco	29
	Vista general	29
	Descripción breve	30
	Representación	30
	Error de tiempo de ejecución	30
Capítulo 4	ADD_***: Adición	31
	Vista general	31
	Descripción breve	32
	Representación	32
	Error de tiempo de ejecución	33

Capítulo 5	AND_***: Función Y	35
	Vista general	35
	Descripción breve	36
	Representación	36
Capítulo 6	ASIN_REAL: Arco seno en la medida de arco	37
	Vista general	37
	Descripción breve	38
	Representación	38
	Error de tiempo de ejecución	38
Capítulo 7	ATAN_REAL: Arco tangencial en la medida de arco	39
	Vista general	39
	Descripción breve	40
	Representación	40
	Error de tiempo de ejecución	40
Capítulo 8	BOOL_TO_***: Conversión de tipo	41
	Vista general	41
	Descripción breve	42
	Representación	42
Capítulo 9	BYTE_TO_***: Conversión de tipo	43
	Vista general	43
	Descripción breve	44
	Representación	44
	Error de tiempo de ejecución	45
Capítulo 10	COS_REAL: Coseno	47
	Vista general	47
	Descripción breve	48
	Representación	48
	Error de tiempo de ejecución	48
Capítulo 11	CTD: Contador hacia abajo	49
	Vista general	49
	Descripción breve	50
	Representación	50
Capítulo 12	CTU: Contador hacia arriba	51
	Vista general	51
	Descripción breve	52
	Representación	52

Capítulo 13	CTUD: Contador hacia arriba/abajo	53
	Vista general	53
	Descripción breve	54
	Representación	55
Capítulo 14	DINT_EXPT_REAL: Exponencialización	57
	Vista general	57
	Descripción breve	58
	Representación	58
	Error de tiempo de ejecución	58
Capítulo 15	DINT_TO_***: Conversión de tipo	59
	Vista general	59
	Descripción breve	60
	Representación	60
	Error de tiempo de ejecución	61
Capítulo 16	DIV_***: División	63
	Vista general	63
	Descripción breve	64
	Representación	64
	Error de tiempo de ejecución	65
Capítulo 17	EQ_***: Igual	67
	Vista general	67
	Descripción breve	68
	Representación	68
	Error de tiempo de ejecución	69
Capítulo 18	EXP_REAL: Función exponencial	71
	Vista general	71
	Descripción breve	72
	Representación	72
	Error de tiempo de ejecución	72
Capítulo 19	F_TRIG: Detección de flancos descendientes	73
	Vista general	73
	Descripción breve	74
	Representación	74
Capítulo 20	GE_***: Mayor igual a	75
	Vista general	75
	Descripción breve	76
	Representación	76
	Error de tiempo de ejecución	76

Capítulo 21	GT_***: Mayor	77
	Vista general	77
	Descripción breve	78
	Representación	78
	Error de tiempo de ejecución	78
Capítulo 22	INT_EXPT_REAL: Exponencialización	79
	Vista general	79
	Descripción breve	80
	Representación	80
	Error de tiempo de ejecución	80
Capítulo 23	INT_TO_***: Conversión de tipo	81
	Vista general	81
	Descripción breve	82
	Representación	82
	Error de tiempo de ejecución	83
Capítulo 24	LE_***: Menor igual a	85
	Vista general	85
	Descripción breve	86
	Representación	86
	Error de tiempo de ejecución	86
Capítulo 25	LIMIT_***: Limitación	87
	Vista general	87
	Descripción breve	88
	Representación	89
	Error de tiempo de ejecución	89
Capítulo 26	LN_REAL: Logaritmo natural	91
	Vista general	91
	Descripción breve	92
	Representación	92
	Error de tiempo de ejecución	92
Capítulo 27	LOG_REAL: Logaritmo respecto a la base 10	93
	Vista general	93
	Descripción breve	94
	Representación	94
	Error de tiempo de ejecución	94
Capítulo 28	LT_***: Menor	95
	Vista general	95
	Descripción breve	96
	Representación	96
	Error de tiempo de ejecución	96

Capítulo 29	MAX_***: Selección de valor máximo	97
	Vista general	97
	Descripción breve	98
	Representación	98
	Error de tiempo de ejecución	99
Capítulo 30	MIN_***: Selección de valor mínimo	101
	Vista general	101
	Descripción breve	102
	Representación	102
	Error de tiempo de ejecución	103
Capítulo 31	MOD_***: Módulo	105
	Vista general	105
	Descripción breve	106
	Representación	106
Capítulo 32	MOVE: Asignación	107
	Vista general	107
	Descripción breve	108
	Representación	108
Capítulo 33	MUL_***: Multiplicación	109
	Vista general	109
	Descripción breve	110
	Representación	110
	Error de tiempo de ejecución	110
Capítulo 34	MUX_***: Multiplexor	111
	Vista general	111
	Descripción breve	112
	Representación	113
Capítulo 35	NE_***: Desigual	115
	Vista general	115
	Descripción breve	116
	Representación	116
	Error de tiempo de ejecución	116
Capítulo 36	NOT_***: Negación	117
	Vista general	117
	Descripción breve	118
	Representación	118

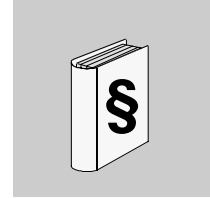
Capítulo 37	OR_***: Función O	119
	Vista general	119
	Descripción breve	120
	Representación	120
Capítulo 38	R_TRIG: Detección de flancos ascendentes	121
	Vista general	121
	Descripción breve	122
	Representación	122
Capítulo 39	REAL_EXPT_REAL: Exponencialización	123
	Vista general	123
	Descripción breve	124
	Representación	124
	Error de tiempo de ejecución	124
Capítulo 40	REAL_TO_***: Conversión de tipo	125
	Vista general	125
	Descripción breve	126
	Representación	127
	Error de tiempo de ejecución	127
Capítulo 41	REAL_TRUNC_***: Conversión de tipo	129
	Vista general	129
	Descripción breve	130
	Representación	130
	Error de tiempo de ejecución	131
Capítulo 42	ROL_***: Rotar a la izquierda	133
	Vista general	133
	Descripción breve	134
	Representación	134
Capítulo 43	ROR_***: Rotar a la derecha	135
	Vista general	135
	Descripción breve	136
	Representación	136
Capítulo 44	RS: Bloque de función biestable, restablecer dominante ..	137
	Vista general	137
	Descripción breve	138
	Representación	138

Capítulo 45	SEL: Selección binaria	139
	Vista general	139
	Descripción breve	140
	Representación	140
Capítulo 46	SHL_***: Desplazar a la izquierda	141
	Vista general	141
	Descripción breve	142
	Representación	142
Capítulo 47	SHR_***: Desplazar a la derecha	143
	Vista general	143
	Descripción breve	144
	Representación	144
Capítulo 48	SIN_REAL: Seno	145
	Vista general	145
	Descripción breve	146
	Representación	146
	Error de tiempo de ejecución	146
Capítulo 49	SQRT_REAL: Raiz cuadrada	147
	Vista general	147
	Descripción breve	148
	Representación	148
	Error de tiempo de ejecución	148
Capítulo 50	SR: Bloque de función biestable, ubicar dominante	149
	Vista general	149
	Descripción breve	150
	Representación	150
Capítulo 51	SUB_***: Sustracción	151
	Vista general	151
	Descripción breve	152
	Representación	152
	Error de tiempo de ejecución	152
Capítulo 52	TAN_REAL: Tangencial	153
	Vista general	153
	Descripción breve	154
	Representación	154
	Error de tiempo de ejecución	154

Capítulo 53	TIME_DIV_***: División de valores temporales	155
	Vista general	155
	Descripción breve	156
	Representación	156
	Error de tiempo de ejecución	156
Capítulo 54	TIME_MUL_***: Multiplicación de valores temporales	157
	Vista general	157
	Descripción breve	158
	Representación	158
	Error de tiempo de ejecución	158
Capítulo 55	TIME_TO_***: Conversión de tipo	159
	Vista general	159
	Descripción breve	160
	Representación	160
	Error de tiempo de ejecución	161
Capítulo 56	TOF: Retardo de desconexión	163
	Vista general	163
	Descripción breve	164
	Representación	164
	Descripción detallada	165
Capítulo 57	TON: Retardo de conexión	167
	Vista general	167
	Descripción breve	168
	Representación	168
	Descripción detallada	169
Capítulo 58	TP: Impulso	171
	Vista general	171
	Descripción breve	172
	Representación	172
	Descripción detallada	173
Capítulo 59	UDINT_EXPT_REAL: Exponencialización	175
	Vista general	175
	Descripción breve	176
	Representación	176
	Error de tiempo de ejecución	176

Capítulo 60	UDINT_TO_***: Conversión de tipo	177
	Vista general	177
	Descripción breve	178
	Representación	178
	Error de tiempo de ejecución	178
Capítulo 61	UINT_EXPT_REAL: Exponencialización	179
	Vista general	179
	Descripción breve	180
	Representación	180
	Error de tiempo de ejecución	180
Capítulo 62	UINT_TO_***: Conversión de tipo	181
	Vista general	181
	Descripción breve	182
	Representación	182
	Error de tiempo de ejecución	182
Capítulo 63	WORD_TO_***: Conversión de tipo	183
	Vista general	183
	Descripción breve	184
	Representación	185
	Error de tiempo de ejecución	185
Capítulo 64	XOR_***: Función O exclusivo	187
	Vista general	187
	Descripción breve	188
	Representación	188
Glosario	189
Índice	215

Información de seguridad



Información importante

AVISO

Lea atentamente estas instrucciones y observe el equipo para familiarizarse con el dispositivo antes de instalarlo, utilizarlo o realizar su mantenimiento. Los mensajes especiales que se ofrecen a continuación pueden aparecer a lo largo de la documentación o en el equipo para advertir de peligros potenciales o para ofrecer información que aclare o simplifique los distintos procedimientos.



La inclusión de este icono en una etiqueta de peligro o advertencia indica un riesgo de descarga eléctrica, que puede provocar lesiones si no se siguen las instrucciones.



Éste es el icono de alerta de seguridad. Se utiliza para advertir de posibles riesgos de lesiones. Observe todos los mensajes que siguen a este icono para evitar posibles lesiones o incluso la muerte.

PELIGRO

PELIGRO indica una situación inminente de peligro que, si no se evita, **provocará** lesiones graves o incluso la muerte.

ADVERTENCIA

ADVERTENCIA indica una posible situación de peligro que, si no se evita, **puede provocar** daños en el equipo, lesiones graves o incluso la muerte.

ADVERTENCIA

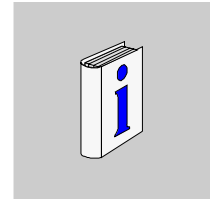
ADVERTENCIA indica una posible situación de peligro que, si no se evita, **puede provocar** lesiones o daños en el equipo.

**TENGA EN
CUENTA**

Sólo el personal de servicio cualificado podrá instalar, utilizar, reparar y mantener el equipo eléctrico. Schneider Electric no asume las responsabilidades que pudieran surgir como consecuencia de la utilización de este material.

© 2007 Schneider Electric. Todos los derechos reservados.

Acerca de este libro



Presentación

Objeto

Esta documentación le ayudará en la configuración de las funciones y los módulos de función.

Campo de aplicación

Esta documentación es válida para Concept 2.6 en Microsoft Windows 98, Microsoft Windows 2000, Microsoft Windows XP y Microsoft Windows NT 4.x.

Nota: Encontrará más información actualizada en el archivo README de Concept.

Documentos relacionados

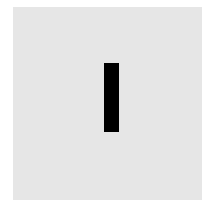
Título	Reference Number
Instrucciones para la instalación de Concept	840 USE 502 03
Manual de usuario de Concept	840 USE 503 03
Concept EFB User Manual	840 USE 505 00
Biblioteca de módulos LL984 de Concept	840 USE 506 03

Puede descargar estas publicaciones técnicas y otra información técnica de nuestro sitio Web en www.telemecanique.com

Comentarios del usuario

Envíe sus comentarios a la dirección electrónica techpub@schneider-electric.com

Generalidades acerca de la biblioteca de bloques IEC



Vista general

Introducción Esta sección contiene informaciones generales acerca de la biblioteca de bloques IEC.

Contenido Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
1	Parametrización de funciones y bloques de función	19

Parametrización de funciones y bloques de función

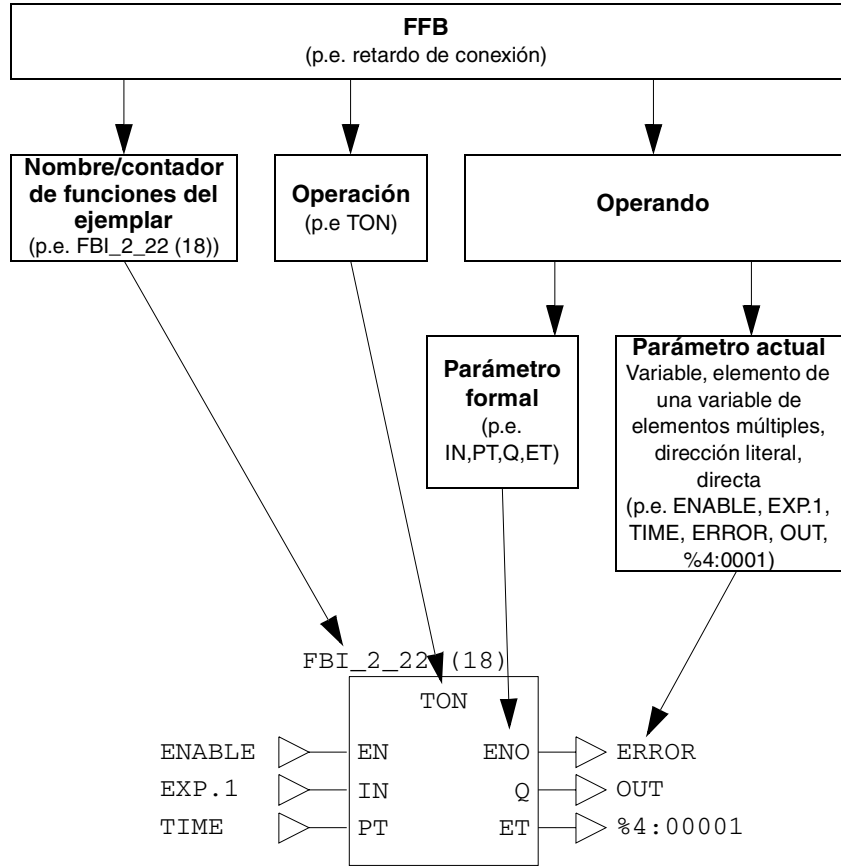


1

Parametrización de funciones y bloques de función

Generalidades

Cada FFB se compone de una operación, los operandos necesarios para la operación y un nombre de ejemplar/contador de funciones.



Operación

La operación establece la funcionalidad que se debe ejecutar por el FFB, p. ej., registro de corrimientos, operaciones de conversión

Operando

El operando determina con lo que se debe ejecutar la operación. En FFBs está compuesto de parámetros formales y parámetros actuales.

Parámetros formales / Parámetros actuales

El parámetro formal es una reserva de lugar para un operando. Durante la parametrización se le asigna un parámetro actual al parámetro formal (parámetro actual).

El parámetro actual puede ser una variable, una variable de elementos múltiples, un elemento de una variable de elementos múltiples, un literal o una dirección directa.

Llamada condicional / incondicional

Cada FFB dispone de la posibilidad de la llamada "condicional" o "incondicional". La condición se realiza mediante una unión delante de la entrada EN.

- EN visualizada
Llamada condicional (sólo si el EN = 1 se procesará FFB)
- EN oculta
Llamada incondicional (el FFB será siempre procesado)

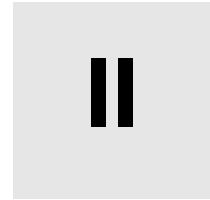
Nota: Si no se parametriza la entrada EN ésta se deberá ocultar. Debido a que las entradas no parametrizadas se ocupan automáticamente con un "0", el FFB no se procesaría jamás en caso contrario.

Nota: En el caso de los bloques de funciones bloqueados (EN = 0) con función de tiempo interna (por ejemplo: DELAY) el tiempo parece seguir corriendo, ya que cuentan con la ayuda del reloj del sistema y no dependen del ciclo del programa ni de que el bloque esté habilitado.

Llamada a funciones y bloques de función en IL y ST

La llamada a funciones y bloques de función en IL (lista de instrucciones) y ST (texto estructurado) deberá consultarlas en los correspondientes capítulos del Manual de usuario.

Descripciones EFB



Vista general

Introducción

Las descripciones EFB están documentadas en orden alfabético.

Nota: El número de entradas de algunos EFBs se puede aumentar mediante modificación vertical del tamaño del símbolo FFB a un máximo de 32. De qué EFBs en concreto se trata, figura en la descripción de los diferentes EFBs.

Contenido

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
2	ABS_***: Formación del valor absoluto	27
3	ACOS_REAL: Arco coseno en la medida de arco	29
4	ADD_***: Adición	31
5	AND_***: Función Y	35
6	ASIN_REAL: Arco seno en la medida de arco	37
7	ATAN_REAL: Arco tangencial en la medida de arco	39
8	BOOL_TO_***: Conversión de tipo	41
9	BYTE_TO_***: Conversión de tipo	43
10	COS_REAL: Coseno	47
11	CTD: Contador hacia abajo	49
12	CTU: Contador hacia arriba	51
13	CTUD: Contador hacia arriba/abajo	53
14	DINT_EXPT_REAL: Exponencialización	57
15	DINT_TO_***: Conversión de tipo	59
16	DIV_***: División	63
17	EQ_***: Igual	67
18	EXP_REAL: Función exponencial	71
19	F_TRIG: Detección de flancos descendientes	73
20	GE_***: Mayor igual a	75
21	GT_***: Mayor	77
22	INT_EXPT_REAL: Exponencialización	79
23	INT_TO_***: Conversión de tipo	81
24	LE_***: Menor igual a	85
25	LIMIT_***: Limitación	87
26	LN_REAL: Logaritmo natural	91
27	LOG_REAL: Logaritmo respecto a la base 10	93
28	LT_***: Menor	95
29	MAX_***: Selección de valor máximo	97
30	MIN_***: Selección de valor mínimo	101
31	MOD_***: Módulo	105
32	MOVE: Asignación	107
33	MUL_***: Multiplicación	109
34	MUX_***: Multiplexor	111

Capítulo	Nombre del capítulo	Página
35	NE_***: Desigual	115
36	NOT_***: Negación	117
37	OR_***: Función O	119
38	R_TRIG: Detección de flancos ascendentes	121
39	REAL_EXPT_REAL: Exponencialización	123
40	REAL_TO_***: Conversión de tipo	125
41	REAL_TRUNC_***: Conversión de tipo	129
42	ROL_***: Rotar a la izquierda	133
43	ROR_***: Rotar a la derecha	135
44	RS: Bloque de función biestable, restablecer dominante	137
45	SEL: Selección binaria	139
46	SHL_***: Desplazar a la izquierda	141
47	SHR_***: Desplazar a la derecha	143
48	SIN_REAL: Seno	145
49	SQRT_REAL: Raíz cuadrada	147
50	SR: Bloque de función biestable, ubicar dominante	149
51	SUB_***: Sustracción	151
52	TAN_REAL: Tangencial	153
53	TIME_DIV_***: División de valores temporales	155
54	TIME_MUL_***: Multiplicación de valores temporales	157
55	TIME_TO_***: Conversión de tipo	159
56	TOF: Retardo de desconexión	163
57	TON: Retardo de conexión	167
58	TP: Impulso	171
59	UDINT_EXPT_REAL: Exponencialización	175
60	UDINT_TO_***: Conversión de tipo	177
61	UINT_EXPT_REAL: Exponencialización	179
62	UINT_TO_***: Conversión de tipo	181
63	WORD_TO_***: Conversión de tipo	183
64	XOR_***: Función O exclusivo	187

ABS_***: Formación del valor absoluto

2

Vista general

Introducción Este capítulo describe el bloque ABS_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	28
Representación	28
Error de tiempo de ejecución	28

Descripción breve

Descripción de la función

La función supone el valor absoluto del valor de entrada y emite el mismo en la salida.

Se pueden procesar los tipos de datos del grupo ANY_NUM.

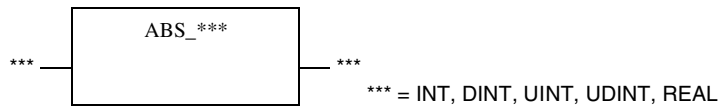
Los tipos de datos del valor de entrada y salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$$\text{OUT} = |\text{IN}|$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	INT, DINT, UINT, UDINT, REAL	Valor de entrada
OUT	INT, DINT, UINT, UDINT, REAL	Valor de salida

Error de tiempo de ejecución

Aviso de error

Si se genera para un parámetro de entrada del tipo de datos REAL un número de coma flotante inadmisiblemente, se emite un aviso de error.

ACOS_REAL: Arco coseno en la medida de arco

3

Vista general

Introducción Este capítulo describe el bloque ACOS_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	30
Representación	30
Error de tiempo de ejecución	30

Descripción breve

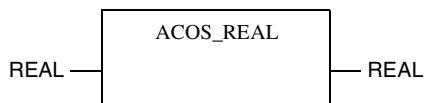
Descripción de la función La función calcula el arco coseno del valor de entrada y emite el mismo en la medida de arco en la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = arc cos IN

Requisitos:

$-1 \leq IN \leq 1$

Ello rige:

$0 \leq OUT \leq \pi$

Descripción de parámetros

Descripción de los parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada
OUT	REAL	Valor de salida en la medida de arco

Error de tiempo de ejecución

Aviso de error

Si durante la ejecución de la función se produce una vulneración del margen de valores en la entrada o si se trata de un número de coma flotante inadmisibles, se emite un aviso de error.

ADD_***: Adición



Vista general

Introducción Este capítulo describe el bloque ADD_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	32
Representación	32
Error de tiempo de ejecución	33

Descripción breve

Descripción de la función

La función suma los valores de entrada del grupo ANY_NUM o del tipo de datos TIME y emite el resultado en la salida.

Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

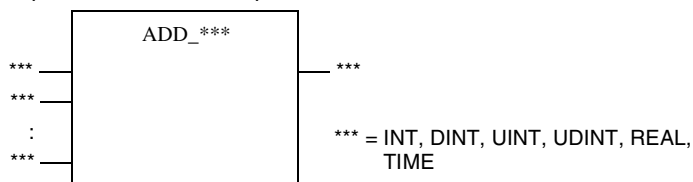
Se puede aumentar el número de entradas para todas las funciones con excepción de ADD_TIME.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

INT, DINT, UINT, UDINT, REAL:

$$\text{OUT} = \text{IN1} + \text{IN2} + \dots + \text{INn}$$

TIME:

$$\text{OUT} = \text{IN1} + \text{IN2}$$

Descripción de parámetros

Descripción de los parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME	Sumando
IN2	INT, DINT, UINT, UDINT, REAL, TIME	Sumando
INn	INT, DINT, UINT, UDINT, REAL	Sumando
OUT	INT, DINT, UINT, UDINT, REAL, TIME	Suma

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error si

- se rebasa el margen de valores en la salida
 - y se dispone de un número de coma flotante inadmisibles para un parámetro de entrada del tipo de datos REAL.
-

AND_***: Función Y

5

Vista general

Introducción Este capítulo describe el bloque AND_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	36
Representación	36

Descripción breve

Descripción de la función La función vincula (según la lógica Y) las secuencias de bits en las entradas y emite el resultado en la salida. La vinculación se realiza por bits.

Se pueden procesar los tipos de datos del grupo ANY_BIT.

Nota: Esta función no está disponible con variables booleanas en el lenguaje de programación LD (Ladder Diagram), ya que allí se puede realizar la misma funcionalidad con contactos.

Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

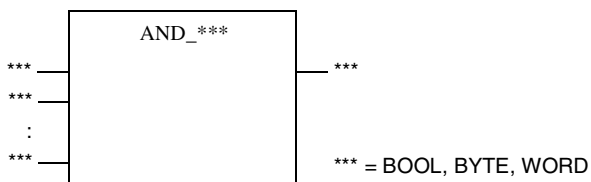
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = IN1 \& IN2 \& INn$

Descripción de parámetros

Descripción de los parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	BOOL, BYTE, WORD	Secuencia de bits de entrada
IN2	BOOL, BYTE, WORD	Secuencia de bits de entrada
INn	BOOL, BYTE, WORD	Secuencia de bits de entrada
OUT	BOOL, BYTE, WORD	Secuencia de bits de salida

ASIN_REAL: Arco seno en la medida de arco

6

Vista general

Introducción Este capítulo describe el bloque ASIN_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	38
Representación	38
Error de tiempo de ejecución	38

Descripción breve

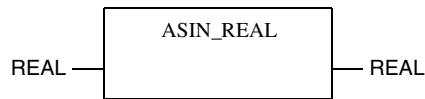
Descripción de la función La función calcula el arco seno del valor de entrada y emite el resultado en la medida de arco en la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = arc sin (IN)

Requisito:

IN -1 ≤ IN ≤ 1

En ello rige:

$$-\frac{\pi}{2} \leq \text{OUT} \leq \frac{\pi}{2}$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada
OUT	REAL	Valor de salida en la medida de arco

Error de tiempo de ejecución

Aviso de error

Si durante la ejecución de la función se produce una vulneración del margen de valores en la entrada o si se trata de un número de coma flotante inadmisibles, se emite un aviso de error.

ATAN_REAL: Arco tangencial en la medida de arco



Vista general

Introducción Este capítulo describe el bloque ATAN_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	40
Representación	40
Error de tiempo de ejecución	40

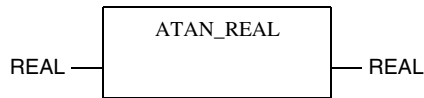
Descripción breve

Descripción de la función La función calcula el arco tangencial del valor de entrada y emite el resultado en la medida de arco en la salida.
Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = arc tan (IN)

En ello rige:

$$-\frac{\pi}{2} \leq \text{OUT} \leq \frac{\pi}{2}$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada
OUT	REAL	Valor de salida en la medida de arco

Error de tiempo de ejecución

Aviso de error

Si la entrada dispone de un número de coma flotante inadmisibles, se emite un aviso de error.

BOOL_TO_***: Conversión de tipo

8

Vista general

Introducción Este capítulo describe el bloque BOOL_TO_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	42
Representación	42

Descripción breve

Descripción de la función La función convierte un valor de entrada del tipo de datos BOOL en un tipo de datos del grupo ANY_NUM o del tipo de datos BYTE, WORD o TIME.

Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

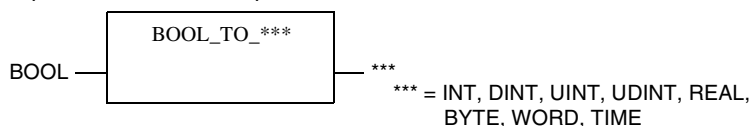
Se ubica el bit con el menor valor del valor de salida al valor de entrada. Todos los demás bits se ubican en cero.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	BOOL	Valor de entrada
OUT	INT, DINT, UINT, UDINT, REAL, BYTE, WORD, TIME	Valor de salida

BYTE_TO_***: Conversión de tipo

9

Vista general

Introducción Este capítulo describe el bloque BYTE_TO_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	44
Representación	44
Error de tiempo de ejecución	45

Descripción breve

Descripción de la función

La función convierte un valor de entrada del tipo de datos BYTE en un tipo de datos del grupo ANY_NUM o del tipo de datos BOOL, WORD o TIME.

Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Nota: El EFB convierte estrictamente según las reglas CEI. Como este EFB ha sido realizado como función genérica, también se han producido algunas transformaciones ilógicas, p.e. BYTE_TO_TIME. En este sentido se debe tener en cuenta que la muestra del bit de entrada es transmitida a la palabra de mayor valor de la palabra de salida.

Durante la conversión del tipo de datos BYTE en un tipo de datos del grupo ANY_NUM o WORD se transmite la muestra de bit de la entrada al bit de menor valor de la salida. Los bits de mayor valor de la salida se ubican en cero.

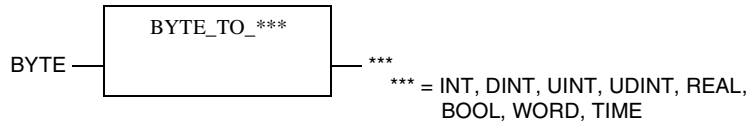
Durante la conversión del tipo de datos BYTE en el tipo de datos BOOL se transmite el bit de menor valor del valor de entrada a la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	BYTE	Valor de entrada
OUT	INT, DINT, UINT, UDINT, REAL, BOOL, WORD, TIME	Valor de salida

Error de tiempo de ejecución

Aviso de error Si durante la conversión al tipo de datos REAL se genera un número de coma flotante inadmisibles, se emite un aviso de error.

COS_REAL: Coseno

10

Vista general

Introducción Este capítulo describe el bloque COS_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	48
Representación	48
Error de tiempo de ejecución	48

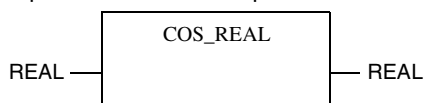
Descripción breve

Descripción de la función La función calcula el coseno del valor de entrada y emite el mismo en la salida. La introducción del valor de entrada se debe realizar en la medida de arco.
Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = cos (IN)

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada en la medida de arco
OUT	REAL	Valor de salida

Error de tiempo de ejecución

Aviso de error

Si la entrada dispone de un número de coma flotante inadmisibles, se emite un aviso de error.

CTD: Contador hacia abajo

11

Vista general

Introducción Este capítulo describe el bloque CTD.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	50
Representación	50

Descripción breve

Descripción de funcionamiento

El módulo de función se utiliza para el conteo regresivo de valores INT.

Los módulos de función para el conteo de valores DINT, UDINT y UINT figuran en la biblioteca Extended.

En caso de señal "1" en la entrada LD se asigna el valor de la entrada PV a la salida CV. En cada transición de "0" a "1" en la entrada CD, el valor de CV se reduce en 1.

Si $CV \leq 0$, la salida Q pasa a "1".

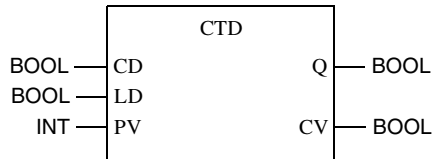
Nota: El contador sólo funciona hasta alcanzar los valores mínimos del CV de salida. No se produce ningún desborde.

Como parámetros adicionales se pueden configurar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
CD	BOOL	Entrada de disparador
LD	BOOL	Carga de datos
PV	INT	Valor de ajuste previo
Q	BOOL	Salida
CV	INT	Valor de cómputo (valor real)

CTU: Contador hacia arriba

12

Vista general

Introducción Este capítulo describe el bloque CTU.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	52
Representación	52

Descripción breve

Descripción de funcionamiento

El módulo de función se utiliza para el conteo progresivo de valores INT.

Los módulos de función para el conteo de valores DINT, UDINT y UINT figuran en la biblioteca Extended.

En caso de señal "1" en la entrada R, se asigna el valor "0" a la salida CV. En cada transición de "0" a "1" en la entrada CU, el valor de CV aumenta en 1.

En caso de $CV \geq PV$ se convierte la salida Q en "1".

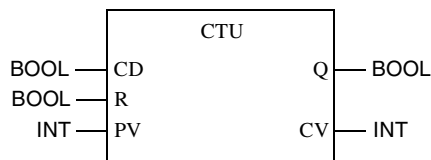
Nota: El contador sólo funciona hasta alcanzar los valores mínimos del CV de salida. No se produce ningún desborde.

Como parámetros adicionales se pueden configurar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
CU	BOOL	Entrada Trigger
R	BOOL	Reset
PV	INT	Valor de ajuste previo
Q	BOOL	Salida
CV	INT	Valor de cómputo (valor real)

CTUD: Contador hacia arriba/ abajo

13

Vista general

Introducción Este capítulo describe el bloque CTUD.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	54
Representación	55

Descripción breve

Descripción de funcionamiento

El módulo de función se utiliza para el conteo progresivo y regresivo de valores INT. Los módulos de función para el conteo de valores DINT, UDINT y UINT figuran en la biblioteca Extended.

En caso de señal "1" en la entrada R, se asigna el valor "0" a la salida CV. En caso de señal "1" en la entrada LD, se asigna el valor de la entrada PV a la salida CV. En cada transición de "0" a "1" en la entrada CU, el valor de CV aumenta en 1. En cada transición de "0" a "1" en la entrada CD, el valor de CV se reduce en 1.

En caso de señal "1" simultánea en las entradas CU y CD, predomina la entrada CU (contador progresivo).

En caso de señal "1" simultánea en las entradas R y LD, predomina la entrada R.

Si $CV \geq PV$, la salida QU pasa a "1".

Si $CV \leq 0$, la salida QD pasa a "1".

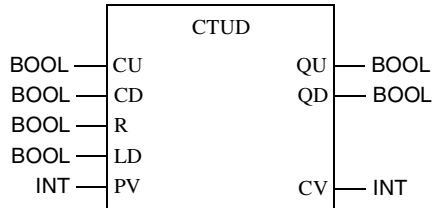
Nota: El contador sólo funciona hasta alcanzar el valor mínimo (contador regresivo) o el valor máximo (contador progresivo) de la salida CV. No se produce ningún desborde.

Como parámetros adicionales se pueden configurar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
CU	BOOL	Entrada de disparador de contador ascendente
CD	BOOL	Entrada de disparador de contador descendente
R	BOOL	Reset
LD	BOOL	Carga de datos
PV	INT	Valor de ajuste previo
QU	BOOL	Indicador ascendente
QD	BOOL	Indicador descendente
CV	INT	Valor de cómputo (valor real)

DINT_EXPT_REAL: Exponencialización

14

Vista general

Introducción Este capítulo describe el bloque DINT_EXPT_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	58
Representación	58
Error de tiempo de ejecución	58

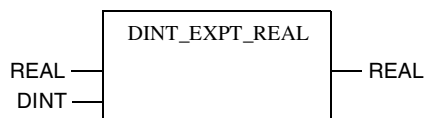
Descripción breve

Descripción de la función La función se utiliza para el cálculo de la exponencialización. El valor en la entrada IN1 (base) se potencia con el valor en la entrada IN2 como exponente y se emite la potencia en la salida.
Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = IN1 \exp IN2$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	REAL	Base
IN2	DINT	Exponente
OUT	REAL	Potencia

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error cuando

- $(IN1 = 0) \& (IN2 < 0)$ es,
 - en IN1 se trata de un número de coma flotante inadmisibile o
 - se trata de un desbordamiento del margen de valores en la entrada.
-

DINT_TO_***: Conversión de tipo

15

Vista general

Introducción Este capítulo describe el bloque DINT_TO_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	60
Representación	60
Error de tiempo de ejecución	61

Descripción breve

Descripción de la función

La función convierte un valor de entrada del tipo de datos DINT en un valor de salida del tipo de datos INT, UDINT, UINT, REAL, TIME, BOOL, BYTE o WORD.

Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Nota: El EFB convierte estrictamente según las reglas CEI. Como este EFB ha sido realizado como función genérica, también se han producido algunas transformaciones ilógicas, p.e. DINT_TO_BOOL.

Durante la conversión del tipo de datos DINT al tipo de datos BOOL, BYTE o WORD se transmiten los bits de menor valor del valor de entrada a la salida.

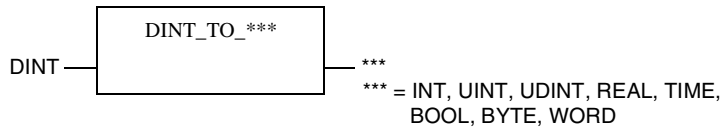
Los valores de entrada negativos no se pueden convertir en los tipos de datos UDINT o UINT.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	DINT	Valor de entrada
OUT	INT, UDINT, UINT, REAL, TIME, BOOL, BYTE, WORD	Valor de salida

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error cuando

- se sobrepasa el margen de valores de la salida o
 - se debe convertir un valor de entrada negativo en un valor de salida UDINT o UINT.
-

Vista general

Introducción Este capítulo describe el bloque DIV_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	64
Representación	64
Error de tiempo de ejecución	65

Descripción breve

Descripción de la función

La función divide el valor en la entrada IN1 entre el valor en la entrada IN2 y emite el resultado en la salida.

Los tipos de datos de los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

En caso de división de los tipos de datos del grupo ANY_INT se corta del resultado un punto detrás de la coma, eventualmente existente en sentido cero, por ejemplo.

$$7 \div 3 = 2$$

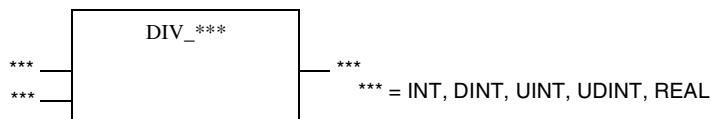
$$(-7) \div 3 = -2$$

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$$\text{OUT} = ((\text{IN1}) \div (\text{IN2}))$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL	Dividendo
IN2	INT, DINT, UINT, UDINT, REAL	Divisor
OUT	INT, DINT, UINT, UDINT, REAL	Cociente

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error si

- $IN2 = 0^0$
 - y se dispone de un número de coma flotante inadmisibles para un parámetro de entrada del tipo de datos REAL.
-

Vista general

Introducción Este capítulo describe el bloque EQ_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	68
Representación	68
Error de tiempo de ejecución	69

Descripción breve

Descripción de la función

La función comprueba las entradas respecto a igualdad, es decir, que la entrada se convierte en "1", cuando en todas las entradas hay igualdad; de otra manera la salida permanece en "0".

Los tipos de datos de todos los valores de entrada deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

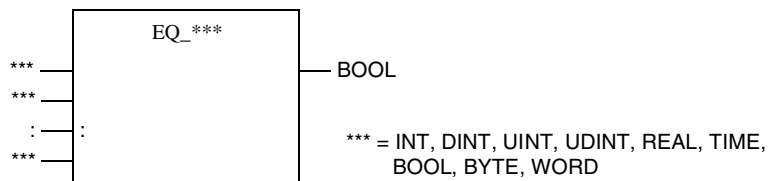
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = 1, wenn (IN1 = IN2) & (IN2 = IN3) & .. & (IN_(n-1) = IN_n)

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	1. Entrada
IN2	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	2. Entrada
INn	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Entrada n
OUT	BOOL	Salida

Error de tiempo de ejecución

Aviso de error

Si se genera para un parámetro de entrada del tipo de datos REAL un número de coma flotante inadmisibles, se emite un aviso de error.

EXP_REAL: Función exponencial

18

Vista general

Introducción Este capítulo describe el bloque EXP_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	72
Representación	72
Error de tiempo de ejecución	72

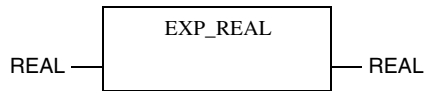
Descripción breve

Descripción de la función La función se utiliza para el cálculo de la función de exponencial. El valor en la entrada se utiliza como exponente para la base y se emite el potencial en la salida. Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = e^{\exp IN}$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Exponente a la base
OUT	REAL	Potencia

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error si

- en la entrada hay un número de coma flotante inadmisibles o
 - se trata de un desbordamiento del margen de valores en la entrada.
-

F_TRIG: Detección de flancos descendientes

19

Vista general

Introducción Este capítulo describe el bloque F_TRIG.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	74
Representación	74

Descripción breve

Descripción de la función El bloque de función se utiliza para de detección de flancos descendientes 1 -> 0.

Nota: Este bloque de función no está disponible en el lenguaje de programación LD (Ladder Diagram), ya que allí existe el "contacto – flanco negativo", el cual tiene la misma funcionalidad.

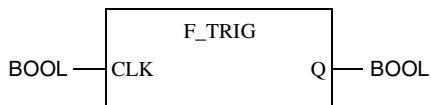
La salida Q se convierte en "1", cuando se realiza un traspaso de "1" a "0" en la entrada CLK. La salida permanece desde una ejecución del bloque de función hasta la siguiente ejecución en "1"; a continuación la salida vuelve a "0".

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
CLK	BOOL	Entrada de ciclo
Q	BOOL	Salida

GE_***: Mayor igual a

20

Vista general

Introducción Este capítulo describe el bloque GE_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	76
Representación	76
Error de tiempo de ejecución	76

Descripción breve

Descripción de la función

La función comprueba los valores de entradas sucesivas respecto a una secuencia descendente o a la igualdad.

Los tipos de datos de todos los valores de entrada deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

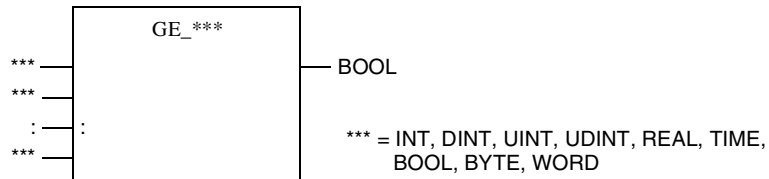
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = 1, cuando $(IN1 \geq IN2) \& (IN2 \geq IN3) \& \dots \& (IN_{(n-1)} \geq IN_n)$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	1. Entrada
IN2	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	2. Entrada
INn	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Entrada n
OUT	BOOL	Salida

Error de tiempo de ejecución

Aviso de error

Si se genera para un parámetro de entrada del tipo de datos REAL un número de coma flotante inadmisibles, se emite un aviso de error.

Vista general

Introducción Este capítulo describe el bloque GT_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	78
Representación	78
Error de tiempo de ejecución	78

Descripción breve

Descripción de la función

La función comprueba los valores de entradas sucesivas respecto a una secuencia descendente.

Los tipos de datos de todos los valores de entrada deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

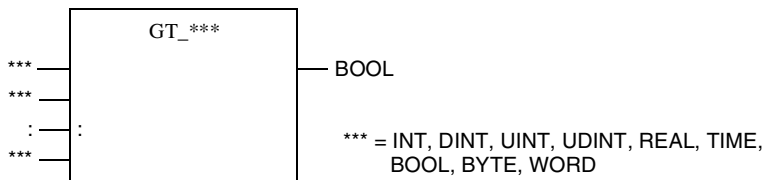
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = 1$, cuando $(IN1 > IN2) \& (IN2 > IN3) \& \dots (IN_{(n-1)} > IN_n)$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	1. Entrada
IN2	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	2. Entrada
INn	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Entrada n
OUT	BOOL	Salida

Error de tiempo de ejecución

Aviso de error

Si se genera para un parámetro de entrada del tipo de datos REAL un número de coma flotante inadmisibles, se emite un aviso de error.

INT_EXPT_REAL: Exponencialización

22

Vista general

Introducción Este capítulo describe el bloque INT_EXPT_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	80
Representación	80
Error de tiempo de ejecución	80

Descripción breve

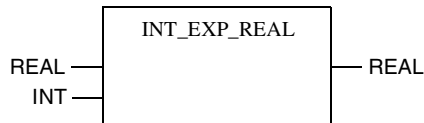
Descripción de la función La función se utiliza para el cálculo de la exponencialización. El valor en la entrada IN1 (base) se potencia con el valor en la entrada IN2 como exponente y se emite la potencia en la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = IN1 \exp(IN2)$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	REAL	Base
IN2	INT	Exponente
OUT	REAL	Potencia

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error, cuando

- $(IN1 = 0) \& (IN2 < 0)$ es,
 - en IN1 se trata de un número de coma flotante inadmisibile o
 - se trata de un desbordamiento del margen de valores en la entrada.
-

INT_TO_***: Conversión de tipo

23

Vista general

Introducción Este capítulo describe el bloque INT_TO_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	82
Representación	82
Error de tiempo de ejecución	83

Descripción breve

Descripción de la función

La función convierte un valor de entrada del tipo de datos INT en un valor de salida del tipo de datos BOOL, BYTE, WORD, DINT, UDINT, UINT, REAL o TIME.

Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Nota: El EFB convierte estrictamente según las reglas CEI. Como este EFB ha sido realizado como función genérica, también se han producido algunas transformaciones ilógicas, p.e. INT_TO_BOOL.

Los valores de entrada negativos no se pueden convertir en los tipos de datos UDINT, UINT o TIME.

En la conversión de un valor de entrada del tipo de datos INT al tipo de datos WORD se transmite la muestra de bits de la entrada inalterada a la salida.

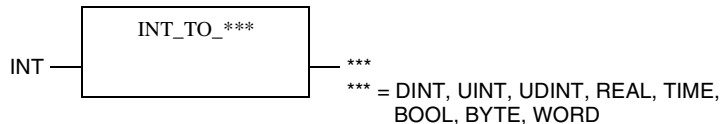
Durante la conversión de un valor de entrada del tipo de datos INT a los tipos de datos BOOL o BYTE se transmiten los bits de menor valor de la entrada a la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	INT	Valor de entrada
OUT	BOOL, BYTE, WORD, DINT, UDINT, UINT, REAL, TIME	Valor de salida

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error cuando

- se sobrepasa el margen de valores de la salida o
 - se debe convertir un valor de entrada negativo en UDINT, UINT o TIME.
-

LE_***: Menor igual a

24

Vista general

Introducción Este capítulo describe el bloque LE_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	86
Representación	86
Error de tiempo de ejecución	86

Descripción breve

Descripción de la función

La función comprueba los valores de entradas sucesivas respecto a una secuencia ascendente o a la igualdad.

Los tipos de datos de todos los valores de entrada deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

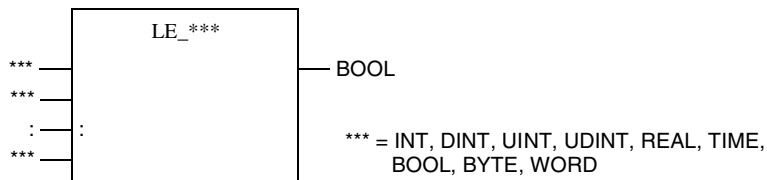
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = 1$, cuando $(IN1 \leq IN2) \& (IN2 \leq IN3) \& \dots \& (IN_{(n-1)} \leq IN_n)$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	1. Entrada
IN2	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	2. Entrada
INn	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Entrada n
OUT	BOOL	Salida

Error de tiempo de ejecución

Aviso de error

Si se genera para un parámetro de entrada del tipo de datos REAL un número de coma flotante inadmisibles, se emite un aviso de error.

LIMIT_***: Limitación

25

Vista general

Introducción Este capítulo describe el bloque LIMIT_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	88
Representación	89
Error de tiempo de ejecución	89

Descripción breve

Descripción de la función

La función traspasa el valor de entrada inalterado (IN) a la salida, cuando el valor de entrada no queda por debajo del valor mínimo (MN) y no sobrepasa el valor máximo (MX). Si el valor de entrada (IN) queda por debajo del valor mínimo (MN), se traspasa el valor mínimo a la salida. Si el valor de entrada (IN) sobrepasa el valor máximo (MX), se traspasa el valor máximo a la salida.

Se pueden procesar los tipos de datos del grupo ANY_ELEM.

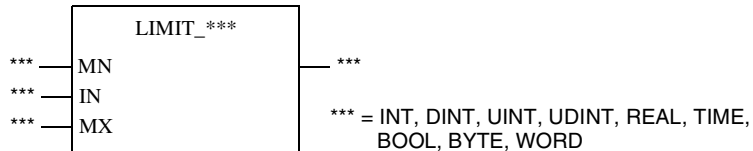
Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = IN, cuando $(IN \geq MN) \& (IN \leq MX)$

OUT = MN, cuando $(IN < MN)$

OUT = MX, cuando $(IN > MX)$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
MN	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	umbral inferior
IN	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Entrada
MX	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	umbral superior
OUT	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Salida

Error de tiempo de ejecución

Aviso de error

Si la entrada dispone de un número de coma flotante inadmisibles, se emite un aviso de error.

LN_REAL: Logaritmo natural

26

Vista general

Introducción Este capítulo describe el bloque LN_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	92
Representación	92
Error de tiempo de ejecución	92

Descripción breve

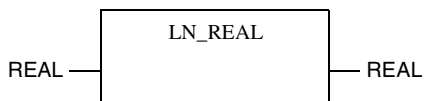
Descripción de la función La función calcula el logaritmo natural del valor de entrada y emite el resultado en la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = \ln(IN)$

Requisito:

$IN > 0$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada
OUT	REAL	Valor de salida

Error de tiempo de ejecución

Aviso de error

Si durante la ejecución de la función se produce una vulneración del margen de valores en la entrada o si se trata de un número de coma flotante inadmisibles, se emite un aviso de error.

LOG_REAL: Logaritmo respecto a la base 10

27

Vista general

Introducción Este capítulo describe el bloque LOG_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	94
Representación	94
Error de tiempo de ejecución	94

Descripción breve

Descripción de la función La función calcula el logaritmo respecto a la base 10 del valor de entrada y emite el resultado en la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = \log (IN)$

Requisito:

$IN > 0$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada
OUT	REAL	Valor de salida

Error de tiempo de ejecución

Aviso de error

Si durante la ejecución de la función se produce una vulneración del margen de valores en la entrada o si se trata de un número de coma flotante inadmisibles, se emite un aviso de error.

Vista general

Introducción Este capítulo describe el bloque LT_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	96
Representación	96
Error de tiempo de ejecución	96

Descripción breve

Descripción de la función

La función comprueba los valores de entradas sucesivas respecto a una secuencia ascendente.

Los tipos de datos de todos los valores de entrada deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

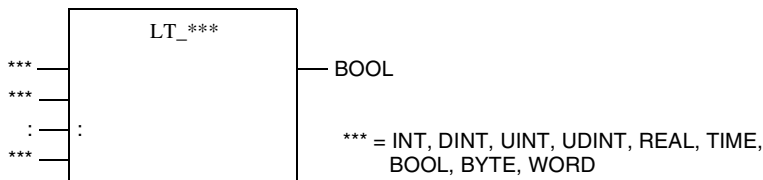
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = 1$, cuando $(IN1 < IN2) \& (IN2 < IN3) \& \dots \& (IN_{(N-1)} < IN_n)$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	1. Valor de entrada
IN2	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	2. Valor de entrada
INn	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Valor de entrada n
OUT	BOOL	Valor de salida

Error de tiempo de ejecución

Aviso de error

Si se genera para un parámetro de entrada del tipo de datos REAL un número de coma flotante inadmisibles, se emite un aviso de error.

MAX_***: Selección de valor máximo

29

Vista general

Introducción Este capítulo describe el bloque MAX_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	98
Representación	98
Error de tiempo de ejecución	99

Descripción breve

Descripción de la función

La función emite el mayor valor de entrada en la salida.

Se pueden procesar los tipos de datos del grupo ANY_ELEM.

Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

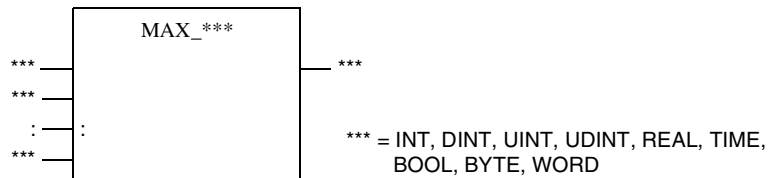
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$$OUT = \text{MAX} \{IN1, IN2, \dots, INn\}$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	1. Valor de entrada
IN2	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	2. Valor de entrada
INn	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Valor de entrada n
OUT	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Valor máximo

Error de tiempo de ejecución

Aviso de error Si se genera para un parámetro de entrada del tipo de datos REAL un número de coma flotante inadmisibles, se emite un aviso de error.

MIN_***: Selección de valor mínimo

30

Vista general

Introducción Este capítulo describe el bloque MIN_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	102
Representación	102
Error de tiempo de ejecución	103

Descripción breve

Descripción de la función

La función emite el menor valor de entrada en la salida.

Se pueden procesar los tipos de datos del grupo ANY_ELEM.

Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

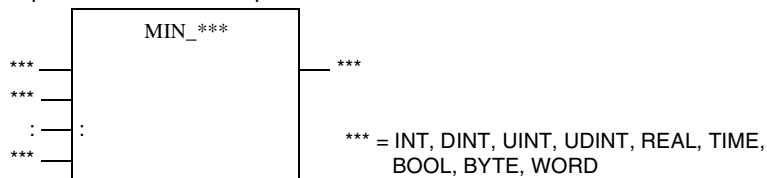
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$$\text{OUT} = \text{MIN} \{ \text{IN1}, \text{IN2}, \dots, \text{INn} \}$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	1. Valor de entrada
IN2	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	2. Valor de entrada
INn	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Valor de entrada n
OUT	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	Valor mínimo

Error de tiempo de ejecución

Aviso de error Si se genera para un parámetro de entrada del tipo de datos REAL un número de coma flotante inadmisibles, se emite un aviso de error.

Vista general

Introducción Este capítulo describe el bloque MOD_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	106
Representación	106

Descripción breve

Descripción de la función La función divide el valor en la entrada IN1 entre el valor en la entrada IN2 y emite el resto de división (módulo) en la salida.

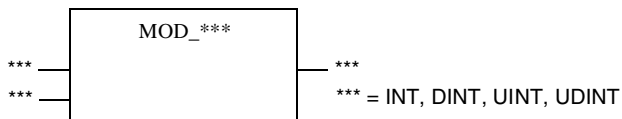
Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = IN1 \bmod IN2$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT	Dividendo
IN2	INT, DINT, UINT, UDINT	Divisor
OUT	INT, DINT, UINT, UDINT	Módulo

MOVE: Asignación

32

Vista general

Introducción Este capítulo describe el bloque MOVE.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	108
Representación	108

Descripción breve

Descripción de la función

La función asigna el valor de entrada a la salida.

La función es genérica, es decir, que el tipo de datos a procesar queda determinado por la variable que se ha aplicado primero a la función.

Si se debe asignar una dirección directa a una variable o viceversa, siempre se debe aplicar primero la variable a la función. No se admite una dirección directa en la entrada y salida de la función, ya que aquí resulta imposible una definición unívoca.

Los tipos de datos del valor de entrada y salida deben ser iguales.

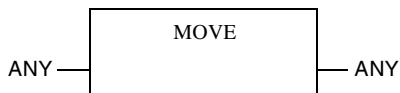
Nota: Esta función no se puede utilizar en el lenguaje de programación LD (Ladder Diagram) con el tipo de datos BOOL, ya que se puede realizar la misma funcionalidad con contactos y bobinas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = IN

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	ANY	Valor de entrada
OUT	ANY	Valor de salida

MUL_***: Multiplicación

33

Vista general

Introducción Este capítulo describe el bloque MUL_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	110
Representación	110
Error de tiempo de ejecución	110

Descripción breve

Descripción de la función

La función multiplica los valores de entrada y emite el resultado en la salida.

Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

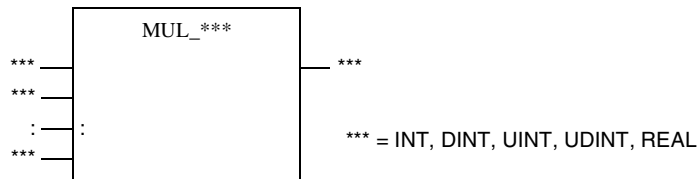
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$$OUT = IN1 \times IN2 \times \dots \times IN_n$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL	Multiplicando (factor)
IN2	INT, DINT, UINT, UDINT, REAL	Multiplicador (factor)
INn	INT, DINT, UINT, UDINT, REAL	Multiplicador (factor)
OUT	INT, DINT, UINT, UDINT, REAL	Producto

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error si

- se trata para un parámetro de entrada del tipo de datos REAL de un número de coma flotante inadmisibles o
- se trata de un desbordamiento del margen de valores en la entrada.

MUX_***: Multiplexor

34

Vista general

Introducción Este capítulo describe el bloque MUX_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	112
Representación	113

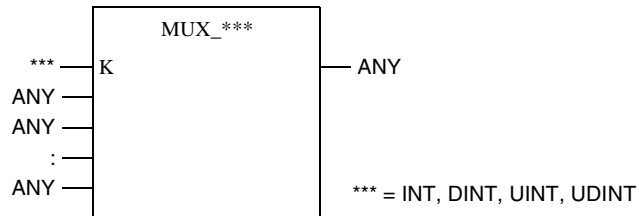
Descripción breve

Descripción de la función	<p>La función traspasa en dependencia del valor en la entrada K la correspondiente entrada a la salida.</p> <p>Se puede aumentar el número de entradas.</p> <p>Como parámetros adicionales se pueden proyectar EN y ENO.</p>
Ejemplo	<p>K = 0: Se traspasa la entrada IN0 a la salida</p> <p>K = 1: Se traspasa la entrada IN1 a la salida</p> <p>K = 5: Se traspasa la entrada IN5 a la salida</p> <p>K = n: Se traspasa la entrada INn a la salida</p>
Tipos de datos	<p>Se pueden procesar los tipos de datos del grupo ANY.</p> <p>Los tipos de datos en las entradas IN0 hasta INn y en la salida deben ser iguales.</p> <p>Para el procesamiento de los diferentes tipos de datos (ANY_INT) en la entrada K se dispone en cada caso de una función especial.</p>

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
K	INT, DINT, UINT, UDINT	Entrada de selección
IN0	ANY	0. Entrada
IN1	ANY	1. Entrada
IN2	ANY	2. Entrada
INn	ANY	Entrada n
OUT	ANY	Salida

Vista general

Introducción Este capítulo describe el bloque NE_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	116
Representación	116
Error de tiempo de ejecución	116

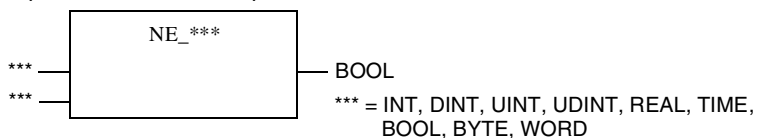
Descripción breve

Descripción de la función La función comprueba los valores de entrada respecto a la desigualdad. Los tipos de datos de los valores de entrada deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial. Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = 1, cuando IN1 < > IN2

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	1. Entrada
IN2	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE, WORD	2. Entrada
OUT	BOOL	Salida

Error de tiempo de ejecución

Aviso de error Si se genera para un parámetro de entrada del tipo de datos REAL un número de coma flotante inadmisibles, se emite un aviso de error.

NOT_***: Negación

36

Vista general

Introducción Este capítulo describe el bloque NOT_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	118
Representación	118

Descripción breve

Descripción de la función La función va negando por bits la secuencia de bits de entrada y emite el resultado en la salida.

Se pueden procesar los tipos de datos del grupo ANY_BIT.

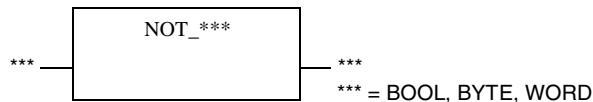
Nota: Esta función no está disponible con variables booleanos en el lenguaje de programación LD (Ladder Diagram), ya que allí se puede realizar la misma funcionalidad con contactos (dispositivo de cierre).

Los tipos de datos del valor de entrada y salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial. Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = NOT IN

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	BOOL, BYTE, WORD	Secuencia de bits de entrada
OUT	BOOL, BYTE, WORD	Secuencia de bits negada

OR_***: Función O

37

Vista general

Introducción Este capítulo describe el bloque OR_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	120
Representación	120

Descripción breve

Descripción de la función La Función vincula (según la lógica O) las secuencias de bits en las entradas y emite el resultado en la salida. La vinculación se realiza por bits.

Se pueden procesar los tipos de datos del grupo ANY_BIT.

Nota: Esta función no está disponible con variables booleanas en el lenguaje de programación LD (Ladder Diagram), ya que allí se puede realizar la misma funcionalidad con contactos.

Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

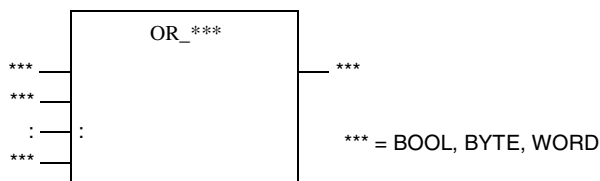
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = IN1 \text{ OR } IN2 \text{ OR } .. \text{ OR } INn$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	BOOL, BYTE, WORD	Secuencia de bits de entrada
IN2	BOOL, BYTE, WORD	Secuencia de bits de entrada
INn	BOOL, BYTE, WORD	Secuencia de bits de entrada
OUT	BOOL, BYTE, WORD	Secuencia de bits de salida

R_TRIG: Detección de flancos ascendentes

38

Vista general

Introducción Este capítulo describe el bloque R_TRIG.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	122
Representación	122

Descripción breve

Descripción de la función El bloque de función se utiliza para de detección de flancos ascendentes 0 -> 1.

Nota: Este bloque de función no está disponible en el lenguaje de programación LD (Ladder Diagram), ya que allí existe el "contacto – flanco positivo", el cual tiene la misma funcionalidad.

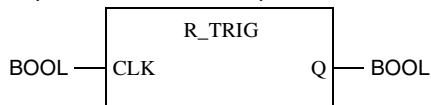
La salida Q se convierte en "1", cuando se realiza un traspaso de "0" a "1" en la entrada CLK. La salida permanece desde una ejecución del bloque de función hasta la siguiente ejecución en "1" (un ciclo); a continuación la salida vuelve a "0".

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
CLK	BOOL	Entrada de ciclo
Q	BOOL	Salida

REAL_EXPT_REAL: Exponencialización

39

Vista general

Introducción Este capítulo describe el bloque REAL_EXPT_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	124
Representación	124
Error de tiempo de ejecución	124

Descripción breve

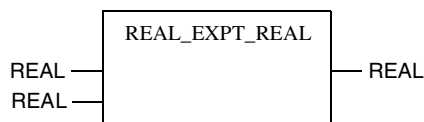
Descripción de la función La función se utiliza para el cálculo de la exponencialización. El valor en la entrada IN1 (base) se potencia con el valor en la entrada IN2 como exponente y se emite la potencia en la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = IN1 \exp IN2$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	REAL	Base
IN2	REAL	Exponente
OUT	REAL	Potencia

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error cuando

- se trata de un número de coma flotante inadmisibles en una de las entradas o
 - se trata de un desbordamiento del margen de valores en la entrada.
-

REAL_TO_***: Conversión de tipo

40

Vista general

Introducción Este capítulo describe el bloque REAL_TO_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	126
Representación	127
Error de tiempo de ejecución	127

Descripción breve

Descripción de la función La función convierte un valor de entrada del tipo de datos REAL en un tipo de datos del grupo ANY_BIT oder ANY_INT o en el tipo de datos TIME.

Nota: El EFB convierte estrictamente según las reglas CEI. Como este EFB ha sido realizado como función genérica, también se han producido algunas transformaciones ilógicas, p.e. REAL_TO_BOOL.

En la conversión según ANY_BIT se transmiten los bits de valor menor del valor de entrada a la salida.

Nota: Para una conversión de REAL a WORD se encuentran disponibles también los bloques R_INT_WORD y R_UINT_WORD de la biblioteca de bloques ANA_IO y el bloque REAL_AS_WORD de la biblioteca de bloques EXTENDED.

Para la conversión según ANY_INT y TIME se redondea según las convenciones de CEI 559.

Como parámetros adicionales se pueden proyectar EN y ENO.

Ejemplo El siguiente ejemplo muestra, cómo se redondea según la CEI 559.

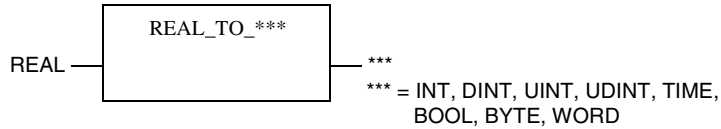
1,4 -> 1
1,5 -> 2
2,5 -> 2
3,5 -> 4
4,5 -> 4
4,6 -> 5

Tipo de datos Los valores de entrada negativos no se pueden convertir en los tipos de datos UDINT, UINT o TIME.
Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada
OUT	INT, DINT, UINT, UDINT, TIME, BOOL, BYTE, WORD	Valor de salida

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error cuando

- se tren la entrada hay un número de coma flotante inadmisibile o
- se sobrepasa el margen de valores de la salida o
- se debe convertir un valor de entrada negativo en UDINT, UINT o TIME.

REAL_TRUNC_***: Conversión de tipo

41

Vista general

Introducción Este capítulo describe el bloque REAL_TRUNC_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	130
Representación	130
Error de tiempo de ejecución	131

Descripción breve

Descripción de la función La Función convierte (con corte en el sentido cero) un valor de entrada del tipo de datos REAL a un tipo de datos ANY_INT.

Como parámetros adicionales se pueden proyectar EN y ENO.

Ejemplo

El siguiente ejemplo muestra, cómo se realiza la conversión.

1,6 -> 1

-1,6 -> -1

1,4 -> 1

-1,4 -> -1

Tipo de datos

Los valores de entrada negativos no se pueden convertir en los tipos de datos UDINT o UINT.

Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada
OUT	INT, DINT, UINT, UDINT	Valor de salida

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error cuando

- se debe convertir un valor de entrada negativo en UDINT oder UINT,
 - se trata de un número de coma flotante inadmisibile en la entrada o
 - se sobrepasa el margen de valores de la salida.
-

ROL_***: Rotar a la izquierda

42

Vista general

Introducción Este capítulo describe el bloque ROL_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	134
Representación	134

Descripción breve

Descripción de la función

La función rota la secuencia de bits en la entrada IN alrededor de n bits (valor en la entrada N) hacia la izquierda en el circuito.

Se pueden procesar los tipos de datos del grupo ANY_BIT.

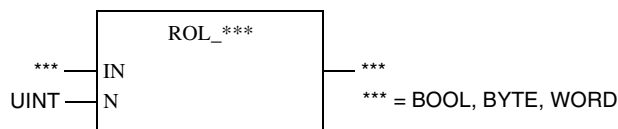
Los tipos de datos de los valores de entrada IN y el del valor de salida OUT deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	BOOL, BYTE, WORD	Secuencia de bits a rotar
N	UINT	Número de los puntos, alrededor de los cuales se debe rotar
OUT	BOOL, BYTE, WORD	Secuencia de bits rotada

ROR_***: Rotar a la derecha

43

Vista general

Introducción Este capítulo describe el bloque ROR_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	136
Representación	136

Descripción breve

Descripción de la función La función rota la secuencia de bits en la entrada IN alrededor de n bits (valor en la entrada N) hacia la derecha en el circuito.

Se pueden procesar los tipos de datos del grupo ANY_BIT.

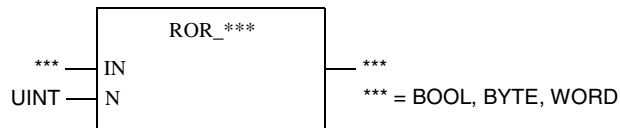
Los tipos de datos de los valores de entrada IN y el del valor de salida OUT deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del módulo:



Descripción de parámetros

Descripción de los parámetros del módulo:

Parámetro	Tipo de datos	Significado
IN	BOOL, BYTE, WORD	Secuencia de bits que se va a rotar
N	UINT	Número de los puntos, alrededor de los cuales se debe rotar
OUT	BOOL, BYTE, WORD	Secuencia de bits rotada

RS: Bloque de función biestable, restablecer dominante

44

Vista general

Introducción Este capítulo describe el bloque RS.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	138
Representación	138

Descripción breve

Descripción de funcionamiento

El módulo de función se utiliza como memoria RS con la característica "Restablecer dominante".

Cuando la entrada S se convierte en "1", la salida Q1 se convierte en "1". Este estado se mantiene aún cuando la entrada S vuelve a ser "0". La salida Q1 sólo se convierte en "0" cuando la entrada R1 se convierte en "1". Si las entradas S y R1 son a la vez "1", la entrada dominante R1 establece la entrada Q1 en "0".

El estado inicial de Q1 durante la primera llamada del módulo de función es "0".

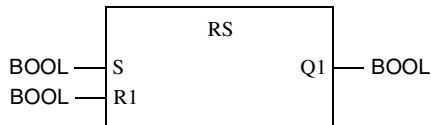
Como parámetros adicionales se pueden configurar EN y ENO.

Nota: Este módulo de función trabaja con una unlocated variable interna, por lo que su comportamiento es de memorización. Esto implica que si la salida "Q1" se conecta con una salida de hardware, cuando se conecte o desconecte la salida del PLC la salida permanecerá en el valor "1".

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
S	BOOL	Ubicar
R1	BOOL	Restablecer (dominante)
Q1	BOOL	Salida

SEL: Selección binaria

45

Vista general

Introducción Este capítulo describe el bloque SEL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	140
Representación	140

Descripción breve

Descripción de la función

La función se utiliza para la selección binaria entre dos valores de entrada.

En función del estado de la entrada G se traspasa, o bien, a la entrada IN0 o la entrada IN1 a la salida OUT.

$G = 0 \rightarrow OUT = IN0$

$G = 1 \rightarrow OUT = IN1$

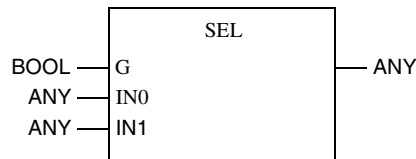
Los tipos de datos de los valores de entrada IN0 y IN1 y del valor de salida OUT deben ser iguales.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
G	BOOL	Entrada de selección
IN0	ANY	Entrada 0
IN1	ANY	Entrada 1
OUT	ANY	Salida

SHL_***: Desplazar a la izquierda

46

Vista general

Introducción Este capítulo describe el bloque SHL_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	142
Representación	142

Descripción breve

Descripción de la función

La función desplaza la secuencia de bits en la entrada IN por n bits (valor en la entrada N) hacia la izquierda.

Desde la derecha se rellena con ceros.

Se pueden procesar los tipos de datos del grupo ANY_BIT.

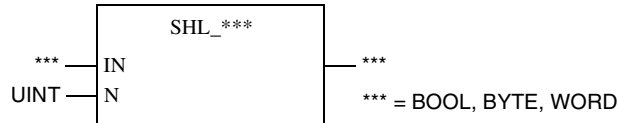
Los tipos de datos de los valores de entrada IN y el del valor de salida OUT deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	BOOL, BYTE, WORD	Secuencia de bits a desplazar
N	UINT	Número de los puntos, alrededor de los cuales se debe desplazar
OUT	BOOL, BYTE, WORD	Secuencia de bits desplazada

SHR_***: Desplazar a la derecha

47

Vista general

Introducción Este capítulo describe el bloque SHR_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	144
Representación	144

Descripción breve

Descripción de la función

La función desplaza la secuencia de bits en la entrada IN por n bits (valor en la entrada N) hacia la derecha.

Desde la izquierda se rellena con ceros.

Se pueden procesar los tipos de datos del grupo ANY_BIT.

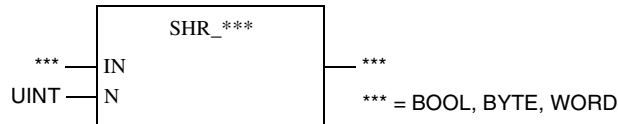
Los tipos de datos de los valores de entrada IN y el del valor de salida OUT deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	BOOL, BYTE, WORD	Secuencia de bits a desplazar
N	BOOL	Número de los puntos, alrededor de los cuales se debe desplazar
OUT	BOOL, BYTE, WORD	Secuencia de bits desplazada

Vista general

Introducción Este capítulo describe el bloque SIN_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	146
Representación	146
Error de tiempo de ejecución	146

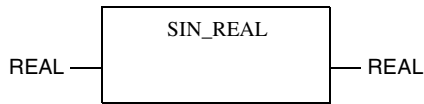
Descripción breve

Descripción de la función La función calcula el seno del valor de entrada y emite el resultado en la salida. La introducción del valor de entrada se debe realizar en la medida de arco.
Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = sin IN

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada en la medida de arco
OUT	REAL	Valor de salida

Error de tiempo de ejecución

Aviso de error Si la entrada dispone de un número de coma flotante inadmisibles, se emite un aviso de error.

SQRT_REAL: Raiz cuadrada

49

Vista general

Introducción Este capítulo describe el bloque SQRT_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	148
Representación	148
Error de tiempo de ejecución	148

Descripción breve

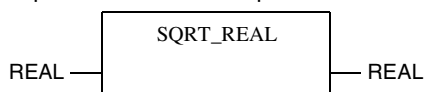
Descripción de la función La función calcula el la raiz cuadrada del valor de entrada y emite el resultado en la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$$\text{OUT} = \sqrt{\text{IN}}$$

Requisito:

$$\text{IN} \geq 0$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada
OUT	REAL	Valor de salida

Error de tiempo de ejecución

Aviso de error

Si durante la ejecución de la función se produce una vulneración del margen de valores en la entrada o si se trata de un número de coma flotante inadmisibles, se emite un aviso de error.

SR: Bloque de función biestable, ubicar dominante

50

Vista general

Introducción Este capítulo describe el bloque SR.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	150
Representación	150

Descripción breve

Descripción de funcionamiento

El módulo de función se utiliza como memoria SR con la característica "Ubicar dominante".

La salida Q1 se convierte en "1" cuando la entrada S1 se convierte en "1". Este estado se mantiene aún cuando la entrada S1 vuelve a ser "0". La salida Q1 sólo vuelve a convertirse en "0" cuando la entrada R se convierte en "1". Si las entradas S1 y R son a la vez "1", la entrada dominante S1 establece la salida Q1 en "1".

El estado inicial de Q1 durante la primera llamada del módulo de función es "0".

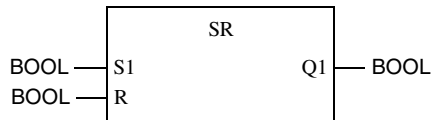
Como parámetros adicionales se pueden configurar EN y ENO.

Nota: Este módulo de función trabaja con una unlocated variable interna, por lo que su comportamiento es de memorización. Esto implica que si la salida "Q1" se conecta con una salida de hardware, cuando se conecte o desconecte la salida del PLC la salida permanecerá en el valor "1".

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
S1	BOOL	Ubicar (dominante)
R	BOOL	Restablecer
Q	BOOL	Salida

SUB_***: Sustracción

51

Vista general

Introducción Este capítulo describe el bloque SUB_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	152
Representación	152
Error de tiempo de ejecución	152

Descripción breve

Descripción de la función

La función resta el valor en la entrada IN2 del valor en la entrada IN1 y emite el resultado en la salida.

Se pueden procesar tipos de datos del grupo ANY_NUM y del tipo de datos TIME.

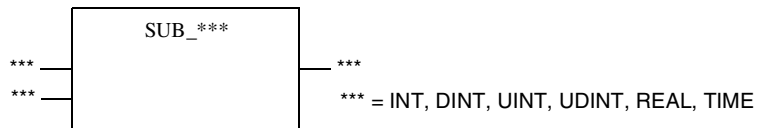
Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$$\text{OUT} = \text{IN1} - \text{IN2}$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	INT, DINT, UINT, UDINT, REAL, TIME	Minuendo
IN2	INT, DINT, UINT, UDINT, REAL, TIME	Sustraendo
OUT	INT, DINT, UINT, UDINT, REAL, TIME	Diferencia

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error si

- se trata para un parámetro de entrada del tipo de datos REAL de un número de coma flotante inadmisibles o
- se trata de un desbordamiento del margen de valores en la entrada.

Vista general

Introducción Este capítulo describe el bloque TAN_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	154
Representación	154
Error de tiempo de ejecución	154

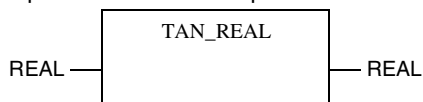
Descripción breve

Descripción de la función La función calcula la tangente del valor de entrada y emite el resultado en la salida. La introducción del valor de entrada se debe realizar en la medida de arco. Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

OUT = tan IN

Requisito:

$$IN \neq \frac{(2n + 1) \times \pi}{2}$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	REAL	Valor de entrada en la medida de arco
OUT	REAL	Valor de salida

Error de tiempo de ejecución

Aviso de error

Si durante la ejecución de la función se produce una vulneración del margen de valores en la entrada o si se trata de un número de coma flotante inadmisibles, se emite un aviso de error.

TIME_DIV_***: División de valores temporales

53

Vista general

Introducción Este capítulo describe el bloque TIME_DIV_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	156
Representación	156
Error de tiempo de ejecución	156

Descripción breve

Descripción de la función La función divide el valor en la entrada IN1 entre el valor en la entrada IN2 y emite el resultado en la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Ejemplo Un punto detrás de la coma, eventualmente existente, se corta para el resultado en sentido cero.

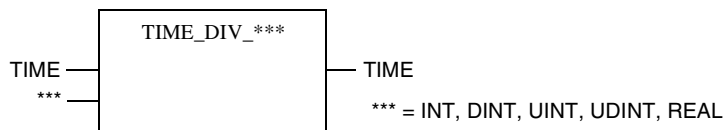
$$67 / 3 = 22$$

$$-67 / 3 = -22$$

Tipo de datos Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Representación

Símbolo Representación del bloque



Fórmula $OUT = IN1 / IN2$

Descripción de parámetros Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	TIME	Dividendo
IN2	INT, DINT, UINT, UDINT, REAL	Divisor
OUT	TIME	Cociente

Error de tiempo de ejecución

Aviso de error Si durante la ejecución de la función se produce un desbordamiento del margen de valores en la salida, se emite un aviso de error.

TIME_MUL_***: Multiplicación de valores temporales

54

Vista general

Introducción Este capítulo describe el bloque TIME_MUL_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	158
Representación	158
Error de tiempo de ejecución	158

Descripción breve

Descripción de la función La función multiplica el valor en la entrada IN1 con el valor en la entrada IN2 y emite el resultado en la salida.

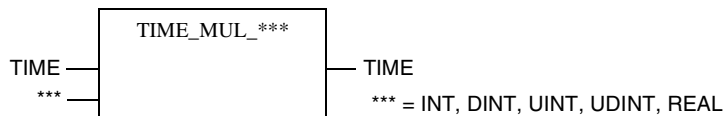
Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = IN1 \times IN2$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	TIME	Multiplicando (factor)
IN2	INT, DINT, UINT, UDINT, REAL	Multiplicador (factor)
OUT	TIME	Producto

Error de tiempo de ejecución

Aviso de error Si durante la ejecución de la función se produce un desbordamiento del margen de valores en la salida, se emite un aviso de error.

TIME_TO_***: Conversión de tipo

55

Vista general

Introducción Este capítulo describe el bloque TIME_TO_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	160
Representación	160
Error de tiempo de ejecución	161

Descripción breve

Descripción de la función

La función convierte un valor de entrada del tipo de datos TIME en un tipo de datos del grupo ANY_BIT o ANY_NUM.

Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Nota: El EFB convierte estrictamente según las reglas CEI. Como este EFB ha sido realizado como función genérica, también se han producido algunas transformaciones ilógicas, p.e. TIME_TO_BOOL.

Durante la conversión de un valor de entrada del tipo de datos TIME en un valor de salida del tipo de datos BOOL, BYTE, WORD, INT o UINT se transmiten siempre los bits de menor valor de la entrada a la salida.

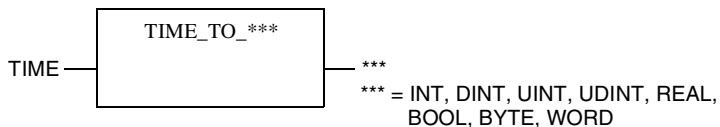
Nota: Para una conversión de TIME a WORD también se dispone del bloque TIME_AS_WORD de la biblioteca de bloques EXTENDED.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	TIME	Valor de entrada
OUT	INT, DINT, UINT, UDINT, REAL, BOOL, BYTE, WORD	Valor de salida

Error de tiempo de ejecución

Aviso de error Si durante la ejecución de la función se produce un desbordamiento del margen de valores en la salida, se emite un aviso de error.

TOF: Retardo de desconexión

56

Vista general

Introducción Este capítulo describe el bloque TOF.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	164
Representación	164
Descripción detallada	165

Descripción breve

Descripción de funcionamiento

El módulo de función se utiliza como retardo de desconexión.

Un flanco 0 -> 1 en la entrada IN provoca un reseteado.

Un flanco 1 -> 0 en la entrada IN inicia la función del temporizador.

Si el tiempo transcurrido (salida ET) alcanza el valor predeterminado en la entrada PT, la salida Q se establece en "0".

El estado inicial de ET durante la primera llamada del módulo de función es "0".

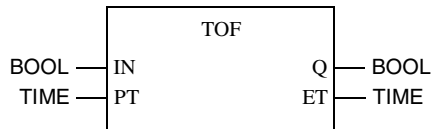
Como parámetros adicionales se pueden configurar EN y ENO.

Nota: La entrada EN no puede utilizarse como función de pausa del módulo de función.
Aun cuando la entrada EN se convierta en "0", se seguirá midiendo el tiempo transcurrido. Si la entrada EN vuelve a convertirse en "1", la salida ET se actualiza y, de este modo, realiza un salto.
Si necesita utilizar la función de pausa, dispone para ello del módulo de función TOF_P de la biblioteca de módulos EXTENDED.

Representación

Símbolo

Representación del módulo:



Descripción de parámetros

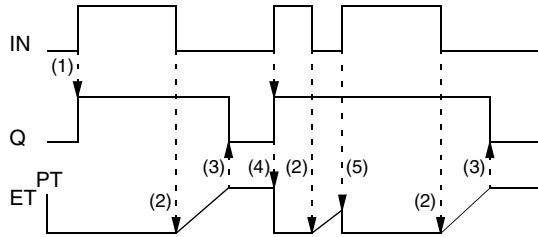
Descripción de los parámetros del módulo:

Parámetro	Tipo de datos	Significado
IN	BOOL	0 -> 1: Reset 1 -> 0: Iniciar temporizador
PT	TIME	Preajuste del tiempo de retardo
Q	BOOL	Salida
ET	TIME	Tiempo transcurrido

Descripción detallada

Cronograma

Representación del retardo de desconexión TOF:



- (1) Si IN se convierte en "1", se convierte Q en "1".
- (2) Si IN se convierte en "0", se inicia el tiempo interno (ET).
- (3) Si el tiempo interno alcanza el valor de PT, Q se convierte en "0".
- (4) Si IN se convierte en "1", se convierte Q en "1" y se detiene/restablece el tiempo interno.
- (5) Si IN se convierte en "1" antes de que el tiempo interno haya alcanzado el valor de PT, se detiene/restablece el tiempo interno, sin que Q se haya convertido en "0".

TON: Retardo de conexión

57

Vista general

Introducción Este capítulo describe el bloque TON.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	168
Representación	168
Descripción detallada	169

Descripción breve

Descripción de funcionamiento

El módulo de función se utiliza como retardo de conexión.

Un flanco 1 -> 0 en la entrada IN provoca un reseteado.

Un flanco 0 -> 1 en la entrada IN inicia la función del temporizador.

Si el tiempo transcurrido (salida ET) alcanza el valor predeterminado en la entrada PT, la salida Q se establece en "1".

El estado inicial de ET durante la primera llamada del módulo de función es "0".

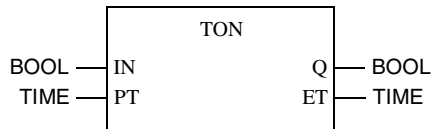
Como parámetros adicionales se pueden configurar EN y ENO.

Nota: La entrada EN no puede utilizarse como función de pausa del módulo de función.
Aun cuando la entrada EN se convierta en "0", se seguirá midiendo el tiempo transcurrido. Si la entrada EN vuelve a convertirse en "1", la salida ET se actualiza y, de este modo, realiza un salto.
Si necesita utilizar la función de pausa, dispone para ello del módulo de función TON_P de la biblioteca de módulos EXTENDED.

Representación

Símbolo

Representación del módulo:



Descripción de parámetros

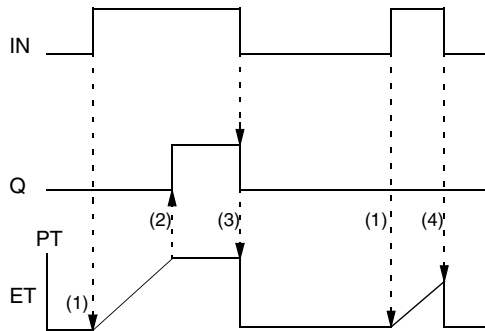
Descripción de los parámetros del módulo:

Parámetro	Tipo de datos	Significado
IN	BOOL	0 -> 1: Iniciar temporizador 1 -> 0: Reset
PT	TIME	Preajuste del tiempo de retardo
Q	BOOL	Salida
ET	TIME	Tiempo transcurrido

Descripción detallada

Cronograma

Representación del retardo de conexión TON:



- (1) Si IN se convierte en "1", se inicia el tiempo interno (ET).
- (2) Si el tiempo interno alcanza el valor de PT, Q se convierte en "1".
- (3) Si IN se convierte en "0", Q se convierte en "0" y se detiene/restablece el tiempo interno.
- (4) Si IN se convierte en "0" antes de que el tiempo interno haya alcanzado el valor de PT, se detiene/restablece el tiempo interno, sin que Q se haya convertido en "1".

TP: Impulso

58

Vista general

Introducción Este capítulo describe el bloque TP.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	172
Representación	172
Descripción detallada	173

Descripción breve

Descripción de la función El bloque de función se utiliza para la generación de un impulso con una duración definida.

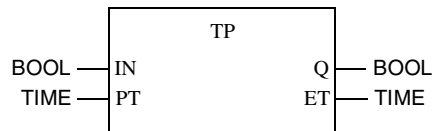
El estado inicial de ET durante la primera llamada del bloque de función es "0".

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

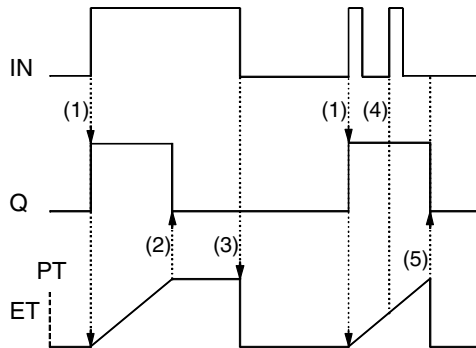
Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	BOOL	Activar impulso
PT	TIME	Ajuste previo de la duración de impulso
Q	BOOL	Salida
ET	TIME	Tiempo interno

Descripción detallada

Diagrama de ciclo

Representación del impulso TP:



- (1) Si IN se convierte en "1", Q se convierte en "1" y se inicia el tiempo interno (ET).
- (2) Si el tiempo interno alcanza el valor de PT, Q se convierte en "0" (independientemente de IN).
- (3) El tiempo interno se detiene/restablece, si IN se convierte en "0".
- (4) Si el tiempo interno aún no ha alcanzado el valor de PT, un ciclo en IN no tendrá influencia sobre el tiempo interno.
- (5) Si el tiempo interno ha alcanzado el valor de PT y si IN es "0", se detiene/restablece el tiempo interno y Q se convierte en "0".

UDINT_EXPT_REAL: Exponencialización

59

Vista general

Introducción

Este capítulo describe el bloque INT_UDINT_EXPT_REAL.

Contenido:

Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	176
Representación	176
Error de tiempo de ejecución	176

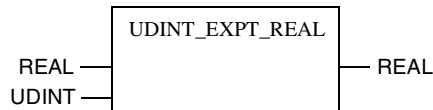
Descripción breve

Descripción de la función La función se utiliza para el cálculo de la exponencialización. El valor en la entrada IN1 (base) se potencia con el valor en la entrada IN2 como exponente y se emite la potencia en la salida.
Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = IN1 \exp IN2$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	REAL	Base
IN2	UDINT	Exponente
OUT	REAL	Potencia

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error cuando

- $(IN1 = 0) \& (IN2 < 0)$ es,
 - en IN1 se trata de un número de coma flotante inadmisibile o
 - se trata de un desbordamiento del margen de valores en la entrada.
-

UDINT_TO_***: Conversión de tipo

60

Vista general

Introducción Este capítulo describe el bloque UDINT_TO_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	178
Representación	178
Error de tiempo de ejecución	178

Descripción breve

Descripción de la función

La función convierte un valor de entrada del tipo de datos UDINT en un valor de salida del tipo de datos DINT, INT, UINT, REAL, TIME, BOOL, BYTE o WORD.

Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Nota: El EFB convierte estrictamente según las reglas CEI. Como este EFB ha sido realizado como función genérica, también se han producido algunas transformaciones ilógicas, p.e. UDINT_TO_BOOL.

Durante la conversión del tipo de datos UDINT al tipo de datos BOOL, BYTE o WORD se transmiten los bits de menor valor del valor de entrada a la salida.

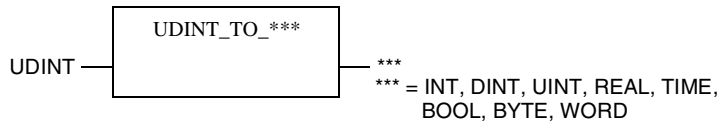
Nota: Para una conversión de UDINT a WORD también se dispone del bloque UDINT_AS_WORD de la biblioteca de bloques EXTENDED.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción del bloque:

Parámetro	Tipo de datos	Significado
IN	UDINT	Valor de entrada
OUT	DINT, INT, UINT, REAL, TIME, BOOL, BYTE, WORD	Valor de salida

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error, cuando se sobrepasa el margen de valores de la salida.

UINT_EXPT_REAL: Exponencialización

61

Vista general

Introducción Este capítulo describe el bloque UDINT_EXPT_REAL.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	180
Representación	180
Error de tiempo de ejecución	180

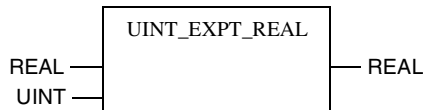
Descripción breve

Descripción de la función La función se utiliza para el cálculo de la exponencialización. El valor en la entrada IN1 (base) se potencia con el valor en la entrada IN2 como exponente y se emite la potencia en la salida.
 Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

$OUT = IN1 \exp IN2$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	REAL	Base
IN2	UINT	Exponente
OUT	REAL	Potencia

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error cuando

- $(IN1 = 0) \& (IN2 < 0)$ es,
 - en IN1 se trata de un número de coma flotante inadmisibile o
 - se trata de un desbordamiento del margen de valores en la entrada.
-

UINT_TO_***: Conversión de tipo

62

Vista general

Introducción Este capítulo describe el bloque UDINT_TO_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	182
Representación	182
Error de tiempo de ejecución	182

Descripción breve

Descripción de la función

La función convierte un valor de entrada del tipo de datos UINT en un valor de salida del tipo de datos BOOL, BYTE, WORD, DINT, INT, UDINT, REAL o TIME.

Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Nota: El EFB convierte estrictamente según las reglas CEI. Como este EFB ha sido realizado como función genérica, también se han producido algunas transformaciones ilógicas, p.e. UINT_TO_BOOL.

En la conversión de un valor de entrada del tipo de datos UINT al tipo de datos WORD se transmite la muestra de bits de la entrada inalterada a la salida.

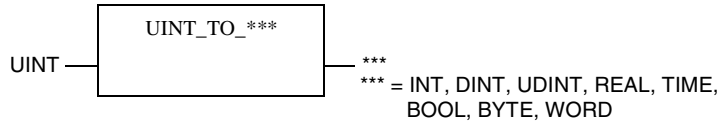
Durante la conversión de un valor de entrada del tipo de datos UINT a los tipos de datos BOOL o BYTE se transmiten los bits de menor valor de la entrada a la salida.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	UINT	Valor de entrada
OUT	BOOL, BYTE, WORD, DINT, INT, UDINT, REAL, TIME	Valor de salida

Error de tiempo de ejecución

Aviso de error

Se emite un aviso de error, cuando se sobrepasa el margen de valores de la salida.

WORD_TO_***: Conversión de tipo

63

Vista general

Introducción Este capítulo describe el bloque WORD_TO_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	184
Representación	185
Error de tiempo de ejecución	185

Descripción breve

Descripción de la función

La función convierte un valor de entrada del tipo de datos WORD a un tipo de datos del grupo ANY_NUM o del tipo de datos BOOL, BYTE o TIME.

Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

Nota: El EFB convierte estrictamente según las reglas CEI. Como este EFB ha sido realizado como función genérica, también se han producido algunas transformaciones ilógicas, p.e. WORD_TO_TIME. En este sentido se debe tener en cuenta que la muestra del bit de entrada es transmitida a la palabra de mayor valor de la palabra de salida.

Durante la conversión del tipo de datos WORD al tipo de datos DINT, UDINT o REAL se transmite la muestra de bit de la entrada al bit de menor valor de la salida. Los bits de mayor valor de la salida se ubican en cero.

Durante la conversión del tipo de datos WORD al tipo de datos BOOL o BYTE, se transmiten los bits de menor valor del valor de entrada a la salida.

Nota: Para una conversión de WORD a REAL se encuentran disponibles también los bloques W_INT_REAL y W_UINT_REAL de la biblioteca de bloques ANA_IO y el bloque WORD_AS_REAL de la biblioteca de bloques EXTENDED.

Nota: Para una conversión de WORD a TIME se encuentra disponible también el bloque WORD_AS_TIME de la biblioteca de bloques EXTENDED.

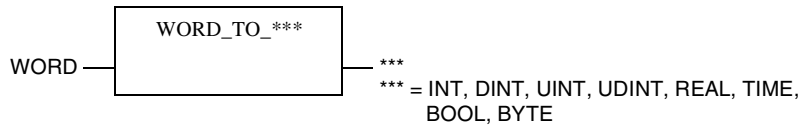
Nota: Para una conversión de WORD a UDINT se encuentra disponible también el bloque WORD_AS_UDINT de la biblioteca de bloques EXTENDED.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN	WORD	Valor de entrada
OUT	INT, DINT, UINT, UDINT, REAL, TIME, BOOL, BYTE	Valor de salida

Error de tiempo de ejecución

Aviso de error

Si durante la conversión al tipo de datos REAL se genera un número de coma flotante inadmisibles, se emite un aviso de error.

XOR_***: Función O exclusivo

64

Vista general

Introducción Este capítulo describe el bloque XOR_***.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción breve	188
Representación	188

Descripción breve

Descripción de la función

La Función vincula (según la lógica exclusiva O) las secuencias de bits en las entradas y emite el resultado en la salida. La vinculación se realiza por bits.

Se pueden procesar los tipos de datos del grupo ANY_BIT.

Los tipos de datos de todos los valores de entrada y el del valor de salida deben ser iguales. Para el procesamiento de los diferentes tipos de datos se dispone de una función especial.

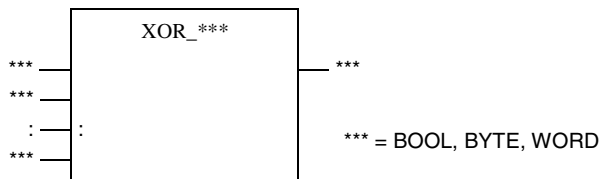
Se puede aumentar el número de entradas.

Como parámetros adicionales se pueden proyectar EN y ENO.

Representación

Símbolo

Representación del bloque



Fórmula

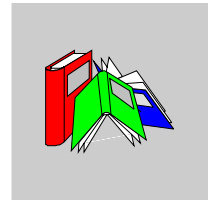
$$\text{OUT} = \text{IN1 XOR IN2 XOR .. XOR INn}$$

Descripción de parámetros

Descripción de parámetros de bloque

Parámetro	Tipo de datos	Significado
IN1	BOOL, BYTE, WORD	Secuencia de bits de entrada
IN2	BOOL, BYTE, WORD	Secuencia de bits de entrada
INn	BOOL, BYTE, WORD	Secuencia de bits de entrada
OUT	BOOL, BYTE, WORD	Secuencia de bits de salida

Glosario



A

Advertencia

Si durante el procesamiento de un FFB o de un paso se detecta un estado crítico (p. ej., valores de entrada críticos o rebasamiento del límite de tiempo), se generará una advertencia que se puede ver con el comando de menú **Online** → **Indicación de eventos....** En el caso de los FFB, la salida ENO permanece en "1".

Ajustes de la transferencia de datos

Ajustes que establecen la forma en que se va a transferir la información desde el equipo de programación al PLC.

ANL_IN

ANL_IN representa el tipo de datos "entrada analógica" y se utiliza para el procesamiento de valores analógicos. Las referencias 3x del módulo de entrada analógica configurado definidas en la lista de componentes E/S se asignan automáticamente al tipo de datos y, por eso, sólo pueden estar ocupadas por Unlocated Variables.

ANL_OUT

ANL_OUT representa el tipo de datos "salida analógica" y se utiliza para el procesamiento de valores analógicos. Las referencias 4x del módulo de salida analógica configurado definidas en la lista de componentes E/S se asignan automáticamente al tipo de datos y, por eso, sólo pueden estar ocupadas por Unlocated Variables.

ANY

En la presente versión, "ANY" incluye los tipos de datos elementales BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME y WORD, así como los tipos de datos derivados de ellos.

ANY_BIT

En la presente versión, "ANY_BIT" incluye los tipos de datos BOOL, BYTE y WORD.

ANY_ELEM	En la presente versión, "ANY_ELEM" incluye los tipos de datos BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME y WORD.
ANY_INT	En la presente versión, "ANY_INT" incluye los tipos de datos DINT, INT, UDINT y UINT.
ANY_NUM	En la presente versión, "ANY_NUM" incluye los tipos de datos DINT, INT, REAL, UDINT y UINT.
ANY_REAL	En la presente versión, "ANY_REAL" incluye el tipo de datos REAL.
Argumento	Significa lo mismo que parámetro actual.
Atrium	El controlador basado en PC se encuentra en una platina AT estándar y se puede utilizar dentro de un equipo central en un slot de bus ISA. El módulo dispone de una placa madre (necesita el controlador SA85) con dos slots para tarjetas PC104. Una de estas tarjetas PC104 se utiliza como CPU y la otra, como controlador INTERBUS.

B

Base de datos del proyecto	Base de datos del equipo de programación que contiene la información de configuración para un proyecto.
Biblioteca	Colección de objetos de software previstos para reutilizarlos al programar nuevos proyectos, o incluso para formar nuevas bibliotecas. Como ejemplos se pueden citar las bibliotecas tipos de módulos de función elementales. Las bibliotecas EFB se pueden subdividir en grupos.
Bits de entrada (referencias 1x)	El estado 1/0 de los bits de entrada se controla mediante los datos de proceso que llegan a la CPU desde un dispositivo de entrada. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">Nota: La x que aparece detrás de la primera cifra del tipo de referencia representa un lugar de almacenamiento de cinco posiciones en la memoria de datos del usuario, por ejemplo, la referencia 100201 significa un bit de entrada en la dirección 201 de la memoria de señal.</div>
Bits de estado	Existe un bit de estado para cada participante con entrada global o entrada/salida específica de datos Peer Cop. Si se ha transferido con éxito un grupo definido de datos dentro del timeout establecido, el bit de estado correspondiente pasará a 1. En caso contrario, este bit pasará a 0 y se borrarán todos los datos pertenecientes a este grupo (a 0).

Bits de salida/ marca (referencias 0x)	Un bit de salida/marca se puede utilizar para controlar los datos de salida reales mediante una unidad de salida del sistema de control o para definir una o varias salidas binarias en la memoria de señal. Nota: La x que aparece detrás de la primera cifra del tipo de referencia representa un lugar de almacenamiento de cinco posiciones en la memoria de datos del usuario, por ejemplo, la referencia 000201 significa un bit de salida o marca en la dirección 201 de la memoria de señal.
Bobina	Una bobina es un elemento LD que transmite, sin modificarlo, el estado de la conexión horizontal de su izquierda a la conexión horizontal de su derecha. Al hacerlo, el estado se guarda en la variable/dirección directa correspondiente.
BOOL	BOOL indica el tipo de datos "booleano". La longitud de los elementos de datos es de 1 bit (en la memoria se guarda en 1 byte). El rango de valores para las variables de este tipo de datos es 0 (FALSE) y 1 (TRUE).
BYTE	BYTE indica el tipo de datos "secuencia de bits 8". La entrada se efectúa en forma de literal de base 2, literal de base 8 o literal de base 16. La longitud de los elementos de datos es de 8 bits. A este tipo de datos no se le puede asignar un rango de valores numérico.

C

Ciclo del programa	Un ciclo del programa se compone de la carga de las entradas, el procesamiento de la lógica del programa y la entrega de las salidas.
Código de sección	El código de sección es el código ejecutable de una sección. El tamaño del código de sección depende principalmente de la cantidad de módulos de la sección.
Código DFB	El código DFB constituye el código DFB ejecutable de una sección. El tamaño del código DFB depende principalmente de la cantidad de módulos de una sección.
Código EFB	El código EFB es el código ejecutable de todos los EFB utilizados. Entre ellos se encuentran también los EFB dentro de DFBs.
Conexión	Conexión de control o de flujo de datos entre objetos gráficos (p. ej., pasos en el editor de SFC o módulos de función en el editor de FBD) dentro de una sección representada gráficamente como una línea.
Conexión local (Local Link)	La conexión de red local es la red que conecta al participante local con otros participantes, ya sea directamente o mediante amplificador de bus.

Conexiones binarias	Conexiones entre salidas y entradas de FFBs del tipo de datos BOOL.
Conexiones serie	En las conexiones serie (COM) se transfiere la información bit a bit.
Constantes	Las constantes son Unlocated Variables a las que se ha asignado un valor que la lógica del programa no puede modificar (protegido contra escritura).
Contacto	Un contacto es un elemento LD que transmite un estado a la conexión horizontal de su derecha. Este estado se obtiene del vínculo booleario AND entre el estado de la conexión horizontal de la izquierda y el estado de la variable/dirección directa correspondiente. Un contacto no modifica el valor de la variable/dirección directa correspondiente.
Contador de funciones	<p>El contador de funciones sirve para identificar inequívocamente una función en un programa o DFB. El contador de funciones no se puede editar y se asigna automáticamente. El contador de funciones tiene siempre la estructura: .n.m</p> <p>n = número de la sección (número correlativo) m = número del objeto FFB dentro de la sección (número correlativo)</p>
Creación de instancias	Creación de una instancia.

D

Datos de instancia DFB	Los datos de instancia DFB son datos internos de los módulos de función utilizados en el programa.
Datos de sección	Son los datos locales de una sección, como por ejemplo literales, conexiones entre módulos, salidas y entradas de módulos no conectados, así como la memoria de estado interna de EFBs.
	<div style="border: 1px solid black; padding: 5px;"><p>Nota: Los datos que aparecen en los DFB de esta sección no son datos de sección.</p></div>
Datos globales	Los datos globales son Unlocated Variables.

DDE (Dynamic Data Exchange) Intercambio de datos dinámico	<p>La interfase DDE permite el intercambio de datos dinámico entre dos programas que se ejecuten bajo Windows. El usuario puede utilizar la interfase DDE en el monitor ampliado para ejecutar sus propias aplicaciones de visualización. Con esta interfase, el usuario (es decir, el cliente DDE) no sólo puede leer los datos del monitor ampliado (el servidor DDE), sino también escribir datos en el PLC a través del servidor. El usuario puede así modificar datos directamente en PLC, mientras verifica y analiza los resultados. Utilizando esta interfase, el usuario puede crear su propia "Graphic-Tool", "Face Plate" o "Tuning Tool" e integrarlas en el sistema. Estas herramientas se pueden escribir en cualquier lenguaje que DDE pueda interpretar, p. ej., Visual Basic o Visual C++. Las herramientas se ejecutan cuando el usuario pulsa uno de los botones del cuadro de diálogo Monitor ampliado. Concept-Graphic-Tool: Mediante la conexión DDE entre Concept y la Herramienta Gráfica Concept se pueden representar las señales de una configuración como cronograma.</p>
Declaración	<p>Mecanismo para establecer la definición de un elemento de lenguaje. Normalmente, una declaración incluye la conexión de un identificador con un elemento de lenguaje y la asignación de atributos como tipos de datos y algoritmos.</p>
Desfragmentación	<p>La desfragmentación elimina huecos indeseados (generados, por ejemplo, al borrar variables no utilizadas) en la memoria.</p>
DFB (Derived Function Block) módulo de función derivado	<p>Un módulo de función derivado representa la llamada de un tipo de módulo de función derivado. Encontrará más detalles acerca de la forma gráfica de la llamada en la definición de "módulo de función (instancia)". A diferencia de las llamadas de los tipos EFB, las llamadas de los tipos DFB se identifican mediante líneas verticales dobles en la parte derecha e izquierda del símbolo de bloque rectangular. El cuerpo de un tipo de módulo de función derivado se diseña en el lenguaje FBD, lenguaje LD, lenguaje ST o en el lenguaje IL, pero sólo en la versión actual del sistema de programación. Las funciones derivadas tampoco se pueden definir aún en la versión actual. Se distingue entre DFBs locales y DFBs globales.</p>
DFB globales	<p>Los DFB globales están disponibles en cada proyecto de Concept. La posición de los DFB globales dependerá de los ajustes del fichero CONCEPT.INI.</p>
DFB locales	<p>Los DFB locales sólo están disponibles en un único proyecto de Concept y se guardan en el directorio DFB bajo el directorio del proyecto.</p>
DINT	<p>DINT representa el tipo de datos "número entero de longitud doble (double integer)". La entrada se efectúa en forma de literal entero, literal de base 2, literal de base 8 o literal de base 16. La longitud de los elementos de datos es de 32 bits. El rango de valores para las variables de este tipo de datos va desde $-2 \exp(31)$ hasta $2 \exp(31) - 1$.</p>

Dirección de participante	La dirección de participante sirve para identificar sin posibilidad de confusión a un participante de red en la ruta de acceso. La dirección se configura directamente en el participante, p. ej., mediante un conmutador rotativo en la parte posterior de los módulos.
Direcciones	Las direcciones (directas) son rangos de memoria en el PLC. Se encuentran en la memoria de señal y pueden estar asignadas a módulos de entrada/salida. Las direcciones directas pueden indicarse (y mostrarse) en los siguientes formatos: <ul style="list-style-type: none">● Formato estándar (400001)● Formato de separador (delimitador) (4:00001)● Formato compacto (4:1)● Formato IEC (QW1)
DP (PROFIBUS)	DP = periferia descentral
Dummy	Fichero vacío compuesto por un encabezado de texto con información general sobre el fichero, como autor, fecha de creación, denominación de EFB, etc. El usuario debe completarlo por medio de otras entradas.

E

Elemento de lenguaje	Cualquier elemento básico en uno de los lenguajes de programación IEC, por ejemplo, un paso en SFC, una instancia de módulo de función en FBD o el valor inicial de una variable.
EN/ENO (Habilitación / Indicación de error)	Si el valor de EN es "0", cuando se ejecute el FFB no se ejecutarán los algoritmos definidos mediante el FFB, y todas las salidas conservarán su valor anterior. En este caso, el valor de ENO se ajustará automáticamente a "0". Si el valor de EN es "1", cuando se ejecute el FFB se ejecutarán los algoritmos definidos mediante el FFB. Una vez que se hayan ejecutado estos algoritmos sin errores, el valor de ENO se ajustará automáticamente a "1". Si se produce un error durante la ejecución de estos algoritmos, ENO pasará a "0" automáticamente. El comportamiento de salida de los FFB no depende de si los FFB se han ejecutado sin EN/ENO o con EN=1. Si está activada la visualización de EN/ENO, habrá que conectar obligatoriamente la entrada EN. De no ser así, no se ejecutará el FFB. La configuración de EN y ENO se activa y desactiva en el cuadro de diálogo de las propiedades de los módulos. Este cuadro de diálogo se abre con los comandos de menú Objetos → Propiedades... o haciendo doble clic en el FFB.

Equipo de programación	Hardware y software que permiten programar, configurar, probar, poner en marcha y buscar errores tanto en las aplicaciones de PLC como en las aplicaciones de sistema descentrales para hacer posible la documentación y el archivado de la fuente. En determinadas circunstancias, el equipo de programación se puede utilizar también para la visualización de procesos.
Error	Si durante el procesamiento de un FFB o de un paso se detecta un error (p. ej., valores de entrada no permitidos o un error de tiempo), se generará un mensaje de error que se puede ver con el comando de menú Online → Indicación de eventos... En el caso de los FFB, la salida ENO se ajustará a "0".
Error de tiempo de ejecución	Errores que se producen durante el procesamiento del programa en el PLC en objetos SFC (p. ej., pasos) o en FFBs. Estos errores pueden ser, por ejemplo, desbordes del rango de valores en contadores o errores de tiempo en pasos.
Estación de E/S DCP	Con un procesador de control distribuido (D908) puede instalar una red descentralizada con un PLC de rango superior. Si se utiliza un D908 con un PLC descentralizado, el PLC de rango superior considerará al PLC descentralizado como una estación de E/S descentralizada. El D908 y el PLC descentralizado se comunican a través de un bus de sistema, lo que proporciona un alto rendimiento con una mínima repercusión en el tiempo de ciclo. El intercambio de datos entre el D908 y el PLC de rango superior se efectúa a 1,5 megabits por segundo a través del bus de E/S descentralizado. Un PLC de rango superior puede trabajar hasta con 31 procesadores D908 (dirección 2-32).
Evaluación	Proceso por el cual se averigua el valor de una función o de las salidas de un módulo de función durante la ejecución del programa.
Expresión	Las expresiones están compuestas por operadores y operandos.

F

FFB (funciones/módulos de función)	Término colectivo para EFB (funciones elementales/módulos de función elementales) y DFB (módulos de función derivados)
Fichero de código fuente (EFB Concept)	El fichero de código fuente es un fichero de fuente C++ convencional. Tras ejecutar el comando de menú Biblioteca → Generar ficheros , este fichero dispondrá de un cuadro de códigos de EFB en el que deberá introducir un código específico para el EFB seleccionado. Para ello, ejecute el comando de menú Objetos → Fuente .

Fichero de copia de seguridad (EFB Concept)	El fichero de copia de seguridad es una copia del último fichero de código fuente. El nombre de este fichero de copia de seguridad es "backup??.c" (se supone que no tiene más de 100 copias del fichero de código fuente). El primer fichero de copia de seguridad tiene el nombre "backup00.c". Si ha realizado modificaciones en el fichero de definición que no provoquen cambios en la interfase en caso de EFBs, podrá ahorrarse la creación del fichero de copia de seguridad editando el fichero de código fuente (Objetos → Fuente). Si crea un fichero de copia de seguridad, puede asignarle el nombre del fichero fuente.
Fichero de definición (EFB Concept)	El fichero de definición contiene información descriptiva general sobre el EFB seleccionado y sus parámetros formales.
Fichero de modelo (EFB Concept)	El fichero de modelo es un fichero ASCII con información de diseño para el editor de FBD Concept y los parámetros para la generación de códigos.
Fichero de prototipo (EFB Concept)	El fichero de prototipo contiene todos los prototipos de las funciones asignadas. A continuación, si la hay, se indicará una definición de tipo de la estructura de estado interna.
Filtro FIR	(Finite Impulse Response Filter) Filtro con respuesta de impulsos finita
Filtro IIR	(Infinite Impulse Response Filter) Filtro con respuesta de impulsos infinita
Formato compacto (4:1)	La primera cifra (la referencia) se separa de la siguiente dirección mediante dos puntos (:) sin poner en la dirección los ceros a la izquierda.
Formato de separador (delimitador) (4:0001)	La primera cifra (la referencia) se separa mediante dos puntos (:) de las siguientes cinco posiciones de la dirección.
Formato estándar (40001)	Justo detrás de la primera cifra (la referencia) está la dirección de cinco posiciones.
Formato IEC (QW1)	En la primera posición de la dirección hay un identificador conforme a IEC, seguido de la dirección de cinco posiciones: <ul style="list-style-type: none">● %0x12345 = %Q12345● %1x12345 = %I12345● %3x12345 = %IW12345● %4x12345 = %QW12345

Función (FUNK)	<p>Unidad organizativa del programa que proporciona exactamente un elemento de datos en la ejecución. Una función no tiene información interna de estado. Si se ejecuta varias veces la misma función con los mismos valores en los parámetros de entrada, se obtienen siempre los mismos valores de salida.</p> <p>Encontrará más detalles acerca de la forma gráfica de las llamadas de funciones en la definición de "módulo de función (instancia)". A diferencia de las llamadas a módulos de función, las llamadas a funciones sólo tienen una única salida sin nombre, ya que éste coincide con el nombre de la función. En FBD, cada llamada se identifica mediante un número inequívoco que le asigna el bloque gráfico; este número se genera de forma automática y no se puede modificar.</p>
Funciones y módulos de función elementales (EFB)	<p>Denominación de las funciones o los módulos de función cuyas definiciones de tipos no han sido formuladas en uno de los lenguajes IEC, es decir, el cuerpo de las definiciones no se puede modificar con el editor DFB (DFB Concept), por ejemplo. Los tipos de EFB se programan en "C" y están disponibles mediante bibliotecas en la forma previa a la compilación.</p>

G

Grupos (EFB)	<p>Algunas bibliotecas de EFB (p. ej., la biblioteca IEC) están subdivididas en grupos. Esto facilita la localización de los EFB, sobre todo en las bibliotecas de grandes dimensiones.</p>
---------------------	---

H

Horizontal	<p>Horizontal significa que la página del texto impreso es más ancha que alta.</p>
-------------------	--

I

IEC 61131-3	<p>Norma internacional: Autómatas programables – Parte 3: Lenguajes de programación.</p>
--------------------	--

- Instrucción (IL)** Las instrucciones son los "comandos" del lenguaje de programación IL. Cada instrucción comienza en una línea nueva y va seguida de un operador o modificador y, si fuera necesario para la operación en cuestión, de uno o más operandos. Si se utilizan varios operandos, éstos se separarán con comas. Antes de la instrucción puede haber una marca, que irá seguida de dos puntos (:). Si hubiera un comentario, deberá ser el último elemento de la línea.
- Instrucción (LL984)** Al programar los controles eléctricos, el usuario deberá implementar instrucciones codificadas operacionalmente en forma de objetos de imagen que están divididos en forma de contactos reconocibles. A nivel de usuario, los objetos de programa proyectados se convierten durante el proceso de carga en códigos operacionales utilizables por el equipo. Estos códigos OP se descodifican en la CPU, y las funciones de firmware del controlador los procesan de tal forma que se puede implementar el control deseado.
- Instrucción (ST)** Las instrucciones son los "comandos" del lenguaje de programación ST. Las instrucciones se deben cerrar mediante el símbolo del punto y coma. En una línea puede haber varias instrucciones (separadas por punto y coma).
- INT** INT representa el tipo de datos "número entero (integer)". La entrada se efectúa en forma de literal entero, literal de base 2, literal de base 8 o literal de base 16. La longitud de los elementos de datos es de 16 bits. El rango de valores para las variables de este tipo de datos va desde $-2 \text{ exp } (15)$ hasta $2 \text{ exp } (15) -1$.
- INTERBUS (PCP)** Para utilizar el canal INTERBUS PCP y el pretratamiento de datos de proceso INTERBUS (PDV) se ha introducido en el configurador de Concept el nuevo tipo de estación de E/S INTERBUS (PCP). A este tipo de estación de E/S se le ha asignado de forma fija el módulo de conexión INTERBUS 180-CRP-660-01. El 180-CRP-660-01 sólo se diferencia del 180-CRP-660-00 en que el rango de E/S en la memoria de señal del controlador es considerablemente mayor.

L

- Ladder Diagram (LD)** Ladder Diagram es un lenguaje de programación gráfico que cumple la norma IEC 1131 y se orienta ópticamente a los "circuitos de corriente" de un esquema de contactos de relés.

Ladder Logic 984 (LL)	<p>En los términos Ladder Logic y Ladder Diagram, la palabra Ladder (contacto) se refiere a la ejecución. A diferencia del esquema de conexiones, en electricidad se utiliza el esquema de contactos para dibujar un circuito eléctrico (con símbolos eléctricos) que representa el desarrollo de los eventos, y no los cables que conectan los elementos entre sí. Una interfase de usuario convencional para controlar las acciones de los dispositivos de automatización admite una interfase de esquema de contactos para que los técnicos electricistas no tengan que aprender ningún lenguaje de programación con el que no estén familiarizados para implementar un programa de control.</p> <p>La estructura del esquema de contactos real permite conectar los elementos eléctricos de tal forma que se genere una salida de control que dependa de un flujo de corriente lógico a través de los objetos eléctricos utilizados, que representan la condición antes exigida de un dispositivo eléctrico físico.</p> <p>De forma sencilla, la interfase de usuario es una pantalla de vídeo generada por la aplicación de programación del PLC que configura un reticulado vertical y horizontal en el que se organizan los objetos de programación. El esquema recibe corriente por la parte izquierda del reticulado y, al unirlo con objetos que están activos, la corriente fluye de izquierda a derecha.</p>
Lenguaje de ejecución (SFC)	<p>Los elementos de lenguaje SFC permiten subdividir una unidad organizativa de un programa de PLC en un número de pasos y transiciones, vinculados entre sí mediante conexiones direccionales. A cada paso le corresponde un número de acciones, y cada transición está vinculada a una condición de transición.</p>
Lenguaje de módulo de función (FBD)	<p>Una o varias secciones que contienen representaciones gráficas de redes de funciones, módulos de función y conexiones.</p>
Lista de componentes de E/S	<p>En la lista de componentes E/S se configuran los módulos de E/S y expertos de las distintas unidades centrales.</p>
Lista de instrucciones (IL)	<p>IL es un lenguaje de programación conforme a IEC 1131 en el que las operaciones como, por ejemplo, llamadas condicionales o incondicionales de módulos de función y funciones, los saltos condicionales o incondicionales, etc., se representan mediante instrucciones.</p>
Literal entero	<p>Los literales enteros sirven para indicar los valores enteros en el sistema decimal. Los valores pueden ir precedidos por el signo (+/-). Los guiones bajos individuales (_) situados entre los números no son significativos.</p> <p>Ejemplo -12, 0, 123_456, +986</p>

Literal real con exponente

Los literales reales con exponente sirven para indicar los valores con coma flotante en el sistema decimal. Los literales reales con exponente se caracterizan por la indicación del punto decimal. El exponente indica la potencia decimal por la que se debe multiplicar el número de partida para obtener el valor que hay que representar. La base puede ir precedida por el signo menos (-). El exponente puede ir precedido por el signo menos o más (+/-). Los guiones bajos individuales (_) situados entre los números no son significativos. (Sólo entre cifras y no inmediatamente delante o detrás de una coma decimal ni tampoco de "E", "E+" o "E-").

Ejemplo

-1.34E-12 ó -1.34e-12

1.0E+6 ó 1.0e+6

1.234E6 ó 1.234e6

Literal tipificado

Si desea establecer usted mismo el tipo de datos de un literal, puede hacerlo con la siguiente construcción: 'Nombre del tipo de datos' #'Valor del literal'.

Ejemplo

INT#15 (tipo de datos: entero, valor: 15),

BYTE#00001111 (tipo de datos: Byte, valor: 00001111)

REAL#23.0 (tipo de datos: Real, valor: 23.0)

Para la asignación del tipo de datos REAL también se puede indicar el valor de la siguiente forma: 23.0.

Indicando una posición de la coma se asigna automáticamente el tipo de datos REAL.

Literales

Los literales sirven para proporcionar directamente valores a las entradas de FFB, las condiciones de transición, etc. La lógica del programa no puede sobrescribir estos valores (están protegidos contra escritura). Se distingue entre literales genéricos y tipificados.

Además, los literales sirven para asignar un valor a las constantes o un valor inicial a una variable.

La entrada se efectúa en forma de literal de base 2, literal de base 8, literal de base 16, literal entero, literal real o literal real con exponente.

Literales de base 16

Los literales de base 16 sirven para indicar valores enteros en el sistema hexadecimal. La base debe identificarse mediante el prefijo 16#. Los valores no pueden ir precedidos por el signo (+/-). Los guiones bajos individuales (_) situados entre los números no son significativos.

Ejemplo

16#F_F ó 16#FF (decimal 255)

16#E_0 ó 16#E0 (decimal 224)

Literales de base 2	<p>Los literales de base 2 sirven para indicar valores enteros en el sistema dual. La base debe identificarse mediante el prefijo 2#. Los valores no pueden ir precedidos por el signo (+/-). Los guiones bajos individuales (_) situados entre los números no son significativos.</p> <p>Ejemplo 2#1111_1111 ó 2#11111111 (decimal 255) 2#1110_0000 ó 2#11100000 (decimal 224)</p>
Literales de base 8	<p>Los literales de base 8 sirven para indicar valores enteros en el sistema octal. La base debe identificarse mediante el prefijo 8#. Los valores no pueden ir precedidos por el signo (+/-). Los guiones bajos individuales (_) situados entre los números no son significativos.</p> <p>Ejemplo 8#3_77 ó 8#377 (decimal 255) 8#34_0 ó 8#340 (decimal 224)</p>
Literales de duración	<p>Las unidades permitidas para la duración (TIME) son días (D), horas (H), minutos (M), segundos (S) y milisegundos (MS) o combinaciones de ellas. La duración se debe identificar con el prefijo t#, T#, time# o TIME#. El "desborde" de la unidad de mayor valor está permitido; por ejemplo, la entrada T#25H15M es válida.</p> <p>Ejemplo t#14MS, T#14.7S, time#18M, TIME#19.9H, t#20.4D, T#25H15M, time#5D14H12M18S3.5MS</p>
Literales genéricos	<p>Si no considera relevante el tipo de datos de un literal, indique solamente el valor para dicho literal. En este caso, Concept asignará al literal un tipo de datos adecuado de forma automática.</p>
Literales reales	<p>Los literales reales sirven para indicar los valores con coma flotante en el sistema decimal. Los literales reales se caracterizan por la indicación del punto decimal. Los valores pueden ir precedidos por el signo (+/-). Los guiones bajos individuales (_) situados entre los números no son significativos.</p> <p>Ejemplo -12.0, 0.0, +0.456, 3.14159_26</p>
Llamada	<p>Proceso por el que se inicia la ejecución de una operación.</p>

Located Variable A las Located Variables se les asigna una dirección de memoria de señal (direcciones de referencia 0x, 1x, 3x, 4x). El valor de estas variables se almacena en la memoria de señal y se puede modificar online con el editor de datos de referencia. Para acceder a estas variables se puede utilizar su nombre simbólico o su dirección de referencia.

Todas las entradas y salidas del PLC están conectadas a la memoria de señal. El acceso del programa a señales periféricas conectadas al PLC se efectúa sólo mediante Located Variables. Los accesos externos a través de las interfaces Modbus o Modbus Plus del PLC, p. ej., por parte de sistemas de visualización, también son posibles mediante Located Variables.

M

Macro Las macros se crean con ayuda del software DFB Concept. Las macros sirven para duplicar las secciones y redes que se utilizan con frecuencia (incluyendo su lógica, sus variables y su declaración de variables). Se distingue entre macros locales y globales.

Las macros tienen las siguientes propiedades:

- Las macros sólo se pueden crear en los lenguajes de programación FBD y LD.
- Las macros sólo contienen una única sección.
- Las macros pueden contener una sección todo lo compleja que se desee.
- Desde el punto de vista técnico, una macro para la que se ha creado un instancia, es decir, una macro insertada en una sección, no se distingue de una sección generada de forma convencional.
- Llamada de DFB en una macro
- Declaración de variables
- Utilización de estructuras de datos propias de la macro
- Aceptación automática de las variables declaradas en la macro
- Valores iniciales para variables
- Creación de múltiples instancias de una macro en todo el programa con variables diferentes
- El nombre de la sección, los nombres de las variables y el nombre de la estructura de datos pueden contener hasta 10 marcas de intercambio distintas (de @0 a @9).

Macros globales Las macros globales están disponibles en todos los proyectos de Concept y se encuentran en el directorio DFB, justo debajo del directorio Concept.

Macros locales Las macros locales sólo están disponibles en un único proyecto de Concept y se guardan en el directorio DFB bajo el directorio del proyecto.

Memoria de programa IEC	La memoria de programa IEC está compuesta por el código de programa, el código EFB, los datos de sección y los datos de instancia del DFB.
Memoria de señal	La memoria de señal es el espacio de memoria para todas las magnitudes a las que se accede a través de referencias (representación directa) en el programa de usuario. Por ejemplo, en la memoria de señal hay bits de entrada, bits de salida/marca, palabras de entrada y palabras de salida/marca.
Memoria total IEC	La memoria total IEC está compuesta por la memoria de programa IEC y los datos globales.
MMI	(Man-Machine-Interface) Interfase hombre-máquina
Modo ASCII	American Standard Code for Information Interchange. El modo ASCII se utiliza para la comunicación con distintos dispositivos centrales. ASCII funciona con 7 bits de datos.
Modo RTU	Remote Terminal Unit (unidad de terminal remota) El modo RTU se utiliza para la comunicación entre el PLC y un ordenador personal compatible con IBM. RTU funciona con 8 bits de datos.
Módulo de función (instancia) (FB)	<p>Un módulo de función es una unidad organizativa del programa que calcula, de acuerdo con la funcionalidad definida en la descripción del tipo de módulo de función, los valores para sus salidas y variable(s) interna(s) cuando se ejecuta como instancia determinada. Todos los valores de las salidas y variables internas de una determinada instancia de módulo de función se conservan de una llamada del módulo de función a la siguiente. Aunque se ejecute varias veces una misma instancia de módulo de función con los mismos argumentos (valores de parámetros de entrada), no se obtendrán necesariamente los mismos valores de salida. Cada instancia de módulo de función se representa gráficamente mediante un símbolo de bloque rectangular. El nombre del tipo de módulo de función está dentro del rectángulo, centrado en la parte superior. El nombre de la instancia del módulo de función también está arriba, pero fuera del rectángulo. Se genera automáticamente al crear una instancia, pero, en caso necesario, el usuario puede modificarlo. Las entradas se representan en la parte izquierda del bloque; las salidas, en la derecha. Los nombres de los parámetros formales de entrada/salida aparecen dentro del rectángulo, en el lugar correspondiente.</p> <p>La descripción anterior de la representación gráfica también es aplicable en general a las llamadas de funciones y a las llamadas de DFB. En las correspondientes definiciones se indican las diferencias.</p>
Módulo SA85	El módulo SA85 es un adaptador Modbus Plus para ordenadores IBM-AT o compatibles.

N

Nombre de instancia

Identificador que pertenece a una determinada instancia de módulo de función. El nombre de instancia sirve para identificar sin posibilidad de confusión un módulo de función en una unidad organizativa del programa. El nombre de instancia se genera automáticamente, pero se puede modificar. El nombre de instancia debe ser único en toda la unidad organizativa del programa, en este caso no se distingue entre mayúsculas y minúsculas. Si ya existe el nombre indicado, aparecerá una advertencia y deberá seleccionar otro nombre. El nombre de instancia se debe formular de acuerdo con la Nomenclatura IEC; si no, aparecerá un mensaje de error. El nombre de instancia generado de manera automática tiene siempre la estructura: FBI_n_m

FBI = instancia de módulo de función
n = número de la sección (número correlativo)
m = número del objeto FFB dentro de la sección (número correlativo)

Nombre del paso

El nombre del paso sirve para identificar inequívocamente a un paso dentro de una unidad organizativa del programa. El nombre del paso se genera automáticamente, pero se puede modificar. El nombre del paso debe ser único en toda la unidad organizativa del programa, si no, se generará un mensaje de error. El nombre de paso generado automáticamente tiene siempre la estructura: S_n_m

S = paso
n = número de la sección (número correlativo)
m = número del paso dentro de la sección (número correlativo)

Nomenclatura IEC (identificador)

Un identificador es una secuencia de letras, números y guiones bajos que debe empezar por una letra o un guión bajo (p. ej., el nombre de un tipo de módulo de función, una instancia, una variable o una sección). Se pueden utilizar las letras propias de los juegos de caracteres nacionales (p. ej., ö, ü, é, ò), excepto en los nombres de proyecto y de DFB.

Los guiones bajos son significativos en los identificadores, p. ej., "A_BCD" y "AB_CD" se interpretan como dos identificadores distintos. No se permiten varios guiones bajos iniciales ni seguidos.

Los identificadores no pueden contener espacios en blanco. Las mayúsculas o minúsculas no son significativas, p. ej., "ABCD" y "abcd" se interpretan como el mismo identificador.

Los identificadores no pueden ser palabras clave.

O

Operador	Un operador es un símbolo para una operación boolearia o aritmética que hay que realizar.
Operando	Un operando es un literal, una variable, una llamada de función o una expresión.

P

Palabras clave	Las palabras clave son combinaciones únicas de caracteres que se utilizan como elementos sintácticos especiales tal como se define en el anexo B de la norma IEC 1131-3. Todas las palabras clave que se utilizan en la norma IEC 1131-3 y, por tanto, en Concept, aparecen enumeradas en el anexo C de la norma IEC 1131-3. Las palabras clave de esta lista no se deberán utilizar para ningún otro fin, p. ej., como nombres de variables, de secciones, de instancias, etc.
Palabras de entrada (referencias 3x)	Una palabra de entrada contiene información que procede de una fuente externa y se representa mediante un número de 16 bits. Un registro 3x también puede contener 16 bits de entrada consecutivos cargados en el registro en formato binario o BCD (decimal codificado en binario). Nota: La x que aparece detrás de la primera cifra del tipo de referencia representa un lugar de almacenamiento de cinco posiciones en la memoria de datos del usuario, por ejemplo, la referencia 300201 significa una palabra de entrada de 16 bits en la dirección 201 de la memoria de señal.
Palabras de salida/marca (referencias 4x)	Una palabra de salida/marca puede utilizarse para almacenar datos numéricos (binarios o decimales) en la memoria de señal, o también para enviar los datos de la CPU a una unidad de salida del sistema de control. Nota: La x que aparece detrás de la primera cifra del tipo de referencia representa un lugar de almacenamiento de cinco posiciones en la memoria de datos del usuario, por ejemplo, la referencia 400201 significa una palabra de salida o marca de 16 bits en la dirección 201 de la memoria de señal.
Parámetro actual	Parámetro de entrada/salida conectado actualmente.
Parámetro de entrada (entrada)	Transmite el argumento correspondiente al ejecutar un FFB.

Parámetro de salida (salida)	Parámetro con el que se devuelve el resultado o los resultados de la evaluación de un FFB.
Parámetros formales	Parámetros de entrada/salida que se utilizan dentro de la lógica de un FFB y se ejecutan como entradas/salidas desde el FFB.
Participante	Un participante es una célula de programación en una red LL984. Una célula o participante estará compuesto por una matriz del tipo 7x11, es decir, 7 filas para 11 elementos.
Participante de red	Un participante es un dispositivo con una dirección (1 a 64) en la red Modbus-Plus.
Participante local de red	El participante local es aquél que se está configurando en este momento.
Paso	Elemento de lenguaje SFC: Situación en la que el comportamiento de un programa sigue, en lo que respecta a sus entradas y salidas, aquellas operaciones que se han definido mediante las correspondientes acciones del paso.
Paso inicial	Primer paso de una cadena secuencial. En todas las cadenas secuenciales debe haber definido un paso inicial. La cadena secuencial se inicia con el paso inicial la primera vez que éste se ejecuta.
PLC	Autómata programable (PLC)
Portapapeles	El portapapeles es una memoria temporal para objetos cortados o copiados. Estos objetos se pueden pegar en secciones. Cada vez que se vuelva a cortar o copiar, se sobrescribirá el contenido anterior del portapapeles.
Procesador Peer	El procesador Peer procesa los ciclos de token y el flujo de datos entre la red Modbus-Plus y la lógica de usuario PLC.
Programa	Máxima unidad organizativa del programa. Un programa se cierra y se carga en un único PLC.
Programar sistema redundante (Hot Standby)	Un sistema redundante está compuesto por dos equipos de PLC con idéntica configuración que se comunican entre sí mediante procesadores redundantes. En caso de que falle el PLC primario, el PLC secundario asume el control del mando. En condiciones normales, el PLC secundario no asume ninguna función de control, sino que comprueba la información de estado para detectar fallos.

Proyecto	<p>Denominación general del nivel superior de una estructura en árbol de un software que determina el nombre de proyecto de orden superior de una aplicación PLC. Después de establecer el nombre del proyecto podrá guardar la configuración del sistema y el programa de control con este nombre. Todos los datos que se originan durante la creación de la configuración y del programa forman parte de este proyecto de nivel superior para la tarea de automatización específica.</p> <p>Denominación general del conjunto completo de la información de programación y planificación en la base de datos del proyecto que representa el código fuente que describe la automatización de una instalación.</p>
Puente	<p>Un puente es un dispositivo que conecta redes. Permite la comunicación entre usuarios de las dos redes. Cada red tiene su propia secuencia de rotación de token. El token no se transmite a través de puentes.</p>

R

REAL	<p>REAL representa el tipo de datos "número con coma flotante". La entrada se efectúa como literal real o como literal real con exponente. La longitud de los elementos de datos es de 32 bits. El rango de valores para variables de este tipo de datos va de +3.402823E+38 a -3.402823E+38.</p>
-------------	---

Nota: Dependiendo del tipo de procesador matemático de la CPU no será posible representar distintas áreas dentro de este rango de valores permitido. Esto será válido para valores que tiendan a CERO y para valores que tiendan a INFINITO. En tales casos, en la modalidad de animación no se mostrará un valor numérico, sino NAN (**Not A Number**) o INF (**INFinite** (infinito)).

Red	<p>Una red es la interconexión de dispositivos en un circuito de información común que se comunican entre sí mediante un mismo protocolo.</p>
Red descentral (DIO)	<p>La programación descentral en la red Modbus Plus permite alcanzar el máximo rendimiento en la transferencia de datos sin necesidad de requisitos especiales en los vínculos. La programación de una red descentral es muy sencilla. Para instalar la red no hace falta crear ninguna lógica adicional de esquema de contactos. Mediante las entradas correspondientes en el procesador Peer Cop se cumplen todos los requisitos para la transferencia de datos.</p>

Referencia

Cada dirección directa es una referencia que comienza por una clave que indica si se trata de una entrada o una salida o de un bit o una palabra. Las referencias que empiezan con la clave 6 representan registros de la memoria extendida de la memoria de señal.

Rango 0x = bits de salida/marca

Rango 1x = bits de entrada

Rango 3x = palabras de entrada

Rango 4x = palabras de salida/marca

Rango 6x = registros en la memoria extendida

Nota: La x que aparece detrás de la primera cifra de cada tipo de referencia representa un lugar de almacenamiento de cinco posiciones en la memoria de datos del usuario, por ejemplo, la referencia 400201 significa una palabra de salida o marca de 16 bits en la dirección 201 de la memoria de señal.

Registro en la memoria extendida (referencia 6x)

Las referencias 6x son palabras de marca en la memoria extendida del PLC. Sólo se pueden utilizar en programas de usuario LL984 y para utilizar una CPU 213 04 o CPU 424 02.

Representación directa

Método para la representación de variables en el programa PLC del que se puede derivar directamente la asignación del lugar de almacenamiento lógico (e indirectamente el lugar de almacenamiento físico).

RIO (Remote I/O)

La E/S remota indica un lugar físico de los dispositivos de control de puntos de E/S en relación con el procesador que los controla. Las entradas/salidas remotas están conectadas al dispositivo de control a través de un cable de comunicación.

S

Salto

Elemento del lenguaje SFC. Los saltos se utilizan para saltar áreas de la cadena secuencial.

Sección	<p>Una sección se puede utilizar, por ejemplo, para describir el funcionamiento de una unidad tecnológica, como un motor.</p> <p>Un programa o DFB está compuesto de una o más secciones. Las secciones se pueden programar con los lenguajes de programación IEC FBD y SFC. Dentro de una sección sólo se puede utilizar uno de los lenguajes de programación mencionados.</p> <p>Cada sección tiene su propia ventana de documento en Concept. Para tener una mejor visión de conjunto, es mejor dividir una sección grande en varias más pequeñas. Para desplazarse dentro de la sección se utiliza la barra de desplazamiento.</p>
Secuencia de bits	Elemento de datos compuesto por uno o varios bits.
Símbolo (icono)	Representación gráfica de distintos objetos en Windows, p. ej., unidades de disco, programas de aplicación y ventanas de documento.
SY/MAX	En los dispositivos de control Quantum, Concept incluye la preparación de la asignación de E/S para módulos de E/S SY/MAX para el control RIO mediante el PLC Quantum. El bastidor remoto SY/MAX tiene un adaptador remoto de E/S en el slot 1 que se comunica mediante un sistema de E/S Modicon S908 R. Los módulos de E/S SY/MAX se incluyen en la asignación de E/S de la configuración de Concept para marcarlos e integrarlos.

T

Texto estructurado (ST)	ST es un lenguaje de texto conforme a IEC 1131, en el que las operaciones, como, por ejemplo, las llamadas de módulos de función y funciones, la ejecución condicionada de instrucciones, la repetición de instrucciones, etc., se representan mediante instrucciones.
TIME	TIME representa el tipo de datos "duración". La entrada se realiza como literal de duración. La longitud de los elementos de datos es de 32 bits. El rango de valores para las variables de este tipo de datos va de 0 a $2^{\text{exp}(32)-1}$. La unidad de medida para el tipo de datos TIME es 1 ms.
Tipo de datos derivado	Los tipos de datos derivados son tipos de datos que se han derivado de los tipos de datos elementales y/o de otros tipos de datos derivados. La definición de los tipos de datos derivados se realiza en el editor de tipos de datos de Concept. Se distingue entre tipos de datos globales y tipos de datos locales.

Tipo de datos genérico	Tipo de datos que sustituye otros tipos de datos.
Tipo de módulo de función	Elemento de lenguaje compuesto por: 1) la definición de una estructura de datos, dividida en variables de entrada, de salida e internas; 2) un conjunto de operaciones que se realizan con los elementos de la estructura de datos cuando se ejecuta una instancia del tipo de módulo de función. Este conjunto de operaciones puede estar formulado en uno de los lenguajes IEC (tipo de DFB) o en "C" (tipo de EFB). Un tipo de módulo de función se puede instanciar (ejecutar) varias veces.
Tipos de datos	<p>La vista general muestra la jerarquía de los tipos de datos tal como se utilizan en las entradas y salidas de funciones y módulos de función. Los tipos de datos genéricos se identifican mediante el prefijo "ANY".</p> <ul style="list-style-type: none">• ANY_ELEM<ul style="list-style-type: none">• ANY_NUM<ul style="list-style-type: none">• ANY_REAL (REAL)• ANY_INT (DINT, INT, UDINT, UINT)• ANY_BIT (BOOL, BYTE, WORD)• TIME• Tipos de datos de sistema (expansiones IEC)• Derivado (de los tipos de datos 'ANY')
Tipos de datos derivados globales	Los tipos de datos derivados globales están disponibles en todos los proyectos de Concept y se encuentran en el directorio DFB, justo debajo del directorio Concept.
Tipos de datos derivados locales	Los tipos de datos derivados locales sólo están disponibles en un único proyecto de Concept y sus DFB locales, y se guardan en el directorio DFB bajo el directorio del proyecto.
Token	La red "Token" controla la propiedad temporal del derecho de transferencia por parte de un único participante. El Token recorre los participantes en una secuencia de direcciones circular (ascendente). Todos los participantes siguen el recorrido del token y pueden recibir todos los datos posibles enviados.
Traffic Cop	Traffic Cop es una lista de componentes que se genera a partir de la lista de componentes del usuario. La lista Traffic Cop se gestiona en el PLC y contiene, además de la lista de componentes del usuario, información de estado de los módulos y las estaciones de E/S, por ejemplo.
Transición	Condición necesaria para que el control pase de uno o más pasos predecesores a uno o más pasos sucesores a lo largo de una conexión direccional.

U

- UDEFB** Funciones/módulos de función elementales definidos por el usuario
Funciones o módulos de función creados con el lenguaje de programación C y disponibles en las bibliotecas de Concept.
- UDINT** UDINT representa el tipo de datos "número entero de longitud doble sin signo (unsigned double integer)". La entrada se efectúa en forma de literal entero, literal de base 2, literal de base 8 o literal de base 16. La longitud de los elementos de datos es de 32 bits. El rango de valores para las variables de este tipo de datos va de 0 a $2^{\text{exp}(32)}-1$.
- UINT** UINT representa el tipo de datos "número entero sin signo (unsigned integer)". La entrada se efectúa en forma de literal entero, literal de base 2, literal de base 8 o literal de base 16. La longitud de los elementos de datos es de 16 bits. El rango de valores para las variables de este tipo de datos va de 0 a $(2^{\text{exp}} 16)-1$.
- Unidad organizativa del programa** Función, módulo de función o programa. Este término se puede referir tanto a un tipo o a una instancia.
- Unlocated Variable** A las Unlocated Variables no se les asigna ninguna dirección de memoria de señal. Por lo tanto, tampoco ocupan ninguna dirección de memoria de señal. El sistema almacena el valor de estas variables, que se puede modificar con el editor de datos de referencia, de forma interna. A estas variables sólo se accede mediante su nombre simbólico.
- Las señales que no necesitan acceso a la periferia, p. ej., resultados intermedios, marcas de sistema, etc., deberían declararse preferiblemente como Unlocated Variables.

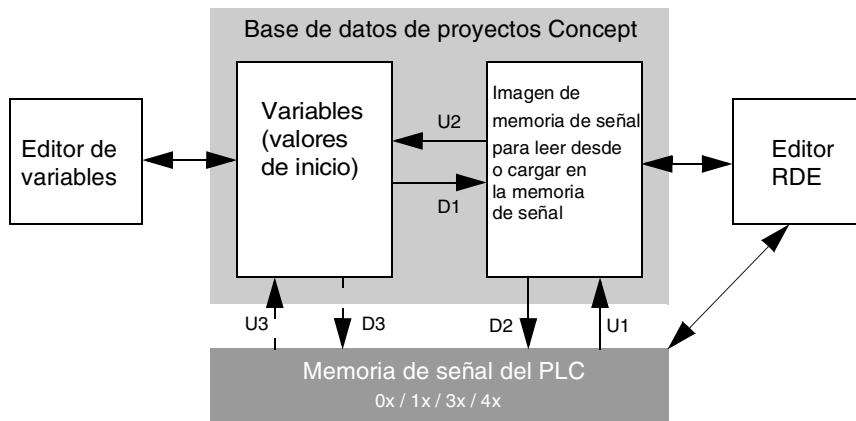
V

- Valor inicial** Valor asignado a una variable durante el inicio del programa. La asignación del valor se realiza en forma de literal.

Variables	<p>Las variables sirven para el intercambio de datos dentro de una misma sección, entre secciones distintas y entre el programa y el PLC.</p> <p>Las variables se componen como mínimo de un nombre de variable y un tipo de datos.</p> <p>Si se asigna una dirección directa (referencia) a una variable, se habla de una Located Variable. Si a una variable no se le asigna ninguna dirección directa, se habla de una Unlocated Variable. Si a la variable se le asigna un tipo de datos derivado, se habla de una variable de elementos múltiples.</p> <p>Además, también existen constantes y literales.</p>
Variables de campo	<p>Variables a las que se les ha asignado un tipo de datos derivado con ayuda de la palabra clave ARRAY (campo). Un campo es una colección de elementos de datos con el mismo tipo de datos.</p>
Variables de elementos múltiples	<p>Variables a las que se ha asignado un tipo de datos derivado definido con STRUCT o ARRAY.</p> <p>Se distingue entre variables de campo y variables estructuradas.</p>
Variables estructuradas	<p>Variables a las que se ha asignado un tipo de dato derivado definido con STRUCT (estructura).</p> <p>Una estructura es una colección de elementos de datos de distintos tipos en general (tipos de datos elementales y/o tipos de datos derivados).</p>
Ventana activa	<p>Ventana seleccionada en este momento. Sólo puede haber una ventana activa en un momento determinado. Cuando se activa una ventana, cambia el color de su barra de título para distinguirla de las demás ventanas. Las ventanas no seleccionadas están inactivas.</p>
Ventana de documento	<p>Ventana dentro de una ventana de aplicación. Dentro de una ventana de aplicación puede haber abiertas varias ventanas de documento al mismo tiempo. No obstante, sólo puede haber una ventana de documento activa. Las ventanas de documento en Concept son, por ejemplo, las secciones, la ventana de mensajes, el editor de datos de referencia y la configuración del PLC.</p>
Ventana de la aplicación	<p>Ventana que contiene el espacio de trabajo, la barra de menús y la barra de herramientas para el programa de la aplicación. El nombre de la aplicación aparece en la barra de título. Una ventana de aplicación puede contener varias ventanas de documento. En Concept, la ventana de aplicación corresponde a un proyecto.</p>
Vertical	<p>Vertical significa que la página del texto impreso es más alta que ancha.</p>

Vista general de la memoria de señal al leer la dotación y al cargar

Vista general:

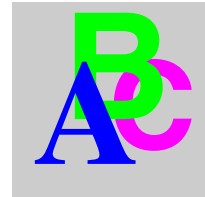
**W****WORD**

WORD representa el tipo de datos "secuencia de bits 16". La entrada se efectúa en forma de literal de base 2, literal de base 8 o literal de base 16. La longitud de los elementos de datos es de 16 bits. A este tipo de datos no se le puede asignar un rango de valores numérico.

Z**Zoom DX**

Esta propiedad permite conectarse a un objeto de programación para observar y, en caso necesario, modificar los valores de los datos.

Índice



A

ABS_***, 27
ACOS_REAL, 29
ADD_***, 31
Adición, 31
AND_***, 35
Arcc coseno en la medida de arco, 29
Arco seno en la medida de arco, 37
Arco tangencial en la medida de arco, 39
Arithmetic
 ADD_***, 31
 DIV_***, 63
 MOD_***, 105
 MOVE, 107
 MUL_***, 109
 SUB_***, 151
 TIME_DIV_***, 155
 TIME_MUL_***, 157
Asignación, 107
ASIN_REAL, 37
ATAN_REAL, 39

B

Bistable
 RS, 137
 SR, 149
Bloque de función
 Parametrización, 19, 20
Bloque de función biestable, restablecer
dominante, 137

Bloque de función biestable, ubicar
dominante, 149
BOOL_TO_***, 41
BYTE_TO_***, 43

C

CEI
 ABS_***, 27
 ACOS_REAL, 29
 ADD_***, 31
 AND_***, 35
 ASIN_REAL, 37
 ATAN_REAL, 39
 BOOL_TO_***, 41
 BYTE_TO_***, 43
 COS_REAL, 47
 CTD, 49
 CTU, 51
 CTUD, 53
 DINT_EXPT_REAL, 57
 DINT_TO_***, 59
 DIV_***, 63
 EQ_***, 67
 EXP_REAL, 71
 F_TRIG, 73
 GE_***, 75
 GT_***, 77
 INT_EXPT_REAL, 79
 INT_TO_***, 81
 LE_***, 85
 LIMIT_***, 87

CEI

LN_REAL, 91
LOG_REAL, 93
LT_***, 95
MAX_***, 97
MIN_***, 101
MOD_***, 105
MOVE, 107
MUL_***, 109
MUX_***, 111
NE_***, 115
NOT_***, 117
OR_***, 119
R_TRIG, 121
REAL_EXPT_REAL, 123
REAL_TO_***, 125
REAL_TRUNC_***, 129
ROL_***, 133
ROR_***, 135
RS, 137
SEL, 139
SHL_***, 141
SHR_***, 143
SIN_REAL, 145
SQRT_REAL, 147
SR, 149
SUB_***, 151
TAN_REAL, 153
TIME_DIV_***, 155
TIME_MUL_***, 157
TIME_TO_***, 159
TOF, 163
TON, 167
TP, 171
UDINT_EXPT_REAL, 175
UDINT_TO_***, 177
UINT_EXPT_REAL, 179
UINT_TO_***, 181
WORD_TO_***, 183
XOR_***, 187

Comparison

EQ_***, 67
GE_***, 75
GT_***, 77
LE_***, 85
LT_***, 95
NE_***, 115
Contador hacia abajo, 49
Contador hacia arriba, 51
Contador hacia arriba/abajo, 53
Conversión de tipo, 41, 43, 59, 81, 125, 129, 159, 177, 181, 183
Converter
 BOOL_TO_***, 41
 BYTE_TO_***, 43
 DINT_TO_***, 59
 INT_TO_***, 81
 REAL_TO_***, 125
 REAL_TRUNC_***, 129
 TIME_TO_***, 159
 UDINT_TO_***, 177
 UINT_TO_***, 181
 WORD_TO_***, 183
COS_REAL, 47
Coseno, 47
Counter
 CTD, 49
 CTU, 51
 CTUD, 53
CTD, 49
CTU, 51
CTUD, 53

D

Desigual, 115
Desplazar a la derecha, 143
Desplazar a la izquierda, 141
Detección de flancos ascendentes, 121
Detección de flancos descendentes, 73
DINT_EXPT_REAL, 57
DINT_TO_***, 59
DIV_***, 63
División, 63
División de valores temporales, 155

E

Edge detection
 F_TRIG, 73
 R_TRIG, 121
EQ_***, 67
EXP_REAL, 71
Exponencialización, 57, 79, 123, 175, 179

F

F_TRIG, 73
Formación del valor absoluto, 27
Función
 Parametrización, 19, 20
Función de exponencial, 71
Función O, 119
Función O exclusivo, 187
Función Y, 35

G

GE_***, 75
GT_***, 77

I

Igual, 67
Impulso, 171
INT_EXPT_REAL, 79
INT_TO_***, 81

L

LE_***, 85
LIMIT_***, 87
Limitación, 87
LN_REAL, 91
LOG_REAL, 93
Logaritmo natural, 91
Logaritmo respecto a la base 10, 93

Logic

AND_***, 35
NOT_***, 117
OR_***, 119
ROL_***, 133
ROR_***, 135
SHL_***, 141
SHR_***, 143
XOR_***, 187
LT_***, 95

M

MAX_***, 97
Mayor, 77
Mayor igual a, 75
Menor, 95
Menor igual a, 85
MIN_***, 101
MOD_***, 105
Módulo, 105
MOVE, 107
MUL_***, 109
Multiplexor, 111
Multiplicación, 109
Multiplicación de valores temporales, 157
MUX_***, 111

N

NE_***, 115
Negación, 117
NOT_***, 117

Numerical

ABS_***, 27
ACOS_REAL, 29
ASIN_REAL, 37
ATAN_REAL, 39
COS_REAL, 47
DINT_EXPT_REAL, 57
EXP_REAL, 71
INT_EXPT_REAL, 79
LN_REAL, 91
LOG_REAL, 93
REAL_EXPT_REAL, 123
SIN_REAL, 145
SQRT_REAL, 147
TAN_REAL, 153
UDINT_EXPT_REAL, 175
UINT_EXPT_REAL, 179

O

OR_***, 119

P

Parametrización, 19, 20

R

R_TRIG, 121
Raíz cuadrada, 147
REAL_EXPT_REAL, 123
REAL_TO_***, 125
REAL_TRUNC_***, 129
Retardo de conexión, 167
Retardo de desconexión, 163
ROL_***, 133
ROR_***, 135
Rotar a la derecha, 135
Rotar a la izquierda, 133
RS, 137

S

SEL, 139
Selección binaria, 139
Selección de valor máximo, 97
Selección de valor mínimo, 101
Selection
 LIMIT_***, 87
 MAX_***, 97
 MIN_***, 101
 MUX_***, 111
 SEL, 139
Seno, 145
SHL_***, 141
SHR_***, 143
SIN_REAL, 145
SQRT_REAL, 147
SR, 149
SUB_***, 151
Sustracción, 151

T

TAN_REAL, 153
Tangencial, 153
TIME_DIV_***, 155
TIME_MUL_***, 157
TIME_TO_***, 159
Timer
 TOF, 163
 TON, 167
 TP, 171
TOF, 163
TON, 167
TP, 171

U

UDINT_EXPT_REAL, 175
UDINT_TO_***, 177
UINT_EXPT_REAL, 179
UINT_TO_***, 181

W

WORD_TO_***, 183

X

XOR_***, 187

