



# Comunicación

Para sistemas de control que  
se programan con SILworX



SAFETY  
NONSTOP



Todos los productos de HIMA nombrados en el presente manual son marcas registradas. Salvo donde se indique lo contrario, esto se aplicará también a los demás fabricantes aquí citados y a sus productos.

Tras haber sido redactadas concienzudamente, las notas y las especificaciones técnicas ofrecidas en este manual han sido compiladas bajo estrictos controles de calidad. En caso de dudas, consulte directamente a HIMA. HIMA le agradecerá que nos haga saber su opinión acerca de p. ej. qué más información debería incluirse en el manual.

Reservado el derecho a modificaciones técnicas. HIMA se reserva asimismo el derecho de actualizar el material escrito sin previo aviso.

Hallará más información en la documentación recogida en el DVD de HIMA y en nuestros sitios web <http://www.hima.com>.

© Copyright 2011, HIMA Paul Hildebrandt GmbH

Todos los derechos reservados.

## Contacto

La dirección de HIMA es:

HIMA Paul Hildebrandt GmbH

Apdo. Postal/Postfach 1261

D-68777 Brühl

Tel.: +49 6202 709-0

Fax: +49 6202 709-107

E-Mail: [info@hima.com](mailto:info@hima.com)

Índice de revisión	Modificaciones	Tipo de modificación	
		técnica	redaccional
4.00	Adaptación a HIMax V4/SILworX V4 Edición en español (traducción)		

## Índice de contenidos

<b>1</b>	<b>Introducción .....</b>	<b>11</b>
1.1	Estructuración y uso del manual .....	11
1.2	Destinatarios .....	12
1.3	Convenciones de representación .....	12
1.3.1	Notas de seguridad .....	12
1.3.2	Notas de uso .....	13
<b>2</b>	<b>Seguridad .....</b>	<b>14</b>
2.1.1	Condiciones de uso .....	14
<b>2.2</b>	<b>Peligros remanentes.....</b>	<b>17</b>
<b>2.3</b>	<b>Medidas de seguridad .....</b>	<b>17</b>
<b>2.4</b>	<b>Información para emergencias .....</b>	<b>17</b>
<b>3</b>	<b>Descripción del producto .....</b>	<b>18</b>
<b>3.1</b>	<b>Protocolo con función orientada a la seguridad (safeethernet).....</b>	<b>18</b>
<b>3.2</b>	<b>Protocolos estándar .....</b>	<b>19</b>
<b>3.3</b>	<b>Redundancia .....</b>	<b>21</b>
<b>3.4</b>	<b>Registro y activación de los protocolos .....</b>	<b>22</b>
<b>3.5</b>	<b>Interfaces Ethernet .....</b>	<b>23</b>
3.5.1	Propiedades de las interfaces Ethernet .....	23
3.5.2	Configuración de las interfaces Ethernet .....	25
3.5.3	Puertos de red utilizados para comunicación Ethernet.....	30
<b>3.6</b>	<b>Interfaces de bus de campo.....</b>	<b>31</b>
3.6.1	Instalación .....	31
3.6.2	Registro y activación .....	33
3.6.3	Asignación de pins de las conexiones D-Sub .....	33
<b>4</b>	<b>safeethernet .....</b>	<b>36</b>
<b>4.1</b>	<b>Qué es safeethernet.....</b>	<b>37</b>
<b>4.2</b>	<b>Configuración de una conexión safeethernet redundante .....</b>	<b>40</b>
4.2.1	Conexión de variables de proceso .....	41
4.2.2	Prueba de la comunicación safeethernet .....	42
<b>4.3</b>	<b>Editor de safeethernet .....</b>	<b>44</b>
<b>4.4</b>	<b>Vista en detalle del editor de safeethernet .....</b>	<b>46</b>
4.4.1	Ficha: Variables de sistema .....	46
4.4.2	Ficha: Fragment Definitions .....	49
<b>4.5</b>	<b>Posibles conexiones safeethernet .....</b>	<b>49</b>
4.5.1	Conexión safeethernet en modo mono (canal 1) .....	49
4.5.2	Conexión safeethernet redundante (canal 1 y canal 2) .....	49
4.5.3	Combinaciones permitidas .....	50
<b>4.6</b>	<b>Parámetros de safeethernet.....</b>	<b>52</b>
4.6.1	Tiempo máximo de ciclo (mínimo tiempo de Watchdog) del sistema de control HIMax .....	52
4.6.2	Tiempo máximo de ciclo del sistema de control HIMatrix .....	53
4.6.3	TimeOut de recepción .....	53
4.6.4	Tiempo de respuesta .....	53
4.6.5	Sync/Async .....	54

4.6.6	ResendTMO .....	54
4.6.7	Acknowledge Timeout .....	55
4.6.8	Production Rate.....	55
4.6.9	Queue.....	55
<b>4.7</b>	<b>Máximo tiempo de reacción para safeethernet .....</b>	<b>56</b>
4.7.1	Cálculo del tiempo máximo de reacción de dos sistemas de control HIMax.....	57
4.7.2	Cálculo del tiempo máximo de reacción en conjunción con un sistema de control HIMatrix .....	57
4.7.3	Cálculo del tiempo máximo de reacción con dos sistemas de control HIMatrix o I/Os remotas.....	58
4.7.4	Cálculo del tiempo máximo de reacción con dos HIMax y un sistema de control HIMatrix .....	58
4.7.5	Cálculo del tiempo máximo de reacción de dos sistemas de control HIMatrix.....	59
4.7.6	Cálculo del tiempo máximo de reacción con dos I/Os remotas.....	60
4.7.7	Cálculo del tiempo máximo de reacción con dos sistemas HIMatrix y un HIMax ..	60
4.7.8	Perfiles safeethernet .....	61
4.7.9	Perfil I (Fast & Cleanroom) .....	62
4.7.10	Perfil II (Fast & Noisy) .....	62
4.7.11	Perfil III (Medium & Cleanroom) .....	63
4.7.12	Perfil IV (Medium & Noisy) .....	63
4.7.13	Perfil V (Slow & Cleanroom).....	64
4.7.14	Perfil VI (Slow & Noisy) .....	64
<b>4.8</b>	<b>Comunicación trascendente a todo el proyecto .....</b>	<b>66</b>
4.8.1	Variantes de la comunicación trascendente a todo el proyecto .....	67
<b>4.9</b>	<b>Comunicación trascendente a todo el proyecto entre SILworX y ELOP II Factory.....</b>	<b>69</b>
4.9.1	Configuración de HIMax en el proyecto SILworX.....	69
4.9.2	Configuración de HIMatrix en ELOP II Factory .....	73
<b>4.10</b>	<b>Panel de control (safeethernet).....</b>	<b>75</b>
4.10.1	Recuadro de lectura (conexión safeethernet) .....	76
<b>5</b>	<b>PROFINET-IO .....</b>	<b>77</b>
<b>5.1</b>	<b>Bloques funcionales de PROFINET-IO .....</b>	<b>77</b>
<b>5.2</b>	<b>Control del estado de consumidor/proveedor (IOxS) .....</b>	<b>78</b>
5.2.1	Variables de control en el controlador HIMA.....	78
5.2.2	Variables de control en el dispositivo de DO de HIMA.....	78
5.2.3	Variables de control en el dispositivo de DI de HIMA.....	78
<b>5.3</b>	<b>PROFIsafe .....</b>	<b>79</b>
5.3.1	Byte de estado y byte de control de PROFIsafe .....	80
5.3.2	F_WD_Time (tiempo de WatchDog de PROFIsafe).....	80
5.3.3	SFRT (Safety Function Response Time).....	82
<b>5.4</b>	<b>Obligaciones para el funcionamiento seguro de PROFIsafe .....</b>	<b>84</b>
5.4.1	Direccionamiento.....	84
5.4.2	Aspectos de la red.....	85
<b>5.5</b>	<b>Controlador PROFINET IO HIMA y PROFIsafe F-Host .....</b>	<b>86</b>
<b>5.6</b>	<b>Ejemplo de PROFINET-IO/PROFIsafe .....</b>	<b>87</b>
5.6.1	Creación de un controlador PROFINET IO HIMA en SILworX.....	87
<b>5.7</b>	<b>Funciones del menú del controlador PROFINET IO.....</b>	<b>92</b>
5.7.1	Ejemplo de estructura en árbol del controlador PROFINET IO .....	92
5.7.2	Controlador PROFINET IO.....	92

5.7.3	Dispositivo PROFINET IO (en el controlador) .....	94
5.7.4	Módulo DAP (Device Access Point Module).....	94
5.7.5	Módulos PROFINET IO de entrada/salida.....	96
5.7.6	Entrada de submódulo .....	97
5.7.7	Salida de submódulo .....	102
5.7.8	Entradas y salidas de submódulo .....	104
5.7.9	Relación de aplicación (propiedades).....	106
5.7.10	Alarm CR (propiedades) .....	107
5.7.11	Input CR (propiedades).....	108
<b>5.8</b>	<b>Dispositivo PROFINET IO HIMA .....</b>	<b>111</b>
<b>5.9</b>	<b>Sistema necesario .....</b>	<b>111</b>
<b>5.10</b>	<b>Ejemplo de PROFINET-IO/PROFIsafe .....</b>	<b>112</b>
5.10.1	Configuración del dispositivo PROFINET IO en SILworX.....	112
<b>5.11</b>	<b>Funciones del menú del dispositivo PROFINET IO .....</b>	<b>115</b>
5.11.1	Función de menú Propiedades .....	115
5.11.2	Módulos PROFINET IO HIMA .....	116
5.11.3	Módulos PROFIsafe HIMA.....	117
5.11.4	Módulo PROFINET-IO/ PROFIsafe .....	118
<b>6</b>	<b>PROFIBUS DP .....</b>	<b>121</b>
<b>6.1</b>	<b>PROFIBUS DP Master de HIMA .....</b>	<b>122</b>
6.1.1	Creación de un PROFIBUS DP Master de HIMA .....	122
<b>6.2</b>	<b>Ejemplo de PROFIBUS DP .....</b>	<b>123</b>
6.2.1	Configuración de PROFIBUS DP Slave .....	123
6.2.2	Configuración de PROFIBUS DP Master .....	125
<b>6.3</b>	<b>Funciones de menú de PROFIBUS DP Master .....</b>	<b>132</b>
6.3.1	Edit.....	132
6.3.2	Función de menú Propiedades .....	133
<b>6.4</b>	<b>Procedimiento de acceso de PROFIBUS-DP.....</b>	<b>137</b>
6.4.1	Protocolo Master/Slave .....	137
6.4.2	Protocolo de Token .....	137
6.4.3	Tiempo de rotación permitido del Token (Ttr).....	137
6.4.4	Cálculo del tiempo de rotación del Token Ttr .....	138
<b>6.5</b>	<b>Ciclo isócrono PROFIBUS DP (a partir de V2 de DP) .....</b>	<b>140</b>
6.5.1	Isochron Mode (a partir de V2 de DP) .....	141
6.5.2	Isochron Mode Sync (a partir de V2 de DP) .....	141
6.5.3	Isochrone Mode Freeze (a partir de V2 de DP) .....	141
<b>6.6</b>	<b>Funciones de menú de PROFIBUS DP Slave (en el Master) .....</b>	<b>142</b>
6.6.1	Creación de un PROFIBUS DP Slave (en el Master) .....	142
6.6.2	Edit.....	142
6.6.3	Propiedades.....	143
<b>6.7</b>	<b>Importación de archivo GSD.....</b>	<b>148</b>
<b>6.8</b>	<b>Edición de los parámetros del usuario .....</b>	<b>149</b>
<b>6.9</b>	<b>Bloques funcionales de PROFIBUS .....</b>	<b>151</b>
6.9.1	Bloque funcional MSTAT .....	152
6.9.2	Bloque funcional RALRM .....	155
6.9.3	Bloque funcional RDIAG .....	159
6.9.4	Bloque funcional RDREC .....	163
6.9.5	Bloque funcional SLACT .....	166
6.9.6	Bloque funcional WRREC .....	169

<b>6.10</b>	<b>Bloques funcionales auxiliares de PROFIBUS .....</b>	<b>172</b>
6.10.1	Bloque funcional auxiliar ACTIVE.....	172
6.10.2	Bloque funcional auxiliar ALARM .....	173
6.10.3	Bloque funcional auxiliar DEID .....	174
6.10.4	Bloque funcional auxiliar ID .....	175
6.10.5	Bloque funcional auxiliar NSLOT .....	176
6.10.6	Bloque funcional auxiliar SLOT .....	176
6.10.7	Bloque funcional auxiliar STDDIAG.....	177
<b>6.11</b>	<b>Códigos de error de los bloques funcionales .....</b>	<b>179</b>
<b>6.12</b>	<b>Panel de control (PROFIBUS DP Master).....</b>	<b>180</b>
6.12.1	Menú contextual (PROFIBUS Master) .....	180
6.12.2	Menú contextual (PROFIBUS Slave) .....	180
6.12.3	Recuadro de lectura (PROFIBUS Master) .....	181
6.12.4	Estado del PROFIBUS DP Master .....	182
6.12.5	Comportamiento del PROFIBUS DP Master .....	182
6.12.6	Función del LED FBx en el PROFIBUS DP Master .....	183
6.12.7	Función del LED FAULT en el PROFIBUS DP Master (sólo HIMax) .....	183
<b>6.13</b>	<b>HIMA PROFIBUS DP Slave .....</b>	<b>184</b>
6.13.1	Creación de un PROFIBUS DP Slave de HIMA.....	184
<b>6.14</b>	<b>Funciones de menú de PROFIBUS DP Slave.....</b>	<b>185</b>
6.14.1	Edit .....	185
6.14.2	Propiedades .....	187
<b>6.15</b>	<b>Panel de control (PROFIBUS DP Slave) .....</b>	<b>189</b>
6.15.1	Recuadro de lectura (PROFIBUS Slave) .....	189
<b>6.16</b>	<b>Función del LED FBx en el PROFIBUS DP Slave .....</b>	<b>190</b>
6.17	Función del LED FAULT en el PROFIBUS DP Slave (sólo HIMax) .....	190
<b>7</b>	<b>Modbus .....</b>	<b>191</b>
<b>7.1</b>	<b>Topología de bus RS485.....</b>	<b>191</b>
7.1.1	Asignación de bornes de H 7506 .....	192
7.1.2	Cable de bus .....	192
7.1.3	Características de transmisión por RS485 .....	193
<b>7.2</b>	<b>HIMA Modbus Master .....</b>	<b>194</b>
7.2.1	Ejemplo de Modbus .....	195
7.2.2	Ejemplo de direccionamiento alternativo de bit/registro .....	199
7.2.3	Funciones de menú de Modbus Master de HIMA .....	201
7.2.4	Códigos de función Modbus del Master .....	204
7.2.5	Slaves Ethernet (Slaves TCP/UDP) .....	211
7.2.6	Modbus Gateway (TCP/UDP Gateway) .....	214
7.2.7	Modbus serie .....	217
7.2.8	Panel de control (Modbus Master) .....	219
7.2.9	Panel de control (Modbus Master -> Slave) .....	220
7.2.10	Función del LED FBx en el Modbus Master .....	221
7.2.11	Función del LED FAULT en el Modbus Master (sólo HIMax) .....	221
<b>7.3</b>	<b>Modbus Slave HIMA .....</b>	<b>222</b>
7.3.1	Configuración del Modbus TCP Slave.....	223
7.3.2	Configuración del Modbus TCP Slave redundante .....	223
7.3.3	Reglas para el Modbus TCP Slave redundante .....	224
7.3.4	Funciones de menú del set Modbus Slave de HIMA.....	225
7.3.5	Asignación de variables de envío/recepción .....	227
7.3.6	Variables de sistema Modbus Slave Set .....	227

7.3.7	Modbus Slave y Modbus Slave Redundant .....	227
7.3.8	Códigos de función Modbus del Modbus Slave HIMA .....	230
7.3.9	Códigos de función específicos de HIMA .....	232
7.3.10	Direccionamiento Modbus por bits y registros .....	234
7.3.11	Offsets para direccionamiento alternativo Modbus .....	236
7.3.12	Panel de control (Modbus Slave) .....	240
7.3.13	Función del LED FBx en el Modbus Slave .....	242
7.3.14	Función del LED FAULT en el Modbus Slave (sólo HIMax) .....	242
7.3.15	Códigos de error de la conexión TCP/IP Modbus .....	242
<b>8</b>	<b>Send &amp; Receive TCP .....</b>	<b>243</b>
<b>8.1</b>	<b>Sistema necesario .....</b>	<b>243</b>
8.1.1	Creación de un protocolo S&R-TCP .....	243
<b>8.2</b>	<b>Ejemplo: Configuración S&amp;R TCP .....</b>	<b>244</b>
8.2.1	Configuración S&R TCP del sistema de control SIMATIC 300 de Siemens .....	246
8.2.2	Configuración S&R TCP del sistema de control HIMax .....	250
<b>8.3</b>	<b>Funciones de menú del protocolo S&amp;R-TCP .....</b>	<b>252</b>
8.3.1	Edit .....	252
8.3.2	Propiedades .....	252
<b>8.4</b>	<b>Funciones de menú de la conexión TCP .....</b>	<b>254</b>
8.4.1	Edit .....	254
8.4.2	Variables de sistema .....	254
8.4.3	Propiedades .....	255
<b>8.5</b>	<b>Intercambio de datos .....</b>	<b>257</b>
8.5.1	Conexiones TCP .....	257
8.5.2	Intercambio de datos cíclico .....	258
8.5.3	Intercambio de datos acíclico con bloques funcionales .....	258
8.5.4	Intercambio de datos cíclico y acíclico simultáneamente .....	258
8.5.5	Control del flujo .....	259
<b>8.6</b>	<b>Sistemas externos con bytes pad .....</b>	<b>259</b>
<b>8.7</b>	<b>Bloques funcionales S&amp;R-TCP .....</b>	<b>260</b>
8.7.1	TCP_Reset .....	261
8.7.2	TCP_Send .....	264
8.7.3	TCP_Receive .....	267
8.7.4	TCP_ReceiveLine .....	271
8.7.5	TCP_ReceiveVar .....	275
<b>8.8</b>	<b>Panel de control (Send/Receive over TCP) .....</b>	<b>280</b>
8.8.1	Recuadro de lectura de parámetros generales .....	280
8.8.2	Recuadro de lectura de conexiones TCP .....	280
8.8.3	Código de error de la conexión TCP .....	281
8.8.4	Tabla de códigos de error adicionales de los bloques funcionales .....	282
8.8.5	Estado de conexión .....	282
8.8.6	Partner Connection State .....	282
<b>9</b>	<b>Protocolo SNTP .....</b>	<b>283</b>
<b>9.1</b>	<b>Cliente SNTP .....</b>	<b>283</b>
<b>9.2</b>	<b>Cliente SNTP (información del servidor) .....</b>	<b>285</b>
<b>9.3</b>	<b>Servidor SNTP .....</b>	<b>285</b>

<b>10</b>	<b>Servidor X-OPC .....</b>	<b>287</b>
10.1	Equipamiento y sistema necesarios.....	287
10.2	Propiedades del servidor X-OPC .....	288
10.3	Propiedades del sistema de control HIMA para la conexión X-OPC .....	289
10.4	Acciones necesarias en caso de modificaciones .....	290
10.5	Forzado de variables globales en módulos de E/S .....	290
10.6	Configuración de una conexión de servidor OPC.....	291
10.6.1	Software necesario:.....	291
10.6.2	Condiciones para operar con servidor X-OPC: .....	291
10.6.3	Instalación en el PC Host .....	292
10.6.4	Configuración del servidor OPC ein SILworX.....	295
10.6.5	Ajustes del servidor OPC en el editor de safeethernet.....	296
10.6.6	Configuración del servidor X-OPC Data Access en SILworX.....	297
10.6.7	Configuración del servidor X-OPC Alarm & Event en SILworX.....	299
10.6.8	Parametrización de los fragmentos y prioridades en SILworX.....	302
10.7	Editor Alarm&Event .....	305
10.7.1	Eventos booleanos .....	305
10.7.2	Eventos escalares .....	306
10.8	Parámetros de propiedades de servidor X-OPC .....	309
10.8.1	Set de Servidor OPC .....	309
10.8.2	Servidor OPC .....	313
10.9	Desinstalación del servidor X-OPC .....	313
<b>11</b>	<b>Interfaz serie síncrona.....</b>	<b>314</b>
11.1	Sistema necesario.....	314
11.2	Diagrama de bloques .....	314
11.3	Conectores hembra D-Sub de FB1 y FB2 .....	315
11.4	Configuración entre submódulo SSI y COM.....	315
11.5	Configuración de la interfaz SSI .....	315
11.5.1	Longitud de cable y frecuencias de reloj recomendadas .....	316
11.6	Indicaciones de aplicación.....	317
<b>12</b>	<b>ComUserTask .....</b>	<b>318</b>
12.1	Sistema necesario.....	318
12.1.1	Creación de un ComUserTask .....	318
12.2	Requisitos .....	319
12.3	Abreviaturas .....	319
12.4	Interfaz CUT en SILworX.....	320
12.4.1	Intervalo regular [ms].....	320
12.4.2	Procesado previo regular .....	320
12.4.3	Procesado posterior regular .....	320
12.4.4	STOP_INVALID_CONFIG.....	320
12.4.5	Variables de la interfaz CUT (CPU<->CUT).....	321
12.4.6	Función de menú Propiedades.....	322
12.4.7	Función de menú Edit .....	323

<b>12.5</b>	<b>Funciones CUT.....</b>	<b>325</b>
12.5.1	Funciones COM-User-Callback .....	325
12.5.2	Funciones COM-User-Library .....	325
12.5.3	Archivos de cabecera .....	325
12.5.4	Áreas de código y de datos y Stack para CUT .....	326
12.5.5	Función de inicio CUCB_TaskLoop .....	326
12.5.6	Interfaces serie RS485/RS232 IF .....	327
12.5.7	UDP/TCP-Socket-IF.....	335
12.5.8	Timer-IF .....	349
12.5.9	Diagnóstico .....	350
<b>12.6</b>	<b>Funciones para SEW .....</b>	<b>351</b>
12.6.1	COM-User-IRQ-Task .....	351
12.6.2	NVRam-IF .....	353
12.6.3	Semaphore-IF .....	354
12.6.4	COM-IO-IF (sólo HM 30).....	357
<b>12.7</b>	<b>Instalación del entorno de desarrollo .....</b>	<b>359</b>
12.7.1	Instalación del entorno Cygwin .....	359
12.7.2	Instalación del compilador de GNU .....	361
<b>12.8</b>	<b>Crear un nuevo proyecto CUT .....</b>	<b>363</b>
12.8.1	Makefiles de CUT .....	364
12.8.2	Edición del código fuente C .....	366
12.8.3	Integración de ComUserTask en el proyecto.....	370
12.8.4	Error al cargar una configuración con CUT .....	373
<b>13</b>	<b>Generalidades .....</b>	<b>374</b>
<b>13.1</b>	<b>Configuración de los bloques funcionales .....</b>	<b>374</b>
13.1.1	Adquisición de bibliotecas de bloques funcionales .....	374
13.1.2	Configuración de los bloques funcionales en el programa del usuario.....	374
13.1.3	Configuración de los bloques funcionales en el árbol de SILworX .....	375
<b>13.2</b>	<b>Intervalo de tiempo de comunicación máximo .....</b>	<b>377</b>
13.2.1	Para sistemas de control HIMax .....	377
<b>13.3</b>	<b>Limitación de carga .....</b>	<b>377</b>
	<b>Anexo .....</b>	<b>379</b>
	<b>Glosario .....</b>	<b>379</b>
	<b>Índice de ilustraciones .....</b>	<b>380</b>
	<b>Índice de tablas .....</b>	<b>383</b>
	<b>Índice alfabético .....</b>	<b>392</b>



## 1 Introducción

El manual de comunicación describe las características y la configuración de los protocolos de comunicación de los sistemas de control HIMatrix y HIMax con función orientada a la seguridad mediante la utilidad de programación SILworX.

Con los protocolos provistos, los sistemas de control HIMax/HIMatrix pueden comunicar entre sí y con sistemas de control de otros fabricantes.

El conocimiento de la normativa y la correcta aplicación de las instrucciones contenidas en este manual por parte de personal cualificado son requisito indispensable para la planificación, el proyecto, la programación, la instalación, la puesta en servicio, el funcionamiento y la conservación de los sistemas de control HIMax/HIMatrix.

En caso de intervenir personal no cualificado, desactivar o puentejar (by-pass) funciones de seguridad o hacer caso omiso de las instrucciones de este manual (con las consiguientes perturbaciones y el menoscabo de las funciones de seguridad), pueden producirse situaciones de serio peligro para las personas, los bienes materiales y el medio ambiente, declinando HIMA toda responsabilidad en tales supuestos.

Los dispositivos de automatización HIMax se proyectan, fabrican y ponen a prueba cumpliendo las pertinentes normas de seguridad. Para usar sólo en los casos de aplicación previstos descritos bajo las condiciones ambientales especificadas.

### 1.1 Estructuración y uso del manual

El contenido de este manual es parte de la descripción del hardware del sistema electrónico programable HIMax.

El manual se divide en los siguientes capítulos principales:

- Introducción
- Seguridad
- Descripción del producto
- Puesta en servicio
- Funcionamiento
- Conservación
- Puesta fuera de servicio
- Transporte
- Desecho

Deberán observarse además los siguientes documentos:

Nombre	Contenido	Documento N. <sup>o</sup>
Manual del sistema HIMax	Descripción del hardware del sistema HIMax	HI 801 141 S
Manual de seguridad de HIMax	Funciones de seguridad del sistema HIMax	HI 801 196 S
Manual de seguridad de HIMatrix	Funciones de seguridad del sistema HIMatrix	HI 800 427 S
Manual de proyecto HIMatrix	Consideración general de proyecto de plantas con sistemas HIMatrix	HI 800 496 S
Manuales de hardware HIMatrix	Descripción del hardware del respectivo sistema de control HIMatrix	HI 800 xxx S
Primeros pasos	Introducción a SILworX	HI 801 194 S

Tabla 1: Manuales vigentes adicionales

Los manuales actuales se hallan siempre en la página web de HIMA: [www.hima.com](http://www.hima.com). Con ayuda del índice de revisión al pie de página podrá compararse la vigencia de los manuales que se tengan respecto a la edición que figura en Internet.

## 1.2 Destinatarios

Este documento va dirigido a planificadores, proyectadores y programadores de equipos de automatización y al personal autorizado para la puesta en servicio, operación y mantenimiento de dispositivos y sistemas. Se presuponen conocimientos especiales en materia de sistemas de automatización con funciones relacionadas con la seguridad.

## 1.3 Convenciones de representación

Para una mejor legibilidad y comprensión, en este documento se usa la siguiente notación:

<b>Negrita</b>	Remarcado de partes importantes del texto.
	Designación de botones de software, fichas e ítems de menús de SILworX sobre los que puede hacerse clic
<i>Cursiva</i>	Variables y parámetros del sistema
<i>Courier</i>	Entradas literales del operador
RUN	Designación de estados operativos en mayúsculas
Cap. 1.2.3	Las referencias cruzadas son enlaces, aun cuando no estén especialmente marcadas como tales. Al colocar el puntero sobre un enlace tal, cambiará su aspecto. Haciendo clic en él, se saltará a la correspondiente página del documento.

Las notas de seguridad y uso están especialmente identificadas.

### 1.3.1 Notas de seguridad

Las notas de seguridad del documento se representan de la siguiente forma.  
Para garantizar mínimos niveles de riesgo, deberá seguirse sin falta lo que indiquen.  
Los contenidos se estructuran en

- Palabra señalizadora: peligro, advertencia, precaución, nota
- Tipo y fuente de peligro
- Consecuencias del peligro
- Prevención del peligro

#### ⚠ PALABRA SEÑALIZADORA



**¡Tipo y fuente de peligro!**

**Consecuencias del peligro**

**Prevención del peligro**

Las palabras señalizadoras significan

- Peligro: su inobservancia originará lesiones graves o mortales
- Advertencia: su inobservancia puede originar lesiones graves o mortales
- Precaución: su inobservancia puede originar lesiones moderadas
- Nota: su inobservancia puede originar daños materiales

**NOTA****¡Tipo y fuente del daño!****Prevención del daño****1.3.2 Notas de uso**

La información adicional se estructura como sigue:

**i**

En este punto figura el texto con la información adicional.

Los trucos y consejos útiles aparecen en la forma:

**SUGERENCIA** En este punto figura el texto con la sugerencia.

## 2 Seguridad

En ningún caso deje sin leer las siguientes informaciones de seguridad, las notas y las instrucciones. Use el producto siempre cumpliendo todas las directivas y las recomendaciones de seguridad.

Este producto se usa con SELV o PELV. El módulo en sí no constituye ninguna fuente de peligro. El uso en áreas explosivas sólo se autoriza si se toman medidas adicionales.

### 2.1.1 Condiciones de uso

Los dispositivos han sido diseñados de forma que cumplan las exigencias de las siguientes normas en materia de compatibilidad electromagnética, clima y medioambiente:

Norma	Contenido
IEC/EN 61131-2	PLCs, parte 2 Características exigidas a los equipos de trabajo y ensayos
IEC/EN 61000-6-2	CEM Norma básica, parte 6-2 Inmunidad a interferencias, entorno industrial
IEC/EN 61000-6-4	Compatibilidad electromagnética (CEM) Norma básica de emisión de interferencias, entorno industrial

Tabla 2: Normas de compatibilidad electromagnética, clima y medio ambiente

Para hacer uso de los sistemas de control HIMax de función relacionada con la seguridad deben cumplirse las siguientes condiciones generales:

Tipo de condición	Contenido de la condición
Clase de protección	Clase de protección II según IEC/EN 61131-2
Polución	Grado de polución II según IEC/EN 61131-2
Altitud de emplazamiento	< 2000 m
Carcasa	Estándar: IP 20/IP 00 Si las normas de aplicación pertinentes (p. ej. EN 60204) así lo exigen, el dispositivo deberá montarse en una carcasa del grado de protección exigido (p. ej. IP 54).

Tabla 3: Condiciones generales

### Condiciones climáticas

Los ensayos más relevantes y los valores límite para las condiciones climáticas se relacionan en la siguiente tabla:

IEC/EN 61131-2	Ensayos climáticos
	Temperatura de trabajo: 0...+60 °C (límites de ensayo: -10...+70 °C)
	Temperatura de almacenamiento: -40...+85 °C
	Frío y calor secos, ensayos de durabilidad: +70 °C/-25 °C, 96 h, acometida de corriente no conectada
	Variaciones de temperatura, ensayos de durabilidad e inmunidad: -25 °C/+70 °C y 0 °C/+55 °C, Acometida de corriente no conectada
	Ciclos con calor húmedo, ensayos de durabilidad: +25 °C/+55 °C, 95% de humedad relativa Acometida de corriente no conectada

Tabla 4: Condiciones climáticas

### Condiciones mecánicas

Los ensayos más relevantes y los valores límite para las condiciones mecánicas se relacionan en la siguiente tabla:

IEC/EN 61131-2	Ensayos mecánicos
	Ensayo de inmunidad frente a vibraciones: 5...9 Hz/3,5 mm de amplitud 9...150 Hz, 1 g, probeta en funcionamiento, 10 ciclos por eje
	Ensayo de inmunidad frente a choques: 15 g, 11 ms, probeta en funcionamiento, 3 choques por eje y dirección (18 choques)

Tabla 5: Ensayos mecánicos

### Condiciones de compatibilidad electromagnética

Para los sistemas con función relacionada con la seguridad se exigen altos niveles frente a interferencias. Los sistemas HIMatrix cumplen estas exigencias según IEC 62061 e IEC 61326-3-1. Véase la columna “Criterio FS” (seguridad funcional).

Normas de ensayos	Ensayos de inmunidad a interferencias	Criterio FS
IEC/EN 61000-4-2	Ensayos de ESD: 6 kV de descarga por contacto, 8 kV al aire	6 kV, 8 kV
IEC/EN 61000-4-3	Ensayos de RFI (10 V/m): 80 MHz...2 GHz, 80% AM Ensayos de RFI (3 V/m): 2 GHz...3 GHz, 80% AM Ensayos de RFI (20 V/m): 80 MHz...1 GHz, 80% AM	- - 20 V/m
IEC/EN 61000-4-4	Ensayos de ráfagas: Tensión de alimentación: 2 kV y 4 kV Líneas de señal: 2 kV	4 kV 2 kV
IEC/EN 61000-4-12	Ensayo con vibraciones atenuadas: 2,5 kV L-, L+/PE 1 kV L+/L -	- -
IEC/EN 61000-4-6	Alta frecuencia, asimétrica: 10 V, 150 kHz...80 MHz, 80% AM 20 V, frecuencias ISM, 80% AM	10 V -
IEC/EN 61000-4-3	Impulsos de 900 MHz	-
IEC/EN 61000-4-5	Tensión transitoria: Tensión de alimentación: 2 kV CM, 1 kV DM Líneas de señal: 2 kV CM, 1 kV DM para E/S de CA	2 kV/1 kV 2 kV

Tabla 6: Ensayos de inmunidad a interferencias

IEC/EN 61000-6-4	Ensayos de emisión de interferencias
EN 55011 Clase A	Emisión de interferencias: irradiada, vinculada al cable

Tabla 7: Ensayos de emisión de interferencias

## Fuente de alimentación

Los ensayos más relevantes y los valores límite para las fuentes de alimentación de los dispositivos se relacionan en la siguiente tabla:

IEC/EN 61131-2	Evaluación de las características de la fuente de corriente continua
	La fuente de alimentación debe cumplir alternativamente las siguientes normas: IEC/EN 61131-2 o SELV (Safety Extra Low Voltage) o PELV (Protective Extra Low Voltage)
	Los cortacircuitos que se usen en los dispositivos HIMax deben responder a lo especificado en este manual.
	Ensayo del rango de tensiones: 24 V CC, -20%...+25% (19,2 V...30,0 V)
	Ensayo de inmunidad contra breves interrupciones de la fuente de alimentación externa: CC, PS 2: 10 ms
	Inversión de polaridad de la fuente de alimentación: Hallará notas al respecto en el correspondiente capítulo del manual del sistema o en la hoja de características de la fuente de alimentación.
	Duración del búfer, ensayo de durabilidad: Ensayo B, 1000 h

Tabla 8: Evaluación de las características de la fuente de corriente continua

## Precauciones contra descargas electrostáticas

Las modificaciones o ampliaciones del sistema, así como la sustitución de módulos, únicamente deberán ser realizadas por personal con conocimientos sobre medidas de protección contra descargas electrostáticas.

### NOTA



¡Las descargas electrostáticas pueden dañar los componentes electrónicos de los sistemas de control!

- Realice estas tareas en un lugar de trabajo antiestático y llevando una cinta de puesta a tierra.
- Guarde bien protegidos los módulos que no tenga en uso (p. ej. en su embalaje original).

Las modificaciones o ampliaciones del cableado del sistema se autorizan sólo a personal con conocimientos sobre medidas de protección contra descargas electrostáticas (ESD).

## 2.2 Peligros remanentes

Un módulo HIMax en sí no representa ninguna fuente de peligro.

Lo siguiente puede conllevar peligros remanentes:

- Errores de realización del proyecto
- Errores en el programa de usuario
- Errores en el cableado

## 2.3 Medidas de seguridad

Respete las normas de seguridad vigentes en el lugar de uso y use la debida indumentaria de seguridad personal.

## 2.4 Información para emergencias

Un sistema de control HIMax forma parte del equipamiento de seguridad de una planta. Si el sistema de control deja de funcionar, la planta adoptará un estado seguro.

En caso de emergencia está prohibida toda intervención que impida la función de seguridad de los sistemas HIMax.

### 3 Descripción del producto

Mediante una conexión de datos, los sistemas HIMax/HIMatrix pueden intercambiar datos de proceso entre sí y con sistemas externos. La configuración de los protocolos descritos en este manual se realiza con la utilidad de programación SILworX.

Para programar sistemas HIMatrix en SILworX, los sistemas HIMatrix deberán tener sistemas operativos a partir de CPU BS V7 y COM BS V12. Es posible establecer conexión **safeethernet** a sistemas HIMatrix con sistemas operativos más antiguos mediante comunicación que trascienda el proyecto. Véase el capítulo 4.8.

En el marco del mantenimiento perfectivo, HIMA desarrolla mejoras de los sistemas operativos. Dichas versiones mejoradas podrá Ud. cargarlas a los módulos con la ayuda de SILworX. Véanse los manuales de sistema HIMax HI 801 141 S, HIMatrix HI 800 495 S y HI 800 494 S.

Las siguientes secciones (3.1 y 3.2) ofrecen una vista general del protocolo orientado a la seguridad **safeethernet** y de los protocolos estándar disponibles.

#### 3.1 Protocolo con función orientada a la seguridad (**safeethernet**)

Todos los sistemas HIMax/HIMatrix pueden comunicar de forma orientada a la seguridad mediante Ethernet conforme a SIL 3. Dicha comunicación de seguridad tiene lugar mediante el protocolo **safeethernet**.

Para el intercambio de datos de proceso orientado a la seguridad entre sistemas de control HIMax y HIMatrix en una red Ethernet se usa el protocolo orientado a la seguridad **safeethernet**.

Descripción del protocolo orientado a la seguridad **safeethernet** a partir del capítulo 4.

#### ⚠ ADVERTENCIA



¡Manipulación de la transmisión de datos orientada a la seguridad!

Daños personales

La empresa usuaria deberá tomar las medidas necesarias para que el Ethernet empleado para **safeethernet** esté suficientemente protegido contra manipulaciones (p. ej. hackers).

El tipo y la cuantía de las medidas deberán acordarse con el ente de inspección oficial que deba emitir su aprobación.

### 3.2 Protocolos estándar

Para la óptima integración del sistema HIMax/HIMatrix con dispositivos de campo y sistemas centralizados de control se dispone de acreditados protocolos estándar adicionales. Aquí encuentran aplicación protocolos de bus de campo y Ethernet.

Toda una serie de protocolos de comunicación permite sólo una transmisión de datos sin función orientada a la seguridad. Estos podrán usarse para las partes de una tarea de automatización que no tengan funciones orientadas a la seguridad.

Elemento	HIMax	HIMatrix L3	HIMatrix L2	Descripción
Módulo o sistema de control requerido	Máximo de 20 módulos de comunicación por sistema de control HIMax.	Módulo de comunicación integrado del sistema de control.	Módulo de comunicación integrado del sistema de control.	Los protocolos estándar se ejecutan en el módulo de comunicación.
Interfaces	Interfaces Ethernet e interfaces de bus de campo de los módulos COM.	Interfaces Ethernet e interfaces de bus de campo del sistema de control.	Interfaces Ethernet e interfaces de bus de campo del sistema de control.	Hallará más información en la Tabla 14.
Máxima cantidad de protocolos estándar	<ul style="list-style-type: none"> <li>▪ 20 por cada sistema de control HIMax.</li> <li>▪ 6<sup>1)</sup> por módulo COM.</li> </ul>	6 <sup>1)</sup>	4 <sup>1)</sup>	Protocolos estándar disponibles. Véase la Tabla 10.
Volumen de datos de proceso de todos los protocolos estándar de un sistema de control	128 kB de envío 128 kB de recepción.	64 kB de envío 64 kB de recepción.	16 kB de envío 16 kB de recepción.	No debe sobrepasarse el máximo volumen de datos de proceso del sistema de control. De hacerlo, se rechazará la parametrización del sistema de control al cargar.

<sup>1)</sup> Servidor X-OPC, cliente SNTP y servidor SNTP no se tienen en cuenta en este cómputo.

Tabla 9: Protocolos estándar

#### ⚠ ADVERTENCIA



**Uso de datos importados no seguros.**

**No deben usarse datos no seguros para las funciones orientadas a la seguridad del programa de usuario.**

Se dispone de los siguientes protocolos estándar:

Protocolo	por módulo HIMax	por HIMatrix	Descripción
PROFINET IO Controller	1	---	Capítulo 5.5
PROFINET IO Device	1	---	Capítulo 5.8
PROFIBUS DP Master	2	1 para F20, 2 para F30, F35, F60	Capítulo 6.1
PROFIBUS DP Slave	1	1	Capítulo 6.13
Modbus Master	1	1	Capítulo 7.2
Modbus Slave	1	1	Capítulo 7.3
S&R TCP	1	1	Capítulo 8
Servidor X-OPC de HIMA X <sup>1)</sup>	---	---	Capítulo 10
ComUserTask	1	1	Capítulo 12

<sup>1)</sup> El servidor X-OPC de HIMA se instala en un PC host y sirve de interfaz de transmisión entre un máximo de 255 sistemas de control HIMax/HIMatrix y sistemas externos que dispongan de una interfaz OPC.

Tabla 10: Protocolos estándar disponibles



Máximo de 1280 conexiones TCP por sistema de control HIMax con 20 módulos COM.  
Máximo de 64 conexiones TCP por sistema de control HIMatrix L3 o módulo COM HIMax.  
Máximo de 32 conexiones TCP por sistema de control HIMatrix L2.

#### **Máxima cantidad de protocolos activados en un HIMatrix o en un módulo COM HIMax**

Se dispone de un máximo de 64 sockets TCP por HIMatrix L3 o módulo COM HIMax.

Ejemplo 1:

Protocolo	Conexiones
1 Modbus Master	44 conexiones TCP Slave
1 Modbus Slave	20 conexiones TCP Master

Tabla 11: Protocolos en un módulo de comunicación

Ejemplo 2:

Protocolo	Conexiones
1 PROFIBUS DP Master	122 conexiones Slave
1 PROFIBUS DP Slave	1 conexión Master

Tabla 12: Protocolos en un módulo de comunicación

### 3.3 Redundancia

El sistema HIMax ha sido concebido como sistema de alta disponibilidad y está preparado también para la redundancia de la comunicación. Una conexión de comunicación será redundante cuando existan dos vías físicas de transmisión iguales y para ello se usen 2 módulos HIMax.

Para HIMatrix se ha previsto la comunicación redundante sólo mediante **safeethernet** y con una interfaz Ethernet.

#### Redundancia mediante safeethernet

La redundancia podrá Ud. configurarla en el editor de **safeethernet**, donde podrá seleccionar las interfaces Ethernet para la segunda vía de transmisión (ver capítulo 4.2).

#### Redundancia mediante protocolos estándar

PROFIBUS DP Master PROFIBUS DP Slave PROFINET IO TCP S&R Modbus Master	La redundancia del correspondiente protocolo estándar deberá Ud. configurarla en el programa de usuario de forma que éste supervise las vías de transmisión redundantes y asigne los datos de proceso redundantemente transmitidos a la respectiva vía.
Modbus Slave	La redundancia puede configurarse en SILworX para HIMax.

### 3.4 Registro y activación de los protocolos

Para HIMax/HIMatrix se dispone de los siguientes protocolos, que pueden activarse de la siguiente manera:

- 
- i** En un X-COM se necesita un código de habilitación de software propio para cada protocolo. Excepción: Para usar 2 Modbus Slaves en un X-COM basta con un solo código de habilitación de software.
- 

Protocolo	Interfaces	Activación
HIMA safeethernet	Ethernet	[1]
SNTP Server/Client	Ethernet	[1]
Servidor X-OPC de HIMA (en el PC host)	Ethernet	[4]
PROFINET IO Controller <sup>1)</sup>	Ethernet	[4]
PROFINET IO Device <sup>1)</sup>	Ethernet	[4]
Modbus TCP Master	Ethernet	[4]
Modbus TCP Slave	Ethernet	[4]
TCP Send/Receive	Ethernet	[4]
PROFIBUS DP Master	FB1 y FB2	[2]
PROFIBUS DP Slave	FB1 o FB2	[2]
Modbus Master RS485	FB1 o FB2	[3]
Modbus Slave RS485	FB1 o FB2	[3]
ComUserTask RS232, RS485, RS422, SSI	FB1 o FB2	[3]

<sup>1)</sup> No disponible para HIMatrix L2

Tabla 13: Protocolos disponibles de HIMax/HIMatrix

- [1]. La función está habilitada por defecto en todos los sistemas HIMax/HIMatrix.
- [2]. Para PROFIBUS Master y PROFIBUS Slave la habilitación tiene lugar mediante la instalación de un submódulo de bus de campo.
- [3]. Para el submódulo de bus de campo RS485 (Modbus RS485) y el submódulo de bus de campo RS232, RS422, SSI (ComUserTask) deberá Ud. obtener asimismo un código de habilitación de software para el protocolo de bus de campo seleccionado.
- [4]. El código de habilitación del software podrá Ud. generarla en el sitio web de HIMA con ayuda del ID de sistema (p. ej. 60000) de su sistema de control. Siga las instrucciones del sitio web [www.hima.com](http://www.hima.com) -> Products -> Registration -> Communication options SILworX

- 
- i** El código de habilitación del software está inseparablemente vinculado a ese ID de sistema. Una licencia puede usarse únicamente para un determinado ID de sistema. Por ello, la habilitación no debería realizarse hasta el momento en que el ID de sistema esté establecido inequívocamente.
- 

Todos los protocolos de Ethernet podrán probarse sin código de habilitación de software durante 5000 horas de servicio.

- 
- i** ¡Pida a tiempo el código de habilitación del software!
- Tras 5000 horas de servicio, la comunicación seguirá en curso hasta detenerse el sistema de control. Después de ello, no será posible iniciar el programa de usuario sin un código válido de habilitación del software para los protocolos planificados en el proyecto (configuración no válida).
-

**Para introducir el código de habilitación de software en SILworX:**

1. Seleccione **Configuration, Resource, License Management** en el árbol.
2. Haga clic con el botón derecho en **License Management** y seleccione **New, License Key** en el menú contextual.  
 La clave de la licencia se agregará.
3. Haga clic con el botón derecho en **License Key** y seleccione **Properties** en el menú contextual.
4. Escriba en el recuadro **Activation Code** el código de habilitación de software generado.

### 3.5 Interfaces Ethernet

La comunicación con sistemas externos tiene lugar a través de interfaces Ethernet de la CPU HIMax y los módulos COM de los sistemas de control HIMatrix. Cada interfaz Ethernet puede procesar simultáneamente varios protocolos.



Mediante las interfaces Ethernet del módulo de bus de sistema X-SB 01, no es posible la comunicación de datos de proceso. Las interfaces Ethernet UP y DOWN se han provisto exclusivamente para conectar los racks HIMax entre sí. Si el bus del sistema se usa en estructura de red, también podrá usarse la interfaz Ethernet DIAG para la conexión de los racks HIMax.

Véase el manual del sistema HI 801 141 S.

#### 3.5.1 Propiedades de las interfaces Ethernet

Propiedad	Módulo de CPU HIMax	Módulo COM HIMax	HIMatrix
Puertos	4	4	4, 2 en caso de F20 y Remote I/Os
Estándar de transmisión	10/100/1000 Base-T, Half y Full Duplex	10/100 Base-T, Half y Full Duplex	10/100 Base-T 1000 Base-T (sólo L3) Half y Full Duplex
Auto Negotiation	Sí	Sí	Sí
Auto-Crossover	Sí	Sí	Sí
Conector hembra	RJ 45	RJ 45	RJ 45
IP-Address	Libremente configurable <sup>1)</sup>	Libremente configurable <sup>1)</sup>	Libremente configurable <sup>1)</sup>
Subnet Mask	Libremente configurable <sup>1)</sup>	Libremente configurable <sup>1)</sup>	Libremente configurable <sup>1)</sup>
Protocolos compatibles	<b>safeethernet</b> Dispositivo programador (PADT), SNTP	<b>safeethernet</b> Protocolos estándar	<b>safeethernet</b> Protocolos estándar Dispositivo programador (PADT), SNTP

<sup>1)</sup> Deberán observarse las reglas de validez general para la asignación de direcciones IP y máscaras de subred.

Tabla 14: Propiedades de las interfaces Ethernet

Cada módulo COM y CPU HIMax y cada sistema de control HIMatrix posee un switch Ethernet con una dirección IP libremente configurable.

El switch Ethernet establece una conexión dirigida entre dos interlocutores de comunicación para la transmisión de datos. Esto evita colisiones y alivia de carga la red.

Para el reenvío dirigido de los datos se crea una tabla de asignación de direcciones MAC/IP (ARP-Cache) y se asignan direcciones MAC a determinadas direcciones IP.

Los paquetes de datos se reenviarán entonces sólo a las direcciones IP listadas en ARP-Cache.



Sustitución de un módulo de CPU/COM HIMax o un sistema de control HIMatrix con idéntica dirección IP.

Si se sustituye un dispositivo para el cual se hubiera configurado un tiempo *ARP Aging Time* = 5 minutos y *MAC Learning* = conservativ, el interlocutor de comunicación adoptará la nueva dirección MAC no antes de 5 minutos y no después de 10 minutos. Durante ese tiempo no será posible la comunicación a través del dispositivo sustituido.

Además del tiempo ARP Aging Time configurable, deberá esperarse el tiempo MAC Aging Time no modificable del switch (unos 10 segundos) hasta que sea posible la comunicación a través del dispositivo sustituido.

### 3.5.2 Configuración de las interfaces Ethernet

La configuración de las interfaces Ethernet se realiza en SILworX mediante la vista en detalle del módulo COM o CPU.

Para los sistemas de control HIMax/HIMatrix, los valores estándar de los parámetros **Parameter Speed Mode** y **Flow Control Mode** están configurados como AutoNeg.



¡Pérdida de la comunicación!

Una configuración desfavorable de los parámetros de Ethernet puede hacer que se pierda la comunicación al dispositivo. Ejecute un reset del dispositivo.

#### Para abrir la vista en detalle del módulo de comunicación:

1. Seleccione **Configuration, Resource, Hardware** en el árbol.
2. Haga clic con el botón derecho y seleccione **Edit** en el menú contextual, para abrir el editor de hardware.
3. Haga clic con el botón derecho en **Communication Module** y seleccione **Detail View** en el menú contextual, para abrir la vista en detalle.



Las entradas de la ficha de propiedades de los módulos COM y CPU deberá Ud. recompilarlas con el programa de usuario y transmitirlas al sistema de control, para hacerlas efectivas de cara a la comunicación del sistema HIMax/HIMatrix.

#### Módulos

Elemento	Descripción
Name	Nombre del módulo de comunicación.
Use Max CPU Load for HH Protocol	<ul style="list-style-type: none"> <li>▪ Activado: Aplicar límite de carga de la CPU tomado del recuadro “Max. CPU Load [%]”.</li> <li>▪ Desactivado: No usar límite de carga de la CPU para safeethernet.</li> </ul>
Max. CPU Load for HH Protocol [%]	<p>Máxima carga de CPU del módulo que se permite que tenga lugar al ejecutar el protocolo safe <b>ethernet</b>.</p> <p><b>i</b> La carga máxima deberá dividirse entre todos los protocolos utilizados que usa este módulo de comunicación.</p>
IP Address	Dirección IP de la interfaz Ethernet
Subnet Mask	Máscara de dirección de 32 bits para subdividir una dirección IP en dirección de red y dirección de host.
Speed Mode	
Flow Control Mode	<p><b>i</b> ¡Se permite sólo elegir el ajuste de “Autoneg”!</p> <p>Si se elige otra opción, el módulo pasará a STOP.</p>
Standard interface	Activado: La interfaz se usa como la interfaz predeterminada para ingresar al sistema.
Default Gateway	Dirección IP de la puerta de enlace predeterminada
Activate Extended Settings	Use los parámetros <i>ARP Aging Time [s]</i> , <i>MAC Learning</i> e <i>IP Forwarding</i> .

Elemento	Descripción
ARP Aging Time [s]	<p>Un módulo COM o CPU guarda las direcciones MAC de sus interlocutores de comunicación en una tabla de asignación de direcciones MAC/IP (cache ARP).</p> <p>Si durante un período entre 1 y 2 veces igual al tiempo <i>ARP Aging Time</i></p> <ul style="list-style-type: none"> <li>- llegan notificaciones del interlocutor de comunicación, la dirección MAC permanecerá en el cache ARP.</li> <li>- no llegan notificaciones del interlocutor de comunicación, se borrará la dirección MAC del cache ARP.</li> </ul> <p>El valor típico para el tiempo <i>ARP Aging Time</i> en una red local es de 5 s...300 s.</p> <p>El usuario no podrá leer el contenido del cache ARP.</p> <p>Si se usan enrutadores o puertas de enlace, adapte (aumente) el tiempo <i>ARP Aging Time</i> de acuerdo a los retardos adicionales para el tramo de ida y de vuelta.</p> <p>Si el tiempo <i>ARP Aging Time</i> es demasiado corto, el módulo COM/CPU borrará del cache ARP la dirección MAC del interlocutor de comunicación y la comunicación sufrirá retrasos o se cancelará. Para una aplicación eficaz, el tiempo ARP Aging Time deberá ser mayor que los tiempos ReceiveTimeouts de los protocolos utilizados.</p> <p>Rango de valores: 1 s...3600 s Valor por defecto: 60 s</p>
MAC Learning	<p>Con MAC Learning y <i>ARP Aging Time</i> el usuario define la rapidez con la que desea que se aprenda una dirección MAC.</p> <p>Son posibles los siguientes ajustes:</p> <ul style="list-style-type: none"> <li>▪ <b>Conservative</b> (recomendado): Si en el cache ARP ya hay registro de direcciones MAC de interlocutores de comunicación, estos registros estarán enclavados durante un mínimo de tiempo <i>ARP Aging Time</i> hasta un máximo de tiempo dos veces igual a <i>ARP Aging Time</i> y no podrán ser sustituidas por otras direcciones MAC. Así se garantiza que no puedan desviarse, voluntaria o involuntariamente, paquetes de datos a participantes ajenos de la red (ARP spoofing).</li> <li>▪ <b>Tolerant</b>: Al recibirse una notificación se comparará la dirección IP de la notificación con los datos que constan en el cache ARP y se sobreescibirá inmediatamente la dirección MAC guardada en el cache ARP con la dirección MAC de la notificación. Use la opción “Tolerant” cuando la disponibilidad de la comunicación sea más importante que el acceso seguro (authorized access) al sistema de control.</li> </ul>
IP Forwarding	<p>Permite a un módulo COM o CPU funcionar como enrutador y reenviar paquetes de datos de otros nodos de la red.</p> <ul style="list-style-type: none"> <li>▪ Activado: El reenvío está activado.</li> <li>▪ Desactivado: El reenvío está desactivado.</li> </ul>

Elemento	Descripción
ICMP Mode	<p>El protocolo ICMP (Internet Control Message Protocol) permite a las capas altas del protocolo detectar estados de error en la capa de red y optimizar la transmisión de los paquetes de datos.</p> <p>Tipos de mensaje del protocolo ICMP compatible con el módulo de CPU:</p> <ul style="list-style-type: none"> <li>▪ No ICMP Responses Todos los comandos de ICMP desactivados. Así se logra un alto grado de seguridad contra posibles sabotajes que pueden producirse a través de la red.</li> <li>▪ Echo Response Si se activa Echo Response, el nodo responderá a un comando Ping. Así podrá constatarse si un nodo está accesible. El grado de seguridad sigue siendo alto.</li> <li>▪ Host Unreachable Irrelevante para el usuario. Sólo para pruebas del fabricante.</li> <li>▪ All Implemented ICMP Responses Todos los comandos de ICMP activados. Así se logra un diagnóstico de errores más exacto en caso de fallos en la red.</li> </ul>

Tabla 15: Parámetros de configuración

### Routings

Elemento	Descripción
Name	Designación del ajuste de enrutado
IP Address	Dirección IP de destino del interlocutor de comunicación (en el enrutado de host directo) o dirección de la red (en el enrutado de subred) Rango de valores: 0.0.0.0...255.255.255.255 Valor por defecto: 0.0.0.0
Subnet Mask	Define el rango de direcciones de destino para un registro de enrutado. 255.255.255.255 (en el caso del enrutado de host directo) o máscara de subred de la red direccionada. Rango de valores: 0.0.0.0...255.255.255.255 Valor por defecto: 255.255.255.255
Gateway	Dirección IP de la puerta de enlace a la red direccionada. Rango de valores: 0.0.0.0...255.255.255.255 Valor por defecto: 0.0.0.1

Tabla 16: Parámetros de enrutado

### Switch Ethernet

Elemento	Descripción
Port	Número del puerto como rotulación de carcasa. Por puerto sólo podrá haber una configuración. Rango de valores: 1...4
Speed [Mbit/s]	10 MBit/s: Tasa de datos 10 MBit/s 100 MBit/s: Tasa de datos 100 MBit/s 1000 MBit/s: Tasa de datos 1000 MBit/s (módulo CPU) Autoneg (10/100/1000): Ajuste automático de baudios Valor por defecto: Autoneg
Flow Control	Full Duplex: Comunicación simultánea en ambos sentidos Half Duplex: Comunicación en un sentido Autoneg: Control automático de la comunicación Valor por defecto: Autoneg
Autoneg también en el caso de los valores fijos	El "Advertising" (transmisión de las características de Speed y Flow-Control) se ejecutará también en caso de obrar valores fijos de Speed y Flow Control. Así otros dispositivos cuyos puertos estén configurados como Autoneg reconocerán la configuración del puerto HIMax.
Limit	Para limitar los paquetes entrantes de tipo Multicast y/o Broadcast. OFF: Sin limitación Broadcast: Limitación de Broadcast (128 kbit/s) Multicast y Broadcast: Limitación de Multicast y Broadcast (1024 kbit/s) Valor por defecto: Broadcast

Tabla 17: Parámetros del switch Ethernet

### VLAN (Port based VLAN)

(función no compatible con sistemas de control HIMatrix)

Para configurar la utilización de VLAN basado en puerto.

- 
- Si se desea la compatibilidad con VLAN, deberá desactivarse VLAN basado en puerto, de forma que todos los puertos puedan comunicar con cualquier otro puerto del switch.
- 

Para cada puerto de un switch podrá Ud. definir a qué otro puerto del switch pueden enviarse los Ethernet Frames recibidos.

La tabla de la ficha VLAN contiene ítems que permiten habilitar o inhabilitar la conexión entre dos puertos dados.

	Eth1	Eth2	Eth3	Eth4
Eth1				
Eth2	habilitada			
Eth3	habilitada	habilitada		
Eth4	habilitada	habilitada	habilitada	
COM	habilitada	habilitada	habilitada	habilitada

Tabla 18: Ficha VLAN

Configuración por defecto: todas las conexiones entre puertos habilitadas

## LLDP

(función no compatible con sistemas de control HIMatrix)

LLDP (Link Layer Discovery Protocol) envía periódicamente por Multicast información sobre el propio dispositivo (p. ej. dirección MAC, nombre del dispositivo, número de puerto) y recibe el mismo tipo de información de los dispositivos contiguos.

Según si se tiene configurado PROFINET en el módulo de comunicación, LLDP usará los siguientes valores:

PROFINET en módulo COM	ChassisID	TTL (Time to Live)
Se utiliza	Nombre de estación	20 s
No se utiliza	Dirección MAC	120 s

Tabla 19: Valores para LLDP

El módulo de comunicación y el procesador admiten LLDP en los puertos Eth1, Eth2, Eth3 y Eth4.

Los siguientes parámetros definen cómo funcionará el puerto correspondiente:

- |              |  |
|--------------|--|
| Off          | LLDP inhabilitado en este puerto   |
| Send         | LLDP envía Ethernet Frames LLDP, los Ethernet Frames recibidos se borrarán sin procesarlos |
| Receive      | LLDP no envía Ethernet Frames LLDP, pero sí procesará Ethernet Frames recibidos            |
| Send/Receive | LLDP envía y procesa Ethernet Frames LLDP recibidos  |
- Configuración por defecto: Send/Receive

## Mirroring

(función no compatible con sistemas de control HIMatrix)

Configura si el módulo duplica paquetes Ethernet en un puerto, de forma que puedan ser leídos por un dispositivo ahí conectado, p. ej. para pruebas.

Los siguientes parámetros definen cómo funcionará el puerto correspondiente:

- |                 |   |
|-----------------|---|
| Off             | Este puerto no participa del Mirroring.                       |
| Egress:         | Los datos salientes de este puerto se duplicarán.             |
| Ingress:        | Los datos entrantes de este puerto se duplicarán.             |
| Ingress/Egress: | Los datos entrantes y salientes de este puerto se duplicarán. |
| Dest Port:      | Los datos duplicados se enviarán a este puerto.               |
- Configuración por defecto: Apagado

### 3.5.3 Puertos de red utilizados para comunicación Ethernet

#### Puertos UDP/Utilización

- 123 SNTP (sincronización entre PES und Remote I/O, así como dispositivos externos)
- 502 Modbus Slave (modificable por el usuario)
- 6010 Safeethernet y OPC
- 8001 Configuración de Remote I/O mediante el sistema PES
- 8000 Programación y manejo con SILworX
- 34964 PROFINET Endpointmapper (necesario para establecer conexión)
- 49152 PROFINET RPC-Server
- 49153 PROFINET RPC-Client

#### Puertos TCP/Utilización

- 502 Modbus Slave (modificable por el usuario)
- Xxx TCP-SR asignado por el usuario



Todos los puertos arriba listados son puertos de destino.  
Los puertos de origen de los módulos de comunicación son variables y no influenciables.  
ComUserTask podrá utilizar cualquier puerto,  
siempre que no esté ya ocupado por otro protocolo.

---

### 3.6 Interfaces de bus de campo

Los submódulos de bus de campo permiten la comunicación a través de las interfaces de bus de campo de X-COM 01 HIMax y de los sistemas de control HIMatrix F20, F30, F35 y F60 CPU 01. Los submódulos de bus de campo son opcionales y se expedien montados de fábrica.

La interfaz de bus de campo FB3 de los sistemas de control HIMatrix se expide de fábrica con RS485 para Modbus (Master o Slave) o asignada como ComUserTask.

A través de las interfaces de bus de campo no es posible la comunicación orientada a la seguridad.

La siguiente sinopsis muestra los submódulos de bus de campo disponibles:

Designación	Descripción
PROFIBUS Master	Submódulo de bus de campo PROFIBUS DP Master
PROFIBUS Slave	Submódulo de bus de campo PROFIBUS DP Slave
Módulo RS485	Submódulo de bus de campo RS485 para usar con Modbus (Master o Slave) o ComUserTask
Módulo RS232	Submódulo de bus de campo RS232 para usar con ComUserTask
Módulo RS422	Submódulo de bus de campo RS422 para usar con ComUserTask
INTERBUS Master <sup>1)</sup>	Submódulo de bus de campo INTERBUS Master
Módulo SSI	Submódulo de bus de campo SSI para usar con ComUserTask

<sup>1)</sup> Sólo para HIMatrix con sistema operativo anterior a CPU BS V7 y COM BS V12  
Compatible sólo con la utilidad de programación ELOP II Factory.

Tabla 20: Submódulos de bus de campo disponibles

#### 3.6.1 Instalación

Los submódulos de bus de campo son opcionales y los instala el fabricante. La definición tiene lugar mediante el número de referencia al hacer el pedido. Además, deberán activarse los protocolos utilizados.

#### ⚠ PRECAUCIÓN



Apertura no autorizada del módulo COM o de HIMatrix

Daños al módulo COM o HIMatrix

Sólo HIMA está autorizada a incorporar nuevos submódulos de bus de campo.

#### 3.6.1.1 Elementos del número de referencia

En las siguientes secciones se describe cómo cambia el número de referencia con la asignación de las interfaces de bus de campo en un sistema de control HIMatrix o HIMax X-COM 01.

Para el número de referencia se asignan cifras a los submódulos de bus de campo. Véase Tabla 1.

Opciones para FB1 y FB2	Descripción
0	No hay submódulos de bus de campo instalados
1	RS485 para Modbus (Master o Slave) o ComUserTask
2	PROFIBUS DP Master
3	PROFIBUS DP Slave
4	INTERBUS Master (sólo en HIMatrix con ELOP II Factory)
5	RS232 para ComUserTask
6	RS422 para ComUserTask
7	SSI para ComUserTask

Tabla 21: Opciones para interfaces de bus de campo FB1 y FB2

### 3.6.1.2 Nº de referencia del módulo COM HIMax

El módulo COM constituye una unidad funcional con la tarjeta de conexión X-CB 001 02. La tarjeta de conexión deberá pedirse por separado.

Al equipar el X-COM 01 con uno o más submódulos de bus de campo, no sólo cambiará el número de referencia sino también la designación del módulo de X-COM 01 a X-COM 010 XY.

La siguiente tabla contiene los componentes disponibles:

Designación	Descripción
X-COM 01	Módulo de comunicación sin submódulos de bus de campo
X-COM 010 XY <sup>1)</sup>	Módulo de comunicación con submódulo de bus de campo
X-CB 001 02	Connector Board

<sup>1)</sup> **X**: Opción para interfaz de bus de campo FB1 según Tabla 21  
**Y**: Opción para interfaz de bus de campo FB2 según Tabla 21

Tabla 22: Componentes HIMax disponibles

La siguiente tabla contiene ejemplos de números de referencia y designaciones:

Nº de referencia	Designación	Submódulo 1 de bus de campo (FB1)	Submódulo 2 de bus de campo (FB2)
98 5260021	X-COM 010 21	PROFIBUS Master (máx. 12 MBit)	RS485
98 5260023	X-COM 010 23	PROFIBUS Master (máx. 12 MBit)	PROFIBUS Slave (máx. 1,5 MBit)
98 5260011	X-COM 010 11	RS485	RS485
98 5260000	X-COM 01	---	---

Tabla 23: Ejemplos de números de referencia y designaciones de módulos COM



HIMA recomienda operar PROFIBUS DP mediante la interfaz de bus de campo FB1 (tasa de transmisión máxima de 12 MBit). A través de la interfaz de bus de campo FB2 se permite una tasa de transmisión máxima de 1,5 MBit.

La designación y el número de referencia (Part-Nr.) constan en la placa de tipo del módulo.

### 3.6.1.3 Números de referencia de los sistemas de control HIMatrix

Los sistemas de control HIMatrix podrán equiparse con submódulos de bus de campo tal y como se indica en la siguiente tabla:

Sistema de control	FB1	FB2	FB3
F20	Libremente equipable	RS485 incorporado	---
F30	Libremente equipable	Libremente equipable	RS485 incorporado
F35	Libremente equipable	Libremente equipable	RS485 incorporado
F60	Libremente equipable	Libremente equipable	---

Tabla 24: Equipamiento de sistemas de control HIMatrix con submódulos de bus de campo

Al elegir un respectivo submódulo de bus de campo cambia el número de referencia:

98 22XY...

**X:** Opción para interfaz de bus de campo FB1 según Tabla 21

**Y:** Opción para interfaz de bus de campo FB2 según Tabla 21

La siguiente tabla contiene ejemplos de números de referencia:

Nº de referencia	Sistema de control	FB1	FB2
98 2210417	F20 01	RS485	---
98 2232415	F30 01	PROFIBUS Slave	PROFIBUS Master
98 2232472	F30 01 SILworX	PROFIBUS Slave	PROFIBUS Master
98 2242416	F35 01	INTERBUS Master	PROFIBUS Master
98 2232497	F35 03 SILworX	PROFIBUS Slave	PROFIBUS Master
98 2212126	F60 CPU 01	RS485	PROFIBUS Master
98 2212137	F60 CPU 01 SILworX	RS485	PROFIBUS Master
98 0012139	F60 CPU 03 SILworX	RS485	PROFIBUS Master

Tabla 25: Ejemplos de números de referencia de los sistemas de control HIMatrix

### 3.6.2 Registro y activación

Las opciones de comunicación se activarán según los submódulos de bus de campo. Véase el capítulo 3.4.

### 3.6.3 Asignación de pins de las conexiones D-Sub

En las tablas subsiguientes se describe la asignación de pins de las interfaces de bus de campo según el submódulo de bus de campo que se haya montado.

#### 3.6.3.1 Submódulo de bus de campo RS485 para Modbus Master, Slave o ComUserTask

Conexión	Señal	Función
1	---	---
2	RP	5 V, desacoplado con diodos
3	RxD/TxD-A	Datos de recepción/envío A
4	CNTR-A	Señal de control A
5	DGND	Potencial de referencia de datos
6	VP	5 V, tensión de alimentación polo +
7	---	---
8	RxD/TxD-B	Datos de recepción/envío B
9	CNTR-B	Señal de control B

Tabla 26: Asignación de pins de las conexiones D-Sub para RS485

### 3.6.3.2 Submódulo de bus de campo PROFIBUS DP Master o Slave

Conexión	Señal	Función
1	---	
2	---	
3	RxD/TxD-A	Datos de recepción/envío A
4	RTS	Señal de control
5	DGND	Potencial de referencia de datos
6	VP	5 V, tensión de alimentación polo +
7	---	---
8	RxD/TxD-B	Datos de recepción/envío B
9	---	---

Tabla 27: Asignación de pins de las conexiones D-Sub para PROFIBUS DP

### 3.6.3.3 Submódulo de bus de campo para INTERBUS

INTERBUS está disponible sólo para sistemas de control HIMatrix con sistema operativo anterior a CPU BS V7 y COM BS V12. Compatible sólo con la utilidad de programación ELOP II Factory.

Conexión	Señal	Función
1	DO	Salida de datos positiva
2	DI	Entrada de datos positiva
3	COM	Conductor común de 0 V
4	---	---
5	---	---
6	DO-	Entrada de datos negativa
7	DI-	Salida de datos negativa
8	---	---
9	---	---

Tabla 28: Asignación de pins de las conexiones D-Sub para INTERBUS

### 3.6.3.4 Submódulo de bus de campo RS232 para ComUserTask

Conexión	Señal	Función
1	---	---
2	TxD	Datos de envío
3	RxD	Datos de recepción
4	---	---
5	DGND	Potencial de referencia de datos
6	---	---
7	RTS	Requerimiento de envío (Request to Send)
8	---	---
9	---	---

Tabla 29: Asignación de pins de las conexiones D-Sub para RS232

### 3.6.3.5 Submódulo de bus de campo RS422 para ComUserTask

Conexión	Señal	Función
1	---	---
2	RP	+5 V desacoplado con diodos
3	RxA	Datos de recepción A
4	TxA	Datos de envío A
5	DGND	Potencial de referencia de datos
6	VP	+5 V tensión de alimentación
7	---	---
8	RxB	Datos de recepción B
9	TxB	Datos de envío B

Tabla 30: Asignación de pins de las conexiones D-Sub para RS422

### 3.6.3.6 Submódulo de bus de campo para SSI

Conexión	Señal	Función
1	D2+	Entrada de datos, canal 2+
2	D1-	Entrada de datos, canal 1-
3	CL2+/D3+	Salida de impulso de desplazamiento canal 2+ o entrada de datos canal 3+
4	CL1+	Salida de impulso de desplazamiento canal 1+
5	GND	Potencial de referencia
6	D1+	Entrada de datos, canal 1+
7	D2-	Entrada de datos, canal 2-
8	CL2-/D3-	Salida de impulso de desplazamiento canal 2- o entrada de datos canal 3-
9	CL1-	Salida de impulso de desplazamiento canal 1-

Tabla 31: Asignación de pins de las conexiones D-Sub para SSI

## 4 safeethernet

Todos los sistemas HIMax/HIMatrix tienen capacidad **safeethernet**. Los mismos pueden comunicar conforme a SIL 3 de forma orientada a la seguridad a través de Ethernet (HIMax 1 Gbit/s, HIMatrix 100 Mbit/s).

Las respectivas interfaces Ethernet de los módulos CPU, COM HIMax y los sistemas de control HIMatrix pueden usarse simultáneamente también para otros protocolos.

La comunicación **safeethernet** entre los sistemas de control puede tener lugar mediante diversas topologías de red de Ethernet. Para aumentar la eficacia y la velocidad de transmisión de datos, adapte Ud. los parámetros del protocolo **safeethernet** a la red Ethernet que utilice.

Estos parámetros pueden ajustarse con ayuda de los denominados perfiles de red. Los ajustes originales de fábrica de los parámetros aseguran la comunicación sin que el usuario tenga que estudiar desde el principio los detalles de la configuración de red.



El protocolo **safeethernet** está orientado a la seguridad y certificado por el ente de inspección TÜV hasta SIL 3 conforme a IEC 61508.

### Equipamiento y sistema necesarios:

Sistema de control HIMA	HIMax con módulo de CPU HIMatrix L2/L3
Activación	Esta función se sirve habilitada por defecto en todos los sistemas HIMax/HIMatrix.

### safeethernet (propiedades):

Elemento	HIMax	HIMatrix L3	HIMatrix L2	Descripción
Módulo o sistema de control requerido	Por HIMax máx. 4 módulos procesadores	Módulo procesador integrado del sistema de control	Módulo procesador integrado del sistema de control	<b>safeethernet</b> se ejecuta en el módulo procesador orientado a la seguridad.
Interfaces Ethernet:	Módulo de CPU 1 Gbit/s Módulo COM 100 Mbit/s	100 Mbit/s	100 Mbit/s	Las interfaces Ethernet utilizadas pueden usarse simultáneamente también para otros protocolos.
Conexiones:	Por HIMax 255	128	64	Conexiones <b>safeethernet</b>
Conexiones redundantes:	Por HIMax 255	128	32 Remote I/O: n. a	Funcionamiento a 2 canales Las conexiones <b>safeethernet</b> redundantes entre sistemas de control HIMax/HIMatrix pueden configurarse en el editor de <b>safeethernet</b> .
Vías de transmisión redundantes	Mediante 2 módulos separados respectivamente	Restricciones por sólo haber un dispositivo	Restricciones por sólo haber un dispositivo	Vías de transmisión <b>safeethernet</b> redundantes.
Volumen de datos de proceso por conexión	1100 bytes	1100 bytes	900 bytes	Por conexión <b>safeethernet</b> .
n. a: no aplicable				

Tabla 32: Protocolo con función orientada a la seguridad (**safeethernet**)

- 
- i** ¡Comunicación trascendente a todo el proyecto!  
Las conexiones **safeethernet** a otro recurso en otro proyecto o en un sistema de control HIMatrix con sistemas operativos anteriores a CPU BS V7 y COM BS V12 pueden configurarse en SILworX. Véase el capítulo 4.8.
- 

## 4.1 Qué es **safeethernet**

En el ámbito de la tecnología de automatización y control de procesos, requisitos como determinismo, fiabilidad, disponibilidad, ampliabilidad y, sobre todo, seguridad son temas centrales.

**safeethernet** es un protocolo para la transmisión de datos orientados a la seguridad hasta el nivel SIL 3 basado en la tecnología Ethernet.

**safeethernet** contiene mecanismos que detectan los siguientes errores y reaccionan frente a ellos de forma orientada a la seguridad:

- Falseamiento de datos transmitidos (bits duplicados, perdidos, alterados)
- Direcciónamiento incorrecto de notificaciones (emisor, receptor)
- Orden incorrecto de datos (repetición, pérdida, sustitución)
- Tiempos incorrectos (retardo, eco)

**safeethernet** toma como base la norma IEEE 802.3.

La transmisión de datos con relevancia para la seguridad utiliza el marco del protocolo Ethernet estándar.

**safeethernet** usa “canales inseguros de transmisión de datos” (Ethernet) según el principio de Black Channel y los monitorea en el emisor y el receptor mediante mecanismos de protocolo orientados a la seguridad. Esto permite utilizar componentes de red Ethernet como concentradores, comutadores y enruteadores (hubs, switches, routers) dentro de una red con función orientada a la seguridad.

**safeethernet** usa las capacidades del Ethernet estándar de forma que se garantice la seguridad y el funcionamiento a tiempo real. Un mecanismo especial de protocolo asegura un comportamiento determinista, incluso en caso de fallo o acceso de participantes de comunicación. Entonces el sistema integrará automáticamente los nuevos componentes en el sistema. Todos los componentes de una red pueden sustituirse sin interrumpir el funcionamiento. El empleo de switches permite definir claramente los tiempos de transmisión. Esto hace que Ethernet pueda funcionar a tiempo real.

También la máxima velocidad posible de transmisión de hasta Gbit/s para datos con función orientada a la seguridad está por encima de los estándares habituales. Como medios de transmisión pueden usarse p. ej. cables de cobre o fibra óptica.

Las conexiones a la Intranet propia de la empresa así como las conexiones a Internet son posibles con **safeethernet**. De esta forma se necesitará una sola red para la transmisión de datos segura y no segura.

- 
- i** La red podrá ser utilizada por otros participantes, siempre y cuando se disponga de suficiente capacidad de transmisión.
-

## ⚠ PELIGRO



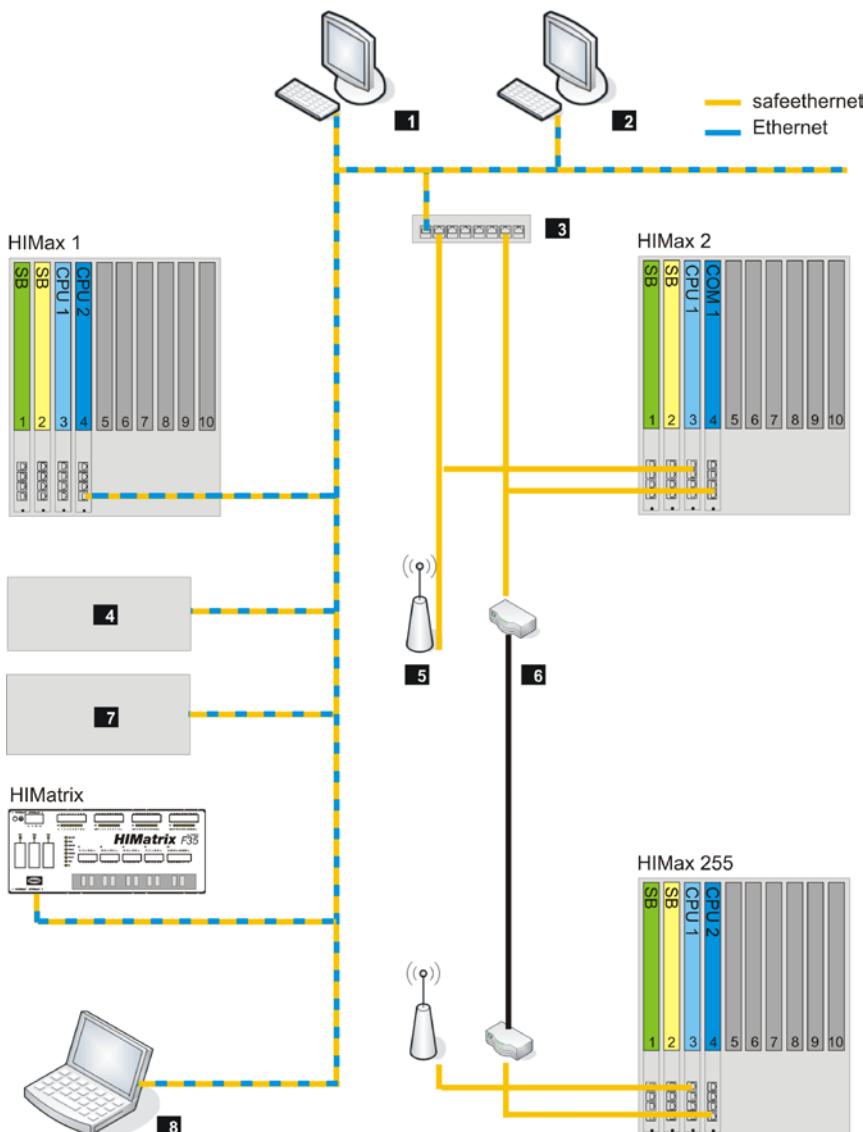
¡Manipulación de la transmisión de datos orientada a la seguridad!

¡Riesgo de daños personales!

La empresa usuaria deberá tomar las medidas necesarias para que el Ethernet empleado para safeethernet esté suficientemente protegido contra manipulaciones (p. ej. hackers).

El tipo y la cuantía de las medidas deberán acordarse con el ente de inspección oficial que deba emitir su aprobación.

safeethernet permite estructuras de sistema flexibles para la automatización descentralizada con tiempos definidos de reacción. Según los requisitos, podrá Ud. distribuir la inteligencia alternativamente de forma central o descentralizada entre los participantes dentro de la red.



**1** PC (sistema de control central DCS)

**2** PADT (SILworX)

**3** Switch

**4** Sistema de control central DCS

**5** Radio, satélite, WLAN

**6** ISDN, DSL, fibra óptica

**7** PLC estándar

**8** PADT (ELOP II Factory)

Fig. 1: Estructuras del sistema

- 
- i ¡Es posible una transición involuntaria al estado seguro!  
Al ensamblar hay que observar que no se creen bucles de red. Los paquetes de datos deberán llegar al sistema de control por una única vía.
-

## 4.2 Configuración de una conexión safeethernet redundante

En este ejemplo se configura una conexión HIMax/HIMax safeethernet redundante.

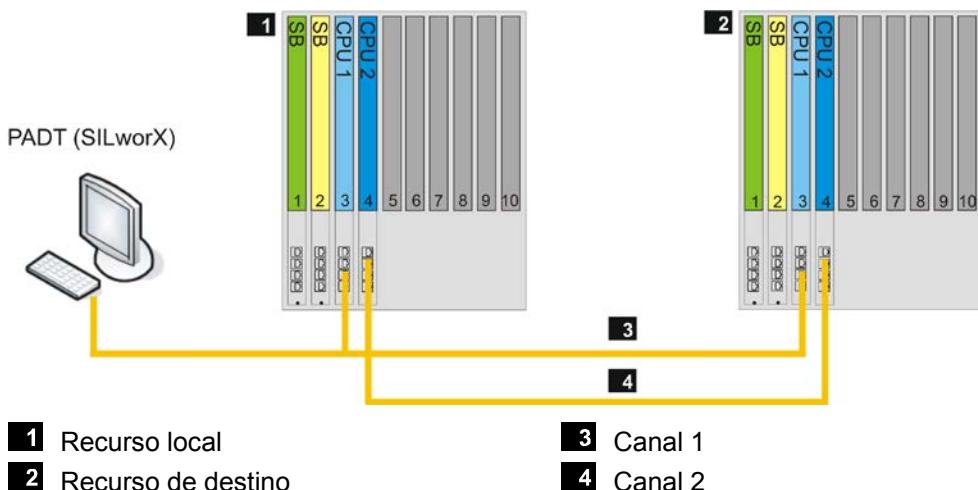


Fig. 2: Disposición para configurar una conexión redundante



Para una conexión safeethernet redundante, HIMA recomienda tender ambas vías de transmisión (canal 1 y canal 2) mediante módulos de comunicación separados. En tal caso, el ancho de banda y el retardo de las correspondientes vías de transmisión deberán ser prácticamente idénticos.

### Creación de una conexión safeethernet

Cree en el editor de safeethernet una conexión safeethernet entre el recurso local y el recurso de destino.

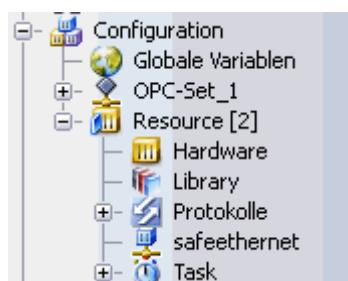


Fig. 3: Árbol del recurso

#### Para abrir el editor de safeethernet del recurso local:

1. Abra Configuration y Resource en el árbol.
  2. Haga clic con el botón derecho en safeethernet y seleccione Edit en el menú contextual.
- En la selección de objetos se encuentran los recursos de destino.

#### Para crear la conexión safeethernet al recurso de destino:

1. En la selección de objetos haga clic en el recurso de destino (**Target Resource**) y arrástrelo con el ratón hasta un lugar libre del área de trabajo del editor de safeethernet.

- i** La ruta de comunicación inversa se asignará automáticamente al recurso de destino en el editor de safeethernet.

#### Para configurar la conexión safeethernet:

1. Seleccione **Ethernet Interfaces Channel 1** del recurso local y del recurso de destino.
2. Seleccione **Ethernet Interfaces Channel 2** del recurso local y del recurso de destino.
3. Seleccione **Network Profile** (p. ej. Fast&Noisy) de la conexión safeethernet.
4. Calcule e introduzca los tiempos **Receive Timeout** y **Response Time** (véase el capítulo 4.6).

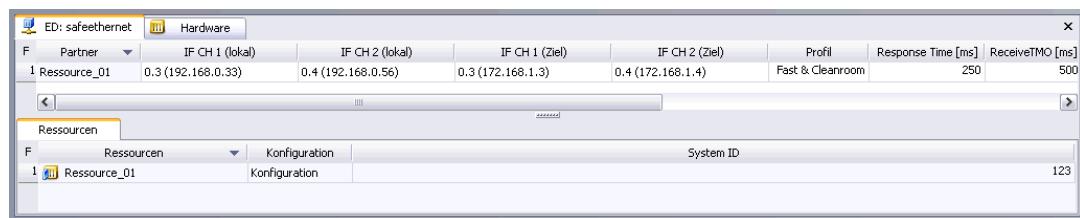


Fig. 4: Valores de los parámetros de una conexión safeethernet redundante

#### 4.2.1 Conexión de variables de proceso

##### Para abrir la vista en detalle de una conexión safeethernet:

Requisitos: El editor de safeethernet del recurso local está abierto.

1. Abra el menú contextual del recurso de destino (**Target Resource**) haciendo clic en él con el botón derecho.
2. Seleccione **Detail View**.
3. Seleccione la ficha **Resource (target) <-> Resource (local)**.

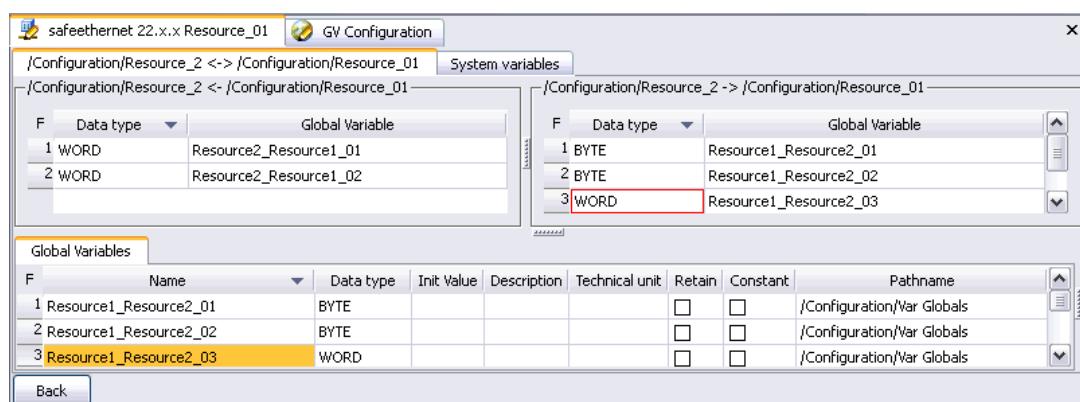


Fig. 5: Vista en detalle del editor de safeethernet

- i** ¡Se podrán usar sólo variables globales del contexto de la configuración, no del contexto del recurso!

**Para agregar variables de envío de safeethernet:**

Las variables de envío se transmiten desde el recurso local al recurso de destino

1. Seleccione el área **Resource (target) -> Resource (local)**.
2. Elija en la selección de objetos una **Global Variable** y arrástrela con el ratón a la columna **Resource (target) -> Resource (local)**.
3. Repita este paso para las demás variables de envío de safeethernet.

**Para agregar variables de recepción de safeethernet:**

Las variables de recepción las recibe el recurso local.

1. Seleccione el área **Resource (target) <- Resource (local)**.
2. Elija en la selección de objetos una **Global Variable** y arrástrela con el ratón a la columna **Resource (target) <- Resource (local)**.
3. Repita este paso para las demás variables de recepción de safeethernet.

**Verifique la conexión safeethernet:**

1. Abra **Configuration, Resource, safeethernet** en el árbol.
2. Haga clic en **Verification** en la barra de acciones y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.



La configuración de la conexión safeethernet deberá recompilarse con el programa de usuario del recurso y del recurso de destino y transmitirse a los sistemas de control, antes de que sea efectiva para la comunicación de HIMax.

#### 4.2.2 Prueba de la comunicación safeethernet

Ponga a cero los indicadores *Wrong Messages* y *Resends* en el panel de control.

1. Haga funcionar el sistema HIMax a plena carga:
  - Todos los protocolos de comunicación estarán en uso (safeethernet y protocolos estándar).
  - Retire los módulos procesadores y vuelva a colocarlos como se describe en el capítulo 4.6.1.
  - Cargue el programa de usuario mediante Reload (modificación de safeethernet; la configuración por Reload no es posible).



Para comprobar la correcta creación de una conexión safeethernet redundante debería Ud. separar una conexión redundante y volver a añadirla, y repetir luego esta operación con la otra. Al hacerlo no deberá producirse ningún error en la comunicación safeethernet.

2. Compruebe en los paneles de control de ambos sistemas de control los indicadores *Wrong Messages* y *Resends*.  
Si los contadores de *Bad Messages* y *Resends*  
= **0**, la configuración de safeethernet estará bien.  
≥ **0**, deberá volver a probarse la configuración safeethernet:
  - Recalcule el tiempo *Receive Timeout* con el máximo tiempo de ciclo.  
Véase el capítulo 4.6.1 y 4.6.3.
  - Varíe el tiempo *Response Time* tal y como se describe en el capítulo 4.6.4.

- 
- i** ¡Otras causas de *bad messages* y *resends*!  
Compruebe la correcta disposición de la red (p. ej. cables, switches, PCs).  
Si la red Ethernet no se usa exclusivamente para **safeethernet**, habrá que comprobar además la tasa de solicitud de la red (probabilidad de colisiones de datos).
-

#### 4.3 Editor de safeethernet

En el editor de safeethernet podrá Ud. crear y configurar las conexiones safeethernet a los interlocutores de comunicación (recursos).

**Para abrir el editor de safeethernet del recurso local:**

1. Abra **Configuration** y **Resource** en el árbol.
  2. Haga clic con el botón derecho en safeethernet y seleccione Edit en el menú contextual.
- El editor de safeethernet contiene el área de trabajo y la selección de objetos.

En el editor de safeethernet podrá Ud. crear y configurar las conexiones safeethernet a los interlocutores de comunicación (recursos). Para ello, arrastre los recursos desde la selección de objetos hasta el área de trabajo.

Para configurar la conexión safeethernet deberá Ud. ajustar los siguientes parámetros de protocolo safeethernet:

Parámetro	Descripción
Partner	Nombre de recurso del miembro vinculado
IF CH...	Interfaces Ethernet disponibles en el recurso local y el recurso de destino. Véase también el capítulo 3.5.
Profile	Combinación de parámetros safeethernet adecuados unos a otros. Véase también el capítulo 4.7.8.
Response Time [ms]	Tiempo hasta llegar confirmación de recepción de una notificación al emisor. Véase también el capítulo 4.6.4.
Receive Timeout [ms]	Tiempo de monitoreo en el PES1, dentro del cual deberá recibirse una respuesta correcta del PES2. Véase también el capítulo 4.6.3.
Resend Timeout [ms]	Tiempo de monitoreo en el PES1, dentro del cual el PES2 deberá haber confirmado la recepción de un paquete de datos. De no ser así, se repetirá el envío del paquete. Véase también el capítulo 4.6.6.
Acknowledge Timeout [ms]	Tiempo tras el cual la CPU deberá haber confirmado como muy tarde un paquete de datos como recibido. Véase también el capítulo 4.6.7.
Prod. Rate	Production Rate: Mínimo intervalo entre dos paquetes de datos. Véase también el capítulo 4.6.8.
Queue	Cantidad de paquetes de datos que pueden enviarse sin confirmación de recepción. Véase también el capítulo 4.6.9.

Parámetro	Descripción	
Freeze Values on Lost Connection [ms]	<p>Reacción de las variables de entrada de esta conexión safeethernet en caso de interrumpirse la conexión.</p> <p>Use initial values      Para las variables de entrada se usarán los valores iniciales.</p> <p>Not limited      Las variables de entrada se congelarán en su valor momentáneo y se usarán hasta que vuelva a establecerse la conexión.</p> <p>Limited      Introducción: haga doble clic en el recuadro desplegable y escriba el tiempo. Las variables de entrada se congelarán en su valor momentáneo y se usarán hasta el time out parametrizado. Despues se usarán los valores iniciales. El time out podrá prolongarse hasta un ciclo de CPU.</p>	
	<p><b>⚠ PRECAUCIÓN</b></p> <p>Para las funciones orientadas a la seguridad implementadas mediante safeethernet sólo se permite utilizar la opción <i>Use Initial Data</i>.</p>	
Fragments per Cycle	<p>Ajuste fijo: Al interlocutor de comunicación se le transmitirá un fragmento por ciclo del sistema de control.</p> <p>Un fragmento de un sistema HIMax es un fragmento <math>\leq 1100</math> bytes Un fragmento de un sistema HIMatrix es un fragmento <math>\leq 900</math> bytes</p>	
Priority of Events	<p>Esta función está activa sólo para la conexión al servidor X-OPC. Con ella se define con qué prioridad requerirá el servidor X-OPC eventos desde el sistema de control.</p> <p>Los fragmentos de prioridad <b>n</b> y los fragmentos de prioridad <b>m</b> se envían en una relación de <b>n</b> a <b>m</b>.</p>	
Priority of State Values	<p>Esta función está activa sólo para la conexión al servidor X-OPC. Con ella se define con qué prioridad requerirá el servidor X-OPC valores de estado desde el sistema de control.</p> <p>Los fragmentos de prioridad <b>n</b> y los fragmentos de prioridad <b>m</b> se envían en una relación de <b>n</b> a <b>m</b>.</p>	
Number of Ignored Warnings	<p>Es la cantidad de advertencias que deben producirse consecutivamente dentro del período <i>Warning Period [ms]</i> hasta que éstas pasan al diagnóstico o a la estadística de errores de comunicación.</p>	
Warnings Period [ms]	<p>0 ms es el único valor admisible por el momento.</p>	
Enable SOE	<p>Valor por defecto: Desactivado.</p>	

Tabla 33: Parámetros de protocolo safeethernet

### Selección de objetos

La selección de objetos ofrece todos los recursos del proyecto a los que puede conectarse ese recurso mediante safeethernet.

- i Para conexiones safeethernet a recursos fuera de un proyecto o a un sistema de control HIMatrix (proyectado en ELOP II Factory) se dispone de una función de exportación (véase el capítulo 4.8).

## 4.4 Vista en detalle del editor de safeethernet

La vista en detalle **Detail View** se referirá siempre al recurso local para el cual se haya iniciado el editor de safeethernet.

### Para abrir la vista en detalle de una conexión safeethernet:

1. Haga clic con el botón derecho en la conexión safeethernet para abrir el menú contextual.
2. Haga clic en **Detail View**.
  - Detail View** contiene las fichas **System Variables**, **Fragment Definitions** y **Resource (local) <-> Resource (target)**.

### 4.4.1 Ficha: Variables de sistema

Con ayuda de las variables de sistema podrá Ud. controlar y evaluar la conexión safeethernet en el programa de usuario.

Variable del sistema	Descripción										
Ack. Frame No.	Contador de recepción (rotativo).										
Number of bad messages	Cantidad de todas las notificaciones defectuosas por cada canal (CRC incorrecto, cabecera incorrecta u otros errores).										
Number of bad messages for the redundant channel	Cantidad de conexiones correctas tras reinicializar la estadística.										
Number of successful connections	Cantidad de las notificaciones perdidas en una de ambas vías de transmisión desde que se reinicializó la estadística. El contador se comprueba sólo hasta el fallo completo de un canal.										
Number of lost messages	Cantidad de notificaciones ubicadas en la cola Early Queue desde que se reinicializó la estadística. Véase también el capítulo 4.6.9.										
Number of lost messages for the redundant channel	Cantidad de notificaciones desecharadas tras reinicializar la estadística.										
Early Queue Usage	Estado actual del canal 1. Se trata del estado actual del canal 1 hasta el momento (Seq-No X-1) de recibirse una notificación con el númer Seq-No X.										
Bad messages	Signatura del layout de datos utilizado en la comunicación.										
Frame No.	Contador de envío (rotativo).										
Channel state	<table border="1"> <thead> <tr> <th>Estado</th><th>Descripción</th></tr> </thead> <tbody> <tr> <td>0</td><td>Sin notificación de estado del canal 1.</td></tr> <tr> <td>1</td><td>Canal 1 bien.</td></tr> <tr> <td>2</td><td>Última notificación defectuosa, la actual está bien.</td></tr> <tr> <td>3</td><td>Error en el canal 1.</td></tr> </tbody> </table>	Estado	Descripción	0	Sin notificación de estado del canal 1.	1	Canal 1 bien.	2	Última notificación defectuosa, la actual está bien.	3	Error en el canal 1.
Estado	Descripción										
0	Sin notificación de estado del canal 1.										
1	Canal 1 bien.										
2	Última notificación defectuosa, la actual está bien.										
3	Error en el canal 1.										
Layout Version	Signatura del layout de datos utilizado en la comunicación.										

Variable del sistema	Descripción																			
Last channel latency	La latencia del canal especifica el retardo entre ambas vías de transmisión redundantes al momento de recibirse notificaciones con idéntico número SeqNo.																			
Last latency of the red. channel	A este propósito se realiza una estadística de latencia media, mínima, máxima y más reciente.																			
Max. channel latency	Si se da “valor mín.” > “valor máx.”, los valores de la estadística no serán válidos.																			
Max. latency of the red. channel	Los valores de latencia más reciente de canal y de latencia media serán entonces 0.																			
Min. channel latency																				
Min. latency of the red. channel																				
Average channel latency																				
Average latency of the red. channel																				
Monotony	Contador de envío de datos útiles (rotativo).																			
New Layout Version	Signatura del nuevo layout de datos.																			
Quality of Channel 1	Estado de la vía principal de transmisión. <table border="1"> <tr> <th>Bit N.º</th> <th>Bit = 0</th> <th>Bit = 1</th> </tr> <tr> <td>0</td> <td>Vía de transmisión no habilitada</td> <td>Vía de transmisión habilitada</td> </tr> <tr> <td>1</td> <td>Vía de transmisión no utilizada</td> <td>Vía de transmisión activamente utilizada</td> </tr> <tr> <td>2</td> <td>Vía de transmisión no conectada</td> <td>Vía de transmisión conectada</td> </tr> <tr> <td>3</td> <td>-</td> <td>Vía de transmisión transmite notificación primero</td> </tr> <tr> <td>4-7</td> <td>Reservado</td> <td>Reservado</td> </tr> </table>		Bit N.º	Bit = 0	Bit = 1	0	Vía de transmisión no habilitada	Vía de transmisión habilitada	1	Vía de transmisión no utilizada	Vía de transmisión activamente utilizada	2	Vía de transmisión no conectada	Vía de transmisión conectada	3	-	Vía de transmisión transmite notificación primero	4-7	Reservado	Reservado
Bit N.º	Bit = 0	Bit = 1																		
0	Vía de transmisión no habilitada	Vía de transmisión habilitada																		
1	Vía de transmisión no utilizada	Vía de transmisión activamente utilizada																		
2	Vía de transmisión no conectada	Vía de transmisión conectada																		
3	-	Vía de transmisión transmite notificación primero																		
4-7	Reservado	Reservado																		
Quality of Channel 2	Estado de la vía redundante de transmisión. Véase el estado del canal 1 (vía principal).																			
Receive Timeout	Tiempo (ms) en el PES1, dentro del cual deberá recibirse una respuesta válida del PES2. Véase también el capítulo 4.6.3.																			
Response Time	Tiempo en milisegundos (ms) hasta llegar confirmación de recepción de una notificación al emisor. Véase también el capítulo 4.6.4.																			
Reset safeethernet statistics	Reinicialización de los valores estadísticos de la conexión de comunicación en el programa de usuario (p. ej. Number of Bad Messages, Channel State, Timestamp for the Last Fault on the Red. Channel [s], Resends). <table border="1"> <tr> <th>Valor</th> <th>Función</th> </tr> <tr> <td>0</td> <td>Sin reinicialización</td> </tr> <tr> <td>1-255</td> <td>Reinicialización de estadística de safeethernet</td> </tr> </table>		Valor	Función	0	Sin reinicialización	1-255	Reinicialización de estadística de safeethernet												
Valor	Función																			
0	Sin reinicialización																			
1-255	Reinicialización de estadística de safeethernet																			
Transmission Control Ch1	Control de transmisión del canal 1 <table border="1"> <tr> <th>Bit 0</th> <th>Función</th> </tr> <tr> <td>FALSE</td> <td>Vía de transmisión habilitada</td> </tr> <tr> <td>TRUE</td> <td>Vía de transmisión bloqueada</td> </tr> </table> <table border="1"> <tr> <th>Bit 1</th> <th>Función</th> </tr> <tr> <td>FALSE</td> <td>Vía de transmisión habilitada para pruebas</td> </tr> <tr> <td>TRUE</td> <td>Vía de transmisión bloqueada</td> </tr> </table> Bit 2...7 reservado.		Bit 0	Función	FALSE	Vía de transmisión habilitada	TRUE	Vía de transmisión bloqueada	Bit 1	Función	FALSE	Vía de transmisión habilitada para pruebas	TRUE	Vía de transmisión bloqueada						
Bit 0	Función																			
FALSE	Vía de transmisión habilitada																			
TRUE	Vía de transmisión bloqueada																			
Bit 1	Función																			
FALSE	Vía de transmisión habilitada para pruebas																			
TRUE	Vía de transmisión bloqueada																			
Transmission Control Ch2	Control de transmisión del canal 2. Véase el control de transmisión del canal 1.																			

Variable del sistema	Descripción	
Connection control	Con esta variable de sistema, la conexión safeethernet puede ser controlada por el programa de usuario.	
	Comando	Descripción
	Autoconnect (0x0000)	Valor por defecto: En caso de perderse la comunicación safeethernet, el sistema de control intentará volver a establecer la conexión en el siguiente ciclo de CPU.
	Toggle Mode 0 (0x0100) Toggle Mode 1 (0x0101)	En caso de perderse la comunicación, ésta podrá restablecerse mediante un cambio controlado por programa del Toggle Modus. <ul style="list-style-type: none"> <li>▪ TOGGLE MODE_0 (0x100) aplicado: Aplique TOGGLE MODE 1 (0x101) para volver a establecer la conexión.</li> <li>▪ TOGGLE MODE 1 (0x101) aplicado: Aplique TOGGLE_MODE_0 (0x100) para volver a establecer la conexión.</li> </ul>
	Disabled (0x8000)	Comunicación safeethernet inhabilitada.
Connection State	"Connection State" evalúa el estado de la comunicación entre dos sistemas de control en el programa de usuario.	
	Estado/Valor	Descripción
	Closed (0)	La conexión está cerrada y no se intenta abrirla.
	Try_open (1)	Se intenta abrir la conexión, pero aún está sin abrir. Este estado es válido tanto para la parte activa como para la parte pasiva.
	Connected (2)	Conexión establecida y en funcionamiento (monitoreo activo de tiempo e intercambio de datos).
Repeats	Cantidad de repeticiones tras reinicializar la estadística.	
Timestamp for the last fault on the red. channel [ms]	Fracción en milisegundos de la marca de tiempo (hora actual del sistema).	
Timestamp for the last fault on the red. channel [s]	Fracción en segundos de la marca de tiempo (hora actual del sistema).	
Timestamp for the last fault [ms]	Fracción en milisegundos de la marca de tiempo (hora actual del sistema).	
Timestamp for the last fault [s]	Fracción en segundos de la marca de tiempo (hora actual del sistema).	
State of the red. channel	Estado actual del canal 2. Se trata del estado actual del canal 2 hasta el momento (Seq-No X-1) de recibirse una notificación con el núm. Seq-No X.	
	Estado	Descripción
	0	Sin notificación de estado del canal 2.
	1	Canal 2 bien.
	2	Última notificación defectuosa, la actual está bien.
	3	Error en el canal 2.

Tabla 34: Ficha de variables de sistema del editor de safeethernet

#### 4.4.2 Ficha: Fragment Definitions

Véase el capítulo 10.6.8.2.

### 4.5 Posibles conexiones safeethernet

Usted podrá establecer conexión safeethernet entre dos sistemas de control HIMax en modo mono o redundante.

Las interfaces Ethernet disponibles para una conexión safeethernet se referirán siempre al recurso (local) para el cual se haya iniciado el editor de safeethernet.

Todas las interfaces Ethernet disponibles de un sistema de control se muestran en los menús desplegables de los parámetros **IF CH....**

Elemento	Descripción
IF CH1 (local)	Interfaz Ethernet (canal 1) del recurso
IF CH2 (local)	Interfaz Ethernet (canal 2) del recurso
IF CH1 (target)	Interfaz Ethernet (canal 1) del miembro vinculado
IF CH2 (target)	Interfaz Ethernet (canal 2) del miembro vinculado

Tabla 35: Interfaces Ethernet disponibles

#### 4.5.1 Conexión safeethernet en modo mono (canal 1)

Para una conexión mono deberá Ud. aplicar las interfaces Ethernet *IF Ch1 (local)* e *IF Ch1 (target)* en el recurso local.

#### 4.5.2 Conexión safeethernet redundante (canal 1 y canal 2)

Son posibles vías de transmisión safeethernet redundantes entre dos sistemas de control HIMax/HIMax.



El ancho de banda y el retardo de las vías de transmisión redundantes deberán ser prácticamente idénticos.

Si en una vía de transmisión los mensajes recibidos llegan demasiado desincronizados o si los mensajes se retrasan en más que el tiempo "Response-Time", el diagnóstico de las vías de transmisión no operará en la forma prevista e interpretará estos retrasos como errores de la vía de transmisión.

Para la evaluación del diagnóstico de vías de transmisión véase la variable de sistema *State of the Red. Channel* y *Channel State*.

Para una conexión redundante se dispone de las siguientes interfaces Ethernet:

- Las interfaces Ethernet *IF CH1 (local)* e *IF CH1 (target)* para el canal 1.
- Las interfaces Ethernet *IF CH2 (local)* e *IF CH2 (target)* para el canal 2.
- Para una conexión redundante a través del canal1 y el canal2 mediante una interfaz Ethernet, seleccione en el editor de safeethernet la misma interfaz Ethernet en el canal 1 *IF CH1 (local)* y en el canal 2 *IF CH2 (local)*.



La ruta de comunicación inversa se asignará automáticamente al recurso de destino en el editor de safeethernet.

#### 4.5.3 Combinaciones permitidas

En la siguiente tabla se representan las combinaciones posibles para una conexión safeethernet redundante.

Canal 1 IF Ch1 (local)/IF Ch1 (target)	Canal 2 IF Ch2 (local)/IF Ch2 (target)
CPU1/CPU1	CPU2/CPU2
CPU1/CPU1	CPU1/CPU1
COM1/COM1	COM2/COM2
CPU1/COM1	CPU2/COM2
CPU1/COM2	CPU2/COM1
CPU1/CPU1	COM1/COM1

Tabla 36: Combinaciones para conexiones safeethernet

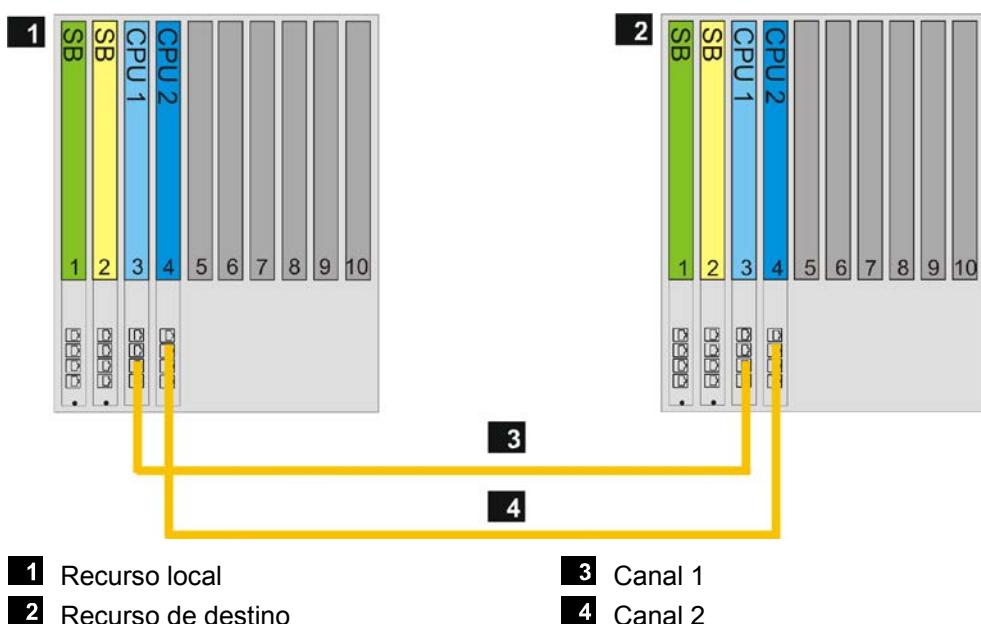


Fig. 6: Conexión redundante entre dos sistemas de control HIMax

La siguiente conexión configurada redundante (mediante un cable) es conveniente de cara a un volumen constante de datos, a pesar de los paquetes de datos perdidos (p. ej. por perturbaciones de CEM).

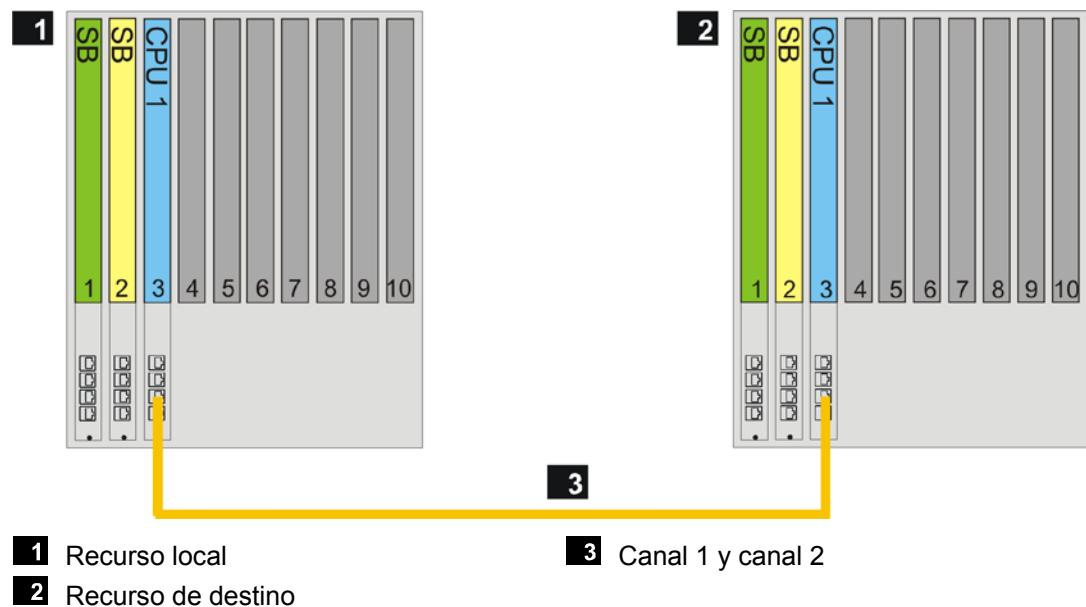


Fig. 7: Conexión redundante de dos sistemas de control HIMax mediante un cable

## 4.6 Parámetros de safeethernet

La comunicación orientada a la seguridad se configura en el editor de safeethernet. Para ello deberá Ud. parametrizar los parámetros descritos en este capítulo.

Para el cálculo de parámetros safeethernet *Receive Timeout* y *Response Time* rige la siguiente condición:

El intervalo de tiempo de comunicación deberá ser lo suficientemente grande como para poder procesar en un ciclo de la CPU todas las conexiones safeethernet.

Véase el capítulo 13.2.

### 4.6.1 Tiempo máximo de ciclo (mínimo tiempo de Watchdog) del sistema de control HIMax

Para determinar el tiempo máximo de ciclo (mínimo tiempo de WatchDog) de un sistema de control HIMax, HIMA recomienda el siguiente procedimiento en un sistema completo con todos los módulos procesadores colocados.

1. Ajuste un tiempo de WatchDog alto para la prueba.
2. Haga funcionar el sistema a plena carga. Deberán estar en uso todas las conexiones de comunicación, tanto mediante safeethernet como mediante protocolos estándares. Lea frecuentemente el tiempo de ciclo en el panel de control y anote las fluctuaciones del tiempo del ciclo.
3. Retire, uno tras otro, cada módulo procesador y vuelva a colocarlos en el rack. Antes de retirar cada módulo procesador, aguarde hasta que el módulo procesador recién agregado se haya sincronizado.



Al agregar un módulo procesador, éste se sincronizará automáticamente con la configuración de los módulos procesadores existentes. El tiempo necesario para la sincronización prolongará el ciclo del sistema de control hasta el máximo tiempo de ciclo.

El tiempo necesario para la sincronización aumentará cuanto mayor sea la cantidad de los módulos procesadores ya sincronizados.

La instalación y el desmontaje de un módulo procesador se describen en el manual X-CPU 01, HI 801 208 S.

4. Lea en el historial de diagnóstico el tiempo de sincronización de n a n+1 módulos procesadores en cada sincronización.
5. Repita estos pasos para el interlocutor de comunicación (segundo sistema de control HiMax). Con el mayor de estos tiempos de sincronización se determinará el tiempo de WatchDog.



Anote los tiempos de sincronización de ambos sistemas de control HiMax.

6. El tiempo mínimo de WatchDog se calcula a partir de:  
mayor tiempo de sincronización + 12 ms de reserva + reserva para las fluctuaciones observadas.

Así se calcula un valor de consigna adecuado para el tiempo de ciclo máximo (mínimo tiempo de WatchDog) para los siguientes cálculos.

**SUGERENCIA** Realice el cálculo del paso 6 para ambos sistemas de control HiMax con los tiempos de sincronización respectivos anotados.

Los tiempos máximos de ciclo (mínimos tiempos de WatchDog) así calculados podrán usarse para el tiempo de WatchDog del recurso respectivo. Véase el manual de seguridad HI 801 196 S.

#### 4.6.2 Tiempo máximo de ciclo del sistema de control HiMatrix

Para determinar el tiempo máximo de ciclo de un sistema de control HiMatrix, HIMA recomienda el siguiente procedimiento:

##### Determinar tiempo máximo de ciclo del sistema de control HiMatrix

1. Haga funcionar el sistema a plena carga. Deberán estar en uso todas las conexiones de comunicación, tanto mediante safeethernet como mediante protocolos estándares. Lea frecuentemente el tiempo de ciclo en el panel de control y anote el tiempo máximo del ciclo.
2. Repita el paso 1 para el interlocutor de comunicación (segundo sistema de control HiMatrix).
3. El mayor de ambos tiempos de ciclo registrados será el máximo tiempo de ciclo buscado.

El tiempo máximo de ciclo queda determinado y se usará para los siguientes cálculos.

#### 4.6.3 TimeOut de recepción

*ReceiveTMO* es el tiempo de monitoreo, en ms, dentro del cual deberá recibirse una respuesta correcta del interlocutor de comunicación.

Si dentro del tiempo *ReceiveTMO* no se recibe respuesta correcta del interlocutor de comunicación, se cerrará la comunicación con función relacionada con la seguridad. Las variables de entrada de esta conexión safeethernet se comportarán tal y como lo defina el parámetro *Freeze Data on Lost Connection [ms]*.

Para las funciones relacionadas con la seguridad implementadas mediante safeethernet sólo se permite utilizar la opción **Use Initial Data**.

Puesto que *ReceiveTMO* es relevante para la seguridad y forma parte del tiempo Worst Case Reaction Time  $T_R$  (para el máximo tiempo de reacción véase el apartado 4.7.1 y ss), *ReceiveTMO* deberá calcularse y registrarse en el editor safeethernet de la siguiente forma.

##### **ReceiveTMO $\geq$ 4 \* Delay + 5 \* max. Cycle Time**

Condición: el intervalo de tiempo de comunicación deberá ser lo suficientemente grande como para poder procesar en un ciclo de la CPU todas las conexiones safeethernet.

Delay: Retardo en la línea de transmisión, p. ej. por switch o satélite

Max. Cycle Time: Tiempo máximo de ciclo de ambos sistemas de control



Si se desea una tolerancia a errores de comunicación, ello podrá lograrse aumentando el tiempo *ReceiveTMO*, siempre y cuando ello sea admisible para el proceso de aplicación desde el punto de vista de los tiempos.

#### 4.6.4 Tiempo de respuesta

El valor de *ResponseTime* es el tiempo, en ms, que transcurre hasta que el emisor de una notificación recibe la confirmación de recepción del destinatario.

Para parametrizar con un perfil safeethernet deberá especificarse un tiempo de respuesta a esperar (*ResponseTime*) para la línea de transmisión según las circunstancias físicas.

El tiempo *ResponseTime* especificado afectará a la configuración de todos los parámetros de la conexión safeethernet, que deberá Ud. calcular de la siguiente manera:

##### **ResponseTime $\leq$ ReceiveTMO / n**

**n = 2, 3, 4, 5, 6, 7, 8.....**

La relación entre *ReceiveTMO* y *Response Time* afecta a la capacidad de tolerancia de errores, p. ej. en caso de pérdida de paquetes de datos (repetición del paquete perdido) o retardos en la línea de transmisión.

En una red donde puedan producirse pérdidas de paquetes de datos, deberá cumplirse la siguiente condición:

$$\text{min. Response Time} \leq \text{ReceiveTMO} / 2 \geq 2 * \text{Delay} + 2,5 * \text{max. Cycle Time}$$

Si se cumple esta condición, podrá interceptarse la pérdida de al menos un paquete de datos sin que se interrumpa la conexión **safeethernet**.



Si no se cumple esta condición, la disponibilidad de una conexión **safeethernet** sólo estará garantizada si se usa en una red libre de perturbaciones y colisiones. ¡En todo caso, ello no representa un problema de seguridad para el módulo procesador!



¡Cerciórese de que el sistema de comunicación cumpla el Response-Time parametrizado! Para los casos en que esto no siempre pueda garantizarse, se dispone de una correspondiente variable de sistema de la conexión para monitorear el Response Time. Si no es sólo en casos aislados que se excede el Response-Time medido a lo largo de la mitad del *ReceiveTMO*, deberá aumentarse el Response Time parametrizado. El *Receive Timeout* deberá adaptarse al Response Time ahora parametrizado.

#### 4.6.5 Sync/Async

**Sync** Por el momento no es compatible.

**Async** Es el valor configurado por defecto.  
Con *Async*, la instancia de protocolo **safeethernet** recibe en la fase de entrada de la CPU y envía, conforme a las reglas de envío, en la fase de salida de la CPU.

#### 4.6.6 ResendTMO

*El tiempo ResendTMO no puede ajustarse manualmente, pero se calcula en base al perfil y al tiempo de respuesta.*

Tiempo de monitoreo en milisegundos (ms) en el PES1, dentro del cual el PES2 deberá haber confirmado la recepción de un paquete de datos. De no ser así, se repetirá el envío del paquete.

Regla: *ResendTMO*  $\leq$  *Receive-Timeout*

En caso de diferente configuración del tiempo *ResendTMO* en ambos interlocutores de comunicación, el interlocutor de protocolo activo (menor SRS) determinará el valor de hecho del tiempo *ResendTMO* de la conexión del protocolo.

#### 4.6.7 Acknowledge Timeout

*AckTMO no puede ajustarse manualmente, pero se calcula en base al perfil y al tiempo de respuesta.*

*AckTMO* es el tiempo dentro del cual la CPU deberá confirmar la recepción de un paquete de datos.

Para una red rápida, el tiempo *AckTMO* es cero, es decir, la recepción del paquete de datos se confirma inmediatamente. Para una red lenta (p. ej. tramo con módem telefónico), el tiempo *AckTMO* será mayor que cero. En este caso se intentará transmitir el mensaje de confirmación junto con los datos de proceso, para reducir la solicitud de la red evitando bloques de direccionamiento y de verificación.

Reglas:

- *AckTMO* debe ser  $\leq$  *ResendTMO*
- *AckTMO* debe ser  $\leq$  *ResendTMO out*, si *Production-Rate* es  $>$  *ResendTMO*.

#### 4.6.8 Production Rate

*ProdRate no puede ajustarse manualmente, pero se calcula en base al perfil y al tiempo de respuesta.*

Mínimo intervalo en milisegundos (ms) entre dos paquetes de datos.

El objetivo de *ProdRate* es limitar el volumen de paquetes de datos a una medida tal que no sobrecargue un canal de comunicación lento. Con ello se logra una tasa de solicitud homogénea del medio de transmisión y se evita la recepción de datos obsoletos en el lado de recepción.

Reglas:

- *ProdRate*  $\leq$  *Receive Timeout*
- *ProdRate*  $\leq$  *Resend-Timeout*, si *Acknowledge-Timeout*  $>$  *Resend-Timeout*



Un valor de “Production Rate” igual a cero significa que con cada ciclo del programa de usuario podrán transmitirse paquetes de datos.

---

#### 4.6.9 Queue

*Queue no puede ajustarse manualmente, pero se calcula en base al perfil y al tiempo de respuesta.*

*Queue* es la cantidad de paquetes de datos que pueden enviarse sin tener que esperar confirmación de recepción.

El valor dependerá de la capacidad de transmisión de la red y de posibles retardos debido a tiempos de ejecución de la red.

Todas las conexiones safeethernet comparten la capacidad de memoria de mensajes disponible de la CPU.

#### 4.7 Máximo tiempo de reacción para safeethernet

En los siguientes ejemplos, las fórmulas de cálculo del máximo tiempo de reacción de una conexión con sistemas de control HIMatrix tendrán validez sólo cuando para estos se haya definido el tiempo de seguridad = 2 \* tiempo de WatchDog. Estas fórmulas son siempre válidas para los sistemas de control HIMax.



El tiempo máximo de reacción admisible dependerá del proceso y habrá de acordarse con el ente de inspección oficial que deba emitir su aprobación.

Términos:

ReceiveTMO:	Tiempo de monitoreo en el sistema PES 1, dentro del cual deberá recibirse una respuesta válida del sistema PES 2. De lo contrario, tras expirar este tiempo se cerrará la comunicación relacionada con la seguridad.
Production Rate:	Separación mínima entre dos envíos de datos.
WatchDog Time:	Máxima duración admisible del ciclo RUN de un sistema de control. La duración del ciclo RUN dependerá de la complejidad del programa de usuario y de la cantidad de las conexiones safeethernet. El tiempo de WatchDog (WDT) consta en las propiedades del recurso.
Worst Case Reaction Time	Máximo tiempo de reacción para la transmisión del cambio de la señal de una entrada física (IN) de un sistema PES 1 hasta el cambio de la salida física (OUT) de un sistema PES 2.
Delay:	Retardo en una línea de transmisión, p. ej. en conexiones por módem o satélite. En una conexión directa puede suponerse en principio un retardo de 2 ms. El retardo que de hecho se dé en la línea de transmisión podrá ser evaluado por el administrador de la red responsable de ello.

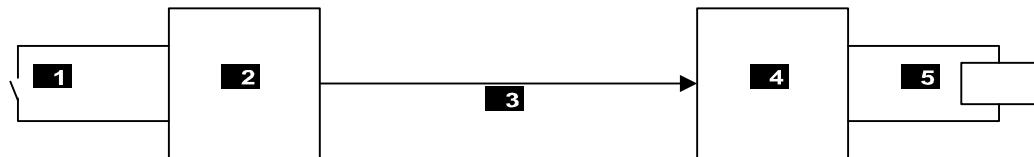
Para los siguientes cálculos de los máximos tiempos de reacción admisibles tienen validez estas condiciones:

- Las señales transmitidas con safeethernet deberán procesarse en los respectivos sistemas de control dentro de un ciclo de CPU.
- Deberán sumarse además los tiempos de reacción de sensores y actuadores.

Los cálculos son válidos también para señales en el sentido opuesto.

#### 4.7.1 Cálculo del tiempo máximo de reacción de dos sistemas de control HIMax

El tiempo máximo de reacción  $T_R$  (Worst Case) desde el cambio de estado de un transductor del sistema de control 1 hasta la reacción de la salida del sistema de control 2 puede calcularse como sigue:



- 1** Entrada
- 2** Sistema de control HIMax 1
- 3** Protocolo con función relacionada con la seguridad
- 4** Sistema de control HIMax 2
- 5** Salida

Fig. 8: Tiempo de reacción en caso de conectar dos sistemas de control HIMax

$$T_R = t_1 + t_2 + t_3$$

$T_R$  Worst Case Reaction Time

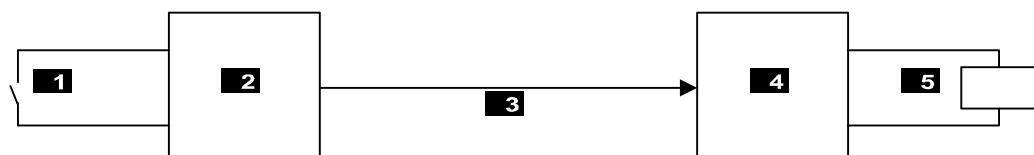
$t_1$  Tiempo de seguridad del sistema de control HIMax 1

$t_2$  *ReceiveTMO*

$t_3$  Tiempo de seguridad del sistema de control HIMax 2

#### 4.7.2 Cálculo del tiempo máximo de reacción en conjunción con un sistema de control HIMatrix

El tiempo máximo de reacción  $T_R$  (Worst Case) desde el cambio de estado de un transductor (IN) del sistema de control HIMax hasta la reacción de la salida (OUT) del sistema de control HIMatrix puede calcularse como sigue:



- 1** Entrada
- 2** Sistema de control HIMax
- 3** Protocolo con función relacionada con la seguridad
- 4** Sistema de control HIMatrix
- 5** Salida

Fig. 9: Tiempo de reacción de un HIMax en conjunción con un sistema de control HIMatrix

$$T_R = t_1 + t_2 + t_3$$

$T_R$  Worst Case Reaction Time

$t_1$  Tiempo de seguridad del sistema de control HIMax

$t_2$  *ReceiveTMO*

$t_3$  2 \* tiempo de WatchDog del sistema de control HIMatrix

#### 4.7.3 Cálculo del tiempo máximo de reacción con dos sistemas de control HIMatrix o I/Os remotas

El tiempo máximo de reacción  $T_R$  (Worst Case) desde el cambio de estado de un transductor (IN) en el primer sistema de control HIMatrix o en las I/O remotas (p. ej. F3 DIO 20/8 01) hasta la reacción de la salida (OUT) en el segundo sistema de control HIMatrix o en la I/O remota puede calcularse como sigue:

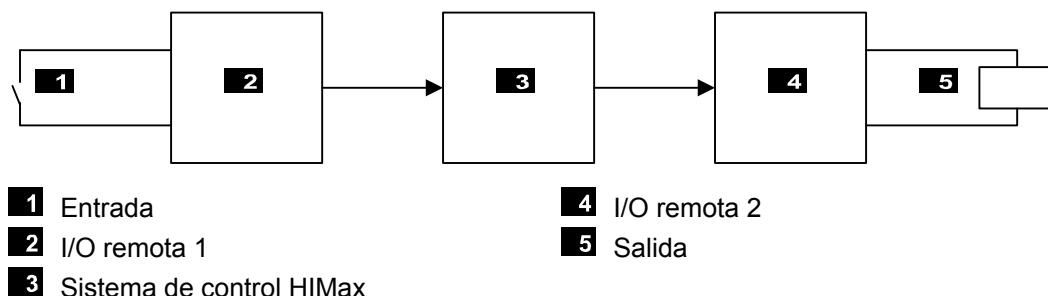


Fig. 10: Tiempo de reacción con dos I/Os remotas y un sistema de control HIMax

$$T_R = t_1 + t_2 + t_3 + t_4 + t_5$$

$T_R$  Worst Case Reaction Time

$t_1$  2 \* tiempo de WatchDog de la I/O remota 1

$t_2$  *ReceiveTMO1*

$t_3$  2 \* tiempo de WatchDog del sistema de control HIMax

$t_4$  *ReceiveTMO2*

$t_5$  2 \* tiempo de WatchDog de la I/O remota 2



Las dos I/Os remotas 1 y 2 también pueden ser idénticas. Los tiempos serán válidos aun cuando en lugar de una I/O remota se utilice un sistema de control HIMatrix.

#### 4.7.4 Cálculo del tiempo máximo de reacción con dos HIMax y un sistema de control HIMatrix

El tiempo máximo de reacción  $T_R$  (Worst Case) desde el cambio de estado de un transductor (IN) en el primer sistema de control HIMax hasta la reacción de la salida (OUT) en el segundo sistema de control HIMax puede calcularse como sigue:

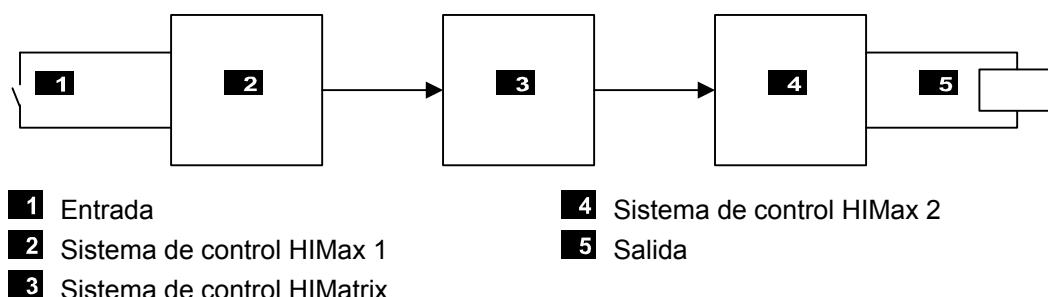


Fig. 11: Tiempo de reacción con dos sistemas de control HIMax y un sistema de control HIMatrix

$$T_R = t_1 + t_2 + t_3 + t_4 + t_5$$

$T_R$  Worst Case Reaction Time

$t_1$  Tiempo de seguridad del sistema de control HIMax 1

$t_2$  ReceiveTMO1

$t_3$  2 \* tiempo de WatchDog del sistema de control HIMatrix

$t_4$  ReceiveTMO2

$t_5$  Tiempo de seguridad del sistema de control HIMax 2



Ambos sistemas de control HIMax 1 y 2 también pueden ser idénticos.

El sistema de control HIMatrix también puede ser un sistema de control HIMax.

#### 4.7.5 Cálculo del tiempo máximo de reacción de dos sistemas de control HIMatrix

El tiempo máximo de reacción  $T_R$  (Worst Case) desde el cambio de estado de un transductor del sistema de control 1 hasta la reacción de la salida del sistema de control 2 puede calcularse como sigue:

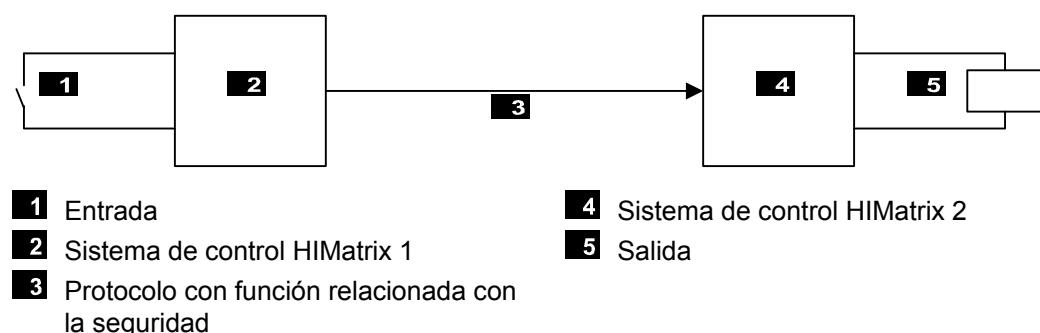


Fig. 12: Tiempo de reacción en caso de conectar dos sistemas de control HIMatrix

$$T_R = t_1 + t_2 + t_3$$

$T_R$  Worst Case Reaction Time

$t_1$  2 \* tiempo de WatchDog del sistema de control HIMatrix 1

$t_2$  ReceiveTMO

$t_3$  2 \* tiempo de WatchDog del sistema de control HIMatrix 2

#### 4.7.6 Cálculo del tiempo máximo de reacción con dos I/Os remotas

El tiempo máximo de reacción  $T_R$  desde el cambio de estado de un transductor (IN) del primer sistema de control HIMatrix o I/O remota (p. ej. F3 DIO 20/8 01) hasta la reacción del segundo sistema de control HIMatrix o I/O remota puede calcularse de la siguiente forma:

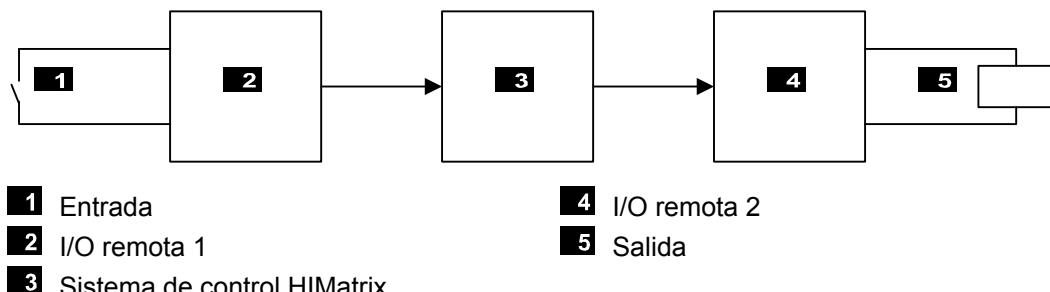


Fig. 13: Tiempo de reacción con I/Os remotas

$$T_R = t_1 + t_2 + t_3 + t_4 + t_5$$

$T_R$  Worst Case Reaction Time

$t_1$  2 \* tiempo de WatchDog de la I/O remota 1

$t_2$  ReceiveTMO<sub>1</sub>

$t_3$  2 \* tiempo de WatchDog del sistema de control HIMatrix

$t_4$  ReceiveTMO<sub>2</sub>

$t_5$  2 \* tiempo de WatchDog de la I/O remota 2

Comentario: las dos I/Os remotas 1 y 2 también pueden ser idénticas. Los tiempos serán válidos aun cuando en lugar de una I/O remota se utilice un sistema de control HIMatrix.

#### 4.7.7 Cálculo del tiempo máximo de reacción con dos sistemas HIMatrix y un HIMax

El tiempo máximo de reacción  $T_R$  desde el cambio de estado de un transductor (IN) del primer sistema de control HIMatrix hasta la reacción de la salida (OUT) del segundo sistema de control HIMatrix puede calcularse de la siguiente forma:

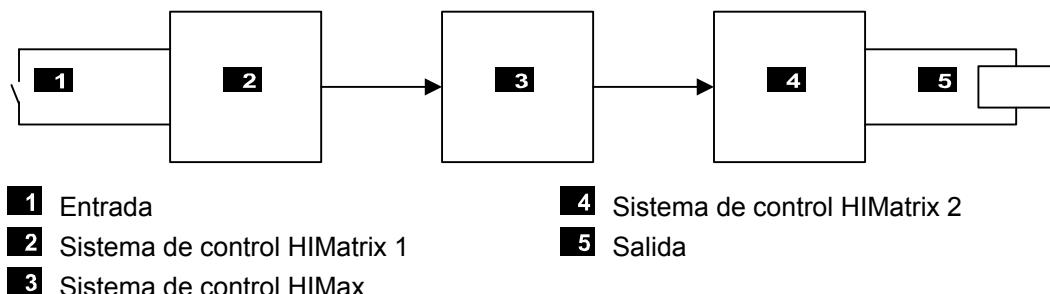


Fig. 14: Tiempo de reacción con dos sistemas de control HIMatrix y un HIMax

$$T_R = t_1 + t_2 + t_3 + t_4 + t_5$$

$T_R$  Worst Case Reaction Time

$t_1$  2 \* tiempo de WatchDog del sistema de control HIMatrix 1

$t_2$  ReceiveTMO<sub>1</sub>

$t_3$  2 \* tiempo de WatchDog del sistema de control HIMax

$t_4$  ReceiveTMO<sub>2</sub>

$t_5$  2 \* tiempo de WatchDog del sistema de control HIMatrix 2

#### 4.7.8 Perfiles safeethernet

Los perfiles **safeethernet** son combinaciones de parámetros adecuados entre sí que se configuran automáticamente al seleccionarse uno de los perfiles **safeethernet**. Para la parametrización deberá Ud. configurar por separado sólo Receive-Timeout y el tiempo Response-Time esperado.

El objetivo de un perfil **safeethernet** es optimizar el caudal de datos en la red teniendo en cuenta las circunstancias físicas.

Requisito para la efectividad de la optimización son las siguientes condiciones:

- El intervalo de tiempo de comunicación deberá ser lo suficientemente grande como para poder procesar en un ciclo de la CPU todas las conexiones **safeethernet**.
- Cuando del tiempo de ciclo promedio de CPU < Response-Time.
- Cuando del tiempo de ciclo promedio de CPU < ProdRate o ProdRate = 0.

#### NOTA



¡Son posibles fallos de la comunicación **safeethernet** o incluso que deje de funcionar!

**Combinaciones inadecuadas de ciclo de CPU, intervalo de comunicación, Response-Time y ProdRate no serán rechazadas en la generación del código ni en la carga por download/reload, pero pueden originar fallos en la comunicación.**

**Compruebe en los paneles de control de ambos sistemas de control los indicadores *bad messages* y *resends*.**

Se dispone de seis perfiles **safeethernet**, de entre los cuales podrá Ud. elegir el perfil **safeethernet** idóneo para su línea de transmisión.

Para la comunicación de datos de proceso con función relacionada con la seguridad se permite solamente usar los perfiles *Fast&Noisy*, *Medium&Noisy* y *Slow&Noisy*.

Fast & Cleanroom      ¡No aptos para comunicación de datos de proceso con función relacionada con la seguridad!

Fast & Noisy

Medium & Cleanroom      ¡No aptos para comunicación de datos de proceso con función relacionada con la seguridad!

Medium & Noisy

Slow & Cleanroom      ¡No aptos para comunicación de datos de proceso con función relacionada con la seguridad!

Slow & Noisy

#### 4.7.9 Perfil I (Fast & Cleanroom)

##### NOTA



¡No aptos para comunicación de datos de proceso con función orientada a la seguridad!  
Para la comunicación de datos de proceso con función orientada a la seguridad se permite solamente usar los perfiles *Fast&Noisy*, *Medium&Noisy* y *Slow&Noisy*.  
Se ofrece por su compatibilidad con versiones anteriores de SILworX y ELOP II Factory.

##### Utilización

¡El perfil *Fast & Cleanroom* es apto para aplicaciones en entornos ideales, p. ej. en laboratorios!

- Para la mayor rapidez de los volúmenes de datos
- Para aplicaciones que requieren de una rápida transmisión de datos
- Para aplicaciones que requieren de un tiempo Worst Case ReactionTime lo menor posible

##### Requisitos de la red

- Fast: Tecnología de 100 Mbits (100 Base TX), tecnología de 1 Gbit
- Clean: Red exenta de perturbaciones.  
Pérdida de datos por sobrecarga de la red, deben evitarse influencias externas o manipulaciones de la red.
- ¡Switches LAN imprescindibles!

##### Características de la vía de comunicación

- Mínimos retardos
- Tiempo esperado de ResponseTime  $\leq$  ReceiveTMO  
(de lo contrario, ERROR en la parametrización)

#### 4.7.10 Perfil II (Fast & Noisy)

##### Utilización

El perfil *Fast & Noisy* es el perfil predeterminado de SILworX para la comunicación safeethernet.

- Para rapidez de los volúmenes de datos
- Para aplicaciones que requieren de una rápida transmisión de datos
- Para aplicaciones que requieren de un tiempo Worst Case Reaction Time lo menor posible

##### Requisitos de la red

- Fast: Tecnología de 100 Mbits (100 Base TX), tecnología de 1 Gbit
- Noisy: Red no exenta de perturbaciones.  
Baja probabilidad de pérdidas de paquetes de datos  
Tiempo para  $\geq$  1 repetición
- ¡Switches LAN imprescindibles!

##### Características de la vía de comunicación

- Mínimos retardos
- Tiempo esperado de ResponseTime  $\leq$  ReceiveTMO / 2  
(de lo contrario, ERROR en la parametrización)

#### 4.7.11 Perfil III (Medium & Cleanroom)

##### NOTA



**¡No aptos para comunicación de datos de proceso con función orientada a la seguridad!**  
Para la comunicación de datos de proceso con función orientada a la seguridad se permite solamente usar los perfiles *Fast&Noisy*, *Medium&Noisy* y *Slow&Noisy*. Se ofrece por su compatibilidad con versiones anteriores de SILworX y ELOP II Factory.

##### Utilización

El perfil *Medium & Cleanroom* es para aplicaciones en una red exenta de perturbaciones que requieren de una transmisión de datos sólo moderadamente rápida.

- Para volúmenes de datos medios
- Apto para redes privadas virtuales (VPN) en las que el intercambio de datos sea lento debido a los dispositivos de seguridad intercalados (firewalls, cifrados) pero exenta de errores.
- Apto para aplicaciones en las que el tiempo Worst Case ReactionTime no revista carácter crítico.

##### Requisitos de la red

- Medio: Tecnología de 10 Mbits (10 Base T), 100 Mbits (100 Base TX) y 1 Gbit
- ¡Switches LAN imprescindibles!
- Clean: Red exenta de perturbaciones.  
Pérdida de datos por sobrecarga de la red, deben evitarse influencias externas o manipulaciones de la red.  
Tiempo para  $\geq 0$  repeticiones.

##### Características de la vía de comunicación

- Retardos moderados
- Tiempo esperado de ResponseTime  $\leq$  ReceiveTMO (de lo contrario, ERROR en la parametrización)

#### 4.7.12 Perfil IV (Medium & Noisy)

##### Utilización

El perfil *Medium & Noisy* es para aplicaciones que requieren de una transmisión de datos sólo moderadamente rápida.

- Para volúmenes de datos medios
- Para aplicaciones que requieren de una transmisión de datos sólo moderadamente rápida
- Apto para aplicaciones en las que el tiempo Worst Case ReactionTime no revista carácter crítico

##### Requisitos de la red

- Medio: Tecnología de 10 Mbits (10 Base T), 100 Mbits (100 Base TX) y 1 Gbit
- ¡Switches LAN imprescindibles!
- Noisy: Red no exenta de perturbaciones.  
Baja probabilidad de pérdidas de paquetes de datos  
Tiempo para  $\geq 1$  repetición

Características de la vía de comunicación

- Retardos moderados
- Tiempo esperado de ResponseTime  $\leq$  ReceiveTMO / 2  
(de lo contrario, ERROR en la parametrización)

#### 4.7.13 Perfil V (Slow & Cleanroom)

##### NOTA



**¡No aptos para comunicación de datos de proceso con función orientada a la seguridad!**

**Para la comunicación de datos de proceso con función orientada a la seguridad se permite solamente usar los perfiles *Fast&Noisy*, *Medium&Noisy* y *Slow&Noisy*. Se ofrece por su compatibilidad con versiones anteriores de SILworX y ELOP II Factory.**

Utilización

El perfil *Slow & Cleanroom* es para aplicaciones en una red exenta de perturbaciones en las que basta con una transmisión de datos lenta.

- Para volúmenes de datos lentos.
- Para aplicaciones en las que es suficiente con una lenta transmisión de datos a sistemas de control (posiblemente alejados) y para casos con condiciones no previsibles de la línea de comunicación.

Requisitos de la red

- Slow: Transmisión de datos por ISDN, línea permanente o radioenlace.
- Clean: Red exenta de perturbaciones.  
Pérdida de datos por sobrecarga de la red, deben evitarse influencias externas o manipulaciones de la red.
- Tiempo para  $\geq 0$  repeticiones.

Características de la vía de comunicación

- Retardos moderados
- Tiempo esperado de ResponseTime = ReceiveTMO (de lo contrario, ERROR en la parametrización)

#### 4.7.14 Perfil VI (Slow & Noisy)

Utilización

El perfil *Slow & Noisy* es para aplicaciones en las que es suficiente con una lenta transmisión de datos a sistemas de control (posiblemente alejados).

- Para volúmenes de datos lentos
- Principalmente para aplicaciones con transmisión de datos por malas líneas telefónicas o enlaces de radio sujetos a perturbaciones.

Requisitos de la red

- Slow: Transmisión de datos por teléfono, satélite, radio, etc.
- Noisy: Red no exenta de perturbaciones.  
Baja probabilidad de pérdidas de paquetes de datos  
Tiempo para  $\geq 1$  repetición

#### Características de la vía de comunicación

- Retardos moderados o altos
- Tiempo esperado de ResponseTime  $\leq$  ReceiveTMO / 2  
(de lo contrario, ERROR en la parametrización)

## 4.8 Comunicación trascendente a todo el proyecto

La comunicación trascendente a todo el proyecto se usa para

- Interconectar recursos de distintos proyectos.
- Interconectar sistemas de control con sistema operativo a partir de V7 (para SILworX) y con sistema operativo anterior a V7 (para ELOP II Factory) mediante safeethernet. Véase el capítulo 4.9.

La comunicación entre ambos proyectos tendrá lugar vía safeethernet y se configurará en el editor de safeethernet.

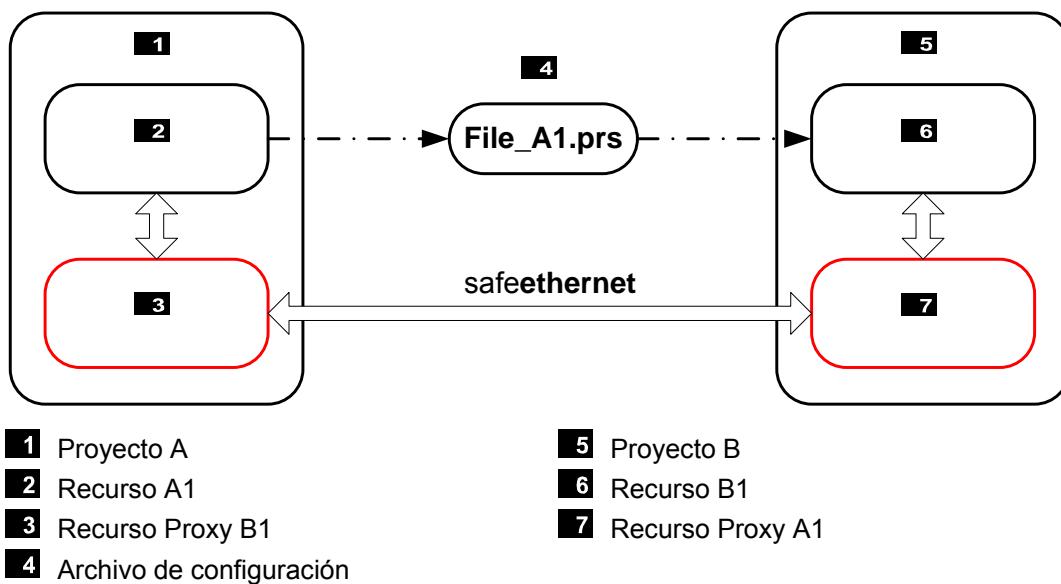


Fig. 15: Conexión safeethernet entre el recurso A1 del proyecto A y el recurso B1 del proyecto B

El proyecto designado como local es el proyecto en el que Ud. realiza la configuración de la conexión safeethernet y en el que se crea el archivo de configuración.

El proyecto designado como de destino (target) es el proyecto al que Ud. importa el archivo de configuración.

El proyecto local y el proyecto de destino son interlocutores de comunicación con el mismo nivel jerárquico en el intercambio de datos.

El respectivo recurso Proxy sirve como comodín para el recurso correspondiente del proyecto externo y se usa para la importación y exportación de conexiones safeethernet.

El recurso *Proxy B1* en el *proyecto A* es el comodín del *recurso B1* del *proyecto B*.

El recurso *Proxy A1* en el *proyecto B* es el comodín del *recurso A1* del *proyecto A*.

En el proyecto local (aquí el *proyecto A*) deberá Ud. crear y configurar manualmente el recurso Proxy (aquí *recurso Proxy B1*). Tras configurar importe el archivo de configuración (aquí *File\_A1.prs*) en el proyecto de destino (aquí del *recurso B1*).

El archivo de configuración *File\_A1.prs* contiene la descripción completa del *recurso A1* para la conexión safeethernet al *recurso B1*. Tras importar el archivo de configuración *File\_A1.prs* al *recurso B1*, se creará automáticamente el *recurso Proxy A1* en el *proyecto B*.

#### 4.8.1 Variantes de la comunicación trascendente a todo el proyecto

En las siguientes dos variantes, los proyectos A y B comunican entre sí vía safeethernet.

El proyecto A es el proyecto local en la primera variante y el proyecto B es el proyecto local en la segunda variante. Básicamente se deja a elección del usuario en cuál de los dos proyectos desea crear la configuración.

Cuesta prácticamente lo mismo efectuar la configuración por cualquiera de los dos medios y se obtiene la misma configuración.

#### Proyecto local A

En el proyecto local A configurará Ud. la comunicación al proyecto de destino B y creará los archivos de configuración. Esto tiene la ventaja de que en el proyecto local solamente deberá crearse manualmente el recurso Proxy B1.

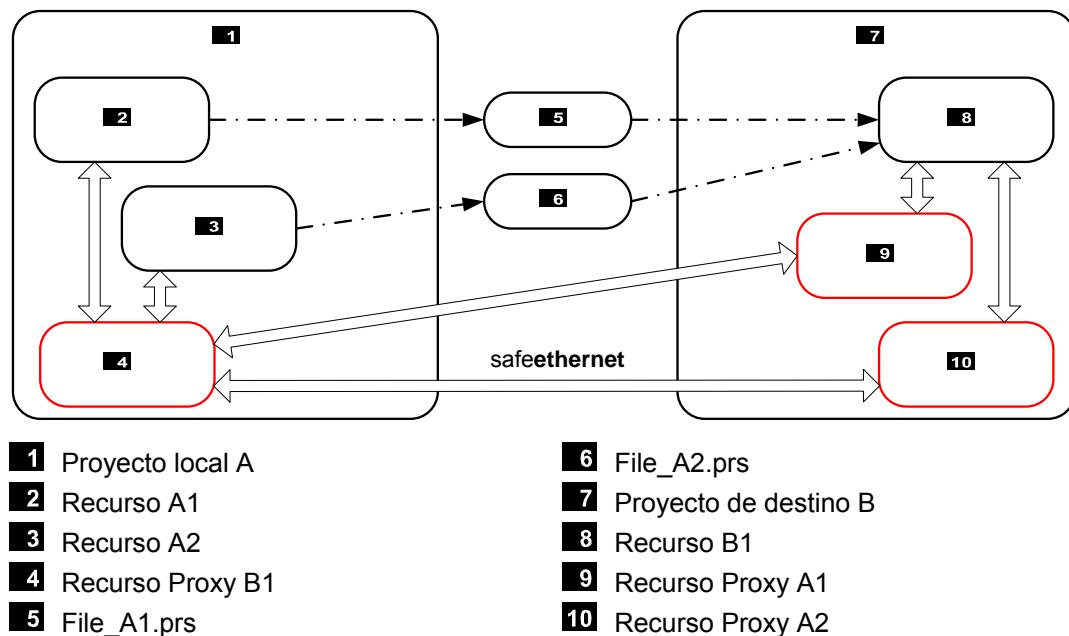


Fig. 16: Variante con el proyecto A como proyecto local

### Proyecto local B

En el proyecto local B configurará Ud. la comunicación al proyecto de destino A y creará los archivos de configuración. Esto tiene el inconveniente de que en el proyecto local B deberán crearse manualmente dos recursos Proxy (A1 y A2).

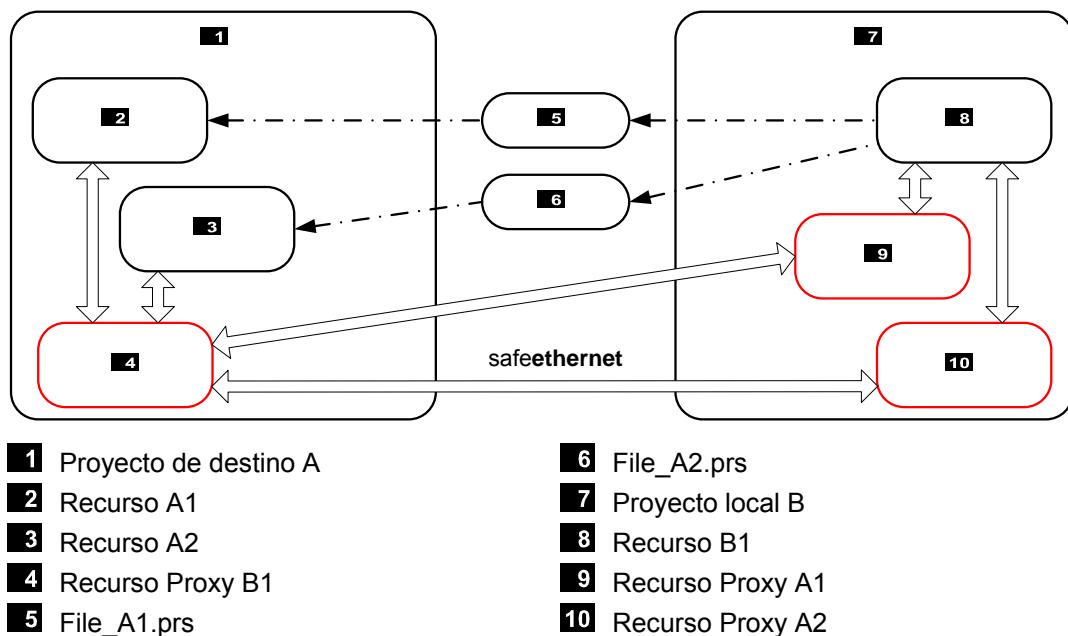


Fig. 17: Variante con el proyecto B como proyecto local

## 4.9 Comunicación trascendente a todo el proyecto entre SILworX y ELOP II Factory

En este ejemplo se configura una conexión safeethernet entre HIMax (SILworX) y HIMatrix (ELOP II Factory).

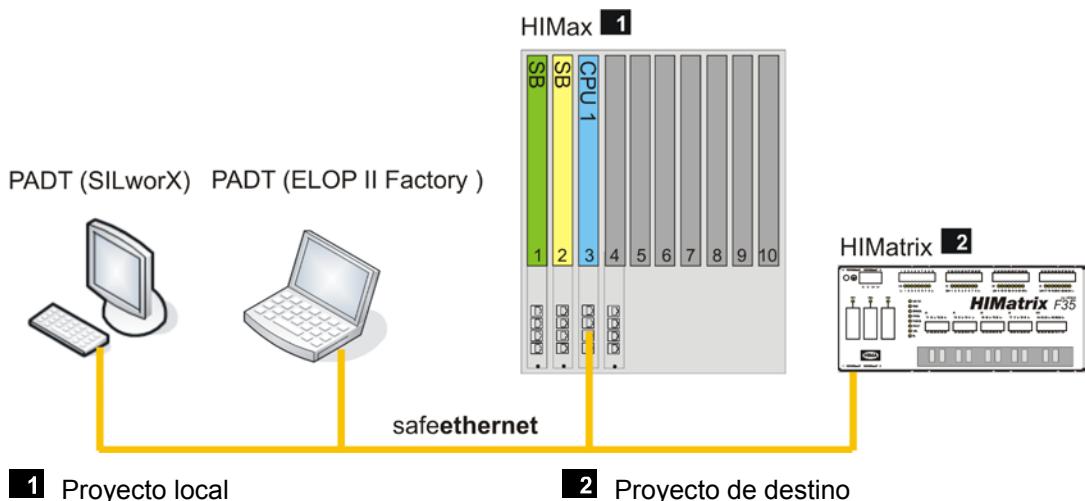


Fig. 18: Comunicación trascendente a todo el proyecto entre SILworX y ELOP II Factory

Abra el recurso del proyecto de destino (HIMatrix) que desee usar en el proyecto local (HIMax) como recurso Proxy.

Determine los siguientes parámetros de este recurso de destino:

- ID de sistema
- Tiempo de seguridad [ms]
- WatchDog Time [ms]
- IP-Address

**i** Estas propiedades del recurso tienen relevancia para la seguridad y están sujetas a restricciones. Hallará indicaciones al respecto en los manuales de seguridad de los respectivos sistemas de control.

### 4.9.1 Configuración de HIMax en el proyecto SILworX

#### 4.9.1.1 Creación del recurso Proxy

Un recurso Proxy sirve como comodín para un recurso en un proyecto externo y se usa para la importación y exportación de conexiones safeethernet.

**Para crear un recurso Proxy en el proyecto local:**

1. Abra el proyecto local en el que desee crear el recurso Proxy.
2. Abra **Configuration** en el árbol.
3. Haga clic con el botón derecho en **Configuration** y seleccione **New, Proxy Resource, ELOP II Factory**.
- Se agregará un nuevo recurso Proxy.

**Para configurar un recurso Proxy en el proyecto local:**

1. Seleccione **Properties** en el menú contextual del recurso Proxy.
2. Escriba un nombre inequívoco en el recuadro **Name**.  
Para el recurso Proxy del proyecto local use el nombre del recurso del proyecto de destino.
3. Introduzca para este recurso Proxy los valores previamente determinados del proyecto de destino **System ID**, **Safety Time [ms]** y **Watchdog Time [ms]**.
4. Haga clic en **OK**. Los demás parámetros pueden dejarse con los valores estándar por defecto.

**Abra la estructura del recurso Proxy:**

1. Haga clic con el botón derecho en **Hardware** y seleccione **Edit, HIMatrix Proxy**.
2. Confirme con **OK** para abrir el editor de hardware del recurso Proxy.



Fig. 19: Recurso Proxy de HIMatrix

3. Haga doble clic en los **COM Module** e introduzca la **IP address** antes determinada para el recurso Proxy.
4. Haga clic en **Save** y luego en **Close**.
5. Repita estos pasos para todos los demás recursos Proxy del proyecto local.

#### 4.9.1.2 Conexión del recurso local al recurso Proxy

Cree en el editor de safeethernet una conexión safeethernet entre el recurso local y el recurso Proxy.

**Para abrir el editor de safeethernet del recurso local:**

1. Abra **Configuration, Resource** en el árbol.
2. Haga clic con el botón derecho en **safeethernet** y seleccione **Edit** en el menú contextual. El recurso Proxy creado se encuentra en la selección de objetos.

**Para crear la conexión safeethernet al recurso Proxy:**

1. En la selección de objetos haga clic en el **recurso Proxy** y arrástrelo con el ratón hasta un lugar libre del área de trabajo del editor de safeethernet.
2. Seleccione las **Interfaces Ethernet** del recurso local y del recurso Proxy.  
Los siguientes parámetros definen el volumen de datos y la tolerancia a errores y colisiones de la conexión safeethernet:
3. Seleccione **Network Profile** (p. ej. Fast&Noisy) de la conexión safeethernet.
4. Calcule e introduzca los tiempos **Receive Timeout** y **Response Time**.

Ejemplo de valores de parámetros de una conexión safeethernet al recurso Proxy:

The screenshot shows the 'safeethernet' configuration window. At the top, there's a table for connection parameters:

	Partner	IF CH 1 (lokal)	IF CH 2 (lokal)	IF CH 1 (ziel)
1	Proxy Ressource	0.3 (192.168.0.5)		0.1 (192.168.0.99)

Below this is a 'Ressourcen' (Resources) table:

	Ressourcen	Konfiguration	System ID
1	HIMax_02	Konfiguration	60000
2	Proxy Ressource		

Fig. 20: Parámetros de una conexión safeethernet al recurso Proxy

#### 4.9.1.3 Conexión de variables de proceso

Conecte las variables de proceso en la vista en detalle de la conexión safeethernet.

##### Para abrir la vista en detalle de una conexión safeethernet:

Requisitos: El editor de safeethernet del recurso local deberá estar abierto.

1. Haga clic con el botón derecho en el **Proxy Resource** y abra el recurso Prox.
2. Seleccione **Detail View** en el menú contextual para abrir la vista en detalle de la conexión safeethernet.
3. Seleccione la ficha **Resource <-> Proxy-Resource**.

##### Para agregar variables de envío de safeethernet:

Las variables de envío se transmiten desde el recurso local al recurso Proxy.

1. Seleccione el área **Resource -> Proxy-Resource**.
2. Elija en la selección de objetos una **Global Variable** y arrástrela con el ratón a la columna **Resource (target) -> Resource (local)**.
3. Repita este paso para las demás variables de envío de safeethernet.

##### Para agregar variables de recepción de safeethernet:

Las variables de recepción las recibe el recurso local.

1. Seleccione el área **Resource <- Proxy-Resource**.
2. Elija en la selección de objetos una **Global Variable** y arrástrela con el ratón a la columna **Resource (target) <- Resource (local)**.
3. Repita este paso para las demás variables de recepción de safeethernet.

#### 4.9.1.4 Exportación del archivo de configuración desde SILworX

La conexión **safeethernet** configurada en SILworX deberá Ud. exportarla como archivo de configuración con la extensión **\*.prs**. Este archivo de configuración podrá importarse en ELOP II Factory para aplicar la conexión **safeethernet** para el sistema de control HIMatrix.

##### Para exportar una conexión **safeethernet**:

1. Dentro del editor de **safeethernet** haga clic en **Proxy Resource** y abra el menú contextual.
2. Seleccione **Export Connection with Proxy Resource** en el menú contextual:  
Se abrirá un típico cuadro de diálogo para guardar archivos.
3. Escriba en el cuadro de diálogo el nombre para el archivo de configuración y guárdelo con la extensión **\*.prs**.
4. Cierre el proyecto local.

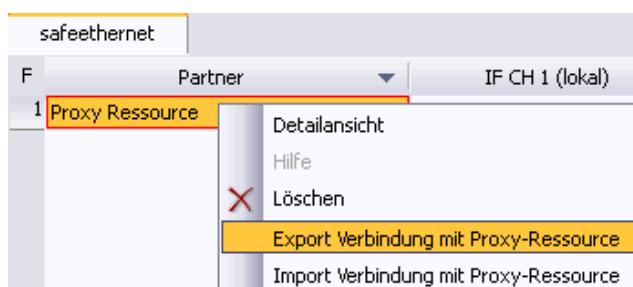


Fig. 21: Exportación de una conexión **safeethernet**

##### Verificación de la conexión **safeethernet**:

1. Abra **Configuration, Resource, safeethernet** en el árbol.
2. Haga clic en **Verification** en la barra de acciones **Action Bar** y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.



La configuración de la conexión **safeethernet** y el programa de usuario del recurso HIMax deberá Ud. recompilarlos y cargarlos al sistema de control para que sea efectiva efectiva la comunicación de HIMax.

#### 4.9.2 Configuración de HIMatrix en ELOP II Factory

**Para importar el archivo de configuración al proyecto de destino (HIMatrix):**

1. Inicie ELOP II Factory.
  2. Abra el proyecto de destino HIMatrix al que desee importar el archivo de configuración.
  3. Seleccione **Target Resource** en el árbol y abra el menú contextual.
  4. Seleccione **Import Connections** en el menú contextual:  
Se abrirá un cuadro de diálogo para cargar un archivo con la extensión \*.prs.
  5. En el cuadro de diálogo seleccione el archivo de configuración que ha creado en el proyecto local HIMax y haga clic en **OK**.
- Tras importar el archivo de configuración, se creará automáticamente el recurso local HIMax como recurso Proxy en el proyecto de destino HIMatrix.

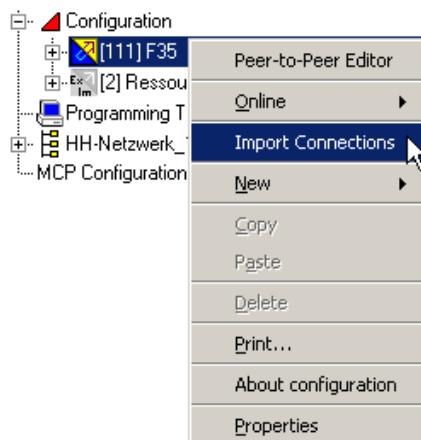


Fig. 22: Importación de conexiones a ELOP II Factory

##### 4.9.2.1 Asignación de señales de proceso de ELOP II Factory

Conecte las señales de proceso en el recurso de destino (HIMatrix).

**Signal Editor** desde la barra de menú **Signals, Editor**.

**Para abrir el editor Peer-to-Peer en ELOP II Factory para el recurso de destino:**

1. Abra **Configuration, Resource, P2P Editor** en el árbol.
2. Introduzca en el editor **P2P Editor** la red HH para esta conexión.
3. En el **P2P Editor** haga clic en **Connect Process Signals**.

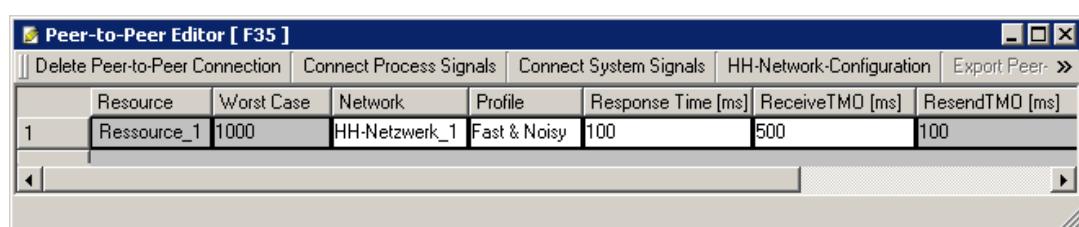


Fig. 23: Editor Peer-to-Peer en ELOP II Factory

- i** Observe que ambos interlocutores de comunicación usen el mismo perfil y los mismos ajustes (se adoptan automáticamente al importar el archivo de configuración).

### Para asignar señales de envío Peer-to-Peer:

Las señales de envío Peer-to-Peer se transmiten desde el recurso HIMatrix al recurso HIMax.

1. Seleccione la ficha **HIMatrix Resource -> HIMax Proxy Resource**.  
La ficha contiene las señales de envío Peer-to-Peer importadas.
2. Haga clic en el editor de señales sobre una **señal de proceso** y arrástrela con el ratón hasta la señal de envío del cuadro de diálogo P2P de señales de proceso a la que desee conectarla.
3. Repita este paso para las demás señales de envío Peer-to-Peer.

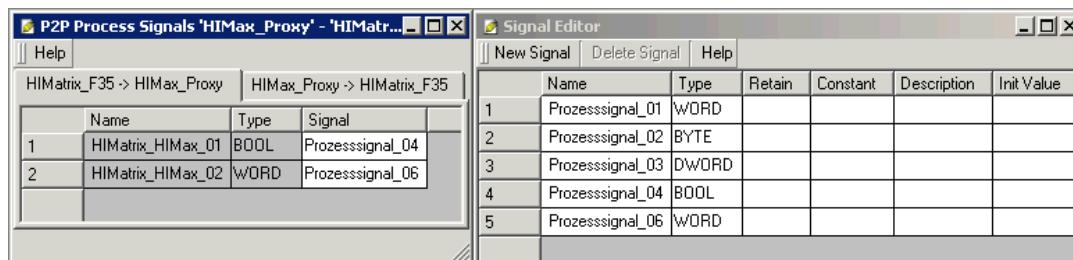


Fig. 24: Asignación de señales de envío en ELOP II Factory

### Para asignar señales de recepción Peer-to-Peer:

Las señales de recepción Peer-to-Peer las recibe el recurso HIMatrix.

1. Register Resource <- Proxy-Resource.  
La ficha contiene las señales de recepción Peer-to-Peer importadas.
2. Haga clic en el editor de señales sobre una **señal de proceso** y arrástrela con el ratón hasta la señal de recepción del cuadro de diálogo P2P de señales de proceso a la que desee conectarla.
3. Repita este paso para las demás señales de recepción Peer-to-Peer.

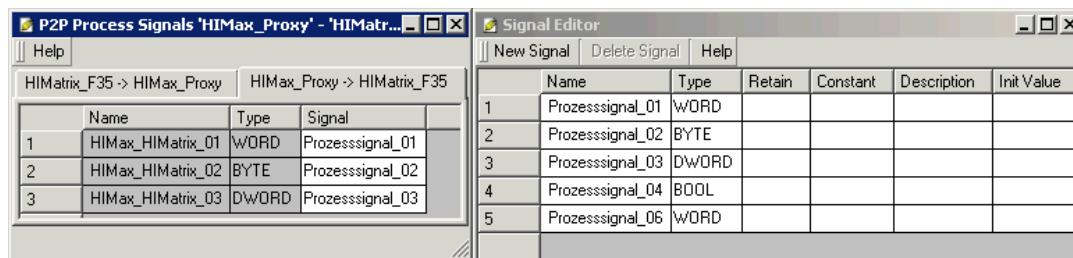


Fig. 25: Asignación de señales de recepción en ELOP II Factory



Hallará más información sobre la conexión de señales de proceso de ELOP II Factory en la ayuda directa en pantalla de ELOP II Factory.



La configuración de la conexión Peer-to-Peer y el programa de usuario del recurso HIMatrix deberá Ud. recompilarlos y cargarlos al sistema de control. Sólo entonces será efectiva la conexión Peer-to-Peer para HIMatrix.

#### 4.10 Panel de control (safeethernet)

El usuario podrá comprobar y controlar la configuración actual de la conexión safeethernet en el panel de control. Además, allí se muestra la información actual del estado de la conexión safeethernet (p. ej. tiempo de ciclo, estado del bus, etc.).

**Para abrir el panel de control para supervisar la conexión safeethernet:**

1. Seleccione **Resource** en el árbol.
2. Seleccione **Online** en el menú contextual del recurso.
3. Escriba sus datos de ingreso en **System Log-in** y abra el panel de control del recurso.
4. Seleccione **safeethernet** en el árbol del panel de control.

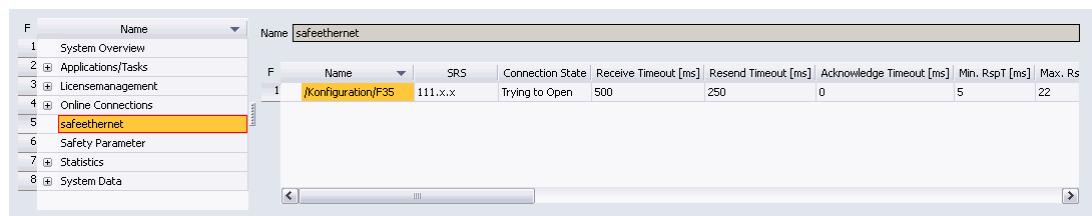


Fig. 26: Panel de control para supervisar la conexión safeethernet

#### Reinicialización de los valores estadísticos:

Con ayuda de la función de menú contextual podrá Ud. poner a cero los datos estadísticos (valores máximos, mínimos del tiempo de ciclo, etc.).

**Para reiniciar los datos estadísticos de la conexión safeethernet:**

1. Seleccione la conexión safeethernet en el árbol.
2. En el menú contextual de la conexión safeethernet seleccione **safeethernet Statistics**.

#### 4.10.1 Recuadro de lectura (conexión safeethernet)

En el recuadro de lectura se mostrarán los siguientes valores de la conexión safeethernet seleccionada:

Elemento	Descripción
Name	Nombre de recurso del interlocutor de comunicación
SRS	Sistema.Rack.Slot
Connection State	Estado de la conexión safeethernet (véase también el capítulo 4.4)
Receive Timeout [ms]	(véase también el capítulo 4.6.4)
Resend timeout [ms]	(véase también el capítulo 4.6.6)
Acknowledge Timeout [ms]	(véase también el capítulo 4.6.7)
Min. RspT [ms]	Valores de hecho de Response-Time en forma de media, mínima, máxima y más reciente. Véase también el capítulo 4.6.4.
Max. RspT [ms]	
Last RspT [ms]	
Av. RspT [ms]	
Bad Messages	Cantidad de notificaciones desecharadas tras reinicializar la estadística.
Reconnections	Cantidad de repeticiones tras reinicializar la estadística.
Number of Successful Connections	Cantidad de conexiones correctas tras reinicializar la estadística.
Early Queue Usage	Cantidad de notificaciones ubicadas Early Queue tras reinicializar la estadística. Véase también el capítulo 4.6.9.
Frame No.	Contador rotativo de envíos
Ack.Frame No.	Contador rotativo de recepciones
Monotony	Contador rotativo de envío de datos útiles
Layout Version	Signatura del punto final de comunicación actual
New Layout Version	Signatura del nuevo punto final de comunicación
Connection Control	Estado del control de conexión. Véase también el capítulo 4.4.
Transmission Control Ch1	Habilitación del canal 1 de la vía de transmisión. Véase también el capítulo 4.4.
Transmission Control Ch2	Habilitación del canal 2 de la vía de transmisión. Véase también el capítulo 4.4.
Quality of Channel 1	Estado del canal 1 de la vía de transmisión. Véase también el capítulo 4.4.
Quality of Channel 2	Estado del canal 2 de la vía de transmisión. Véase también el capítulo 4.4.
Late Received Redundant Messages	En caso de vías de transmisión redundantes. Cantidad de notificaciones recibidas con retraso tras reinicializar la estadística.
Lost Redundant Messages	En caso de vías de transmisión redundantes. Cantidad de las notificaciones recibidas sólo en una de ambas vías de transmisión desde que se reinicializó la estadística.
Protocol Version	0–1 Antigua versión de protocolo para HIMatrix con sistema operativo de CPU < V7 2 Nueva versión de protocolo para HIMax y HIMatrix con sistema operativo de CPU ≥ V7

Tabla 37: Recuadro de lectura de conexión safeethernet

## 5 PROFINET-IO

PROFINET-IO es el protocolo de transmisión basado en tecnología Ethernet de la organización de usuarios de PROFIBUS para la automatización.

Igual que en PROFIBUS-DP, en PROFINET-IO los dispositivos de campo descentralizados se integran mediante una descripción de dispositivos de campo (archivo GSDML) en SILworX.

El controlador PROFINET-IO de HIMA cumple el grado de conformidad A y es compatible con la comunicación Non-Real-Time (NRT) y Real Time (RT) a los dispositivos PROFINET-IO. La comunicación Real-Time se usa para el intercambio de datos crítico desde el punto de vista del tiempo y la comunicación Non-Real-Time para las operaciones no críticas desde el punto de vista del tiempo (p. ej. lectura y escritura acíclica).

Sólo podrá obtenerse una conexión PROFINET-IO redundante configurando un segundo dispositivo o controlador PROFINET-IO y adaptándolo en el programa de usuario.

### 5.1 Bloques funcionales de PROFINET-IO

Para el intercambio de datos acíclico se dispone en SILworX de bloques funcionalmente idénticos a los bloques funcionales de PROFIBUS-DP.

Se dispone de los siguientes bloques funcionales de PROFINET-IO:

Bloque funcional	Descripción de la función
MSTAT 6.9.1	Control del estado del controlador mediante el programa de usuario
RALRM 6.9.2	Lectura de los mensajes de alarma de los dispositivos
RDREC 6.9.4	Lectura de los registros de datos de los dispositivos
SLACT 6.9.5	Control del estado de los dispositivos mediante el programa de usuario
WRREC 6.9.6	Escritura de los registros de datos de los dispositivos

Tabla 38: Sinopsis de bloques funcionales de PROFINET-IO

Los bloques funcionales de PROFINET-IO los parametrizará Ud. igual que los bloques funcionales de PROFIBUS-DP. Véase a partir del capítulo 6.9.

## 5.2 Control del estado de consumidor/proveedor (IOxS)

Con las variables de sistema descritas en este capítulo puede controlarse el estado de consumidor/proveedor (IOxS) mediante el programa de usuario. Si no se desea el control mediante el programa de usuario, a las variables de salida de un dispositivo deberá asignárseles una constante con el valor TRUE. Entonces los estados serán "GOOD" tan pronto como el módulo de comunicación reciba valores de proceso válidos desde el módulo procesador.

En la siguientes figura se ilustra el intercambio de variables de sistema entre el controlador HIMA y un dispositivo DO o DI respectivamente.

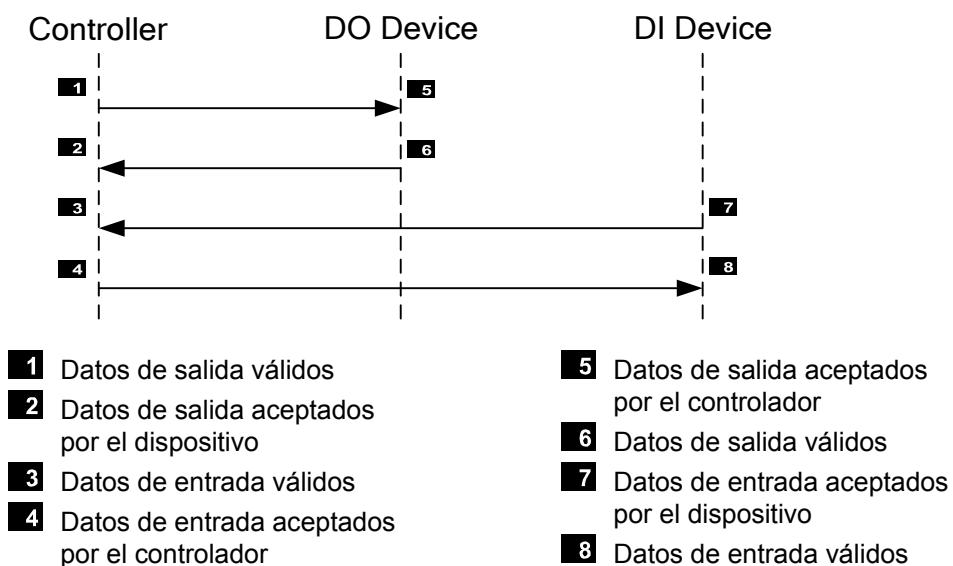


Fig. 27: Control del estado de consumidor/proveedor (IOxS)

### 5.2.1 Variables de control en el controlador HIMA

Con las variables de salida *Valid Output Data* **1** e *Input Data Accepted by Controller* **4** podrá controlarse el estado de consumidor/proveedor (IOxS) mediante el programa de usuario.

Con las variables de entrada *Output Data Accepted by Device* **2** y *Valid Input Data* **3** podrá leerse el estado de consumidor/proveedor (IOxS) mediante el programa de usuario.

### 5.2.2 Variables de control en el dispositivo de DO de HIMA

Con la variable de salida *Valid Output Data* **6** podrá controlarse el estado de consumidor/proveedor (IOxS) mediante el programa de usuario.

Con la variable de entrada *Output Data Accepted by Device* **5** podrá leerse el estado de consumidor/proveedor (IOxS) mediante el programa de usuario.

### 5.2.3 Variables de control en el dispositivo de DI de HIMA

Con la variable de salida *Input Data Accepted by Device* **7** podrá controlarse el estado de consumidor/proveedor (IOxS) mediante el programa de usuario.

Con la variable de entrada *Valid Input Data* **8** podrá leerse el estado de consumidor/proveedor (IOxS) mediante el programa de usuario.

### 5.3 PROFIsafe

¡Se presupone que se conoce la especificación PROFIsafe de PNO!

PROFIsafe usa el protocolo PROFINET para la transmisión de datos orientada a la seguridad hasta el nivel SIL 3 en base a la tecnología Ethernet.

El protocolo PROFIsafe se sobrepone al protocolo PROFINET y contiene los datos útiles seguros, además de información de las copias de seguridad. Los datos seguros PROFIsafe se transmiten junto con datos PROFINET sin relevancia para la seguridad a través del protocolo PROFINET subordinado.

PROFIsafe usa “canales no seguros de transmisión de datos” (Ethernet) de forma similar al principio Black-Channel para transmitir datos seguros. De esta forma se intercambian los datos seguros PROFIsafe entre F-Host y F-Device.

#### PROFIsafe en sistemas de control HIMA

Según la especificación PROFIsafe, F-Host enviará paquetes de notificaciones repetidos hasta que F-Device responda al F-Host con una confirmación de la recepción. Sólo entonces enviará F-Host un nuevo paquete a F-Device.

Mediante cada paquete PROFIsafe repetido se transmitirá el estado momentáneo del valor del proceso. Puede por tanto suceder que la misma señal de proceso tenga valores distintos en los paquetes repetidos.

PROFIsafe ha sido diseñado en los dispositivos HIMA por el lado de recepción de forma tal que los valores de proceso se adopten sólo una vez con la primera recepción de un paquete.

Los valores de proceso de los paquetes repetidos (con igual número consecutivo de paquete) se desecharán.

En caso de cortarse la conexión, las variables de valores de proceso de PROFIsafe adoptarán el valor inicial una vez haya transcurrido el tiempo *F\_WD\_Time*.

Para que un determinado valor de proceso sea recibido por la otra parte (F-Host/F-Device), el valor de proceso deberá estar presente sin variar durante al menos el siguiente tiempo:

$$2 * F\_WD\_Time + F\_WD\_Time2$$

El usuario deberá parametrizar el sistema PROFIsafe de forma tal que el tiempo *SFRT* (Safety Function Response Time) sea adecuado para las respectivas funciones de seguridad.

Véanse las fórmulas de cálculo en el capítulo 5.3.3.

- 
- i** Para un comportamiento conforme a PROFIsafe deberán cumplirse estas condiciones:
- Los valores iniciales de las variables de valores de proceso estarán configurados como “0”
  - El parámetro *AutoAwaitFParamsOnConnLoss* deberá estar desactivado.  
Véase el capítulo 5.11.

¡La configuración por defecto de SILworX cumple el comportamiento conforme a PROFIsafe!

---

### 5.3.1 Byte de estado y byte de control de PROFIsafe

Las dos variables de sistema Control-Byte y Status-Byte están contenidas en cada submódulo PROFIsafe y se intercambian en la comunicación entre F-Host y F-Device. Véase el capítulo 5.7.6.3 y el capítulo 5.11.4.

El byte de control de PROFIsafe se escribe en F-Host y se lee en F-Device.  
El byte de estado de PROFIsafe se escribe en F-Device y se lee en F-Host.

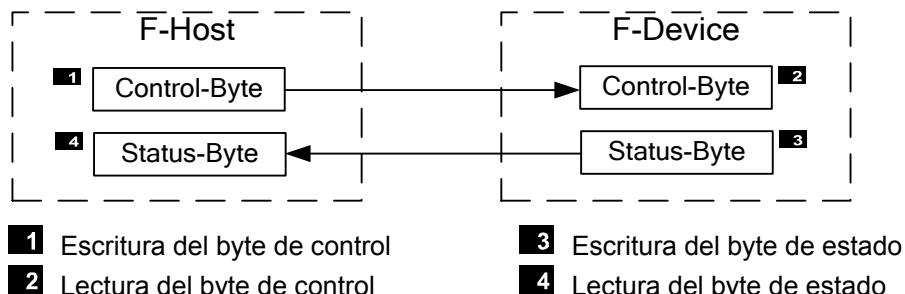


Fig. 28: Byte de estado y byte de control de PROFIsafe



Las variables de sistema Control-Byte y Status-Byte tienen en HIMA funciones adicionales que divergen de la especificación PROFIsafe. Ver Tabla 48 y Tabla 65.

### 5.3.2 F\_WD\_Time (tiempo de WatchDog de PROFIsafe)

Para una conexión PROFIsafe en servicio entre un F-Device y un F-Host de HIMA tiene validez la siguiente desigualdad:

$$\begin{aligned}
 F\_WD\_Time > \\
 3 * \text{tiempo de ciclo de CPU} * \text{cantidad de intervalos de comunicación} + \\
 2 * \text{intervalo de producción de controlador PROFINet}^1 + \\
 1 * DAT (\text{F-Device Acknowledgement Time}) + \\
 2 * \text{tiempo interno de bus de F-Device} + \\
 2 * \text{intervalo de producción de dispositivo PROFINet}^1 + \\
 2 * \text{retardo Ethernet}
 \end{aligned}$$

<sup>1)</sup> El intervalo de producción del controlador PROFINet y del dispositivo PROFINet son por lo general iguales y se calculan como sigue:  
factor de reducción \* factor SendClock \* 31,25 µs



DAT (F-Device Acknowledgement Time) y el tiempo de bus de dispositivo interno constan en la descripción de dispositivo del F-Device de su fabricante.

En el caso de los dispositivos F-Devices HIMax y HIMatrix-L3,  
 $DAT = DAT_{out} = DAT_{in} = 2 * WDT-CPU$

### 5.3.2.1 Comentario sobre F\_WD\_Time (tiempo de WatchDog de PROFIsafe)

1. DAT (F-Device Acknowledgement Time) es el tiempo que un F-Device necesita para reaccionar con una respuesta a una notificación recibida de PROFIsafe. F-Device es la unidad segura (en HIMA el módulo de CPU) que ejecuta F-Device-Stack. Particularmente en el caso de los sistemas/dispositivos modulares, aquí no están contenidos los tiempos de las funciones y los componentes sin función orientada a la seguridad.  
Esta definición del tiempo DAT difiere de la especificación PROFIsafe V2.5c, capítulo 9.3.3., en los siguientes puntos:
  - DAT no contiene los tiempos para el bus interno de F-Device.
  - DAT no contiene la parte del intervalo de producción del dispositivo PROFINet.
  - DAT no contiene los retardos, como son p. ej. filtrado de los valores de entrada/salida, retardos físicos de entrada/salida, etc.
  - DAT significa DATin (entrada) o DATout (salida), según conexión.
  - Para todos los tiempos se usarán los máximos respectivos.
2. El tiempo de bus interno de F-Device es en HIMatrix-L3 y en HIMax el tiempo (*max. Number of Communication Time Slices - 1*) \* WDT CPU.
3. Requisitos: F-Device funciona cíclicamente y su tiempo DAT es  $DAT = 2 * \text{max. cycle}$ .
  - F-Device **no funciona con communication time slices**.  
Si  
*HIMA CPU cycle time \* number of communication time slices < F-Device cycle time*, para *HIMA CPU cycle* deberá sumarse al cálculo del tiempo *F\_WD\_Time* también *Delta = tiempo de ciclo de F-Device - F-Device cycle time \* number of communication time slices*.
  - F-Device funciona **con communication time slices**.  
Si  
*(HIMA CPU cycle time \* number of communication time slices) < F-Device cycle time \* number of F-Device communication time slices*, para *HIMA CPU cycle time* deberá sumarse al cálculo de *F\_WD\_Time* también *Delta = (F-Device cycle time \* number of communication time slices) - (HIMA CPU cycle time \* number of communication time slices)*.

### 5.3.3 SFRT (Safety Function Response Time)

#### 5.3.3.1 Cálculo del tiempo SFRT entre un F-Device y un F-Host HIMA

El tiempo SFRT admisible para una conexión PROFIsafe entre un F-Device y un F-Host HIMA con salida local se calcula como sigue:

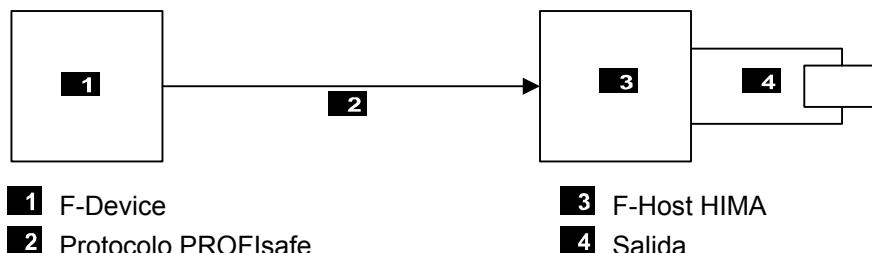


Fig. 29: Tiempo de reacción entre un F-Device y un F-Host HIMA

$$\text{SFRT} \leq \text{MaxDataAgeIn} + 2 * \text{F_WD_TIME} + \text{MaxDataAgeOut} + \text{Tu}$$

Comentarios:

En caso de HIMax/HIMatrix-L3 y uso local de los datos, es decir, cuando no salen en una E/S HIMax, el tiempo de tolerancia a errores del módulo de CPU podrá sustituirse por  $2 * \text{WDT-CPU}$ . Véase también *Comentario sobre los cálculos del tiempo SFRT* en el capítulo 5.3.3.3.

#### 5.3.3.2 Cálculo del tiempo SFRT con F-Devices y un F-Host HIMA

El tiempo SFRT admisible para una conexión PROFIsafe entre un F-Host HIMA y un F-Device con salida local se calcula como sigue:

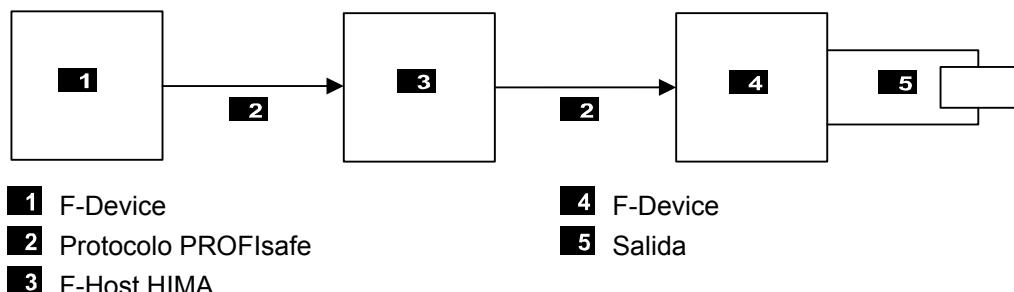


Fig. 30: Tiempo de reacción con un F-Host HIMA y dos F-Devices

$$\text{SFRT} \leq \text{MaxDataAgeIn} + 2 * \text{F_WD_TIME(input)} + 2 * \text{WDZ-CPU} + 2 * \text{F_WD_TIME(output)} + \text{MaxDataAgeOut} + \text{Tu}$$

#### 5.3.3.3 Comentario sobre los cálculos del tiempo SFRT (Safety Function Response Time)

1. Definición de **SFRT** según IEC61784-3 Ed.2
2. Todos los demás retardos adicionales en el programa del usuario (p. ej. por bloques funcionales TOF, TON, etc.) o de módulos (p. ej. por filtrado de salida/entrada, relés, etc.) deberán sumarse.
3. **MaxDataAgeIn** es la antigüedad máxima de un valor de proceso leído de una entrada física, el cual un F-Device inserta en una notificación PROFIsafe. En todo caso, sólo la parte no contenida ya en DATin.

**i**

- En HIMatrix-L3 deberá ponerse MaxDataAgeIn a 0 ms.  
En un sistema HIMax, MaxDataAgeIn deberá
- ser  $FTT\ CPU^1) - 2 * WDZ-CPU - DATin$  para entradas físicas.
  - ponerse a 0 ms para datos formados por el programa del usuario.
- 

#### 4. **MaxDataAgeOut**

es el tiempo de reacción Worst-Case de un F-Output-Device o F-Host para

- a) salida de valores de proceso recibidos en una salida física,
- b) excitación de las salidas físicas tras transcurrir el tiempo  $F\_WD\_Time$  y
- c) desactivación de las salidas físicas en caso de fallar el dispositivo.

- En HIMatrix-L3 es  $MaxDataAgeOut = 3 * WDZ-CPU$  (para salidas físicas)
- En HIMax es  $MaxDataAgeOut = WDZ-CPU + FTZ-CPU^1)$  (para salidas físicas)
- Al transcurrir el tiempo  $F\_WD\_TIME$ , HIMax y HIMatrix-L3 reaccionan sin errores como muy tarde tras  $2 * WDT-CPU$ .
  - Si el F-Host/F-Device (en el caso del módulo de CPU HIMA) falla justo antes de esta reacción, las salidas quedarán en un estado sin energía/excitación en HIMatrix-L3 tras el tiempo  $WDT\ CPU$ . En HIMax esto sucede como muy tarde tras el tiempo  $FTT\ CPU^1) - WDT\ CPU$ , pues el módulo de salida llevará ya un tiempo  $WDT\ CPU$  sin recibir ninguna notificación.
  - Suponiendo que haya sólo **un** error, podrá restarse  $1 * WDT\ CPU$  de  $MaxDataAgeOut$  para HIMatrix-L3 y HIMax.

5. **Tu** es el mínimo de  $DATin$ ,  $DATout$ ,  $WDT\ CPU$ . Teóricamente puede aplicarse para  $DATin$  y  $DATout$  la mitad respectivamente, siempre que el fabricante haya especificado la inexactitud con la que el dispositivo supervisa el tiempo  $F\_WD\_Time$ . Si el dispositivo funciona cíclicamente, entonces  $DAT = 2 * max. device cycle$ .

En HIMatrix-L3 HIMA y F-Devices HIMax es  $DATout = DATin = 2 * WDT\ CPU$ .

6. **F\_WD\_TIME**, véase el capítulo 5.3.2.

<sup>1)</sup> Tiempo de tolerancia de errores del módulo de CPU

## 5.4 Obligaciones para el funcionamiento seguro de PROFIsafe

### 5.4.1 Direccionamiento

La red PROFIsafe corresponde en HIMA a la red Ethernet PROFINET a través de la cual se transmiten las notificaciones PROFIsafe. Como red se entiende aquí una red lógica, que puede extenderse mediante varias redes físicas parciales.

Se tendrá una separación adecuada de la red para PROFIsafe cuando no sea posible que las notificaciones PROFIsafe sobrepasen la separación de la red.

Esto será el caso p. ej. cuando se use un enrutador de base IP y las redes estén conectadas a distintas interfaces Ethernet del enrutador.

No se tendrá una separación de red segura para PROFIsafe cuando p. ej. las redes se conecten mediante un enrutador monopuerto, switches, concentradores o puentes Ethernet.

Incluso cuando se usen switches administrables y las redes PROFIsafe estén separadas p. ej. mediante VLANs basadas en puertos, debería no obstante preferirse la inequivocidad en el direccionamiento. Así se impedirá que p. ej. durante los trabajos de mantenimiento o ampliación se realicen por descuido conexiones entre redes PROFIsafe.

Deberá cumplirse lo siguiente para el direccionamiento de dispositivos PROFIsafe:

- Debe estar asegurado que las direcciones F de los módulos/dispositivos PROFIsafe de una red PROFIsafe sean inequívocos.
- Para la seguridad del direccionamiento se recomienda además que las direcciones F sean inequívocas incluso en redes PROFIsafe separadas.
- En la puesta en servicio y en caso de modificaciones de las funciones de seguridad, deberá comprobarse que la función de seguridad use, más allá de PROFIsafe, las entradas y salidas correctas de los dispositivos PROFIsafe correctos.
- Los módulos F PROFIsafe deberán configurarse de manera tal (p.ej asignándoles una dirección F o un tiempo F\_WD\_Time adecuados) que en una red PROFIsafe los módulos de igual longitud de entrada y de salida tengan diferentes signaturas CRC1. El usuario podrá leer la signatura CRC1 en SILworX.

#### Comentario:

Esto estará garantizado de todas formas para los módulos F afectados cuando en una red PROFIsafe se use un solo F-Host y los parámetros F de los módulos F de una red PROFIsafe se diferencien sólo en la dirección F.

Para que no se genere accidentalmente una signatura CRC1 igual, deberá prestarse atención a que p. ej. los parámetros *F\_WD\_Time*, *F\_Prm\_Flag1/2* sean iguales para todos los módulos F y que los módulos F no usen i-Par-CRC.

### Riesgo de dispositivos PROFIsafe con igual longitud de datos de entrada y de salida

Para el uso de dispositivos PROFIsafe es obligatorio que las longitudes de entrada F-INPUT y de salida F-OUTPUT de la misma conexión PROFIsafe sean desiguales.

De lo contrario es posible que PROFIsafe no detecte errores de direccionamiento de los componentes estándar y/o de la tecnología de transmisión estándar y que ello dé lugar a errores en el comportamiento orientado a la seguridad.

En el caso de los módulos F de HIMA que se configuren como parte del sistema de control HIMA, deberán elegirse diferentes longitudes de entrada F-Input y de salida F-Output.

Para prevenir el riesgo de mal comportamiento de las funciones orientadas a la seguridad, use únicamente módulos F-Input o F-Output. ¡No use módulos F-Input/Output!

#### 5.4.2 Aspectos de la red

La red utilizada para la transmisión de notificaciones PROFIsafe deberá presentar suficiente calidad de transmisión y suficiente disponibilidad.



Si PROFIsafe detecta una menor calidad de transmisión no detectada por los dispositivos de transmisión estándar (Ethernet), se desencadenará una reacción de seguridad.

Tras una reacción de seguridad causada por una insuficiente calidad de transmisión, deberán subsanarse los problemas y volver a garantizar una suficiente calidad de transmisión. Sólo tras tomar las debidas medidas a este propósito deberá Ud. acusar el error como atendido (reset) y habilitar PROFIsafe para su nueva puesta en marcha. Ello se realiza mediante la señal “Operator-Acknowledge” o “Reset”.

#### ⚠ ADVERTENCIA



Se permite aplicar “Operator-Acknowledge” y “Reset” únicamente cuando ya no hay ningún estado peligroso sin resolver.

Una red PROFIsafe deberá protegerse frente a intervenciones no autorizadas (DoS, hackers, ...). Las medidas a tomar se acordarán con la autoridad supervisora. Esto es particularmente importante cuando se usan técnicas inalámbricas para la transmisión. Hallará más información en la especificación PROFIsafe V2.5c, tablas 23 y 24.

#### Disponibilidad con respecto a las notificaciones insertadas

Los paquetes de notificaciones pueden guardarse p. ej. mediante componentes de red tales como switches para insertarlos (enviarlos) más tarde<sup>1)</sup>. Estos paquetes de notificaciones darán lugar a una desconexión si son más antiguos (véase número consecutivo Tabla 48 y Tabla 65) que el último paquete de notificaciones recibido por el dispositivo PROFIsafe.

<sup>1)</sup> Presunción de error para comunicación orientada a la seguridad

**5.5****Controlador PROFINET IO HIMA y PROFIsafe F-Host**

En esta capítulo se describen las propiedades del controlador PROFINET IO HIMA y PROFIsafe F-Host así como las funciones de menús y los cuadros de diálogo de SILworX necesarios para configurar el controlador PROFINET IO HIMA y PROFIsafe Host.

**Sistema necesario para controlador PROFINET IO**

Equipamiento y sistema necesarios:

Elemento	Descripción
Sistema de control	HIMax con módulo COM HIMatrix L3
Módulo de CPU	Las interfaces Ethernet del módulo procesador no pueden usarse para PROFINET IO.
Módulo COM	Ethernet 10/100BaseT.
Activación	Para habilitar se necesitará código de habilitación de software. Véase el capítulo 3.4.

Tabla 39: Equipamiento y sistema necesarios para controlador PROFINET IO

**Propiedades de controlador PROFINET IO y PROFIsafe Host**

Propiedades	Descripción
Con función orientada a la seguridad	No
Velocidad de transmisión	100 Mbit/s Full Duplex
Vía de transmisión	Interfaces Ethernet de los módulos COM Interfaces Ethernet utilizadas aptas simultáneamente también para otros protocolos.
Categoría de conformidad	El controlador PROFINET IO cumple las exigencias de la categoría de conformidad A.
Real Time Class	Categoría RT 1
Máx. cantidad de controladores PROFINET IO	Para cada módulo COM podrá configurar Ud. un controlador PROFINET IO.
Máx. cantidad de relaciones de aplicación (AR) de dispositivos PROFINET IO	Un controlador PROFINET IO puede establecer una relación de aplicación (AR) con un máximo de 64 dispositivos PROFINET IO.
Cantidad de relaciones de comunicación (CRs por cada AR)	Estándar: 1 entrada CR, 1 salida CR, 1 alarma CR
Máx. longitud de datos de proceso de una relación de comunicación (CR)	Salida: máx. 1440 bytes Entrada: máx. 1440 bytes
Cadencia de envío	Possible mediante el ajuste de <i>Reduction Rate</i> al nivel del dispositivo.
<b>Para PROFIsafe son válidas las siguientes propiedades</b>	
Máx. cantidad de F-Hosts (HIMax)	1024
Máx. cantidad de F-Hosts (HIMatrix L3)	512
Máx. longitud de datos de proceso de una relación de comunicación (CR)	Salida: máx. 123 bytes de datos útiles + 5 bytes <sup>1)</sup> Entrada: máx. 123 bytes de datos útiles + 5 bytes <sup>1)</sup>
Máx. tamaño de los datos útiles HIMax: HIMatrix L3:	1024 x 123 bytes = 125 952 bytes 512 x 123 bytes = 62 976 bytes
<sup>1)</sup> 5 bytes de datos de administración (bytes de control y estado y CRC)	

Tabla 40: Propiedades del controlador PROFINET IO

## 5.6 Ejemplo de PROFINET-IO/PROFIsafe

En este ejemplo se describe cómo configurar en un sistema de control HIMax un controlador PROFINET IO HIMA que tiene una conexión a un dispositivo PROFINET IO.

El dispositivo PROFINET IO dispone de un módulo PROFINET IO y un módulo PROFIsafe. En el controlador PROFINET IO HIMA deberá configurarse el dispositivo PROFINET IO tal y como esté montado de hecho.

### 5.6.1 Creación de un controlador PROFINET IO HIMA en SILworX

**Para crear un nuevo controlador PROFINET IO HIMA:**

1. Seleccione **Configuration, Resource, Protocols** en el árbol.
2. Seleccione **New, PROFINET-IO Controller** en el menú contextual de los protocolos, para agregar un nuevo controlador PROFINET IO-IO.
3. En el menú contextual del controlador PROFINET IO seleccione **Properties**.
4. Escriba en el recuadro **Name** el nombre deseado para el controlador.
5. Seleccione **COM Module**.

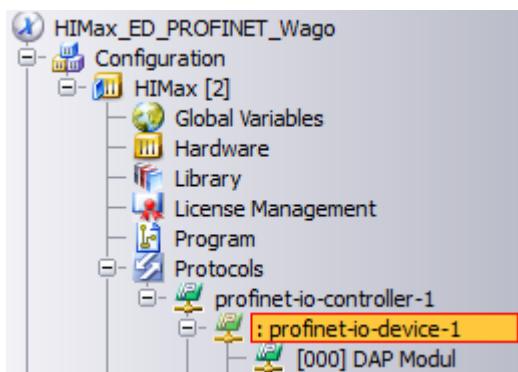


Fig. 31: Controlador PROFINET IO en el árbol de HIMax

#### 5.6.1.1 Configuración del dispositivo en el controlador PROFINET IO HIMA

**Para crear un dispositivo PROFINET IO HIMA en el controlador PROFINET IO:**

1. Seleccione **New, PROFINET-IO Device** en el menú contextual del controlador PROFINET IO-IO.

**Para leer el archivo de biblioteca GSDML de una fuente de datos externa (p. ej. CD, llave USB, internet):**

1. Selecione **Configuration, Resource, Protocols, PROFINET-IO Controller, GSDML Library** en el árbol.
2. En el menú contextual de la biblioteca GSDML seleccione **New** y agregue el archivo GSDML correspondiente al dispositivo PROFINET IO.

**Para cargar el archivo GSDML para un nuevo dispositivo PROFINET IO:**

1. Selecione **Configuration, Resource, Protocols, PROFINET-IO Controller, PROFINET-IO Device** en el árbol.
2. Selecione **Properties** en el menú contextual y abra la ficha **Parameter**.
3. Escriba en el recuadro **Name** el nombre deseado para el dispositivo.
4. Escriba la dirección IP del dispositivo PROFINET-IO Device en el recuadro **IP Address**.
5. Selecione en el menú desplegable de **GSDML File** el archivo de biblioteca GSDML correspondiente al dispositivo PROFINET IO-IO y cierre la ficha **Properties**.

**Para seleccionar punto de acceso DAP (Device Access Point) del dispositivo PROFINET IO:**

1. Seleccione **Protocols, PROFINET-IO Controller, PROFINET-IO Device, DAP Module** en el árbol.
2. Seleccione en el menú contextual **Edit** y elija el registro de datos (*DAP = Device Access Point*) adecuado para el dispositivo PROFINET IO.

- i** El archivo de biblioteca GSDML contiene frecuentemente varios (*DAP = Device Access Point*) de un fabricante.

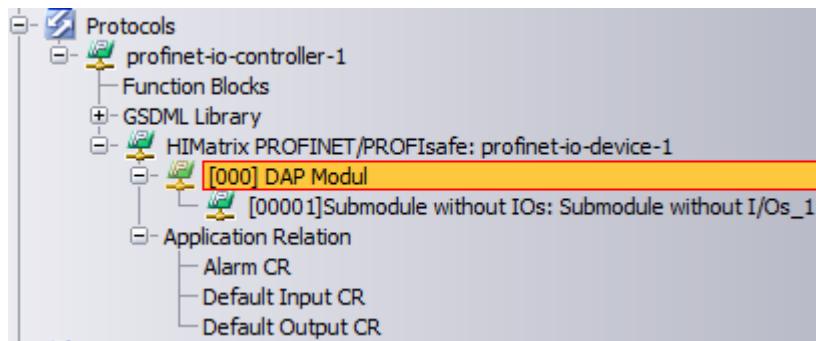


Fig. 32: Device Access Point (DAP) para el dispositivo PROFINET IO

**Para configurar los slots de módulo:**

1. Abra **Protocols, PROFINET-IO Device** en el árbol.
2. Seleccione en el menú contextual de PROFINET-IO Device **New** para abrir la lista de módulos.
3. Seleccione para el dispositivo PROFINET IO los módulos adecuados de la lista de módulos y confírmelos con **Add Module(s)**.

**Para numerar los módulos de dispositivo PROFINET IO:**

El **Device Access Point (DAP) module** tiene como slot fijo el número 0. Todos los demás módulos de dispositivo PROFINET IO deberá Ud. numerarlos.

1. Seleccione en el menú contextual de PROFINET-IO Device Modul **Properties**.
2. Escriba en el recuadro **Slot** los slots de módulos del dispositivo tal y como esté montado de hecho el dispositivo PROFINET IO.
3. Repita este paso para los demás **PROFINET IO device modules**.

Las fichas **Model** y **Features** indican más características del archivo GSDML.

**Para configurar la relación de aplicación:**

1. Abra **PROFIsafe-IO Device, Application Relation** en el árbol.
2. Haga clic con el botón derecho en **Default Input CR** y en el menú contextual seleccione **Properties**.
3. Adapte el parámetro del factor de reducción (póngalo p. ej. a 4).
4. Haga clic con el botón derecho en **Default Output CR** y en el menú contextual seleccione **Properties**.
5. Adapte el parámetro del factor de reducción (póngalo p. ej. a 4).

### 5.6.1.2 Configuración del módulo Device Access Point (DAP)

**Para configurar el módulo Device Access Point (DAP):**

1. Seleccione [000] DAP Module, [xxxxx] DAP Submodule.
2. Haga clic con el botón derecho en [xxxxx] DAP Submodule y en el menú contextual seleccione Edit.
3. En el cuadro de diálogo Edit seleccione la ficha System Variables y asigne a la variable de salida *Input Data Accepted by Controller* una variable global con el valor inicial TRUE, en caso de no desearse el control del estado de consumidor/proveedor con lógica especial del programa de usuario.



Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

**Ajuste de los parámetros de cabecera p. ej. para un DAP con ajustes de alarma.**

**Para ajustar los parámetros de cabecera del Device Access Point (DAP) para el dispositivo PROFINET IO:**

1. Seleccione Protocols, PROFINET-IO Controller, PROFINET-IO Device, DAP Module, [xxxxx] DAP Submodule, Alarm Settings (Header): Parameters en el árbol.
2. Seleccione Properties en el menú contextual.
3. Escriba en el recuadro Name el nombre deseado de parámetros de cabecera.
4. Con el botón Edit se abrirá un cuadro de diálogo en el que podrá Ud. editar los ajustes de interfaces y diagnóstico/alarmas.

### 5.6.1.3 Configuración de módulos de dispositivo PROFINET IO



La suma de las variables, en bytes, deberá ser igual al tamaño del respectivo módulo, en bytes.

**Para configurar el módulo de dispositivo PROFINET IO:**

1. Seleccione [001] PROFINET-IO Device Module, [xxxxx] PROFINET-IO Device Submodule.
2. Haga clic con el botón derecho en [xxxxx] Submodule y seleccione Edit en el menú contextual.
3. En el cuadro de diálogo Edit seleccione la ficha Process Variables.
4. Elija en la selección de objetos la variable adecuada y arrástrela con el ratón al área Input Signals.
5. Haga clic con el botón derecho en una zona libre del área Input Signals y seleccione New Offsets en el menú contextual, para generar los offsets de las variables.
6. En el cuadro de diálogo Edit seleccione la ficha System Variables y asigne a las variables de salida *Input Data Accepted by Controller* y *Valid Output Data* una variable global con el valor inicial TRUE, en caso de no desearse el control del estado de consumidor/proveedor.



Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

### 5.6.1.4 Configuración de módulos de dispositivo IO PROFIsafe

- i** La suma de las variables, en bytes, deberá ser igual al tamaño del respectivo módulo, en bytes.

#### Para configurar el módulo de dispositivo IO PROFIsafe:

1. Seleccione **[001] PROFIsafe-IO Device Module, [xxxxx] PROFIsafe-IO Device Submodule**.
2. Haga clic con el botón derecho en **[xxxxx] Submodule** y seleccione **Edit** en el menú contextual.
3. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.
4. Elija en la selección de objetos la variable adecuada y arrástrela con el ratón al área **Input Signals**.
5. Haga clic con el botón derecho en una zona libre del área **Input Signals** y seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.
6. En el cuadro de diálogo **Edit** seleccione la ficha **System Variables** y asigne a las variables de salida *Input Data Accepted by Controller* y *Valid Output Data* una variable global con el valor inicial TRUE, en caso de no desearse el control del estado de consumidor/proveedor.

- i** Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

#### Para configurar los parámetros F:

1. Seleccione **[001] PROFIsafe-IO Device Module, [xxxxx] PROFIsafe-IO Device SubModule, F-Parameter**.
2. Haga clic con el botón derecho en **F-Parameter** y en el menú contextual seleccione **Properties**.
3. Ajuste los siguientes parámetros:
  - **F\_Dest\_Add**: ajuste de la dirección de destino del módulo de dispositivo
  - **F\_WD\_Time**: tiempo de WatchDog de la conexión a este módulo de dispositivo

#### Verificación de la configuración PROFINET-IO

1. Abra **Configuration, Resource, Protocols, PROFINET-IO Controller** en el árbol.
2. Haga clic con el botón derecho en **PROFINET-IO Controller** y seleccione **Verification** en el menú contextual.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.

- i** Recompile el recurso y cárguelo a los sistemas de control, para que la configuración sea efectiva para la comunicación de PROFINET IO.

### 5.6.1.5 Determinación de los dispositivos PROFINET IO en la red

#### Para hallar el dispositivo PROFINET IO en la red Ethernet:

1. Ingrese (ejecute Modul-Login) al módulo de comunicación con el **PROFINET IO Controller**.
2. Seleccione la vista en línea **PROFINET-IO Controller**, **PROFINET-IO Station** en el árbol.
3. Seleccione **Get PROFINET IO Network Stations** en el menú contextual.
  - Se mostrará una lista de todos los dispositivos PROFINET de la red de este controlador PROFINET IO.

#### Para configurar el dispositivo PROFINET IO en la vista en línea:

1. En la lista de dispositivos haga clic con el botón derecho en el dispositivo PROFINET IO a configurar para modificar sus ajustes.
2. Con la función de menú contextual **Name the PROFINET IO Device** podrá Ud. definir el nombre del dispositivo.
  - Cerciórese de que el nombre de dispositivo PROFINET IO sea el mismo que el utilizado al planificar el proyecto (se permiten sólo minúsculas).
3. Con la función de menú contextual **Network Settings** definirá Ud. IP-Address, Subnet-Mask y Gateway.



El nombre de dispositivo PROFINET IO y los ajustes de red del dispositivo PROFINET IO deberán estar registrados en el controlador PROFINET IO, pues de lo contrario no será posible la comunicación.

## 5.7 Funciones del menú del controlador PROFINET IO

### 5.7.1 Ejemplo de estructura en árbol del controlador PROFINET IO

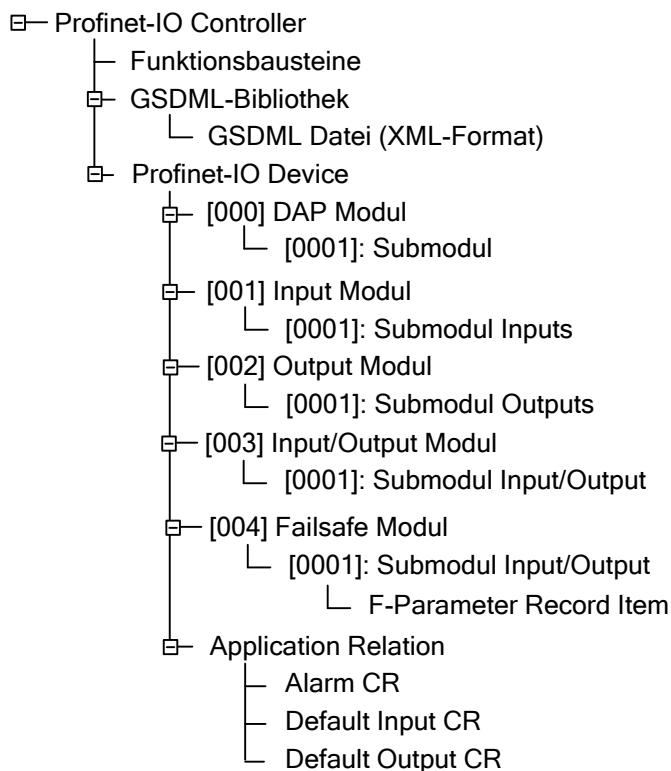


Fig. 33: Estructura en árbol del controlador PROFINET IO

### 5.7.2 Controlador PROFINET IO

Con la función de menú **Properties** del menú contextual del controlador PROFINET IO se abre el cuadro de diálogo **Properties**. Este cuadro de diálogo contiene las siguientes fichas:

#### 5.7.2.1 Ficha PROFINET-IO (propiedades)

Elemento	Descripción
Type	Controlador PROFINET IO
Name	Nombre de dispositivo del controlador PROFINET IO
Process Data Refresh Rate [ms]	Tiempo de actualización, en ms, para el intercambio de datos del protocolo entre COM y CPU. Si para el tiempo de actualización <i>Refresh Rate</i> se ha elegido cero o un valor menor que el tiempo de ciclo del sistema de control, el intercambio de datos tendrá lugar lo más rápidamente posible. Rango de valores: 4...(2 <sup>31</sup> - 1) Valor por defecto: 0
Force Process Data Consistency	Activado: Transferencia de todos los datos del protocolo desde CPU a COM dentro de un ciclo de la CPU. Desactivado: Transferencia de todos los datos del protocolo desde CPU a COM repartida a lo largo de varios ciclos de la CPU, con 1100 bytes por sentido de transmisión de datos. Con ello es posible reducir también el tiempo de ciclo. Valor por defecto: Activado

Elemento	Descripción
Module	Selección del módulo COM en el que se ejecuta este protocolo.
Activate Max. $\mu$ P Budget	Activado: Aplicar el límite $\mu$ P-Budget tomado del recuadro <i>Max. <math>\mu</math>P-Budget in [%]</i> . Desactivado: No se usará límite de $\mu$ P-Budget para este protocolo.
Max. $\mu$ P budget in [%]	Máximo valor de $\mu$ P-Budget del módulo que se permite que tenga lugar durante la ejecución del protocolo. Rango de valores: 1...100% Valor por defecto: 30%
RPC Port Server	Remote Procedure Call Port Rango de valores: 1024...65535 Valor por defecto: 49152 ¡RPC Port Server y RPC Port Client deberán ser diferentes!
RPC Port Client	Remote Procedure Call Port Rango de valores: 1024...65535 Valor por defecto: 49153 ¡RPC Port Server y RPC Port Client deberán ser diferentes!
F_Source_Add	Dirección del controlador (F-Host). En una red PROFIsafe el usuario deberá usar direcciones inequívocas de controladores/dispositivos PROFIsafe. Véase también IEC61784-3-3 V2.5c, capítulo 9.7

Tabla 41: Ficha PROFINET-IO (propiedades)

### 5.7.3 Dispositivo PROFINET IO (en el controlador)

Con la función de menú **Properties** del menú contextual del controlador PROFINET IO se abre el cuadro de diálogo **Properties**, que contiene las siguientes fichas:

#### 5.7.3.1 Ficha “Parameters” (propiedades)

Elemento	Descripción
Name	Nombre de dispositivo del dispositivo PROFINET IO
IP Address	Dirección IP del interlocutor de comunicación. Rango de valores: 0.0.0.0...255.255.255.255 Valor por defecto: 192.168.0.99 No use direcciones IP asignadas. Véase el capítulo 3.5.3.
Subnet Mask	Máscara de subred de la subred direccionada en la que se halla el dispositivo. Rango de valores: 0.0.0.0...255.255.255.255 Valor por defecto: 255.255.255.0
Gsdml File	GSDML es la abreviatura de Generic Station Description Markup Language, un lenguaje descriptivo basado en XML. El archivo GSDML contiene los datos maestros del dispositivo PROFINET

Tabla 42: Ficha “Parameters” (propiedades)

#### Fichas “Model” y “Features”

Las fichas **Model** y **Features** indican más características del archivo GSDML, como p. ej. *Manufacturer Name*, *Device Description* o *Supported Factors*. Estos datos son información de ayuda para parametrizar el dispositivo y no pueden ser modificados.

### 5.7.4 Módulo DAP (Device Access Point Module)

Dentro de un dispositivo PROFINET se creará siempre un módulo para la conexión del bus (DAP: Device Access Point). El módulo DAP está predeterminado y no podrá borrarse.

Con la función de menú **Properties** del menú contextual del módulo DAP se abre el cuadro de diálogo **Properties**, que contiene las siguientes fichas:

#### Ficha “Parameters” (propiedades)

Elemento	Descripción
Name	Nombre del módulo DAP
Slot	No modificable. Valor por defecto: 0

Tabla 43: Ficha “Parameters” (propiedades)

#### Fichas “Model” y “Features”

Las fichas **Model** y **Features** indican más características del archivo GSDML, como p. ej. *Module ID*, *Hardware/Software Version*. Estos datos son información de ayuda para parametrizar el dispositivo y no pueden ser modificados.

#### 5.7.4.1 Submódulo DAP (propiedades)

Con la función de menú **Properties** del menú contextual del submódulo DAP se abre el cuadro de diálogo **Properties**, que contiene las siguientes fichas:

##### Ficha “Parameters”

Elemento	Descripción
Name	Nombre del submódulo de entrada
Sub-Slot	Valor por defecto: 1
IO Data CR, Inputs	Selección de la relación de comunicación (CR) en la que se desea que transmitan las entradas del submódulo. - None - Default Input CR
Input Data Accepted by Controller	Selección de la relación de comunicación (CR) en la que se desea que se transmita el estado de consumidor (CS) del submódulo. - None - Default Output CR

Tabla 44: Ficha “Parameters”

##### Fichas “Model” y “Features”

Las fichas **Model** y **Features** indican más características del archivo GSDML, como p. ej. *Submodul ID*, *Data Length*. Estos datos son información de ayuda para parametrizar el dispositivo y no pueden ser modificados.

#### 5.7.4.2 DAP Submodule (Edit)

Con la función de menú **Edit** del menú contextual de los submódulos de entrada se abre el cuadro de diálogo **Edit**, que contiene las siguientes fichas:

##### Ficha “System Variables”

La ficha **System Variables** ofrece las siguientes variables de sistema, las cuales permiten evaluar o controlar el estado del submódulo PROFINET IO en el programa del usuario.

Elemento	Descripción
Valid input data	True      Datos de entrada válidos GOOD False     Datos de entrada no válidos BAD
Input Data Accepted by Controller	True      Datos de entrada válidos GOOD False     Datos de entrada no válidos BAD

Tabla 45: Ficha “System Variables”



Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

#### 5.7.4.3 Parámetros de cabecera

Algunos dispositivos contienen los denominados parámetros de cabecera, en los que podrán activarse y desactivarse p. ej. diagnósticos, alarmas e interfaces.

### 5.7.5 Módulos PROFINET IO de entrada/salida

Un módulo PROFINET IO de entrada/salida puede tener varios submódulos. Los controladores PROFINET IO HIMax tienen un submódulo en cada módulo de entrada/salida PROFINET IO.

En los módulos de entrada PROFINET IO se registran las variables de entrada del controlador PROFINET IO enviadas por el dispositivo PROFINET IO.

En los módulos de salida PROFINET IO se registran las variables de salida del controlador PROFINET IO enviadas al dispositivo PROFINET IO.

#### Para crear los módulos PROFINET IO necesarios:

1. Abra **Configuration, Resource, Protocols, PROFINET-IO Device** en el árbol.
2. Seleccione en el menú contextual de **PROFINET IO Device, New**.
3. Seleccione los módulos adecuados.

Con la función de menú **Properties** del menú contextual de los módulos de entrada/salida PROFINET IO se abre el cuadro de diálogo **Properties**, que contiene las siguientes fichas:

#### Ficha “Parameters”

Elemento	Descripción
Name	Nombre del módulo PROFINET IO de entrada/salida
Slot	0 hasta 32767 Valor por defecto: 1

Tabla 46: Ficha “Parameters” del módulo PROFINET IO de entrada/salida

#### Fichas “Model” y “Features”

Las fichas **Model** y **Features** indican más características del archivo GSDML, como p. ej. *Module ID, Hardware/Software Version*. Estos datos son información de ayuda para parametrizar el dispositivo y no pueden ser modificados.

### 5.7.6 Entrada de submódulo

Con los parámetros de los submódulos se ajustan las relaciones de comunicación y la reacción de los módulos en caso de interrupciones de la conexión.

#### 5.7.6.1 Entrada de submódulo (propiedades)

Con la función de menú **Properties** del menú contextual del *submódulo de entrada* se abre el cuadro de diálogo **Properties**, que contiene las siguientes fichas:

Ficha “Parameters”

Elemento	Descripción	
Name	Nombre del submódulo de entrada	
Sub-Slot	No modificable para controlador PROFINET IO HIMax. Valor por defecto: 1	
IO Data CR, Inputs	Selección de la relación de comunicación (CR) en la que se desea que transmitan las entradas del submódulo. - None - Default Input CR	
Input data accepted by Controller	Selección de la relación de comunicación (CR) en la que se desea que se transmita el estado de consumidor (CS) del submódulo. - None - Default Output CR	
Shared Input	Activado	Varios controladores PROFINET IO tendrán acceso a las entradas.
	Desactivado	Sólo un controlador PROFINET IO tendrá acceso a las entradas.
Input Values if IO CR is Disconnected	Reacción de las variables de entrada de este submódulo PROFINET IO en caso de interrumpirse la conexión.  Conservar el último valor  Aplicar valores iniciales	Las variables de entrada se congelarán en su valor momentáneo y se usarán hasta que vuelva a establecerse la conexión.  Para las variables de entrada se usarán los valores iniciales.

Tabla 47: Ficha “Parameters”

#### Fichas “Model” y “Features”

Las fichas **Model** y **Features** indican más características del archivo GSDML, como p. ej. *Submodule ID*, *Hardware/Software Version* o *Data Length*. Estos datos son información de ayuda para parametrizar el dispositivo y no pueden ser modificados.

### 5.7.6.2 Entrada de submódulo (editar)

Con la función de menú **Edit** del menú contextual de los submódulos de entrada se abre el cuadro de diálogo **Edit**. Este cuadro de diálogo contiene las siguientes fichas:

#### Ficha “System Variables”

La ficha **System Variables** ofrece las siguientes variables de sistema, las cuales permiten evaluar el estado del submódulo PROFINET IO en el programa del usuario.

Elemento	Descripción	
Valid Input Data	True	Datos de entrada válidos GOOD
	False	Datos de entrada no válidos BAD
Input Data Accepted by Controller	True	Datos de entrada válidos GOOD
	False	Datos de entrada no válidos BAD
<b>Los siguientes parámetros están disponibles sólo para módulos PROFIsafe</b>		
Valid Output Data	True	Datos de salida válidos GOOD
	False	Datos de salida no válidos BAD
Output Data Accepted by Device	True	Datos de salida válidos GOOD
	False	Datos de salida no válidos BAD



Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

Elemento	Descripción
PROFIsafe Control	<p>PROFIsafe envía desde el controlador, con cada notificación, el byte de control PROFIsafe al dispositivo, el cual podrá aplicarse en el programa del usuario. Véase también el capítulo 5.3.1.</p> <p>Bit 0    iPar_EN_C: Para cargar nuevos “iParameters” al dispositivo F, iPar_EN_C deberá ponerse como TRUE para desbloquear así el dispositivo F. Mientras iPar_EN_C sea TRUE, se intercambiarán valores seguros “0” entre F-Host y F-Device.</p> <p>Bit 1    OA_C: Operator Acknowledge. Tras un error de PROFIsafe (p. ej. time-out o error de suma CRC), el bit deberá ponerse como TRUE al menos durante un ciclo de PROFIsafe. Si se desea reiniciar una conexión PROFIsafe, el acuse y la habilitación por parte del operador (Operator-Acknowledge) no deberá tener lugar hasta haberse subsanado todo estado peligroso.</p> <p>Bit 2    Reservado</p> <p>Bit 3    Reservado</p> <p>Bit 4    Activate_FV_C: FALSE: Se intercambiarán valores de proceso entre F-Host y F-Device. TRUE: Se intercambiarán valores seguros “0” entre F-Host y F-Device.</p> <p>Bit 5    Reservado</p> <p>Bit 6    Reservado</p> <p>Bit 7    Reset_Comm: Reset de la comunicación PROFIsafe, el bloque de protocolo se pone en el estado inicial. El bit deberá ser TRUE hasta que se lea el estado PROFIsafe relee el bit 2 <i>Reset_Comm</i> como TRUE.</p>
PROFIsafe RoundTrip Time last	Para un F-Host éste es el tiempo entre el envío de una notificación de datos (con número consecutivo N) y la recepción del respectivo acuse de confirmación (Acknowledgment) (con número consecutivo N), medido en milisegundos.

Elemento	Descripción
PROFIsafe Status	<p>PROFIsafe recibe en el host, con cada notificación, el byte de estado PROFIsafe, el cual podrá evaluarse en el programa del usuario.</p> <p>Bit 0      iPar_OK_S TRUE: conservar nuevos “iParameters” FALSE: sin modificaciones</p> <p>Bit 1      OA_Req_S Operator Acknowledge Requested.</p> <p>Bit 2      Reset_Comm es el valor Reset_Comm leído del byte del control del host. Este bit señala si ha llegado el Reset_Comm.</p> <p>Bit 3      FV_activated_S</p> <p>Bit 4      Toggle_h</p> <p>Bit 5      Device_Fault TRUE: El F-Device ha comunicado un error de dispositivo. FALSE: El F-Device no ha comunicado un error de dispositivo.</p> <p>Bit 6      WD_timeout TRUE: El F-Device ha comunicado un time-out de WD o en el F-Host ha expirado el time-out de host. FALSE: Ni en el F-Device ni en el F-Host se ha llegado a un time-out.</p> <p>Bit 7      CRC TRUE: El F-Device ha comunicado un error de suma de verificación CRC o en el F-Host se ha producido un error de CRC. FALSE: Ni en el F-Device ni en el F-Host se ha producido un error de CRC.</p>

Tabla 48: Ficha “System Variables”

En la ficha **Process Variables** se escriben las variables de entrada.

### 5.7.6.3 F-Parameters de la entrada de submódulo (sólo para módulos PROFIsafe)

Los dispositivos PROFIsafe F-Devices necesitan “F-Parameters” normalizados para el intercambio seguro de datos de proceso. El F-Device no establecerá la comunicación hasta que se hayan ajustado “F-Parameters” válidos. Los parámetros difuminados en gris no podrán editarse, siendo en parte predefinidos por el archivo GSDML o calculados automáticamente.

Elemento	Descripción
Name	Nombre del módulo
Index	Índice del módulo
F_Par_Version	Compatible sólo con el modo V2. Se deniega el modo V1. Se determina mediante el archivo GSDML.
F_Source_Add	¡La dirección F_Source_Address del F-Host deberá ser inequívoca dentro de la red PROFIsafe! Rango de valores: 1 hasta 65534
F_Dest_Add	¡La dirección F_Destination_Address del F-Devices deberá ser inequívoca dentro de la red PROFIsafe! Rango de valores: 1 hasta 65534
F_WD_Time	Tiempo de WatchDog Rango de valores: 1 ms hasta 65534 ms
F_iPar_CRC	Escriba F_iPar_CRC del F-Device en este recuadro.
F_SIL	Lectura del nivel SIL 0 – SIL1 1 – SIL2 2 – SIL3 3 – NoSIL Se determina mediante el archivo GSDML.
F_Check_iPar	Lectura de iParameter CRC Se determina mediante el archivo GSDML.
F_Block_ID	Estructura de los F-Parameters Se determina mediante el archivo GSDML.
F_CRC_Length	Especifica si se usa CRC de 3 bytes o CRC de 4 bytes. Se determina mediante el archivo GSDML.
F_Par_CRC	Lectura de F-Parameter CRC (CRC1) Se calcula a partir de los F-Parameters actuales

Tabla 49: Parámetros F de entrada de submódulo (propiedades)

## 5.7.7 Salida de submódulo

Con los parámetros de los submódulos se ajustan las relaciones de comunicación y la reacción de los módulos en caso de interrupciones de la conexión.

### 5.7.7.1 Salida de submódulo (propiedades)

Con la función de menú **Properties** del menú contextual del *Output Submodules* se abre el cuadro de diálogo **Properties**, que contiene las siguientes fichas:

#### Ficha “Parameters”

Elemento	Descripción
Name	Nombre del submódulo de salida
Sub-Slot	No modificable para controlador PROFINET IO HIMax. Valor por defecto: 1
IO Data CR, Outputs	Selección de la relación de comunicación (CR) en la que se desea que transmitan las salidas del submódulo. - None - Default Input CR
Output Data Accepted by Device	Selección de la relación de comunicación (CR) en la que se desea que se transmita el estado de consumidor (CS) del submódulo. - None - Default Output CR

Tabla 50: Ficha “Parameters”

#### Fichas “Model” y “Features”

Las fichas **Model** y **Features** indican más características del archivo GSDML, como p. ej. *Submodule ID*, *Hardware/Software Version* o *Data Length*. Estos datos son información de ayuda para parametrizar el dispositivo y no pueden ser modificados.

### 5.7.7.2 Salida de submódulo (editar)

Con la función de menú Edit del menú contextual del submódulo de salida se abre el cuadro de diálogo **Edit**, que contiene las siguientes fichas:

#### Ficha “System Variables”

La ficha **System Variables** ofrece las siguientes variables de sistema, las cuales permiten evaluar el estado del submódulo PROFINET IO en el programa del usuario.

Elemento	Descripción	
Valid Output Data	True	Datos de salida válidos GOOD
	False	Datos de salida no válidos BAD
Output Data Accepted by Device	True	Datos de salida válidos GOOD
	False	Datos de salida no válidos BAD
<b>Los siguientes parámetros están disponibles sólo para módulos PROFIsafe</b>		
Valid Input Data	True	Datos de entrada válidos GOOD
	False	Datos de entrada no válidos BAD
Input Data Accepted by Controller	True	Datos de entrada válidos GOOD
	False	Datos de entrada no válidos BAD
Hallará más parámetros de módulos PROFIsafe en Tabla 48.		

Tabla 51: Ficha “System Variables”



Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

En la ficha **Process Variables** se escriben las variables de salida

### 5.7.7.3 F-Parameters de la salida de submódulo (sólo para módulos PROFIsafe)

F-Parameters se describen en el capítulo 5.7.6.3.

## 5.7.8 Entradas y salidas de submódulo

Con los parámetros de los submódulos se ajustan las relaciones de comunicación y la reacción de los módulos en caso de interrupciones de la conexión.

### 5.7.8.1 Entradas y salidas de submódulo (propiedades)

Con la función de menú **Properties** del menú contextual de los *submódulos de entrada y salida (Input/Output Submodule)* se abre el cuadro de diálogo **Properties**, que contiene las siguientes fichas:

Ficha “Parameters”

Elemento	Descripción
Name	Nombre del submódulo de entrada/salida
Sub-Slot	No modificable para controlador PROFINET IO HIMax. Valor por defecto: 1
IO Data CR, Inputs	Selección de la relación de comunicación (CR) en la que se desea que transmitan las entradas del submódulo. - None - Default Input CR
IO Data CR, Outputs	Selección de la relación de comunicación (CR) en la que se desea que transmitan las salidas del submódulo. - None - Default Output CR
Input Data Accepted by Controller	Selección de la relación de comunicación (CR) en la que se desea que se transmita el estado de consumidor (CS) del submódulo. - None - Default Output CR
Output Data Accepted by Device	Selección de la relación de comunicación (CR) en la que se desea que se transmita el estado de consumidor (CS) del submódulo. - None - Default Input CR
Input Values When IO CR is Disconnected	- Conservar el último valor - Aplicar valores iniciales

Tabla 52: Ficha “Parameters”

### Fichas “Model” y “Features”

Las fichas **Model** y **Features** indican más características del archivo GSDML, como p. ej. *Submodule ID*, *Hardware/Software Version* o *Data Length*. Estos datos son información de ayuda para parametrizar el dispositivo y no pueden ser modificados.

### 5.7.8.2 Entradas y salidas de submódulo (editar)

Con la función de menú **Edit** del menú contextual de los submódulos de entrada/salida se abre el cuadro de diálogo **Edit**, que contiene las siguientes fichas:

#### Ficha “System Variables”

La ficha **System Variables** ofrece las siguientes variables de sistema, las cuales permiten evaluar el estado del submódulo PROFINET IO en el programa del usuario.

Elemento	Descripción	
Valid Output Data	True	Datos de salida válidos GOOD
	False	Datos de salida no válidos BAD
Output Data Accepted by Device	True	Datos de salida válidos GOOD
	False	Datos de salida no válidos BAD
Valid Input Data	True	Datos de entrada válidos GOOD
	False	Datos de entrada no válidos BAD
Input Data Accepted by Controller	True	Datos de entrada válidos GOOD
	False	Datos de entrada no válidos BAD
Los parámetros de módulos PROFIsafe se describen en la Tabla 48.		

Tabla 53: Ficha “System Variables”

En la ficha **Process Variables** se escriben las variables de entrada y las variables de salida en sus áreas respectivas



Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

### 5.7.8.3 F-Parameters de entradas/salidas de submódulo (sólo para módulos PROFIsafe)

F-Parameters se describen en el capítulo 5.7.6.3.

### 5.7.9 Relación de aplicación (propiedades)

Una relación de aplicación (AR: Application Relation) es una estructura lógica para el intercambio de datos entre controlador y dispositivo. La transmisión de datos dentro de la relación de aplicación de este ejemplo tiene lugar (véase Fig. 34) mediante las relaciones de comunicación predeterminadas (Alarm CR, Default Input CR y Default Output CR). Estas relaciones de comunicación están ya predeterminadas por defecto en los módulos de entrada y salida.

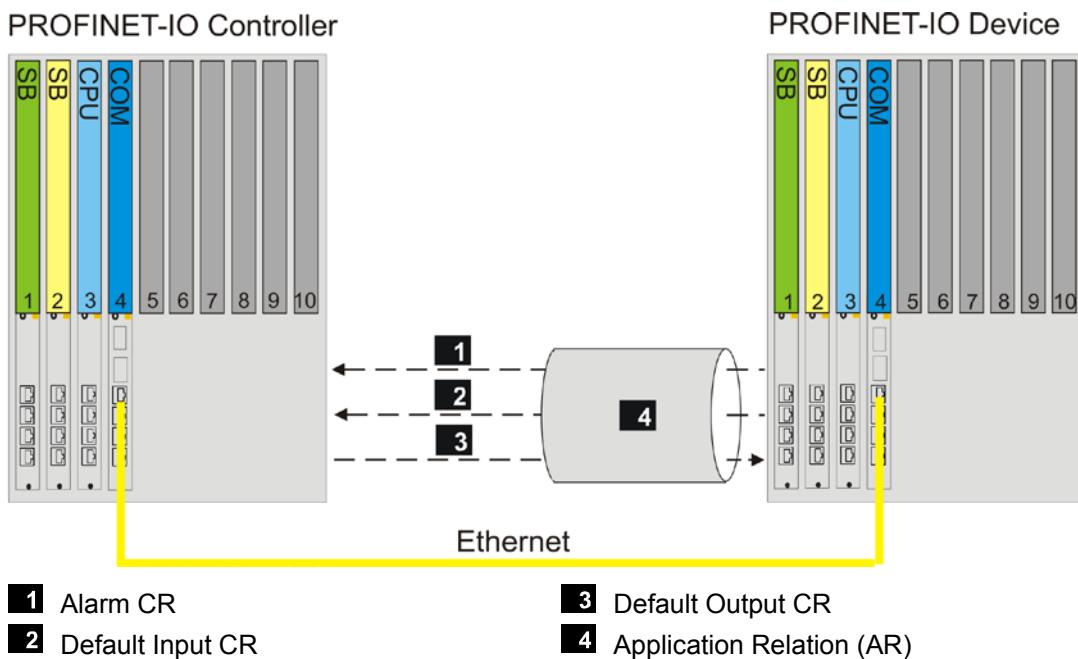


Fig. 34: Comunicación mediante PROFINET/PROFIsafe

Con la función de menú **Properties** del menú contextual de *Application Relation* se abre el cuadro de diálogo **Properties**.

Elemento	Descripción
Name	No modificable
AR UUID	Indicativo para la identificación inequívoca de la relación de aplicación (AR). No modificable
Connection Establishment Timeout Factor	Con este parámetro se calcula el tiempo (desde la perspectiva de un dispositivo PROFINET IO) que podrá transcurrir como máximo, al establecer conexión, entre el envío de la respuesta al requerimiento de conexión y la recepción de una nueva solicitud desde el controlador PROFINET IO. Rango de valores: 1...1000 (x 100 ms) Valor por defecto: 600
Supervisor may adopt the AR	Definición de si un supervisor de PROFINET IO puede aplicar la relación de aplicación (AR). 0 No permitido 1 Permitido Valor por defecto: 0

Tabla 54: Relación de aplicación (propiedades)

### 5.7.10 Alarm CR (propiedades)

Dentro de una relación de aplicación podrán establecerse varias relaciones de comunicación (CRs: Communication Relations).

Mediante la relación de comunicación Alarm CR, un dispositivo PROFINET IO transmite alarmas al controlador PROFINET IO.

Con la función de menú **Properties** del menú contextual de Application Relation se abre el cuadro de diálogo **Properties**, que contiene las siguientes fichas:

Elemento	Descripción							
Name	No modificable							
VLAN ID, High Priority	Para asegurar la separación, a cada LAN virtual (VLAN) se le asignará un número único. Un dispositivo perteneciente a la VLAN con ID = 1 podrá comunicar con cualquier otro dispositivo de la misma VLAN, pero no con un dispositivo de otra VLAN con ID = 2, 3, ... Para el rango de valores véase también IEC-61158-6: <table> <tr> <td>0x000</td> <td>Sin VLAN</td> </tr> <tr> <td>0x001</td> <td>VLAN estándar</td> </tr> <tr> <td>0x002 hasta 0xFFFF</td> <td>Véase IEEE 802.1 Q</td> </tr> </table> Valor por defecto: 0		0x000	Sin VLAN	0x001	VLAN estándar	0x002 hasta 0xFFFF	Véase IEEE 802.1 Q
0x000	Sin VLAN							
0x001	VLAN estándar							
0x002 hasta 0xFFFF	Véase IEEE 802.1 Q							
VLAN ID, Low Priority	Véase la descripción en VLAN ID, High Priority Valor por defecto: 0							
Alarm Priority	Usar la prioridad del usuario Ignorar la prioridad del usuario	Se usará la prioridad asignada por el usuario. Se ignorará la prioridad asignada por el usuario. La alarma generada será siempre de baja prioridad.						
Alarm Resends	Máxima cantidad de envíos del dispositivo, en caso de no responder el controlador. Rango de valores: 3 hasta 15 Valor por defecto: 10							
Alarm Timeout Factor	Con el "RTA Timeout Factor" se calcula el tiempo de timeout del dispositivo que se permite que transcurra como máximo desde el envío de un "RTA_Data(Alarm)-Frame" y la recepción de "RTA_ACK-Frame" como confirmación. $\text{RTA Timeout} = \text{RTA Timeout Factor} \times 100 \text{ ms}$ Rango de valores: 1...65535 Valor por defecto: 5							

Tabla 55: Alarm CR (propiedades)

### 5.7.11 Input CR (propiedades)

Mediante la relación de comunicación Input CR, un dispositivo PROFINET IO transmite variables al controlador PROFINET IO.

Con la función de menú **Properties** del menú contextual de Default Input CR se abre el cuadro de diálogo **Properties**. Este cuadro de diálogo contiene los siguientes parámetros:

Elemento	Descripción								
Name	Cualquier nombre inequívoco para una CR de entrada Default Input CR no es modificable								
Type	1 (no modificable)								
Send Clock Factor	Send Clock Factor determina el reloj de envío de la transmisión cíclica de una CR de entrada/salida. <b>Send Clock = Send Clock Factor x 31,25 µs</b> Rango de valores: 1...128 Valor por defecto: 32								
Reduction Factor	El factor de reducción permite reducir el tiempo real de ciclo del envío de datos de una CR de entrada/salida a Send Clock. El tiempo real de ciclo de los datos se calcula a partir de: <b>Send Cycle = Reduction Factor x Send Clock</b> Rango de valores: 1...16384 Valor por defecto: 32 (según dispositivo)								
Watchdog Factor	Con Watchdog Factor se calcula el tiempo que, desde la perspectiva de un consumidor de CR de entrada/salida, se permite que transcurra como máximo entre la recepción de dos frames: <b>Watchdog Time = WatchdogFactor x Send Cycle</b> Rango de valores: 1...7680 Valor por defecto: 3								
VLAN-ID	Para asegurar la separación, a cada LAN virtual (VLAN) se le asignará un número único. Un dispositivo perteneciente a la VLAN con ID = 1 podrá comunicar con cualquier otro dispositivo de la misma VLAN, pero no con un dispositivo de otra VLAN con ID = 2, 3, ... Para el rango de valores véase también IEC-61158-6: <table style="margin-left: 20px;"> <tr> <td>0x000</td> <td>Sin VLAN</td> </tr> <tr> <td>0x001</td> <td>VLAN estándar</td> </tr> <tr> <td>0x002 hasta</td> <td>Véase IEEE 802.1 Q</td> </tr> <tr> <td>0xFFFF</td> <td></td> </tr> </table> Valor por defecto: 0	0x000	Sin VLAN	0x001	VLAN estándar	0x002 hasta	Véase IEEE 802.1 Q	0xFFFF	
0x000	Sin VLAN								
0x001	VLAN estándar								
0x002 hasta	Véase IEEE 802.1 Q								
0xFFFF									

Tabla 56: Input CR (propiedades)

### 5.7.11.1 Input CR (editar)

Con la función de menú **Edit** del menú contextual de Default Input CR se abre el cuadro de diálogo **System Variables** con las siguientes variables de sistema.

Elemento	Descripción	
	Valor	Descripción
Data Status Input CR	0	<p>State Primary describe el canal principal en caso de conexiones redundantes 1 = Primary 0 = Backup En caso de conexiones mono: 1 = Connected 0 = Not Connected</p>
	1	No se usa
	2	<p>Data Valid "Invalid" se aplica en la fase de puesta en marcha o cuando la aplicación no es capaz de comunicar errores mediante IOPS 1 = Valid 0 = Invalid</p>
	3	No se usa
	4	<p>Process State Tiene sólo carácter informativo, la validez real de los datos se comunica mediante IOPS. 1 = Run 0 = Stop</p>
	5	<p>Problem Indicator En caso de detectarse problemas, los detalles se indicarán mediante los datos de diagnóstico de Alarm-CR. 1 = Regular operation 0 = Problem detected</p>
	6	No se usa
	7	No se usa

Tabla 57: Input CR (editar)

### 5.7.11.2 Output CR (propiedades)

Dentro de una relación de aplicación podrán establecerse varias relaciones de comunicación (CRs: Communication Relations).

Mediante la relación de comunicación Output CR, el controlador PROFINET IO transmite variables al dispositivo PROFINET IO.

Con la función de menú **Properties** del menú contextual de Output CR se abre el cuadro de diálogo **Properties**, que contiene los siguientes parámetros:

Elemento	Descripción								
Name	Cualquier nombre inequívoco para una CR de salida Default Output CR no es modificable								
Type	2 (no modificable)								
Send Clock Factor	Send Clock Factor determina el reloj de envío de la transmisión cíclica de una CR de entrada/salida. <b>Send Clock = Send Clock Factor x 31,25 µs</b> Rango de valores: 1...128 Valor por defecto: 32								
Reduction Factor	Para ajustar la frecuencia de transmisión. El factor de reducción permite reducir el tiempo real de ciclo del envío de datos de una CR de entrada/salida. El tiempo real de ciclo de los datos se calcula a partir de: <b>Send Cycle = Reduction Factor x Send Clock</b> Rango de valores: 1...16384 Valor por defecto: 32								
Watchdog Factor	Con Watchdog Factor se calcula el tiempo que, desde la perspectiva de un consumidor de CR de entrada/salida, se permite que transcurra como máximo entre la recepción de dos frames: <b>Watchdog Time = WatchdogFactor x Send Cycle</b> Rango de valores: 1...7680 Valor por defecto: 3								
VLAN-ID	Para asegurar la separación, a cada LAN virtual (VLAN) se le asignará un número único. Un dispositivo perteneciente a la VLAN con ID = 1 podrá comunicar con cualquier otro dispositivo de la misma VLAN, pero no con un dispositivo de otra VLAN con ID = 2, 3, ... Para el rango de valores véase también IEC-61158-6: <table style="margin-left: 20px;"> <tr> <td>0x000</td> <td>Sin VLAN</td> </tr> <tr> <td>0x001</td> <td>VLAN estándar</td> </tr> <tr> <td>0x002 hasta</td> <td>Véase IEEE 802.1 Q</td> </tr> <tr> <td>0xFFFF</td> <td></td> </tr> </table> Valor por defecto: 0	0x000	Sin VLAN	0x001	VLAN estándar	0x002 hasta	Véase IEEE 802.1 Q	0xFFFF	
0x000	Sin VLAN								
0x001	VLAN estándar								
0x002 hasta	Véase IEEE 802.1 Q								
0xFFFF									

Tabla 58: Output CR (propiedades)

## 5.8 Dispositivo PROFINET IO HIMA

En esta capítulo se describen las propiedades del dispositivo PROFINET IO HIMA, así como las funciones de menús y los cuadros de diálogo de SILworX necesarios para configurar el controlador PROFINET IO HIMA.

## 5.9 Sistema necesario

Equipamiento y sistema necesarios:

Elemento	Descripción
Sistema de control	HIMax con módulo COM HIMatrix L3
Módulo de CPU	Las interfaces Ethernet del módulo procesador no pueden usarse para PROFINET IO.
Módulo COM	Ethernet 10/100BaseT
Activación	Para habilitar se necesitará código de habilitación de software. Véase el capítulo 3.4.

Tabla 59: Equipamiento y sistema necesarios para controlador PROFINET IO

### Propiedades de dispositivo PROFINET IO:

Elemento	Descripción
Con función orientada a la seguridad	No
Velocidad de transmisión	100 Mbit/s Full Duplex
Vía de transmisión	Interfaces Ethernet de los módulos COM Interfaces Ethernet utilizadas aptas simultáneamente también para otros protocolos.
Categoría de conformidad	El dispositivo PROFINET IO cumple las exigencias de la categoría de conformidad A.
Real Time Class	Categoría RT 1
Máx. cantidad Dispositivos PROFINET IO	Para cada módulo COM podrá Ud. configurar un dispositivo PROFINET IO.
Máx. cantidad de relaciones de aplicación (AR) al controlador PROFINET IO	Un dispositivo PROFINET IO puede establecer un máximo de 5 relaciones de aplicación (AR) con un controlador PROFINET IO.
Máx. cantidad de relaciones de comunicación (CRs por cada AR)	Estándar: 1 entrada, 1 salida, 1 alarma
Máx. longitud de datos de proceso de todos los módulos PROFINET IO configurados	Salida: máx. 1440 bytes Entrada: máx. 1440 bytes
Priorización de los datos	Possible mediante el ajuste de <i>Reduction Rate</i> al nivel del dispositivo.
<b>Para PROFIsafe son válidas las siguientes propiedades</b>	
Máx. cantidad de F-Devices por módulo COM (HIMax y HIMatrix L3)	63
Máx. longitud de datos de proceso de una relación de comunicación (CR)	Salida: máx. 123 bytes de datos útiles + 5 bytes <sup>1)</sup> Entrada: máx. 123 bytes de datos útiles + 5 bytes <sup>1)</sup>
Máx. tamaño de los datos útiles HIMax: HIMatrix L3	1024 x 123 bytes = 125 952 bytes 512 x 123 bytes = 62 976 bytes
<sup>1)</sup> 5 bytes de datos de administración (bytes de control y estado y CRC)	

Tabla 60: Propiedades del controlador PROFINET IO

## 5.10 Ejemplo de PROFINET-IO/PROFIsafe

En el siguiente capítulo se describe la configuración del dispositivo PROFINET-IO/PROFIsafe HIMA.

### 5.10.1 Configuración del dispositivo PROFINET IO en SILworX

**Para crear un nuevo dispositivo PROFINET IO HIMA:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. En el menú contextual de los protocolos seleccione **New, PROFINET-IO Device** para agregar un nuevo dispositivo PROFINET IO.
3. En el menú contextual del dispositivo PROFINET IO seleccione **Properties**.
4. Escriba en el recuadro **Name** el nombre deseado para el dispositivo PROFINET IO.
5. **Seleccione COM Module.**

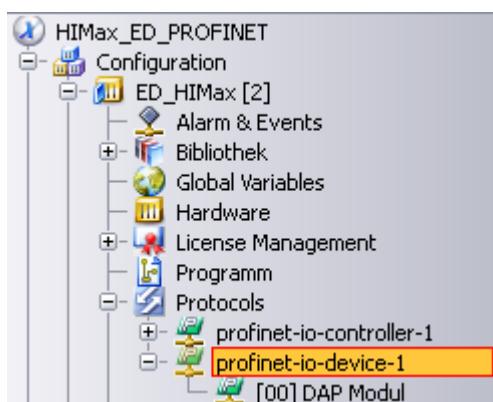


Fig. 35: Dispositivo PROFINET IO en la estructura de árbol de HIMax

**Para crear los módulos PROFINET IO necesarios:**

1. Abra **Configuration, Resource, Protocols, PROFINET-IO Device** en el árbol.
2. Seleccione en el menú contextual de PROFINET-IO Device **New**.
3. Seleccione los módulos adecuados para este ejemplo.

Módulo PROFINET-IO/PROFIsafe	Slot
"In 1 Byte"	1
"Safe Out 1 Byte"	2

**Para numerar los módulos de dispositivo PROFINET IO:**

1. Haga clic con el botón derecho en el primer **módulo de dispositivo PROFINET IO** y seleccione **Properties** en el menú contextual.
2. Escriba **1** en el recuadro **Slot**.
3. Repita este paso para los demás **módulos de dispositivo PROFINET IO** y numérelos consecutivamente.



**Numere** los módulos tal y como esté montado de hecho el dispositivo PROFINET IO.

**¡El siguiente paso es sólo necesario para los módulos PROFIsafe!**

4. Escriba la dirección del módulo PROFIsafe en el recuadro **PROFIsafe\_F\_Destination\_Address**.

### 5.10.1.1 Configuración del módulo de entrada del dispositivo PROFINET IO

- i** La suma de las variables, en bytes, deberá ser igual al tamaño del respectivo módulo, en bytes.

**Para configurar el módulo de entrada [01] "In 1 Byte":**

1. Seleccione en el dispositivo PROFINET IO-IO el módulo de entrada **[01] "In 1 Byte"**.
2. Haga clic con el botón derecho en **[01] In 1 Byte** y seleccione **Edit** en el menú contextual.
3. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.
4. Elija en la selección de objetos la variable adecuada y arrástrela con el ratón al área **Input Signals**.
5. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Inputs Signals**.
6. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.
7. En el cuadro de diálogo **Edit** seleccione la ficha **System Variables** y asigne a la variable de salida *Input Data Accepted by Controller* una variable global con el valor inicial TRUE, en caso de no desearse el control del estado de consumidor/proveedor.

- i** Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

### 5.10.1.2 Configuración del módulo de salida del dispositivo PROFIsafe

**Para configurar el módulo de salida [02] "Safe Out 1 Byte":**

1. Seleccione en el dispositivo PROFIsafe el módulo de salida **[02] "Safe Out 1 byte"**.
2. Haga clic con el botón derecho en **[02] Safe Out 1 Byte** y seleccione **Edit** en el menú contextual.
3. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.
4. Elija en la selección de objetos la variable adecuada y arrástrela con el ratón al área **Output Signals**.
5. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Signals**.
6. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.
7. En el cuadro de diálogo **Edit** seleccione la ficha **System Variables** y asigne a las variables de salida *Valid Output Variable* e *Input Data Accepted by Device* una variable global con el valor inicial TRUE, en caso de no desearse el control del estado de consumidor/proveedor.

- i** Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

### 5.10.1.3 Verificación de la configuración del dispositivo PROFINET IO

#### **Verificación de la configuración del dispositivo PROFINET IO:**

1. Seleccione **Configuration, Resource, Protocols, PROFINET-IO Device** en el árbol.
2. Haga clic en **Verification** en la barra de acciones y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.



La configuración del dispositivo PROFINET IO deberá Ud. recompilarla con el programa de usuario de recurso del dispositivo PROFINET IO y cargarla a los sistemas de control, para que sea efectiva para la comunicación de PROFINET IO.

---

## 5.11 Funciones del menú del dispositivo PROFINET IO

### 5.11.1 Función de menú Propiedades

Con la función de menú **Properties** del menú contextual del dispositivo PROFINET IO se abre el cuadro de diálogo **Properties**.

Elemento	Descripción	
Type	Dispositivo PROFINET IO	
Name	Cualquier nombre inequívoco para un dispositivo PROFINET IO	
Process Data Refresh Rate [ms]	Intervalo de actualización, en ms, para el intercambio de datos del protocolo entre COM y CPU. Si para el intervalo de actualización se ha elegido cero o un valor menor que el tiempo de ciclo del sistema de control, el intercambio de datos tendrá lugar lo más rápidamente posible. Rango de valores: 4...( $2^{31} - 1$ ) – Valor por defecto: 0	
Force Process Data Consistency	Activado: Desactivado:	Transferencia de todos los datos del protocolo desde CPU a COM dentro de un ciclo de la CPU. Transferencia de todos los datos del protocolo desde CPU a COM repartida a lo largo de varios ciclos de la CPU, con 1100 bytes por sentido de transmisión de datos. Con ello es posible reducir también el tiempo de ciclo. Valor por defecto: Activado
Module	Selección del módulo COM en el que se ejecuta este protocolo.	
Activate Max. $\mu$ P-Budget	Activado: Desactivado:	Aplicar el límite $\mu$ P-Budget tomado del recuadro <i>Max. <math>\mu</math>P-Budget in [%]</i> . No se usará límite de $\mu$ P-Budget para este protocolo.
Max. $\mu$ P-Budget in [%]	Máximo valor de $\mu$ P-Budget del módulo que se permite que tenga lugar durante la ejecución del protocolo. Rango de valores: 1...100% Valor por defecto: 30%	
RPC Port Server	Remote Procedure Call Port Rango de valores: 1024...65535 Valor por defecto: 49152 ¡RPC Port Server y RPC Port Client deberán ser diferentes!	
RPC Port Client	Remote Procedure Call Port Rango de valores: 1024...65535 Valor por defecto: 49153 ¡RPC Port Server y RPC Port Client deberán ser diferentes!	
AutoAwaitFParamsOn ConnLoss	<b>Este parámetro se usa sólo para módulos PROFIsafe.</b> Tras cada corte de la conexión al F-Host deberán cargarse de nuevo los F-Parameters al F-Device. Para hacer más fácil la puesta en servicio de PROFIsafe podrá activarse este parámetro. En tal caso, el F-Device aplicará automáticamente los F-Parameters necesarios al reiniciarse tras un corte de conexión. <b>Tras la puesta en servicio, este parámetro deberá desactivarse para un comportamiento del sistema conforme a PROFIsafe.</b> Activado: Desactivado:	
	F-Device adopta automáticamente el estado <i>Wait for F Parameters</i> . F-Device deberá ser puesto por el usuario en el estado <i>Wait for F Parameters</i> mediante un comando en línea. Valor por defecto: Desactivado	

Tabla 61: Propiedades generales del dispositivo PROFINET IO

### 5.11.2 Módulos PROFINET IO HIMA

En el dispositivo PROFINET IO HIMA se dispone de los siguientes módulos PROFINET IO.

Módulo PROFINET IO	Máx. tamaño de las variables de entrada	Máx. tamaño de las variables de salida
"In 1 Byte"	1 byte	
"In 2 Bytes"	2 bytes	
"In 4 Bytes"	4 bytes	
"In 8 Bytes"	8 bytes	
"In 16 Bytes"	16 bytes	
"In 32 Bytes"	32 bytes	
"In 64 Bytes"	64 bytes	
"In 128 Bytes"	128 bytes	
"In 256 Bytes"	256 bytes	
"In 512 Bytes"	512 bytes	
"In 1024 Bytes"	1024 bytes	
"In-Out 1 Byte"	1 byte	1 byte
"In-Out 2 Bytes"	2 bytes	2 bytes
"In-Out 4 Bytes"	4 bytes	4 bytes
"In-Out 8 Bytes"	8 bytes	8 bytes
"In-Out 16 Bytes"	16 bytes	16 bytes
"In-Out 32 Bytes"	32 bytes	32 bytes
"In-Out 64 Bytes"	64 bytes	64 bytes
"In-Out 128 Bytes"	128 bytes	128 bytes
"In-Out 256 Bytes"	256 bytes	256 bytes
"In-Out 512 Bytes"	512 bytes	512 bytes
"Out 1 Byte"		1 byte
"Out 2 Bytes"		2 bytes
"Out 4 Bytes"		4 bytes
"Out 8 Bytes"		8 bytes
"Out 16 Bytes"		16 bytes
"Out 32 Bytes"		32 bytes
"Out 64 Bytes"		64 bytes
"Out 128 Bytes"		128 bytes
"Out 256 Bytes"		256 bytes
"Out 512 Bytes"		512 bytes
"Out 1024 Bytes"		1024 bytes

Tabla 62: Módulos PROFINET IO

### 5.11.3 Módulos PROFIsafe HIMA

En el dispositivo PROFINET IO HIMA se dispone de los siguientes módulos PROFIsafe.

Módulo PROFIsafe	Máx. tamaño de las variables de entrada	Máx. tamaño de las variables de salida
“Safe In 1 Byte”	1 byte	
“Safe In 2 Bytes”	2 bytes	
“Safe In 4 Bytes”	4 bytes	
“Safe In 8 Bytes”	8 bytes	
“Safe In 16 Bytes”	16 bytes	
“Safe In 32 Bytes”	32 bytes	
“Safe In 64 Bytes”	64 bytes	
“Safe In 123 Bytes”	123 bytes	
“Safe In-Out 1 Byte”	1 byte	1 byte
“Safe In-Out 2 Bytes”	2 bytes	2 bytes
“Safe In-Out 4 Bytes”	4 bytes	4 bytes
“Safe In-Out 8 Bytes”	8 bytes	8 bytes
“Safe In-Out 16 Bytes”	16 bytes	16 bytes
“Safe In-Out 32 Bytes”	32 bytes	32 bytes
“Safe In-Out 64 Bytes”	64 bytes	64 bytes
“Safe In-Out 123 Bytes”	123 bytes	123 bytes
“Safe Out 1 Byte”		1 byte
“Safe Out 2 Bytes”		2 bytes
“Safe Out 4 Bytes”		4 bytes
“Safe Out 8 Bytes”		8 bytes
“Safe Out 16 Bytes”		16 bytes
“Safe Out 32 Bytes”		32 bytes
“Safe Out 64 Bytes”		64 bytes
“Safe Out 123 Bytes”		123 bytes

Tabla 63: Módulos PROFIsafe

#### Para crear un módulo PROFINet o PROFIsafe:

1. Abra **Configuration, Resource, Protocols, PROFINET-IO Device** en el árbol.
2. Seleccione en el menú contextual de PROFINET-IO Device **Insert Modules**.
3. Seleccione un módulo adecuado para transmitir los datos de proceso necesarios.
4. Haga clic con el botón derecho en el módulo seleccionado y elija **Edit**.
  - Introduzca las variables de entrada o salida en la ficha **Process Variables**.
  - En la ficha **System Variables** asigne a las variables de salida *Input Data Accepted by Device* y *Valid Output Variable* una variable global con el valor inicial TRUE, en caso de no desearse el control del estado de consumidor/proveedor.



Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

#### El siguiente ajuste es sólo necesario para los módulos PROFIsafe.

- En la ficha **Properties** escriba el *Slot* y la dirección *F\_Destination\_Address*.

### 5.11.4 Módulo PROFINET-IO/ PROFIsafe

Con los parámetros de los módulos se ajustan las relaciones de comunicación y la reacción de los módulos en caso de interrupciones de la conexión.

#### Propiedades

Con la función de menú **Properties** del menú contextual de los módulos se abre el cuadro de diálogo **Properties**. Este cuadro de diálogo contiene las siguientes fichas:

Elemento	Descripción
Name	Nombre del módulo de dispositivo
Slot	0 hasta 32767
Module ID	Número inequívoco
Sub-slot	Cantidad de sub-slots
Process data behavior	Valor que adoptan los datos de proceso tras un corte de conexión - Conservar los últimos datos de proceso válidos - Aplicar los valores iniciales
Length of IO Input Data	1...123
Length of IO Output Data	1...123
PROFIsafe F_Destination_Address	¡La dirección F_Destination_Address del F-Devices deberá ser inequívoca dentro de la red PROFIsafe! Rango de valores: 1 hasta 65534

Tabla 64: Propiedades generales del módulo de dispositivo

#### Edit

Con la función de menú **Edit** del menú contextual de los submódulos se abre el cuadro de diálogo **Edit**.

La ficha **System Variables** ofrece las siguientes variables de sistema, las cuales permiten evaluar el estado del submódulo en el programa del usuario.



Estas variables de sistema podrán usarse para el control del estado de consumidor/proveedor. Véase el capítulo 5.2.

Elemento	Descripción	
Valid Output Data	True	Datos de salida válidos GOOD
	False	Datos de salida no válidos BAD
Output Data Accepted by Device	True	Datos de salida válidos GOOD
	False	Datos de salida no válidos BAD
Valid Input Data	True	Datos de entrada válidos GOOD
	False	Datos de entrada no válidos BAD
Input Data Accepted by Device	True	Datos de entrada válidos GOOD
	False	Datos de entrada no válidos BAD

Elemento	Descripción
<b>Los siguientes parámetros se usan sólo para módulos PROFIsafe</b>	
PROFIsafe Control	<p>El byte de control PROFIsafe transmitido por el controlador puede leerse en el dispositivo. Véase también el capítulo 5.3.1.</p> <p>Bit 0 iPar_EN_DC La habilitación del controlador desbloquea el dispositivo para poder cargar nuevos “iParameters” al dispositivo.</p> <p>Bit 1 OA_Req_DC Operator Acknowledge del byte de control del host.</p> <p>Bit 2 Reset_Comm es el valor Reset_Comm leído del byte del control del host</p> <p>Bit 3 activate_FV_DC FALSE: Se intercambiarán valores de proceso entre F-Host y F-Device. TRUE: Se intercambiarán valores seguros “0” entre F-Host y F-Device.</p> <p>Bit 4 Toggle_d Bit Toggle del F-Device</p> <p>Bit 5 Cons_nr_R El número consecutivo se aplicará siempre que Toggle_d Bit presente una modificación entre dos bytes de control consecutivos, es decir, con independencia de si se produce un error o no.</p> <p>Bit 6 F_ParamValid TRUE: Parametrización F realizada FALSE: Otros</p> <p>Bit 7 F_Param_ConfiguredTwice TRUE: F-Device se ha parametrizado al menos dos veces con “F-Parameter” distintos FALSE: Otros</p>
PROFIsafe F_iPar_CRC	<p>iParameters son parámetros individuales o de la tecnología propia de F-Device. iPar_CRC es una resultante de la configuración de F-Device.</p> <p><b>i</b> Es responsabilidad del usuario ajustar el valor correcto de iPar_CRC tras aplicar los “iParameters” y cambiar luego al modo de producción real.</p>
PROFIsafe F_SIL	<p>0 SIL1 1 SIL2 2 SIL3 3 Sin SIL</p>
PROFIsafe RoundTrip Time last	PROFIsafe debe determinar en un F-Host el valor RoundTripTimeLast. Para un F-Host éste es el tiempo que transcurre entre el envío de una notificación de datos (con número consecutivo N) y la recepción del respectivo acuse de confirmación (Acknowledgment) (con número consecutivo N), medido en milisegundos.

Elemento	Descripción																
PROFIsafe Status	<p>PROFIsafe envía desde el dispositivo, con cada notificación, el byte de estado PROFIsafe al controlador, el cual podrá aplicarse en el programa de usuario del dispositivo.</p> <p>Véase también el capítulo 5.3.1.</p> <table> <tr> <td>Bit 0</td> <td>iPar_OK_DS Conservar nuevos “iParameters”.</td> </tr> <tr> <td>Bit 1</td> <td>Device_Fault_DS TRUE: Error de dispositivo FALSE: No hay errores de dispositivo Se considera sólo a partir del estado PROFIsafe 21 <i>Await Message</i>.</td> </tr> <tr> <td>Bit 2</td> <td>Reservado</td> </tr> <tr> <td>Bit 3</td> <td>Reservado</td> </tr> <tr> <td>Bit 4</td> <td>FV_activated_DS Valor seguro “Fail-safe” activado</td> </tr> <tr> <td>Bit 5</td> <td>Reservado</td> </tr> <tr> <td>Bit 6</td> <td></td> </tr> <tr> <td>Bit 7</td> <td>Reset_Comm El bloque de protocolo se pone en el estado inicial.</td> </tr> </table>	Bit 0	iPar_OK_DS Conservar nuevos “iParameters”.	Bit 1	Device_Fault_DS TRUE: Error de dispositivo FALSE: No hay errores de dispositivo Se considera sólo a partir del estado PROFIsafe 21 <i>Await Message</i> .	Bit 2	Reservado	Bit 3	Reservado	Bit 4	FV_activated_DS Valor seguro “Fail-safe” activado	Bit 5	Reservado	Bit 6		Bit 7	Reset_Comm El bloque de protocolo se pone en el estado inicial.
Bit 0	iPar_OK_DS Conservar nuevos “iParameters”.																
Bit 1	Device_Fault_DS TRUE: Error de dispositivo FALSE: No hay errores de dispositivo Se considera sólo a partir del estado PROFIsafe 21 <i>Await Message</i> .																
Bit 2	Reservado																
Bit 3	Reservado																
Bit 4	FV_activated_DS Valor seguro “Fail-safe” activado																
Bit 5	Reservado																
Bit 6																	
Bit 7	Reset_Comm El bloque de protocolo se pone en el estado inicial.																

Tabla 65: Cuadro de diálogo “Edit” de submódulos

En la ficha **Process Variables** se escriben las variables de entrada.

## 6 PROFIBUS DP

PROFIBUS DP es un estándar abierto e internacional de bus, de uso generalizado en aplicaciones que requieren de rápidos tiempos de reacción con volúmenes de datos especialmente pequeños.

El PROFIBUS DP Master de HIMA y el PROFIBUS DP Slave de HIMA cumplen los criterios de la norma europea EN 50170 [7] y de la norma mundial de validez general IEC 61158 para PROFIBUS DP.

El PROFIBUS DP Master de HIMA puede intercambiar datos con los PROFIBUS DP Slaves de forma cíclica y acíclica.

Para el intercambio de datos acíclico se dispone en SILworX de diversos bloques funcionales. Con estos bloques funcionales podrá Ud. adaptar el PROFIBUS DP Master de HIMA y los PROFIBUS DP Slaves óptimamente a las necesidades de su proyecto.

Sólo podrá obtenerse una conexión PROFIBUS DP redundante configurando un segundo PROFIBUS DP Master/Slave y adaptándolo en el programa de usuario.

- PROFIBUS DP Master (véase el capítulo 6.1)
- PROFIBUS DP Slave (véase el capítulo 6.13)

## 6.1 PROFIBUS DP Master de HIMA

En esta capítulo se describen las propiedades del PROFIBUS DP Master de HIMA, así como las funciones de menús y los cuadros de diálogo de SILworX necesarios para configurar el PROFIBUS DP Master de HIMA.

### Equipamiento y sistema necesarios:

Elemento	Descripción
Sistema de control HIMA	HIMax con módulo COM HIMatrix a partir de V7 de S.Op. de CPU y V12 de S.Op. de COM
Módulo COM	En la interfaz de bus de campo serie utilizada (FB1 ó FB2), el módulo COM deberá estar equipado con un submódulo PROFIBUS DP Master opcional de HIMA. Véase el capítulo 3.6.
Activación	Para la habilitación mediante el submódulo de bus de campo véase el capítulo 3.4.

Tabla 66: Equipamiento y sistema necesarios



Para HIMax, HIMA recomienda operar PROFIBUS DP mediante la interfaz de bus de campo FB1 (tasa de transmisión máxima de 12 Mbit). A través de la interfaz de bus de campo FB2 se permite una tasa de transmisión máxima de 1,5 Mbit.

### Propiedades de PROFIBUS DP Master:

Elemento	Descripción
Tipo de PROFIBUS DP Master de HIMA	DP-V1 Master Class 1 con funciones DP-V2 adicionales
Velocidad de transmisión	9,6 kbit/s...12 Mbit/s
Dirección de bus	0...125
Máx. cantidad de PROFIBUS DP Masters	Por cada HIMatrix F20, F30, F35, F60 o módulo COM HIMax podrá Ud. configurar dos PROFIBUS DP Masters. Por cada HIMatrix F20 sólo un PROFIBUS DP Master.
Máx. cantidad de PROFIBUS DP Slaves	Podrán configurarse hasta 122 Slaves por recurso (en todas las instancias de protocolo Master). Restricción: a un segmento de bus sin repetidor podrán conectarse un máximo de 31 Slaves.
Máx. longitud de datos de proceso de un Slave	DP-Output =: máx. 244 bytes DP-Input =: máx. 244 bytes

Tabla 67: Propiedades de PROFIBUS DP Master

Según la norma se permite un total de tres repetidores, lo que admite un máximo de 122 participantes de bus por cada interfaz serie de un Master.

### 6.1.1 Creación de un PROFIBUS DP Master de HIMA

#### Para crear un nuevo PROFIBUS DP Master de HIMA:

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Seleccione **New, PROFIBUS DP Master** en el menú contextual de los protocolos, para agregar un nuevo PROFIBUS DP Master.
3. En el menú contextual de PROFIBUS DP Master, seleccione **Properties, General**.
4. Seleccione **Module y Interfaces**.

## 6.2 Ejemplo de PROFIBUS DP

En este ejemplo tenemos un PROFIBUS DP Master de HIMA que intercambia variables con un PROFIBUS DP Slave de HIMA.

Aquí se explica cómo crear y parametrizar el PROFIBUS DP Master de HIMA y el PROFIBUS DP Slave de HIMA.

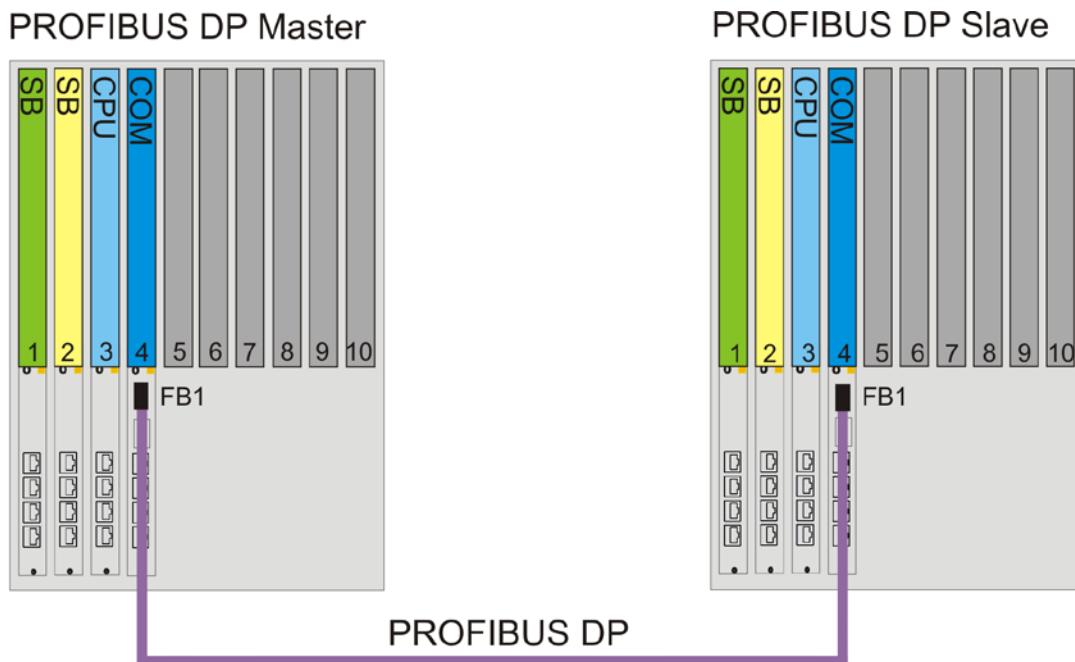


Fig. 36: Comunicación mediante PROFIBUS DP

Los módulos COM de ambos sistemas de control HIMax deberán estar equipados con el respectivo submódulo PROFIBUS DP en la interfaz de bus de campo 1. Véase el capítulo 3.6.

Para este ejemplo, deberá Ud. crear las siguientes variables globales en SILworX:

Variable global	Tipo
PB_Slave_Master1	UINT
PB_Slave_Master2	DWORD
PB_Slave_Master3	DWORD
PB_Slave_Master4	BYTE
PB_Master_Slave1	DWORD
PB_Master_Slave2	BYTE

### 6.2.1 Configuración de PROFIBUS DP Slave

Configuración del PROFIBUS DP Slave.

**Para crear un nuevo PROFIBUS DP Slave de HIMA:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Seleccione **New, PROFIBUS DP Slave** en el menú contextual de los protocolos, para agregar un nuevo PROFIBUS DP Slave.
3. Seleccione **Edit** en el menú contextual de PROFIBUS DP Slave.
4. En la ficha **Properties** seleccione los **módulos COM** y las **interfaces** (p. ej. FB1).

**Asignación de variables en el PROFIBUS DP Slave de HIMA:**

1. Seleccione **Edit** en el menú contextual de PROFIBUS DP Slave.
2. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.



La dirección inicial de las variables de E/S del PROFIBUS DP Slave de HIMA comienza siempre con 0. Si el PROFIBUS DP Master (de otro fabricante) espera un valor inicial mayor, deberán insertarse variables “postizas” (dummies) antes de las variables útiles.

**Salidas en el PROFIBUS DP Slave de HIMA**

Nombre	Tipo	Offset	Variable global
PB_Slave_Master1	UINT	0	PB_Slave_Master1
PB_Slave_Master2	DWORD	2	PB_Slave_Master2
PB_Slave_Master3	DWORD	6	PB_Slave_Master3
PB_Slave_Master4	BYTE	10	PB_Slave_Master4

Tabla 68: Salidas del PROFIBUS DP Slave de HIMA

1. En la selección de objetos, arrastre las variables globales a enviar con el ratón hasta el área **Output Variables**.



Las variables de salida del PROFIBUS DP Slave de HIMA de este ejemplo constan de **cuatro variables** con un total de **11 bytes**. La variable de salida con el menor offset tiene la dirección inicial **0**.

2. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Variables**.
3. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.

**Entradas en el PROFIBUS DP Slave de HIMA**

Nombre	Tipo	Offset	Variable global
PB_Master_Slave1	DWORD	14	PB_Master_Slave1
PB_Master_Slave2	BYTE	18	PB_Master_Slave2

Tabla 69: Entradas del PROFIBUS DP Slave de HIMA

1. Elija en la selección de objetos las variables globales a recibir y arrástrelas con el ratón al área **Input Variables**.



Las variables de entrada del PROFIBUS DP Slave de HIMA de este ejemplo constan de **dos variables** con un total de **5 bytes**. La variable de entrada con el menor offset tiene la dirección inicial **0**.

2. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Input Variables**.
3. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.

**Verificación de la configuración de PROFIBUS DP Slave:**

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Slave** en el árbol.
2. Haga clic en **Verification** en la barra de acciones y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.



La configuración del PROFIBUS DP Slave deberá Ud. recompilarla con el programa de usuario de recurso del PROFIBUS DP Slave y cargarla a los sistemas de control, para que sea efectiva para la comunicación de PROFIBUS DP.

### 6.2.2 Configuración de PROFIBUS DP Master

**Para crear un nuevo PROFIBUS DP Master de HIMA:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Seleccione **New, PROFIBUS DP Master** en el menú contextual de los protocolos, para agregar un nuevo PROFIBUS DP Master.
3. En el menú contextual de PROFIBUS DP Master, seleccione **Properties, General**.
4. En la ficha **General** seleccione los **módulos COM** y las **interfaces** (p. ej. FB1).



Los siguientes pasos sirven para configurar el PROFIBUS DP Slave de HIMax en el PROFIBUS DP Master de HIMax.

**Para crear un nuevo PROFIBUS DP Slave de HIMax en el PROFIBUS DP Master de HIMax:**

1. Seleccione **New, PROFIBUS DP Slave** en el menú contextual de PROFIBUS DP Master.

**Para leer el archivo GSD para el nuevo PROFIBUS DP Slave:**

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS Slave** en el árbol.
2. Seleccione **Read GSD File** en el menú contextual de PROFIBUS DP Master y elija el archivo GSD correspondiente al PROFIBUS Slave (p. ej. hax100ea.gsd).



Los archivos GSD para sistemas de control HIMax los encontrará en el sitio web de HIMA [www.hima.com](http://www.hima.com).

### 6.2.2.1 Creación de los módulos PROFIBUS DP de HIMax

En el PROFIBUS DP Master deberá Ud. configurar la cantidad de los bytes a transmitir realmente. Esto lo logrará seleccionando *módulos* hasta alcanzar la configuración física de los Slaves.

- 
- i** Es irrelevante la cantidad de módulos que se usen para alcanzar la cantidad necesaria de bytes, siempre y cuando no se exceda la cantidad máxima de 32 módulos.  
Para no dificultar innecesariamente la configuración del PROFIBUS DP Master, la cantidad elegida de módulos debería ser baja.
- 

#### Para crear los módulos PROFIBUS DP necesarios:

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS Slave** en el árbol.
2. En la barra de menús **PROFIBUS DP Master**, seleccione **Add modules**.
3. Seleccione los módulos adecuados para este ejemplo, al fin de recibir **11 bytes** del PROFIBUS DP Slave y enviar **3 bytes**.

#### Para numerar los módulos de PROFIBUS DP:

1. Haga clic con el botón derecho en el primer **módulo PROFIBUS DP** y seleccione **Properties** en el menú contextual.
2. Escriba **0** en el recuadro **Slot**.
3. Repita este paso para los demás módulos **PROFIBUS DP** y numérelos consecutivamente.

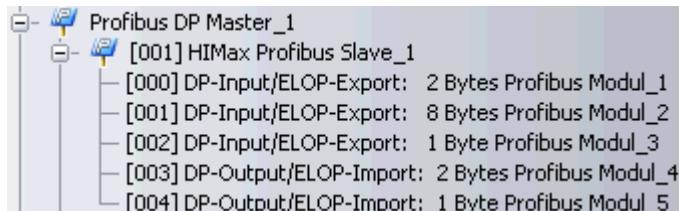


Fig. 37: PROFIBUS DP Slave HIMax con módulos

- 
- i** **Numere** los módulos PROFIBUS DP HIMax comenzando por 0 en orden ascendente y sin dejar huecos.

El orden de los módulos PROFIBUS DP es irrelevante de cara a la función. Sin embargo, para mayor claridad deberían crearse ordenados por módulos de entrada DP y módulos de salida DP.

### 6.2.2.2 Configuración de los módulos de entrada/salida



La suma de las variables, en bytes, deberá ser igual al tamaño del respectivo módulo, en bytes.

#### Para configurar el módulo de entrada [000] DP-Input/ELOP-Export: 2 Bytes:

1. Seleccione en PROFIBUS DP Slave el módulo de entrada **[000] DP-Input/ELOP-Export: 2 Bytes**.
2. Haga clic con el botón derecho en el módulo de entrada y seleccione **Edit** en el menú contextual.
3. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.
4. Elija en la selección de objetos la variable adecuada y arrástrela con el ratón al área **Input Signals** del módulo de entrada **[000] DP-Input/ELOP-Export: 2 Bytes**.

Nombre	Tipo	Offset	Variable global
PB_Slave_Master1	UINT	0	PB_Slave_Master1

Tabla 70: Variables del módulo de entrada [000] DP-Input/ELOP-Export: 2 bytes

5. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Inputs Signals**.
6. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.

#### Para configurar el módulo de entrada [001] DP-Input/ELOP-Export: 8 Bytes:

1. Seleccione en PROFIBUS DP Slave el módulo de entrada **[001] DP-Input/ELOP-Export: 8 bytes**.
2. Haga clic con el botón derecho en el módulo de entrada y seleccione **Edit** en el menú contextual.
3. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.
4. Elija en la selección de objetos la variable adecuada y arrástrela con el ratón al área **Inputs Signals** del módulo de entrada **[001] DP-Input/ELOP-Export: 8 bytes**.

Nombre	Tipo	Offset	Variable global
PB_Slave_Master2	DWORD	0	PB_Slave_Master2
PB_Slave_Master3	DWORD	4	PB_Slave_Master3

Tabla 71: Variables del módulo de entrada [001] DP-Input/ELOP-Export: 8 bytes

5. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Inputs Signals**.
6. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.

#### Para configurar el módulo de entrada [002] DP-Input/ELOP-Export: 1 byte:

1. Seleccione en PROFIBUS DP Slave el módulo de entrada **[002] DP-Input/ELOP-Export: 1 byte**.
2. Haga clic con el botón derecho en el módulo de entrada y seleccione **Edit** en el menú contextual.
3. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.
4. Elija en la selección de objetos la variable adecuada y arrástrela con el ratón al área **Inputs Signals** del módulo de entrada **[002] DP-Input/ELOP-Export: 1 byte**.

Nombre	Tipo	Offset	Variable global
PB_Slave_Master4	BYTE	0	PB_Slave_Master4

Tabla 72: Variables del módulo de entrada [002] DP-Input/ELOP-Export: 1 byte

5. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del **Inputs Signals**.
6. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.

**Para configurar el módulo de salida [003] DP-Output/ELOP-Import 2 Bytes:**

1. Seleccione en el PROFIBUS DP Slave el módulo de salida **[003] DP-Output/ELOP-Import 2 Bytes**.
2. Haga clic con el botón derecho en el módulo de salida y seleccione **Edit** en el menú contextual.
3. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.
4. Elija en la selección de objetos la variable adecuada y arrástrela con el ratón al área **Output Signals** del módulo de salida **[003] DP-Output/ELOP-Import 2 Bytes**.

Nombre	Tipo	Offset	Variable global
PB_Master_Slave1	UINT	0	PB_Master_Slave1

Tabla 73: Variables del módulo de salida [003] DP-Output/ELOP-Import: 2 bytes

5. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Signals**.
6. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.

**Para configurar el módulo de salida [004] DP-Output/ELOP-Import 1 Byte:**

1. Seleccione en el PROFIBUS DP Slave el módulo de salida **[004] DP-Output/ELOP-Import 1 Byte**.
2. Haga clic con el botón derecho en el módulo de salida y seleccione **Edit** en el menú contextual.
3. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.
4. Elija en la selección de objetos la variable adecuada y arrástrela con el ratón al área **Output Signals** del módulo de salida **[004] DP-Output/ELOP-Import 1 Byte**.

Nombre	Tipo	Offset	Variable global
PB_Master_Slave2	BYTE	0	PB_Master_Slave2

Tabla 74: Variables del módulo de salida [004] DP-Output/ELOP-Import: 1 byte

5. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Signals**.
6. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.

### 6.2.2.3 Creación de datos de usuario en el PROFIBUS DP Master

**Para crear datos de usuario en el PROFIBUS DP Master:**

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS DP Slave** en el árbol.
2. Haga clic con el botón derecho en **PROFIBUS DP Slave** y en el menú contextual seleccione **Properties**.
3. Seleccione la ficha **Data** y haga clic en el botón **Edit** junto a los datos de usuario.

En el recuadro de datos de usuario de 32 bytes de longitud se definen *la dirección inicial* y *la cantidad de variables de los bloques*. Véase también el capítulo 6.8.

4. Para este ejemplo deberán introducirse los siguientes datos de usuario:
  - 4, para que se reciban **cuatro variables** desde el PROFIBUS DP Master.
  - 2 para que se envíen **dos variables** desde el PROFIBUS DP Master.
 La dirección inicial del bloque de entrada y salida comienza por **0** respectivamente.

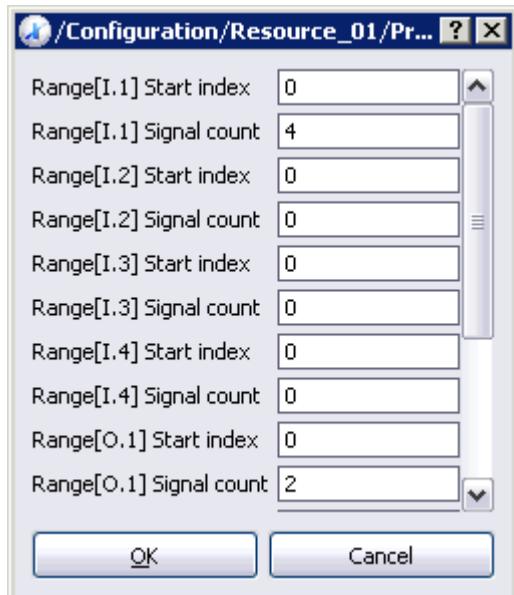


Fig. 38: Recuadro de datos de usuario

#### Verificación de la configuración de PROFIBUS DP Slave:

1. Abra Configuration, Resource, Protocols, PROFIBUS DP Master en el árbol.
2. Haga clic en **Verification** en la barra de acciones y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.

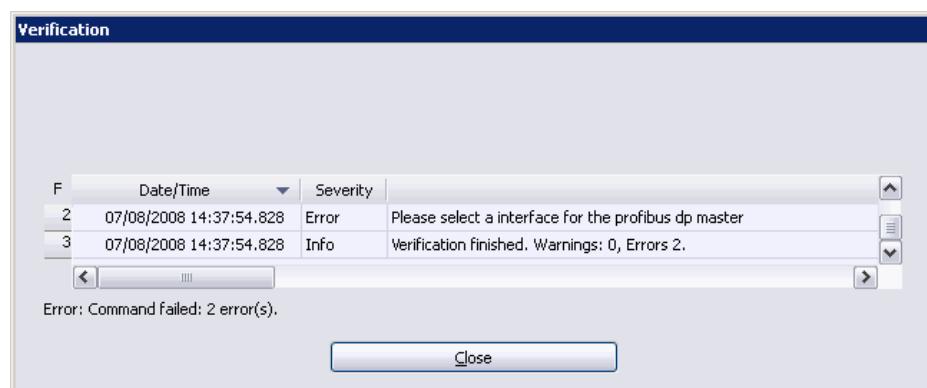


Fig. 39: Cuadro de diálogo “Verification”



La configuración del PROFIBUS DP Master deberá Ud. recompilarla con el programa de usuario de recurso del PROFIBUS DP Master y cargarla a los sistemas de control, para que sea efectiva para la comunicación de PROFIBUS DP.

### 6.2.2.4 Optimización de los parámetros de PROFIBUS DP

En principio, con los parámetros de PROFIBUS predeterminados por defecto debería tener lugar la comunicación PROFIBUS sin problemas. No obstante, optimizando los ajustes podrá Ud. obtener un intercambio de datos más rápido y una mejor detección de errores.

#### Para determinar el tiempo Target Rotation Time TTR [ms]

1. Abra el panel de control del PROFIBUS DP Master HIMax.
2. En la estructura de árbol del panel de control seleccione **PROFIBUS DP Master** y lea el tiempo **Target Rotation Time TTR [ms]** real. Anote ese valor.

#### Para determinar los parámetros necesarios de PROFIBUS DP Slave

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS DP Slave** en el árbol.
2. Haga clic con el botón derecho en **HIMax PROFIBUS Slave** y seleccione **Properties**.
3. Seleccione la ficha **Features** y lea los parámetros **Min. Slave Interval MSI [ms]** de ese PROFIBUS DP Slave. Anote ese valor.
4. Seleccione la ficha **Transfer Rate** y lea el parámetro **Max. Tsdr** de la velocidad de transmisión utilizada. Anote ese valor.

#### Para introducir los parámetros determinados:

1. Haga clic con el botón derecho en **PROFIBUS DP Master** y en el menú contextual **Properties**.
2. Seleccione la ficha **Timings**.

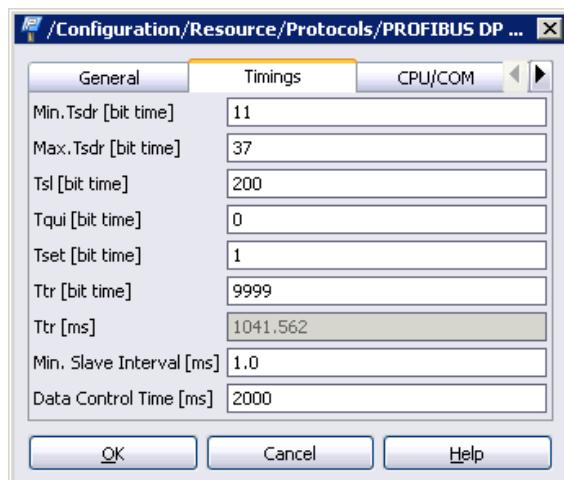


Fig. 40: Propiedades de PROFIBUS DP Master

3. Convierta el valor leído **Max. Tsdr** en el **bit Time** e intodúzcalo.
4. Convierta el valor leído **Target Rotation Time TTR [ms]** en el **bit Time** e intodúzcalo, sumándole 1/3 de margen de seguridad en el recuadro **Target Rotation Time TTR [ms]**.
5. Introduzca el valor leído **Min. Slave Interval MSI [ms]**.



Si hay varios Slaves configurados, deberá usarse el mayor valor de MaxTsdr [bit time] y el mayor valor de Min. Slave Intervall [ms].

6. El tiempo de monitoreo de datos útiles [ms] deberá ajustarse a un valor  $\geq 6 \cdot Ttr [ms]$ .

**Introducción del tiempo de WatchDog para PROFIBUS-DP Slave**

1. Abra Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS DP Slave en el árbol.
2. Haga clic con el botón derecho en HIMax PROFIBUS Slave y seleccione Properties.

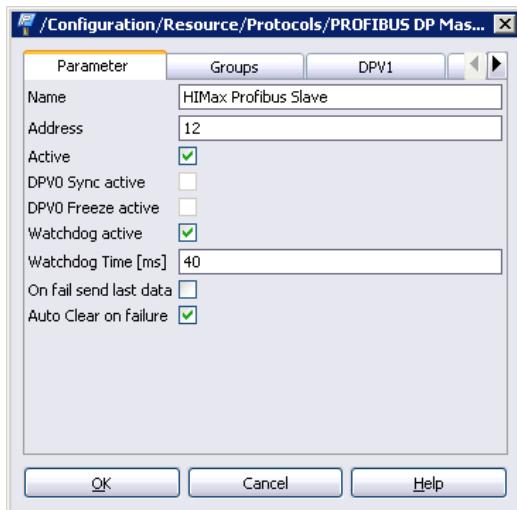


Fig. 41: Propiedades de PROFIBUS DP Slave

3. Seleccione la ficha **Parameters** y marque la casilla de verificación **Watchdog active**.
4. Introduzca un tiempo de WatchDog [ms]  $\geq 6 \cdot T_{tr}$  [ms] en el recuadro **Watchdog Time [ms]**.



La configuración de PROFIBUS DP Master y Slave deberá Ud. recompilarla con el programa de usuario de recurso del PROFIBUS DP Master y Slave y cargarla a los sistemas de control, para que sea efectiva para la comunicación de PROFIBUS DP.

## 6.3 Funciones de menú de PROFIBUS DP Master

### 6.3.1 Edit

Con la función de menú **Edit** del menú contextual de PROFIBUS DP Master se abre el cuadro de diálogo **Edit**.

La ficha **System Variables** ofrece las siguientes variables de sistema, las cuales permiten evaluar el estado del PROFIBUS DP Master en el programa del usuario.

Elemento	Descripción																
Error Counter	Cantidad de errores tras reinicializar la estadística.																
Baud Rate	Velocidad (bit/s) a la que opera el bus.																
Bus Error	<p>Si se produce un error de bus, en la variable de sistema <i>Bus Error</i> se aplicará un código de error. Un código de error tal permanecerá aplicado hasta subsanar el error de bus.</p> <table border="1"> <thead> <tr> <th>Código</th><th>Significado</th></tr> </thead> <tbody> <tr> <td>0</td><td>OK, sin errores de bus</td></tr> <tr> <td>1</td><td>Error de dirección: La dirección del Master ya existe en el bus.</td></tr> <tr> <td>2</td><td>Fallo del bus Fallo registrado en el bus (p. ej. bus no terminado correctamente, varios participantes transmitiendo al mismo tiempo).</td></tr> <tr> <td>3</td><td>Error de protocolo Paquete incorrectamente codificado recibido.</td></tr> <tr> <td>4</td><td>Error de hardware El hardware ha comunicado un error, p. ej. por haberse ajustado tiempos insuficientes.</td></tr> <tr> <td>5</td><td>Error desconocido El Master ha cambiado de estado por razones desconocidas</td></tr> <tr> <td>6</td><td>Reset de controlador En caso de graves fallos del bus, se reinicializará el chip del controlador.</td></tr> </tbody> </table> <p>Para evaluar la variable de estado <i>Bus Error</i> en el programa del usuario, ésta deberá vincularse a una variable.</p>	Código	Significado	0	OK, sin errores de bus	1	Error de dirección: La dirección del Master ya existe en el bus.	2	Fallo del bus Fallo registrado en el bus (p. ej. bus no terminado correctamente, varios participantes transmitiendo al mismo tiempo).	3	Error de protocolo Paquete incorrectamente codificado recibido.	4	Error de hardware El hardware ha comunicado un error, p. ej. por haberse ajustado tiempos insuficientes.	5	Error desconocido El Master ha cambiado de estado por razones desconocidas	6	Reset de controlador En caso de graves fallos del bus, se reinicializará el chip del controlador.
Código	Significado																
0	OK, sin errores de bus																
1	Error de dirección: La dirección del Master ya existe en el bus.																
2	Fallo del bus Fallo registrado en el bus (p. ej. bus no terminado correctamente, varios participantes transmitiendo al mismo tiempo).																
3	Error de protocolo Paquete incorrectamente codificado recibido.																
4	Error de hardware El hardware ha comunicado un error, p. ej. por haberse ajustado tiempos insuficientes.																
5	Error desconocido El Master ha cambiado de estado por razones desconocidas																
6	Reset de controlador En caso de graves fallos del bus, se reinicializará el chip del controlador.																
Average cycle time	Tiempo promedio medido del ciclo de bus, en milisegundos.																
Last cycle time	Tiempo medido del ciclo de bus, en milisegundos.																
Master State	<p>Indica el estado momentáneo del protocolo.</p> <p>0: OFFLINE 1: STOP 2: CLEAR 3: OPERATE</p> <p>Para evaluar la variable de estado <i>Master State</i> en el programa del usuario, éste deberá vincularse a una variable.</p>																
Maximum Cycle Time	Tiempo máximo medido del ciclo de bus, en milisegundos.																
Min. Slave Interval	Intervalo mínimo de Slave medido de un Slave asignado a este Master.																
Minimum Cycle Time	Tiempo mínimo medido del ciclo de bus, en milisegundos.																
Target Rotation Time	Tiempo de rotación planificado de Token.																

Tabla 75: Variables de sistema del PROFIBUS DP Master

### 6.3.2 Función de menú Propiedades

Con la función de menú **Properties** del menú contextual de PROFIBUS DP Master se abre el cuadro de diálogo **Properties**.

Este cuadro de diálogo contiene las siguientes fichas:

#### 6.3.2.1 Ficha “General”

Elemento	Descripción																																												
Type	PROFIBUS DP Master																																												
Name	Cualquier nombre inequívoco para un PROFIBUS DP Master																																												
Module	Selección del módulo COM en el que se ejecuta este protocolo.																																												
Max. µP Budget	Activado: Aplicar el límite µP-Budget tomado del recuadro <i>Max. µP-Budget in [%]</i> . Desactivado: No se usará límite de µP-Budget para este protocolo.																																												
Max. µP budget in [%]	Máximo valor de µP-Budget del módulo que se permite que tenga lugar durante la ejecución del protocolo.  Rango de valores: 1...100% Valor por defecto: 30%																																												
Behavior on CPU/COM Connection Loss	En caso de cortarse la conexión del módulo procesador al módulo de comunicación, se inicializarán las variables de entrada o se usarán sin variar en el módulo procesador, según lo que se defina en este parámetro (p. ej. si se extrae el módulo de comunicación con la comunicación en curso). <table> <tr> <td>Adopt initial data</td> <td>Las variables de entrada se reinicializarán a sus valores iniciales.</td> </tr> <tr> <td>Retain Last Value</td> <td>Las variables de entrada conservarán su último valor.</td> </tr> </table>	Adopt initial data	Las variables de entrada se reinicializarán a sus valores iniciales.	Retain Last Value	Las variables de entrada conservarán su último valor.																																								
Adopt initial data	Las variables de entrada se reinicializarán a sus valores iniciales.																																												
Retain Last Value	Las variables de entrada conservarán su último valor.																																												
Address	Dirección de estación del Master. Se permite que en el bus exista una sola dirección de estación del Master.  Rango de valores: 0...125 Valor por defecto: 0																																												
Interface	Interfaz COM que se desea usar para el Master. Rango de valores: FB1, FB2																																												
Baud Rate	Velocidad (bit/s) a la que opera el bus. Posibles valores: <table border="1"> <tr> <th>Valor</th> <th>Baud Rate</th> <th>FB1</th> <th>FB2</th> </tr> <tr> <td>9600</td> <td>9,6 kbit/s</td> <td>X</td> <td>X</td> </tr> <tr> <td>19200</td> <td>19,2 kbit/s</td> <td>X</td> <td>X</td> </tr> <tr> <td>45450</td> <td>45,45 kbit/s</td> <td>X</td> <td>X</td> </tr> <tr> <td>93750</td> <td>93,75 kbit/s</td> <td>X</td> <td>X</td> </tr> <tr> <td>187500</td> <td>187,5 kbit/s</td> <td>X</td> <td>X</td> </tr> <tr> <td>500000</td> <td>500 kbit/s</td> <td>X</td> <td>X</td> </tr> <tr> <td>1500000</td> <td>1,5 Mbit/s</td> <td>X</td> <td>X</td> </tr> <tr> <td>3000000</td> <td>3 Mbit/s</td> <td>X</td> <td>-</td> </tr> <tr> <td>6000000</td> <td>6 Mbit/s</td> <td>X</td> <td>-</td> </tr> <tr> <td>12000000</td> <td>12 Mbit/s</td> <td>X</td> <td>-</td> </tr> </table>	Valor	Baud Rate	FB1	FB2	9600	9,6 kbit/s	X	X	19200	19,2 kbit/s	X	X	45450	45,45 kbit/s	X	X	93750	93,75 kbit/s	X	X	187500	187,5 kbit/s	X	X	500000	500 kbit/s	X	X	1500000	1,5 Mbit/s	X	X	3000000	3 Mbit/s	X	-	6000000	6 Mbit/s	X	-	12000000	12 Mbit/s	X	-
Valor	Baud Rate	FB1	FB2																																										
9600	9,6 kbit/s	X	X																																										
19200	19,2 kbit/s	X	X																																										
45450	45,45 kbit/s	X	X																																										
93750	93,75 kbit/s	X	X																																										
187500	187,5 kbit/s	X	X																																										
500000	500 kbit/s	X	X																																										
1500000	1,5 Mbit/s	X	X																																										
3000000	3 Mbit/s	X	-																																										
6000000	6 Mbit/s	X	-																																										
12000000	12 Mbit/s	X	-																																										

Tabla 76: Propiedades generales de PROFIBUS DP Master

## 6.3.2.2 Ficha “Timings”

Elemento	Descripción
MinTsdr [bit time]	Min. Station Delay Time: Tiempo mínimo que un PROFIBUS DP Slave debe esperar antes de responder. Rango de valores: 11...1023 Valor por defecto: 11
MaxTsdr [bit time]	Max. Station Delay Time: Tiempo de que un PROFIBUS DP Slave dispone como máximo para responder. Max Tsdr $\geq$ Tsdr (del Slave conectado con el mayor valor de Tsdr) Los valores MaxTsdr de los Slaves se leen desde los archivos GSD y se encuentran en el cuadro de diálogo <b>Properties</b> de los Slaves en la ficha <b>Transfer Rate</b> . Rango de valores: 37...65535 Valor por defecto: 100
Tsl [bit time]	Slot Time Máximo período de tiempo que el Master espera a una respuesta del Slave. $Tsl > MaxTsdr + 2*Tset + Tqui + 13$ Rango de valores: 37...16383 Valor por defecto: 200
Tqui [bit time]	Quiet Time for Modulator (tiempo de pausa del modulador) Tiempo que se permite a un participante para conmutar de envío a recepción. Rango de valores: 0...493 Valor por defecto: 0
Tset [bit time]	Setup Time Tiempo de reacción frente a un evento Rango de valores: 1...494 Valor por defecto: 1
Ttr [bit time]	Tiempo planificado para una rotación de Token Tiempo máximo planificado de que se dispone para una rotación de Token. Una estimación baja de Ttr se obtiene por cálculo. Véase el capítulo 6.4.4. Rango de valores: 256...16777215 Valor por defecto: 9999
Ttr [ms]	Tiempo de rotación real de Token, en ms
Min. Slave Intervall [ms]	Tiempo que debe transcurrir como mínimo entre dos interrogaciones cíclicas de un Slave. El Master cumplirá con este tiempo y nunca se transgredirá por abajo. Es posible que el ciclo de PROFIBUS DP se prolongue si el modo isócrono está inactivo y la cantidad de telegramas acíclicos aumenta en un ciclo. El valor de <i>Min. Slave Interval</i> del Slave se lee del archivo GSD y se encuentra en el cuadro de diálogo <b>Properties</b> del Slave, en la ficha <b>Features</b> . En el modo isócrono, el valor <i>Min. Slave Interval</i> define el tiempo del ciclo isócrono. El modo isócrono se activará al activarse “Isochron Mode Sync” o “Freeze”. Véase también el tiempo de actualización entre CPU y COM (ficha CPU/COM). Rango de valores: 0...65535 (en pasos de 100 µs) Valor por defecto: 10 (= 1 ms)

Elemento	Descripción
User Data Monitoring Time [ms]	Período dentro del cual el Master deberá comunicar su estado momentáneo en el bus. Valor de referencia: User Data Monitoring Time = WDT del Slave Rango de valores: 0...65535 (en pasos de 10 ms) Valor por defecto: 200 (= 2000 ms)

Tabla 77: Ficha “Timings” en el cuadro de diálogo de propiedades de PROFIBUS DP Master

### 6.3.2.3 Ficha CPU/COM

Los valores predefinidos para los parámetros permiten un intercambio de datos de PROFIBUS DP lo más rápido posible entre el módulo COM (COM) y el módulo CPU (CPU) en el sistema de control HIMax. Estos parámetros solamente deberían modificarse cuando se necesite reducir la tasa de trabajo de CPU y COM y ello sea admisible para el proceso.

- i** Se recomienda que los parámetros los modifiquen sólo programadores experimentados. Un aumento del tiempo de actualización de COM y CPU implica también un aumento del tiempo real de actualización de los datos de PROFIBUS DP. Examine los requisitos de tiempo del equipo o la instalación.

Observe asimismo el parámetro *Min. Slave Interval [ms]*, el cual define el tiempo de actualización de los datos de PROFIBUS DP desde y al PROFIBUS DP Slave. Su valor podrá aumentarse en función del tiempo de actualización de COM/CPU.

Elemento	Descripción
Process Data Refresh Rate [ms]	Tiempo de actualización, en ms, para el intercambio de datos del protocolo entre COM y CPU. Si para el tiempo de actualización <i>Refresh Rate</i> se ha elegido cero o un valor menor que el tiempo de ciclo del sistema de control, el intercambio de datos tendrá lugar lo más rápidamente posible. Rango de valores: 0...(2 <sup>31</sup> - 1) Valor por defecto: 0
Force Process Data Consistency	Activado: Transferencia de todos los datos del protocolo desde CPU a COM dentro de un ciclo de la CPU.  Desactivado: Transferencia de todos los datos del protocolo desde CPU a COM repartida a lo largo de varios ciclos de la CPU, con 1100 bytes por sentido de transmisión de datos. Con ello es posible reducir también el tiempo de ciclo.  Valor por defecto: Activado

Tabla 78: Ficha “CPU/COM” en el cuadro de diálogo de propiedades de PROFIBUS DP Master

## 6.3.2.4 Ficha “Other”

Elemento	Descripción
Max. Number of Resends	Máxima cantidad de repeticiones de envío de un Master cuando un Slave no responde. Rango de valores: 0...7 Valor por defecto: 1
Highest Active Address	Highest Station Address (HSA) Mayor dirección de estación esperable de un Master. Los Masters con direcciones de estación más allá del valor HSA no se incluirán en la rotación del Token. Rango de valores: 0...125 Valor por defecto: 125
Isochron Mode Sync	Isochron Mode Sync permite una regulación síncrona de cadencia en Master y Slave, así como una activación simultánea de las salidas físicas de varios Slaves. Si “Isochron Mode Sync” está activo, el Master transmitirá el comando de control “Sync” como telegrama difundido a todos los Slaves. Tan pronto como los Slaves compatibles con el modo isócrono reciban el comando “Sync”, estos conmutarán los datos del programa de usuario simultáneamente a las salidas físicas. Los valores de las salidas físicas permanecerán congelados hasta el próximo comando “Sync”. El tiempo del ciclo viene definido por “Min. Slave Intervall”. Condición: Ttr < Min. Slave Intervall  Valor por defecto: Desactivado
Isochron Mode Freeze	“Isochron Mode Freeze” permite una adopción simultánea de los datos de entrada de varios Slaves. Si “Isochron Mode Freeze” está activo, el Master transmitirá el comando de control “Freeze” como telegrama difundido a todos los Slaves. Tan pronto como los Slaves compatibles con el modo isócrono reciban el comando “Freeze”, las variables de las entradas físicas quedarán congeladas en su valor momentáneo. Entonces el Master podrá leer los valores. Los datos de entrada no se actualizarán hasta un siguiente comando “Freeze”. El tiempo del ciclo viene definido por “Min. Slave Intervall”. Condición: Ttr < Min. Slave Intervall  Valor por defecto: Desactivado
Auto Clear on Error	El Master adoptará el estado CLEAR cuando deje de funcionar un Slave que tenga aplicado “Auto Clear on Error”.  Valor por defecto: Desactivado
Time Master	El Master es asimismo Master de tiempo y transmite periódicamente la hora del sistema a través del bus. Valor por defecto: Desactivado
Clock Sync Interval [ms]	Intervalo de sincronización de reloj. Período dentro del cual el Master de tiempo deberá transmitir la hora del sistema por el bus. Rango de valores: 0...65535 (en pasos de 10 ms) Valor por defecto: 0 = sin Time Master

Tabla 79: Otras propiedades de PROFIBUS DP Master

## 6.4 Procedimiento de acceso de PROFIBUS-DP

El procedimiento de acceso al bus ofrece a cada participante de bus una ventana definida de tiempo, dentro de la cual el participante deberá completar su tarea de comunicación.

### 6.4.1 Protocolo Master/Slave

La asignación de bus entre un PROFIBUS-DP Master y un PROFIBUS-DP Slave se asegura mediante el procedimiento Master/Slave.

Un PROFIBUS-DP Master activo comunica con PROFIBUS-DP Slaves pasivos.

El PROFIBUS-DP Master que tiene el Token tiene el derecho a transmitir y puede comunicar con los PROFIBUS-DP Slaves que tiene asignados. El Master asigna a un Slave el bus durante un tiempo determinado, dentro del cual el Slave deberá responder.

### 6.4.2 Protocolo de Token

La asignación de bus entre dispositivos de automatización (Master Cat. 1) y/o dispositivos programadores (Master Cat. 2) se asegura mediante "Token Passing".

Todos los PROFIBUS-DP Master conectados conjuntamente a un bus constituyen un círculo de rotación de Token. El PROFIBUS-DP Master activo que tenga el Token asumirá durante ese tiempo la función de Master en el bus.

Los PROFIBUS-DP Masters se ordenan en el círculo de rotación del Token-Ring por número ascendente de dirección de estación, pasándose el Token entre ellos en ese orden hasta el PROFIBUS-DP Master con la mayor dirección de estación.

Éste pasará el Token al Master con la menor dirección de estación para cerrar el círculo de rotación del Token.

El tiempo de rotación del Token es el tiempo de una rotación del Token a lo largo de todos los PROFIBUS-DP Masters. Ttr (Target Rotation Time) es el tiempo máximo permitido para una rotación del Token.

### 6.4.3 Tiempo de rotación permitido del Token (Ttr)

Valores de referencia para diversas velocidades de transmisión.

Al configurar el PROFIBUS DP Master observe que una parte de los parámetros de la ficha **Timings** dependerá de la velocidad de transmisión (Baud-Rate) elegida en la ficha **General**. Para la primera (inicial) configuración use los valores de referencia indicados en la siguiente tabla. En un paso ulterior se podrán modificar los valores.

	9,6 k	19,2 k	45,45 k	93,75 k	187,5 k	500 k	1,5 M	3 M	6 M	12 M
MinTsdr	11	11	11	11	11	11	11	11	11	11
MaxTsdr	60	60	400	60	60	100	150	250	450	800
Tsl bit time	100	100	640	100	100	200	300	400	600	1000
Tqui bit time	0	0	0	0	0	0	0	3	6	9
Tset bit time	1	1	95	1	1	1	1	4	8	16

Tabla 80: Valores de referencia para tiempo de rotación de Token con diversas velocidades de transmisión

Todos los tiempos se indican en Tbits (1Tbit = 1/[bit/s]).

MinTsdr tiene al menos 11 Tbit de longitud, ya que un carácter consta de 11 bits (1 bit de inicio, 1 bit de parada, 1 bit de paridad, 8 bits de datos).

Tiempo de transmisión para un carácter

Baud Rate	Tbit bit = 1 / Baud-Rate	Tiempo
9600 bit/s	$1 / 9600 = 104,166 \mu s$	$11 * 104,166 \mu s = 114,583 ms$
6 Mbit/s	$1 / 6 * 10^6 = 166,667 ns$	$11 * 166,667 ns = 1,833 \mu s$

Tabla 81: Tiempo de transmisión para un carácter con diversas velocidades de transmisión

#### 6.4.4 Cálculo del tiempo de rotación del Token Ttr

Calcule el mín. tiempo de rotación del Token Ttr como sigue:

$$Ttr_{min} = n * (198 + T1 + T2) + b * 11 + 242 + T1 + T2 + Tsl$$

Elemento	Significado
n	Cantidad de Slaves activos
b	Cantidad de bytes de datos de E/S de los Slaves activos (entrada y salida)
T0	$35 + 2 * Tset + Tqui$
T1	Si $T0 < MinTsdr$ : $T1 = MinTsdr$ Si $T0 > MinTsdr$ : $T1 = T0$
T2	Si $T0 < MaxTsdr$ : $T2 = MaxTsdr$ Si $T0 > MaxTsdr$ : $T2 = T0$
Tsl	Máximo período de tiempo de slot que el Master espera a una respuesta del Slave
198	Doble de la cabecera del telegrama con longitud variable (para requerimiento y respuesta)
242	Global_Control, FDL_Status_Req y Token-Passing

Tabla 82: Elementos de cálculo del tiempo de rotación del Token



Esta estimación del tiempo de rotación del Token *Ttr* será válida solamente cuando opere un solo Master en el bus, no deban repetirse envíos y no se transmitan datos acíclicos.

En ningún caso ajuste un valor de *Ttr* menor al calculado con la fórmula anterior, pues entonces no podrá garantizarse ya un funcionamiento exento de errores. HIMA recomienda ajustar el doble o el triple del valor calculado.

#### Ejemplo de cálculo del tiempo de rotación del Token Ttr

Dada la siguiente configuración:

5 Slaves activos

(n = 5)

20 bytes de datos de E/S por Slave

(b = 100)

Para una velocidad de transmisión de 6 Mbit/s se tomaron las siguientes constantes de tiempo en la Tabla 82:

- $MinTsdr = 11 T_{bit}$
- $MaxTsdr = 450 T_{bit}$
- $Tsl \text{ bit time} = 600 T_{bit}$
- $Tqui \text{ bit time} = 6 T_{bit}$
- $Tset \text{ bit time} = 8 T_{bit}$

$$T_0 = 35 + 2 * T_{set} + T_{qui}$$

$$T_0 = 35 + 2 * 8 + 6$$

$$\mathbf{T_0 = 57 \text{ T}_{bit}}$$

Puesto que  $T_0 > \text{MinTsdr}$ :  $T_1 = T_0 = 57 \text{ T}_{bit}$

Puesto que  $T_0 < \text{MaxTsdr}$ :  $T_2 = \text{MaxTsdr} = 450 \text{ T}_{bit}$

Use los valores obtenidos para la fórmula del mínimo tiempo de rotación del Token:

$$T_{tr_{min}} = n * (198 + T_1 + T_2) + b * 11 + 242 + T_1 + T_2 + T_{sl}$$

$$T_{tr_{min}} = 5 (198 + 57 + 450) + 100 * 11 + 242 + 57 + 450 + 600$$

$$T_{tr_{min}} [\text{T}_{bit}] = 5974 \text{ T}_{bit}$$

Resultado:

$$T_{tr_{min}} [\mu\text{s}] = 5974 \text{ T}_{bit} * 166,67 \text{ ns} = 995,68 \mu\text{s}$$



- El sistema comprueba el valor de  $T_{tr}$  al introducirlo en el cuadro de diálogo.
- Si el valor de  $T_{tr}$  introducido es menor que el valor calculado por SILworX, se generará un mensaje de error en el libro de registro. Se propondrá además un valor mínimo para  $T_{tr}$ .
- Si se han seleccionado *Isochron Mode Sync* o *Isochron Mode Freeze*, el tiempo del ciclo vendrá definido por el parámetro *MinSlaveInterval*. Entonces el valor de  $T_{tr}$  deberá ser en todo caso menor que el valor de *Minimum Slave Interval*.
- El incumplimiento de esta condición en el modo isócrono generará un mensaje de error.

**6.5****Ciclo isócrono PROFIBUS DP (a partir de V2 de DP)**

El ciclo PROFIBUS DP consta aquí de una parte fija cíclica y una parte de telegrama acíclica condicionada por eventos.

La parte acíclica del telegrama en un ciclo PROFIBUS DP podrá prolongarse en función de ello, lo que en determinadas aplicaciones (como, p. ej., la tecnología de actuadores) no es deseable.

Para obtener un tiempo de ciclo constante ( $t_{\text{const}}$ ), se activará el modo isócrono en el Master, en el cual es el parámetro *Min. Slave Interval [ms]* el que define el tiempo de ciclo constante ( $t_{\text{const}}$ ). El ciclo isócrono de PROFIBUS DP así parametrizado tiene una exactitud de cadencia con un margen de error < 10 ms.

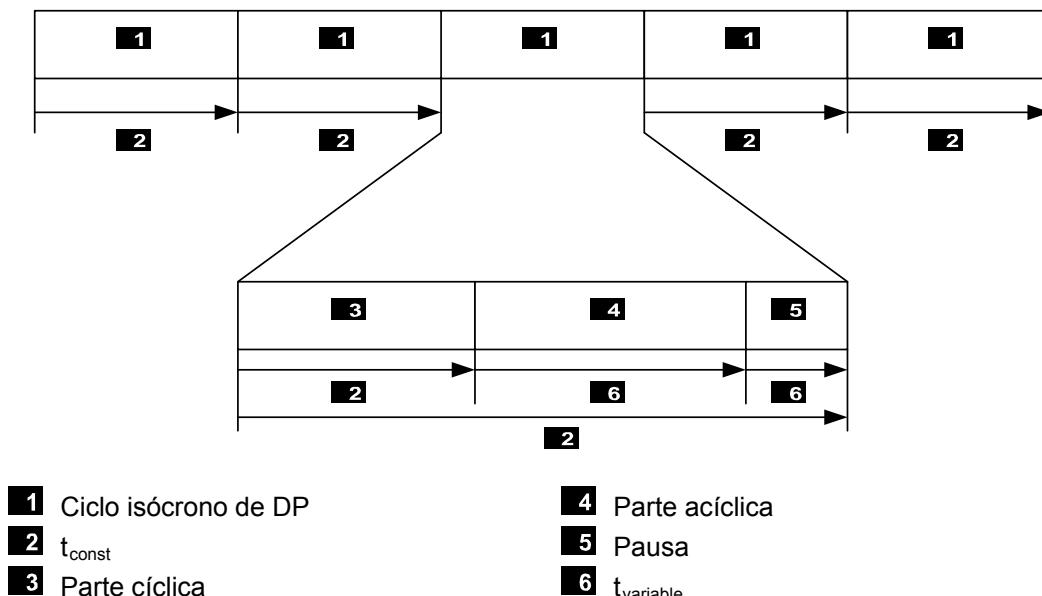


Fig. 42: Ciclo isócrono PROFIBUS DP

Para averiguar la parte cíclica, el usuario deberá calcular el mínimo tiempo de rotación del Token.

Además deberá reservarse un intervalo suficiente (por lo general el doble o el triple del mínimo tiempo de rotación del Token  $Ttr$ ) para la parte acíclica. Cuando no se necesite el tiempo de reserva se aplicará una pausa antes del siguiente ciclo, para mantener constante el tiempo del ciclo (véase también el capítulo 6.4.3, tiempo de rotación del Token  $Ttr$ ).



El Master se parametrizará mediante *Min. Slave Interval [ms]* con el tiempo de ciclo de DP averiguado.

Para que el modo isócrono sea efectivo, deberá activarse al menos uno de los dos parámetros “Isochronous Sync Mode” o “Isochronous Freeze Mode” en el Master.

Se permite que en el bus opere un solo Master en modo isócrono. No se admiten más Masters.

### 6.5.1 Isochron Mode (a partir de V2 de DP)

Esta función permite una regulación síncrona en Master y Slave, independientemente de la tasa de trabajo solicitada del bus. El ciclo de bus se sincronizará hasta un margen de error < 10 ms. Ello permite implementar operaciones de posicionamiento de alta precisión.



Las ventajas del *modo isócrono* podrán ser utilizadas con restricciones incluso por Slaves no compatibles (DP-V0-Slaves) con el *modo isócrono*. Para ello deberá Ud. activar *Slaves Sync* y/o *Freeze* y asignarlos al grupo 8.

Por lo general se usan simultáneamente los modos “Sync” y “Freeze”.

---

### 6.5.2 Isochron Mode Sync (a partir de V2 de DP)

*Isochron Mode Sync* permite una regulación síncrona de cadencia en Master y Slave, así como una activación simultánea de las salidas físicas de varios Slaves.

### 6.5.3 Isochrone Mode Freeze (a partir de V2 de DP)

*Isochron Mode Freeze* permite una adopción simultánea de los datos de entrada de varios Slaves.

## 6.6 Funciones de menú de PROFIBUS DP Slave (en el Master)

### 6.6.1 Creación de un PROFIBUS DP Slave (en el Master)

Para crear un nuevo PROFIBUS DP Master HIMA en un PROFIBUS DP Slave:

1. Abra Configuration, Resource, Protocols, Profibus DP Master en el árbol.
2. En el menú contextual de Profibus DP Master seleccione New, PROFIBUS Slave, para agregar un nuevo PROFIBUS Slave.

### 6.6.2 Edit

Con la función de menú **Edit** del menú contextual del PROFIBUS DP Slave se abre el cuadro de diálogo **System Variables**.

La ficha **System Variables** ofrece las siguientes variables de sistema, las cuales permiten evaluar el estado del PROFIBUS DP Slave en el programa del usuario.

Elemento	Descripción	
Activation control	Cambiando de 0 a 1 se desactiva el Slave. Cambiando de 1 a 0 se activa un Slave previamente desactivado. Activado = 0 Desactivado = 1	
PNO Ident Number	Número de 16 bits asignado por la Organización de Usuarios "PROFIBUS DP Nutzerorganisation e.V." y que identifica inequívocamente un producto (dispositivo de campo).	
Standard Diagnosis	Mediante el diagnóstico estándar, el Slave comunica su estado actual al Master. Esta variable contiene siempre el diagnóstico estándar recibido por último. Los parámetros corresponden al telegrama de diagnóstico según IEC 61158.	
Connection Count	El recuento se incrementa con cada nueva conexión. Se cuenta desde la puesta a cero del contador	
Connection State	Valor	Descripción
		0 Desactivado: Se cargan los lotes de parámetros para estos Slaves, pero los Slaves se ignoran completamente. Los valores de entrada se ponen a sus valores iniciales, en el bus no se registra ninguna actividad relativa a estos Slaves.
		1 Inactivo (no conectado): Si (ya) no es posible contactar con un Slave, los datos de entrada se pondrán a los valores iniciales. Para cada Slave podrán elegirse las siguientes opciones: <ul style="list-style-type: none"> <li>▪ El Master sigue enviando datos de salida al Slave o</li> <li>▪ El Master intenta volver a parametrizar el Slave.</li> </ul>
	2	Activo (conectado): Los Slaves intercambian datos de E/S con la CPU
Counter Slave Alarm	Cantidad de alarmas presentadas hasta el momento. Se cuenta desde la puesta a cero del contador.	
Standard Diagnosis Count	Cantidad de mensajes de diagnóstico presentados hasta el momento. Se cuenta desde la puesta a cero del contador.	

Tabla 83: Variables de sistema del PROFIBUS DP Slave

### 6.6.3 Propiedades

Con la función de menú **Properties** del menú contextual del PROFIBUS DP Slave se abre el cuadro de diálogo **Properties**. Este cuadro de diálogo contiene las siguientes fichas:

#### 6.6.3.1 Ficha “Parameters”

Elemento	Descripción
Name	Nombre del Slave
Adress	Dirección del Slave Rango de valores: 0...125 Valor por defecto: 0
Active	Estado del Slave Sólo un Slave activo podrá comunicar con un PROFIBUS DP Master. Valor por defecto: Activado
DPV0 Sync active	<i>Sync Mode</i> permite una activación simultánea de las salidas de varios Slaves DP-V0. <b>Atención:</b> En el caso de Slaves DP-V2 que funcionen en modo <i>Isochronous Sync Mode</i> , este recuadro deberá estar desactivado. Valor por defecto: Desactivado
DPV0 Freeze active	<i>Freeze Mode</i> permite una adopción simultánea de los datos de entrada de varios Slaves DP-V0. <b>Atención:</b> En el caso de Slaves DP-V2 que funcionen en modo <i>Isochronous Freeze Mode</i> , este recuadro deberá estar desactivado. Valor por defecto: Desactivado
Watchdog active	Si se selecciona esta opción, el Slave detectará si un Master deja de funcionar y adoptará en consecuencia el estado seguro. Valor por defecto: Desactivado
WatchDog Time [ms]	La casilla de verificación “Watchdog active” deberá estar activada. Si dentro de este periodo no se ha producido ningún intercambio de datos entre Master y Slave, el Slave se desactivará y pondrá todos los datos de salida DP-Output a su valor inicial. 0 = Desactivado Valor de referencia: WatchDog Time del Slave > 6 * Ttr Rango de valores: 0...65535 Valor por defecto: 0
On failure send last data	FALSE: En caso de error se cortará la conexión y volverá a establecerse. TRUE: En caso de error se siguen enviando datos, aun sin confirmación del Slave. Valor por defecto: Desactivado
Auto Clear on Failure	Si <i>Auto Clear on Failure</i> es TRUE en el Master y en este Slave, el Master pondrá todo el PROFIBUS DP en estado seguro en caso de que este Slave falle y deje de funcionar. Valor por defecto: Activado

Tabla 84: Ficha “Parameters” del PROFIBUS DP Slave

### 6.6.3.2 Ficha “Groups”

En esta ficha podrán agruparse los Slaves en distintos grupos. Los comandos de control globales Sync y Freeze podrán entonces dirigirse específicamente a uno o más grupos.

Elemento	Descripción	
Miembro del grupo 1	Miembro del grupo 1	Valor por defecto: Desactivado
Miembro del grupo 2	Miembro del grupo 2	
Miembro del grupo 3	Miembro del grupo 3	
Miembro del grupo 4	Miembro del grupo 4	
Miembro del grupo 5	Miembro del grupo 5	
Miembro del grupo 6	Miembro del grupo 6	
Miembro del grupo 7	Miembro del grupo 7	
Miembro del grupo 8	Miembro del grupo 8	

Tabla 85: Ficha “Groups” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave

### 6.6.3.3 Ficha DP-V1

En esta ficha se encuentran parámetros recién definidos a partir de V1 de DP. En el caso de los Slaves DP-V0, no podrá seleccionarse nada en este punto. Los parámetros compatibles con el Slave se reconocen en la columna “Supported”.

Elemento	Descripción
DP-V1	Si el modo DP-V1 no está activado, tampoco podrán usarse las demás funciones DP-V1. El Slave actuará entonces como un Slave DP-V0. Es posible que entonces deban modificarse también los datos de parametrización (véase el manual del Slave) Valor por defecto: Desactivado
Failsafe	Si se activa este modo, el Master en estado CLEAR no enviará ceros como datos de salida al Slave, sino un paquete de datos vacío (paquete “Failsafe”). El Slave reconoce en base a ello que ahora deberá aplicar los datos de salida seguros (no necesariamente todos ceros) en las salidas. Valor por defecto: Desactivado
Isochron Mode	Esta función permite una regulación síncrona en Master y Slave, independientemente de la tasa de trabajo solicitada del bus. El ciclo de bus se sincronizará hasta un margen de error < 1 ms. Esto permite implementar operaciones de posicionamiento de alta precisión. Valor por defecto: Desactivado
Publisher Active	Esta función se necesita para el tráfico de datos transversal del Slave. Esto permite la comunicación directa (y por ende más rápida) entre los Slaves mediante difusión (broadcast) sin rodeos a través del Master. Valor por defecto: Desactivado
Prm Block Struct. Supp.	El Slave es compatible con datos parametrizados estructurados (sólo lectura). Valor por defecto: Desactivado
Check. Cfg.-Mode	Control reducido de configuración si está activado “Check Cfg.-Mode”, pudiendo operar el Slave sin la configuración completa. Esta casilla debería estar desactivada para la puesta en servicio. Valor por defecto: Desactivado

Tabla 86: Ficha DP-V1 en el cuadro de diálogo de propiedades de PROFIBUS DP Slave

#### 6.6.3.4 Ficha “Alarms”

En esta página podrán activarse alarmas. Sin embargo, sólo en el caso de Slaves DP-V1, estando activado DP-V1 y siempre que el Slave sea compatible con la alarma. Las alarmas compatibles se reconocen por la marca en la casilla de la columna “**Supported**”. Si una alarma es obligatoria, ello se reconoce en la columna “**Required**”.

Elemento	Descripción	
Update Alarm	Alarma en caso de modificar los parámetros de un módulo.	Valor por defecto: Desactivado
Status Alarm	Alarma en caso de modificarse el estado de un módulo.	
Vendor Alarm	Alarma específica del fabricante.	
Diagnostic Alarm	Alarma en caso de ciertos eventos tales como cortocircuito, exceso de temperatura, etc. en un módulo.	
Process Alarm	Alarma en caso de eventos importantes en el proceso.	
Pull & Plug Alarm	Alarma en caso de desenchufar o enchufar un módulo.	

Tabla 87: Ficha “Alarms” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave

#### 6.6.3.5 Ficha “Data”

Esta ficha contiene información sobre las longitudes de datos compatibles y los datos del usuario (datos de parametrización avanzada).

Elemento	Descripción
Max. Input Length	Longitud máxima de los datos de entrada.
Max. Output Length	Longitud máxima de los datos de salida.
Max. Data Length	Longitud total máxima de los datos de entrada y salida.
User Data Length	Longitud de los datos del usuario.
User Data	Datos de parametrización. Es recomendable no editar aquí. Es más cómodo usar el cuadro de diálogo <i>User Parameters</i> , capítulo 6.8.
Max. Diag. Data Length	Longitud máxima de los datos de diagnóstico que el Slave envía.

Tabla 88: Ficha “Data” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave

### 6.6.3.6 Ficha “Model”

Esta página contiene diversa información que se explica por sí misma:

Elemento	Descripción
Model	Denominación dada por el fabricante al PROFIBUS DP Slave
Manufacturer	Fabricante del dispositivo de campo
Ident. Number	ID del Slave definido por la Organización de Usuarios de PROFIBUS (PNO)
Revision	Número de revisión del PROFIBUS DP Slave
Hardware Release	Número de edición del PROFIBUS DP Slave
Software Release	Número de edición del software del PROFIBUS DP Slave
GSD file name	Nombre del archivo GSD
Info Text	Información complementaria acerca del PROFIBUS DP Slave

Tabla 89: Ficha “Model” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave

### 6.6.3.7 Ficha “Features”

Elemento	Descripción
Modular Station	TRUE: Estación modular FALSE: Estación compacta
First slot number	La numeración de los módulos (slots) deberá comenzar con este numero y ser consecutiva.
Max Modules	Cantidad máxima de módulos que puede albergar una estación modular.
Support for “Set Slave Add”	Slave compatible con la asignación dinámica de direcciones.
Min. Slave Intervall [ms]	Tiempo que debe transcurrir como mínimo entre dos invocaciones cíclicas a un Slave.
Diag. Update	Cantidad de ciclos de Polling hasta que el diagnóstico del Slave refleja el estado actual.
Support for WDBase1ms	Slave compatible con 1 ms como base de tiempo para WatchDog.
Support for DP V0 Sync	Slave compatible con DP-V0 Sync
Support for DP V0 Freeze	Slave compatible con DP-V0 Freeze
DP V1 Data Types	Slave compatible con tipos de datos DP-V1.
Extra Alarm SAP	Slave compatible con SAP 50 para confirmación de alarmas.
Alarm Seq. Mode Count	Especifica cuántas alarmas puede procesar simultáneamente el Slave. Cero significa una alarma de cada tipo.

Tabla 90: Ficha “Features” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave

### 6.6.3.8 Ficha “Baud Rates”

En esta ficha se encuentran las velocidades de transmisión compatibles con el Slave, así como el respectivo valor *MaxTsdr*.

*MaxTsdr* es el tiempo dentro del cual deberá responder como máximo el Slave a un requerimiento del Master. El rango de valores dependerá del Slave y de la velocidad de transmisión y se halla entre 15 y 800 Tbit.

Elemento	Descripción
9,6 k	MaxTsdr = 60
19,2 k	MaxTsdr = 60
31,25 k	No compatible
45,45 k	MaxTsdr = 60
93,75 k	MaxTsdr = 60
187,5 k	MaxTsdr = 60
500 k	MaxTsdr = 70
1,5 M	MaxTsdr = 75
3 M	MaxTsdr = 90
6 M	MaxTsdr = 100
12 M	MaxTsdr = 120

Tabla 91: Ficha “Baud Rates” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave

### 6.6.3.9 Ficha “Acyclic”

Esta ficha contiene algunos parámetros para la transmisión de datos acíclica:

Elemento	Descripción
Support for C1 Read/Write	Slave compatible con la transmisión de datos acíclica.
C1 Read/Write required	Transmisión de datos acíclica obligatoria para el Slave.
C1 Max Data Length [Byte]	Longitud máxima de un paquete de datos acíclico.
C1 Response Timeout [ms]	Time-out para transmisión de datos acíclica.

Tabla 92: Ficha “Acyclic” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave

## 6.7 Importación de archivo GSD

El archivo GSD contiene los datos de parametrización del PROFIBUS DP Slave.

**Para leer el archivo GSD para el nuevo PROFIBUS DP Slave:**

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS Slave** en el árbol.
2. Seleccione **Read GSD File** en el menú contextual de PROFIBUS DP Master y elija el archivo GSD correspondiente al PROFIBUS Slave (p. ej. hax100ea.gsd).



Los archivos GSD para sistemas de control HIMax/HIMatrix los encontrará en el sitio web de HIMA [www.hima.com](http://www.hima.com).

La exactitud del archivo GSD es responsabilidad del fabricante del respectivo dispositivo de campo.

Los archivos GSD para HIMax (hax100ea.gsd) y HIMatrix (hix100ea.gsd) contienen los siguientes módulos:

Módulos de entrada de PROFIBUS DP Master	Tipo	Cantidad
DP-Input/ELOP-Export	Byte	1
DP-Input/ELOP-Export	Bytes	2
DP-Input/ELOP-Export	Bytes	4
DP-Input/ELOP-Export	Bytes	8
DP-Input/ELOP-Export	Bytes	16
DP-Input/ELOP-Export	Word	1
DP-Input/ELOP-Export	Words	2
DP-Input/ELOP-Export	Words	4
DP-Input/ELOP-Export	Words	8
DP-Input/ELOP-Export	Words	16
Módulos de salida de PROFIBUS DP Master	Tipo	Cantidad
DP-Output/ELOP-Import	Byte	1
DP-Output/ELOP-Import	Bytes	2
DP-Output/ELOP-Import	Bytes	4
DP-Output/ELOP-Import	Bytes	8
DP-Output/ELOP-Import	Bytes	16
DP-Output/ELOP-Import	Word	1
DP-Output/ELOP-Import	Words	2
DP-Output/ELOP-Import	Words	4
DP-Output/ELOP-Import	Words	8
DP-Output/ELOP-Import	Words	16

Tabla 93: Archivo GSD del PROFIBUS DP Slave HIMax/HIMatrix

## 6.8 Edición de los parámetros del usuario

En el recuadro de datos de usuario se definen la **dirección inicial** y la **cantidad de variables** de los bloques.

Además, en el PROFIBUS DP Master deberá Ud. configurar la cantidad de los bytes a transmitir realmente. Esto se realiza seleccionando los módulos PROFIBUS DP desde el archivo GSD del PROFIBUS DP Slave (véase también el capítulo 6.2.2).

**Para abrir el cuadro de diálogo "User Parameters":**

1. Abra **Configuration/Resource/Protocols PROFIBUS DP Master** en el árbol.
2. Haga clic con el botón derecho en **PROFIBUS Slave** y seleccione **Properties**.
3. Haga clic en la ficha **Data** y seleccione el **botón...** junto a los datos del usuario.



La composición del cuadro de diálogo **Edit User Parameters** viene definida por el archivo GSD del Slave.

### Composición básica del recuadro de datos de usuario de 32 bytes:

La composición del recuadro de datos de usuario de 32 bytes es la siguiente:

Los 32 bytes están en **ocho grupos**, con **cuatro bytes por grupo**.

Los bloques 1...4 definen cuáles y qué cantidad de variables recibe el PROFIBUS DP Master desde el PROFIBUS DP Slave.

Los bloques 5...8 definen cuáles y qué cantidad de variables envía el PROFIBUS DP Master al PROFIBUS DP Slave.

Los dos primeros bytes de cada bloque especifican la dirección inicial para la primera variable a leer o escribir de un bloque.

Los dos últimos bytes de cada bloque especifican la cantidad de variables que se desea enviar o recibir.

### Configuración de los datos del usuario en diversos bloques:

Normalmente no es necesario repartir las variables (datos del usuario) en diversos bloques. Basta con definir cada primer bloque de variables de entrada y de salida y leer o escribir los datos *en bloque*.

Sin embargo, en aplicaciones donde es necesario leer y escribir solamente variables seleccionadas, podrán definirse hasta cuatro bloques de variables de entrada y de salida respectivamente.

### Ejemplo

PROFIBUS DP Master envía y recibe las siguientes variables desde PROFIBUS DP Slave:

1er bloque: **4** variables de entrada a partir de la dirección inicial **0**.

2º bloque: **6** variables de entrada a partir de la dirección inicial **50**.

4º bloque :**9** variables de entrada a partir de la dirección inicial **100**.

5º bloque: **2** variables de salida a partir de la dirección inicial **10**.

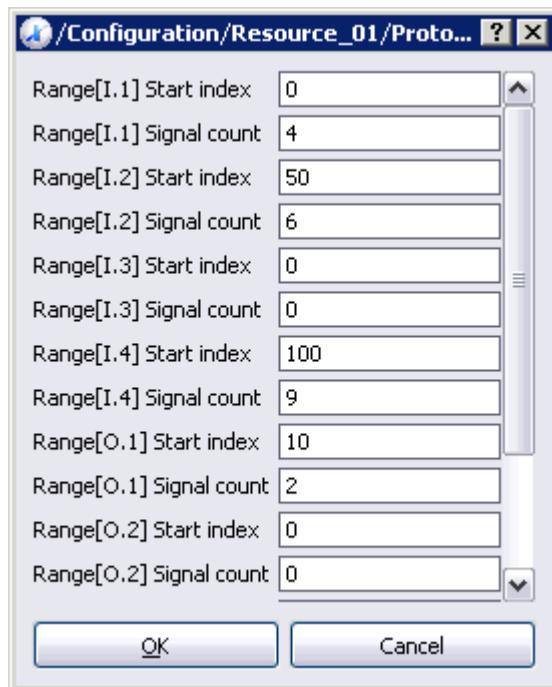
**Configuración de datos de usuario en el PROFIBUS DP Master:**

Master Import/Slave Export	Dirección inicial	Cantidad de variables
1er bloque (byte 0...3)	0,0	0,4
2er bloque (byte 4...7)	0,50	0,6
3er bloque (byte 8...11)	0,0	0,0
4er bloque (byte 12...15)	0,100	0,9

Tabla 94: Ejemplo: Bloques 1...4 del recuadro de datos de usuario

Master Export/Slave Import	Dirección inicial	Cantidad de variables
5er bloque (byte 16...19)	0,10	0,2
6er bloque (byte 20...23)	0,0	0,0
7er bloque (byte 24...27)	0,0	0,0
8er bloque (byte 28...31)	0,0	0,0

Tabla 95: Ejemplo: Bloques 1...4 del recuadro de datos de usuario

Cuadro de diálogo *Edit User Parameters* de un PROFIBUS DP Slave HIMatrix o HIMax.Fig. 43: Cuadro de diálogo *Edit User Parameters*

## 6.9 Bloques funcionales de PROFIBUS

Con los bloques funcionales de PROFIBUS podrá Ud. adaptar óptimamente a las necesidades de su proyecto el PROFIBUS DP Master de HIMA y los PROFIBUS DP Slaves asignados a éste.

Los bloques funcionales los parametrizará Ud. en el programa de usuario de forma tal que las funciones del Master y de los Slaves (alarmas, datos de diagnóstico, estados) puedan leerse y aplicarse en el programa de usuario.



Los bloques funcionales son necesarios sólo para aplicaciones especiales. ¡Para el tráfico normal de datos cíclico entre Master y Slave no se necesitan estos bloques funcionales! Para la configuración básica de los bloques funcionales de PROFIBUS DP véase el capítulo 13.1.

Se dispone de los siguientes bloques funcionales:

Bloque funcional	Descripción de la función	Apto a partir del nivel de DP
MSTAT 6.9.1	Control del estado del Master mediante el programa de usuario	DP-V0
RALRM 6.9.2	Lectura de los mensajes de alarma de los Slaves	DP-V1
RDIAG 6.9.3	Lectura de los mensajes de diagnóstico de los Slaves	DP-V0
RDREC 6.9.4	Lectura de los registros de datos acíclicos de los Slaves	DP-V1
SLACT 6.9.5	Control del estado de los Slaves mediante el programa de usuario	DP-V0
WRREC 6.9.6	Escriftura de los registros de datos acíclicos de los Slaves	DP-V1

Tabla 96: Sinopsis de los bloques funcionales de PROFIBUS DP



Los PROFIBUS DP Master de HIMA funcionan con el nivel DP-V1.  
Los PROFIBUS DP Slaves de HIMA funcionan con el nivel DP-V0.  
Observe que, por tanto, no todos los bloques funcionales del HIMA PROFIBUS DP Master podrán gobernar un PROFIBUS DP Slave de HIMA.

### 6.9.1 Bloque funcional MSTAT

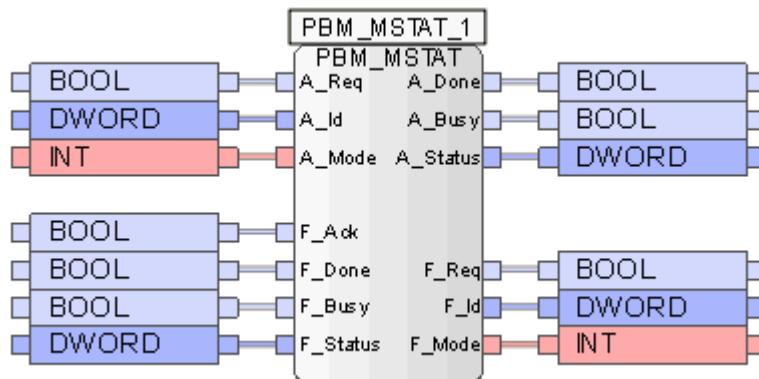


Fig. 44: Bloque funcional MSTAT

Con el bloque funcional **MSTAT** (a partir de V0 de DP) el PROFIBUS DP Master podrá ser gobernado por el programa del usuario. Así, el PROFIBUS DP Master podrá aplicarse a una entrada física mediante un interruptor mecánico o ponerse en uno de los siguientes estados mediante un temporizador:

- 0: OFFLINE
- 1: STOP
- 2: CLEAR
- 3: OPERATE

- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo “A” quiere decir “Application”.

Entradas A	Descripción	Tipo
A_Req	El flanco positivo da inicio al bloque	BOOL
A_ID	ID de Master (no se usa)	DWORD
A_Mode	El PROFIBUS DP Master podrá ponerse en los siguientes estados 0: OFFLINE 1: STOP 2: CLEAR 3: OPERATE	INT

Tabla 97: Entradas A del bloque funcional MSTAT

Salidas A	Descripción	Tipo
A_Done	TRUE: El PROFIBUS DP Master se ha puesto en el estado definido en la entrada A_Mode.	BOOL
A_Busy	TRUE: No ha finalizado el cambio de estado del PROFIBUS DP Master.	BOOL
A_Status	Código de error o estado (véase el capítulo 6.11)	DWORD

Tabla 98: Salidas A del bloque funcional MSTAT

### Entradas y salidas del bloque funcional con el prefijo F:

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional MSTAT en la estructura en árbol. El prefijo "F" quiere decir "Field".



La conexión del bloque funcional MSTAT en el árbol (en la carpeta de bloques funcionales) al bloque funcional MSTAT (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá Ud. crearlas previamente en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **MSTAT** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **MSTAT** en el árbol.

F-Inputs	Tipo
F_ACK	BOOL
F_DONE	BOOL
F_BUSY	BOOL
F_STATUS	DWORD

Tabla 99: Entradas F del bloque funcional MSTAT

Conecte las *F-Outputs* del bloque funcional **MSTAT** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **MSTAT** en el árbol.

F-Outputs	Tipo
F_REQ	BOOL
F_ID	DWORD
F_MODE	INT

Tabla 100: Salidas F del bloque funcional MSTAT

### Para crear el bloque funcional MSTAT en la estructura de árbol:

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **MSTAT**.
3. Haga clic con el botón derecho en el bloque funcional **MSTAT** y seleccione **Edit**.
  - Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **MSTAT** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **MSTAT** en el programa de usuario.

Entradas	Tipo
M_ID	DWORD
MODE	INT
REQ	BOOL

Tabla 101: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **MSTAT** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **MSTAT** en el programa de usuario.

Salidas	Tipo
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Tabla 102: Variables de salida del sistema

**Para hacer uso del bloque funcional MSTAT:**

1. Ponga en el programa de usuario la entrada *A\_Mode* en el estado deseado.  
Si no se aplica *A\_Mode*, tras el paso 2 se generará un código de error en la salida *A\_Status* y el PROFIBUS DP Master no se aplicará.
2. Ponga en el programa de usuario la entrada *A\_Req* en TRUE.



*El bloque funcional reacciona a un cambio de flanco positivo en A\_Req.*

- La salida *A\_Busy* es TRUE hasta que está ejecutado el comando MSTAT. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Done* a TRUE.



Si no es posible aplicar el modo predefinido, se generará un código de error en la salida *A\_Status*.

El modo actual del Master puede verse en la variable Master-Status (véase el capítulo 6.10).

### 6.9.2 Bloque funcional RALRM

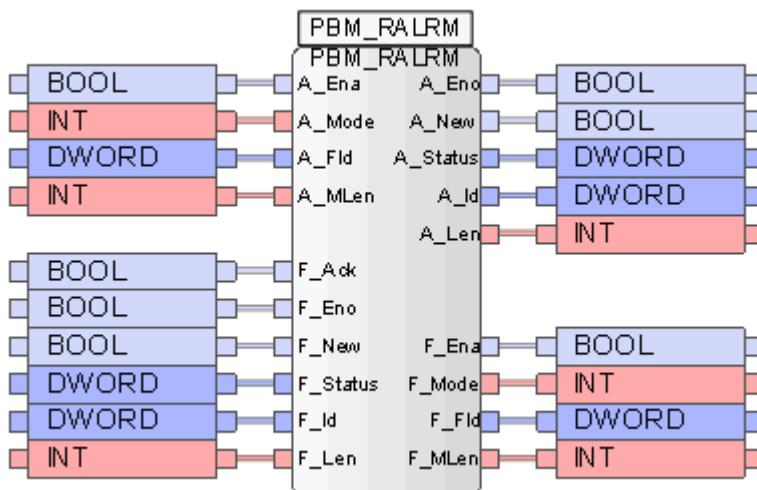


Fig. 45: Bloque funcional RALRM

El bloque funcional **RALRM** (a partir de DP-V1) sirve para evaluar alarmas.

Las alarmas son una forma especial de mensajes de diagnóstico que deben tratarse con prioridad. Las alarmas comunican a la aplicación eventos importantes que requieren de reacciones por parte de la aplicación (p. ej. WRREC). En todo caso, esto dependerá de cada fabricante, pudiendo consultarse en el manual de dispositivo del PROFIBUS DP Slave.

Mientras el bloque funcional **RALRM** esté activo, éste esperará a mensajes de alarma de los Slaves. Si se recibe una alarma, la salida **A\_NEW** cambiará a TRUE durante al menos un ciclo y los datos de la alarma podrán leerse por telegrama de alarma. Antes de la siguiente alarma, **A\_NEW** cambiará a FALSE durante al menos un ciclo. Todas las alarmas se confirman explícitamente. No se pierde ninguna alarma.

En caso de usarse varios bloques funcionales **RALRM**, el programa de usuario deberá crearse de forma tal que cada vez sólo un bloque funcional **RALRM** esté activo.

- 
- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).
- 

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo "A" quiere decir "Application".

Entradas A	Descripción	Tipo
A_Ena	Con TRUE se habilitará el bloque funcional	BOOL
A_Mode	No se usa	INT
A_FId	No se usa	DWORD
A_MLen	Máxima longitud esperada de los datos de alarma recibidos, en bytes	INT

Tabla 103: Entradas A del bloque funcional RALRM

Salidas A	Descripción	Tipo
A_Eno	TRUE: Bloque funcional activo FALSE: Bloque funcional no activo	BOOL
A_New	TRUE: Se ha recibido una nueva alarma FALSE: No hay ninguna nueva alarma	BOOL
A_Status	Código de error o estado (véase el capítulo 6.11)	DWORD
A_ID	N.º de ID del Slave que ha desencadenado la alarma	DWORD
A_Len	Longitud de los datos de alarma recibidos, en bytes	INT

Tabla 104: Salidas A del bloque funcional RALRM

**Entradas y salidas del bloque funcional con el prefijo F:**

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional RALRM en la estructura en árbol. El prefijo "F" quiere decir "Field".



La conexión del bloque funcional **RALRM** en el árbol (en la carpeta de bloques funcionales) al bloque funcional **RALRM** (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá Ud. crearlas previamente en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **RALRM** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **RALRM** en el árbol.

F-Inputs	Tipo
F_ACK	BOOL
F_ENO	BOOL
F_NEW	BOOL
F_STATUS	DWORD
F_ID	DWORD
F_LEN	INT

Tabla 105: Entradas F del bloque funcional RALRM

Conecte las *F-Outputs* del bloque funcional **RALRM** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **RALRM** en el árbol.

F-Outputs	Tipo
F_Ena	BOOL
F_MODE	INT
F_FID	DWORD
F_MLEN	INT

Tabla 106: Salidas F del bloque funcional RALRM

**Para crear el bloque funcional RALRM en la estructura de árbol:**

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **RALRM**.
3. Haga clic con el botón derecho en el bloque funcional **RALRM** y seleccione **Edit**.
  - Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **RALRM** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **RALRM** en el programa de usuario.

Entradas	Tipo
EN	BOOL
F_ID	DWORD
MLEN	INT
MODE	INT

Tabla 107: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **RALRM** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **RALRM** en el programa de usuario.

Salidas	Tipo
ACK	BOOL
ENO	BOOL
ID	DWORD
LEN	INT
NEW	BOOL
STATUS	DWORD

Tabla 108: Variables de salida del sistema

En la ficha “Process variables” del bloque funcional **RALRM** en la estructura de árbol habrá que definir variables cuya estructura sea adecuada a los datos de las alarmas. Si no se definen variables, podrán solicitarse datos de alarmas, pero no podrán leerse.

Un mensaje de alarma tiene al menos 4 bytes. Los primeros cuatro bytes del mensaje de alarma contienen los datos de alarma estándar.

Para una evaluación simplificada de las alarmas estándar, HIMA pone el bloque funcional auxiliar **ALARM** (véase el capítulo 6.10) a su disposición. Si desea Ud. usar dicho bloque, agrupe los primeros cuatro bytes en una variable del tipo DWORD y entregue esta variable a la entrada *I/N* del bloque funcional auxiliar **ALARM**.



Si un telegrama de alarma contiene más bytes de los definidos en la ficha “Data”, se aceptará sólo la cantidad de bytes definida. El resto se cortará.

Datos de alarma	Descripción
Byte 0	Longitud del mensaje de alarma, en bytes (4...126)
Byte 1	ID del tipo de alarma 1: Alarma de diagnóstico 2: Alarma de proceso 3: Alarma "Pull" 4: Alarma "Plug" 5: Alarma de estado 6: Alarma de actualización 31: Fallo de una extensión de un Master o Slave 32...126: Específ. del fabricante: El significado deberá consultarse en la descripción del fabricante del dispositivo.
Byte 2	N.º de slot del componente que ha desencadenado la alarma
Byte 3	0: No hay más información 1: Alarma entrante, perturbación en el slot 2: Alarma saliente, no hay más perturbación en el slot 3: Alarma saliente, el slot sigue perturbado
Byte 4 hasta 126	El significado deberá consultarse en la descripción del fabricante del dispositivo

Tabla 109: Datos de alarma



La estructura de las alarmas estándar (bytes 0...3) está normalizada y es idéntica en todos los fabricantes. Para los bytes 4...126 utilizados específicamente según cada fabricante, consulte el manual de dispositivo del PROFIBUS DP Slave.

Tenga en cuenta que los dispositivos conformes a la norma DP-V0 no son compatibles con telegramas de alarma.

#### Para hacer uso del bloque funcional RALRM:

1. En el programa de usuario, defina para la entrada *A\_Mlen* la cantidad esperada como máximo de datos de alarma, en bytes. Durante el funcionamiento no podrá modificarse *A\_Mlen*.
2. Ponga en el programa de usuario la entrada *A\_Ena* en TRUE.



A diferencia de los otros bloques funcionales, el bloque funcional **RALRM** estará activo sólo mientras la entrada *A\_Ena* sea TRUE.

Si el bloque se ha iniciado correctamente, la salida *A\_Eno* cambiará a TRUE. Si no es posible iniciar el bloque, se generará un código de error en la salida *A\_Status*.

Si se produce una nueva alarma, la salida *A\_New* cambiará a TRUE durante al menos un ciclo. Durante ese tiempo, las salidas contendrán los datos del Slave que haya desencadenado la alarma, los cuales podrán evaluarse.

Después, la salida *A\_New* cambiará de nuevo a FALSE durante al menos un ciclo. Las salidas *A\_Id* y *A\_Len* volverán a ponerse a cero para que se pueda volver a recibir y evaluar el siguiente mensaje de alarma.

### 6.9.3 Bloque funcional RDIAG

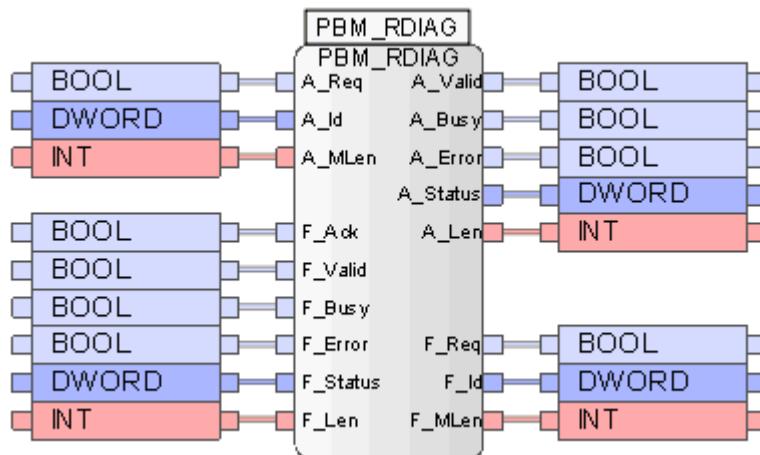


Fig. 46: Bloque funcional RDIAG

El bloque funcional **RDIAG** (a partir de V0 de DP) sirve para leer el mensaje de diagnóstico actual (6 bytes...240 bytes) de un Slave.

En el PROFIBUS DP Master de HIMA se podrán tener activos simultáneamente tantos bloques **RDIAG** como se dese.

- 
- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).
- 

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo "A" quiere decir "Application".

Entradas A	Descripción	Tipo
A_Req	El flanco positivo da inicio al requerimiento de un mensaje de diagnóstico	BOOL
A_ID	N.º de ID del Slave (véase el capítulo 6.10)	DWORD
A_MLen	Máxima longitud esperada del mensaje de diagnóstico a leer, en bytes	INT

Tabla 110: Entradas A del bloque funcional RALRM

Salidas A	Descripción	Tipo
A_Valid	Se ha recibido un nuevo mensaje de diagnóstico que es válido	BOOL
A_Busy	TRUE: No ha finalizado aún la lectura	BOOL
A_Error	TRUE: Se ha producido un error durante la lectura	BOOL
A_Status	Código de error o estado (véase el capítulo 6.11)	DWORD
A_Len	Longitud de los datos de diagnóstico leídos, en bytes	INT

Tabla 111: Salidas A del bloque funcional RALRM

### Entradas y salidas del bloque funcional con el prefijo F:

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional **RDIAG** en la estructura en árbol. El prefijo "F" quiere decir "Field".



La conexión del bloque funcional **RDIAG** en el árbol (en la carpeta de bloques funcionales) al bloque funcional **RDIAG** (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá Ud. crearlas previamente en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **RDIAG** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **RDIAG** en el árbol.

F-Inputs	Tipo
F_ACK	BOOL
F_VALID	BOOL
F_BUSY	BOOL
F_ERROR	BOOL
F_Status	DWORD
F_LEN	INT

Tabla 112: Entradas F del bloque funcional RDIAG

Conecte las *F-Outputs* del bloque funcional **RDIAG** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **RDIAG** en el árbol.

F-Outputs	Tipo
F_Req	BOOL
F_Id	DWORD
F_MLen	INT

Tabla 113: Salidas F del bloque funcional RDIAG

### Para crear el bloque funcional RDIAG en la estructura de árbol:

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **RDIAG**.
3. Haga clic con el botón derecho en el bloque funcional **RDIAG** y seleccione **Edit**.
  - Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **RDIAG** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **RDIAG** en el programa de usuario.

Entradas	Tipo
ID	DWORD
MLEN	INT
REQ	BOOL

Tabla 114: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **RDIAG** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **RDIAG** en el programa de usuario.

Salidas	Tipo
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
Estado	DWORD
VALID	BOOL

Tabla 115: Variables de salida del sistema

### Datos de diagnóstico

En la ficha **Data** habrá que definir variables cuya estructura deberá ser adecuada a los datos de diagnóstico. Un mensaje de diagnóstico tiene como mínimo seis bytes y como máximo 240 bytes. Los primeros cuatro bytes del mensaje de diagnóstico contienen el diagnóstico estándar.

Para una evaluación simplificada del diagnóstico estándar, HIMA pone el bloque funcional auxiliar **STDDIAG** (véase el capítulo 6.10) a su disposición. Si desea Ud. usar dicho bloque, agrupe los primeros cuatro bytes en una variable del tipo DWORD y entregue esta variable a la entrada */N* del bloque funcional auxiliar **STDDIAG**.

- 
- i** Si un telegrama de diagnóstico contiene más bytes de los definidos en la ficha "Data", se aceptará sólo la cantidad de bytes definida. El resto se cortará.
- 

Datos de diagnóstico	Descripción
Byte 0	Los bytes 0...3 contienen el diagnóstico estándar. El diagnóstico estándar podrá decodificarse como variable de tipo DWORD con el bloque funcional auxiliar <b>STDDIAG</b> .
Byte 1	
Byte 2	
Byte 3	Dirección de bus del Master al que se le ha asignado un Slave.
Byte 4	Byte High (ID del fabricante)
Byte 5	Byte Low (ID del fabricante)
Byte 6...240	El significado deberá consultarse en la descripción del fabricante del dispositivo

Tabla 116: Datos de diagnóstico

- 
- i** Los Slaves de HIMA comunican un telegrama de diagnóstico de 6 bytes de longitud. El significado de los bytes está estandarizado.

En los Slaves de otros fabricantes los primeros 6 bytes son funcionalmente idénticos. Hallará más información sobre telegramas de diagnóstico en la descripción de Slave del fabricante.

---

### Para hacer uso del bloque funcional RDIAG:

1. Aplique en el programa de usuario la dirección de Slave a la entrada *A\_ID*.
2. En el programa de usuario, defina para la entrada *A\_Mlen* la cantidad esperada como máximo de datos de diagnóstico, en bytes.
3. Ponga en el programa de usuario la entrada *A\_Req* en TRUE.

- 
- i El bloque funcional reacciona a un cambio de flanco positivo en *A\_Req*.
- 

La salida *A\_Busy* será TRUE hasta que esté ejecutado el requerimiento de diagnóstico. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Valid* o *A\_Error* a TRUE.

Si el telegrama de diagnóstico es válido, la salida *A\_Valid* cambiará a TRUE. Los datos de diagnóstico podrán evaluarse mediante las variables definidas en la ficha “Data”. La salida *A\_Len* contiene la cantidad de datos de diagnóstico, en bytes, que se han leído realmente.

Si no pudo leerse correctamente el telegrama de diagnóstico, la salida *A\_Error* cambiará a TRUE y se generará un código de error en la salida *A\_Status*.

#### 6.9.4 Bloque funcional RDREC

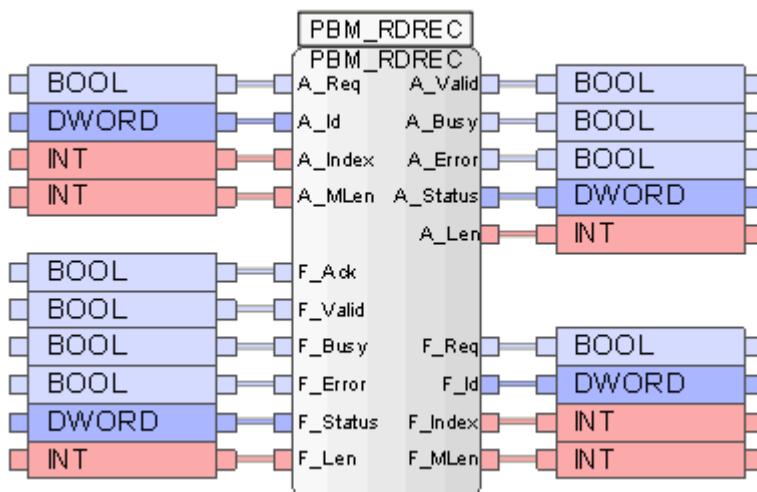


Fig. 47: Bloque funcional RDREC

El bloque funcional **RDREC** sirve para la lectura acíclica de un registro de datos de un Slave direccionado en la salida *A\_Index*. Consulte en el manual de uso del Slave cuáles son los datos que podrán leerse.

Esta función está definida a partir de V1 de DP y es opcional.

En el PROFIBUS DP Master de HIMA puede haber simultáneamente hasta 32 bloques **RDREC** y/o **WRREC** activos.



Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo "A" quiere decir "Application".

Entradas A	Descripción	Tipo
A_Req	El flanco positivo da inicio al requerimiento de lectura.	BOOL
A_Id	N.º de ID del Slave, véase el capítulo 6.10	DWORD
A_Index	N.º de registro de datos del registro de datos a leer. El significado deberá consultarse en la descripción del fabricante del dispositivo.	INT
A_MLen	Máxima longitud de los datos a leer, en bytes.	INT

Tabla 117: Entradas A del bloque funcional RDREC

Salidas A	Descripción	Tipo
A_Valid	Se ha recibido un nuevo registro de datos que es válido.	BOOL
A_Busy	TRUE: No ha finalizado aún la lectura.	BOOL
A_Error	TRUE: Se ha producido un error FALSE: Sin errores	BOOL
A_Status	Código de error o estado, véase el capítulo 6.11	DWORD
A_Len	Longitud de la información leída del registro de datos, en bytes	INT

Tabla 118: Salidas A del bloque funcional RDREC

### Entradas y salidas del bloque funcional con el prefijo F

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional **RDREC** en la estructura en árbol. El prefijo "F" quiere decir "Field".



La conexión del bloque funcional **RDREC** en el árbol (en la carpeta de bloques funcionales) al bloque funcional RDREC (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá Ud. crearlas previamente en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **RDREC** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **RDREC** en el árbol.

F-Inputs	Tipo
F_Ack	BOOL
F_Valid	BOOL
F_Busy	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Tabla 119: Entradas F del bloque funcional RDREC

Conecte las *F-Outputs* del bloque funcional **RDREC** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **RDREC** en el árbol.

F-Outputs	Tipo
F_Req	BOOL
F_Id	DWORD
F_Index	INT
F_Mlen	INT

Tabla 120: Salidas F del bloque funcional RDREC

### Para crear el bloque funcional RDREC en la estructura de árbol:

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **RDREC**.
3. Haga clic con el botón derecho en el bloque funcional **RDREC** y seleccione **Edit**.
  - Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **RDREC** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **RDREC** en el programa de usuario.

Entradas	Tipo
ID	DWORD
INDEX	INT
MLEN	INT
REQ	BOOL

Tabla 121: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **RDREC** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **RDREC** en el programa de usuario.

Salidas	Tipo
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Tabla 122: Variables de salida del sistema

Datos	Descripción
No hay variables predefinidas	En la ficha <i>Process Variables</i> podrá definirse la estructura de datos que se desee, debiendo ésta en todo caso ser adecuada a la estructura del registro de datos. Para la estructura del registro de datos véase el manual de uso del fabricante del Slave.

Tabla 123: Datos

#### Para hacer uso del bloque funcional RDREC:

1. Aplique en el programa de usuario la dirección de Slave a la entrada *A\_ID*.
2. En el programa de usuario aplique el índice específico del Slave para el registro de datos (manual del fabricante) en la entrada *A\_Index*.
3. Aplique en el programa de usuario la longitud del registro de datos a leer en la entrada *A\_Len*.
4. Ponga en el programa de usuario la entrada *A\_Req* en TRUE.



El bloque funcional reacciona a un cambio de flanco positivo en *A\_Req*.

La salida *A\_Busy* será TRUE hasta que esté ejecutado el requerimiento del registro de datos. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Valid* o *A\_Error* a TRUE.

Si el registro de datos es válido, la salida *A\_Valid* cambiará a TRUE. El registro de datos podrá evaluarse mediante las variables definidas en la ficha “Data”. La salida *A\_Len* contiene la longitud del registro de datos, en bytes, que se ha leído realmente.

Si no pudo leerse correctamente el registro de datos, la salida *A\_Error* cambiará a TRUE y se generará un código de error en la salida *A\_Status*.

### 6.9.5 Bloque funcional SLACT

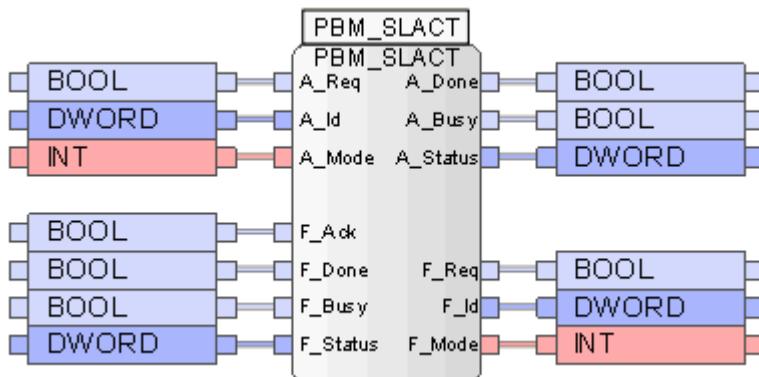


Fig. 48: Bloque funcional SLACT

El bloque funcional **SLACT** (a partir de V0 de DP) sirve para activar y desactivar un Slave desde el programa de usuario del PROFIBUS DP Master. Así, el Slave podrá aplicarse a una entrada física del PROFIBUS DP Master mediante un interruptor mecánico o ponerse en uno de los siguientes estados mediante un temporizador.

$\neq 0$ : activo

$= 0$ : no activo

En caso de usarse varios bloques funcionales **SLACT**, el programa de usuario deberá crearse de forma tal que cada vez sólo un bloque funcional **SLACT** esté activo.



Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo "A" quiere decir "Application".

Entradas A	Descripción	Tipo
A_Req	El flanko positivo da inicio al bloque	BOOL
A_Id	N.º de ID del Slave (véase el capítulo 6.10)	DWORD
A_Mode	Estado en el que desea ponerse el PROFIBUS DP Slave: $\neq 0$ : activo (conectado) $= 0$ : no activo (desactivado)	INT

Tabla 124: Entradas A del bloque funcional SLACT

Salidas A	Descripción	Tipo
A_Done	TRUE: El PROFIBUS DP Slave se ha puesto en el estado definido en la entrada "A_Mode".	BOOL
A_Busy	TRUE: No ha finalizado el cambio de estado del PROFIBUS DP Slave.	BOOL
A_Status	Código de error o estado (véase el capítulo 6.11)	DWORD

Tabla 125: Salidas A del bloque funcional SLACT

### Entradas y salidas del bloque funcional con el prefijo F:

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional **SLACT** en la estructura en árbol. El prefijo "F" quiere decir "Field".



La conexión del bloque funcional SLACT en el árbol (en la carpeta de bloques funcionales) al bloque funcional SLACT (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá Ud. crearlas previamente en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **SLACT** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **SLACT** en el árbol.

F-Inputs	Tipo
F_Ack	BOOL
F_Done	BOOL
F_Busy	BOOL
F_Status	DWORD

Tabla 126: Entradas F del bloque funcional SLACT

Conecte las *F-Outputs* del bloque funcional **SLACT** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **SLACT** en el árbol.

F-Outputs	Tipo
F_Req	BOOL
F_Id	DWORD
F_Mode	INT

Tabla 127: Salidas F del bloque funcional SLACT

### Para crear el bloque funcional SLACT en la estructura de árbol:

1. Abra Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New en el árbol.
2. Seleccione el bloque funcional **SLACT**.
3. Haga clic con el botón derecho en el bloque funcional **SLACT** y seleccione **Edit**.
  - Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **SLACT** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **SLACT** en el programa de usuario.

Entradas	Tipo
ID	DWORD
MODE	INT
REQ	BOOL

Tabla 128: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **SLACT** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **SLACT** en el programa de usuario.

Salidas	Tipo
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Tabla 129: Variables de salida del sistema

**Para hacer uso del bloque funcional SLACT:**

1. Ponga en el programa de usuario la entrada *A\_Mode* en el estado deseado.
2. Aplique en el programa de usuario el ID con la dirección de Slave a la entrada *A\_ID*.
3. Ponga en el programa de usuario la entrada *A\_Req* en TRUE.



*El bloque funcional reacciona a un cambio de flanco positivo en A\_Req.*

La salida *A\_Busy* es TRUE hasta que está ejecutado el comando SLACT. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Done* a TRUE.

En la salida *A\_Status* se señalizará el modo Slave-Mode, en caso de que dicho modo se haya podido aplicar.

En la salida *A\_Status* se señalizará un código de error, en caso de que el modo Slave-Mode no se haya podido aplicar.

### 6.9.6 Bloque funcional WRREC

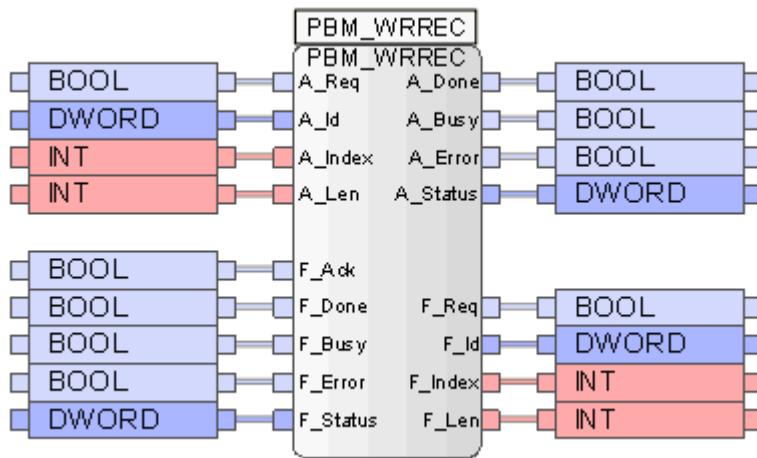


Fig. 49: Bloque funcional WRREC

El bloque funcional **WRREC** (a partir de V1 de DP) sirve para la escritura acíclica de un registro de datos a un Slave direccionado en la salida *A\_Index*. Consulte en el manual de uso del Slave cuáles son los datos que podrán escribirse.

En el PROFIBUS DP Master de HIMA puede haber simultáneamente hasta 32 bloques **RDREC** y/o **WRREC** activos.



Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo “A” quiere decir “Application”.

Entradas A	Descripción	Tipo
A_Req	El flanco positivo da inicio al requerimiento de escritura de un registro de datos.	BOOL
A_ID	N.º de ID del Slave (véase el capítulo 6.10)	DWORD
A_Index	N.º de registro de datos del registro de datos a escribir. El significado deberá consultarse en la descripción del fabricante del dispositivo.	INT
A_Len	Longitud del registro de datos a escribir, en bytes	INT

Tabla 130: Entradas A del bloque funcional WRREC

Salidas A	Descripción	Tipo
A_Done	TRUE: El bloque funcional ha finalizado la operación de escritura.	BOOL
A_Busy	TRUE: El bloque funcional no ha finalizado aún la operación de escritura.	BOOL
A_Error	TRUE: Durante la escritura se ha producido un error.	BOOL
A_STATUS	Código de error o estado (véase el capítulo 6.11)	DWORD

Tabla 131: Salidas A del bloque funcional WRREC

### Entradas y salidas del bloque funcional con el prefijo F:

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional **WRREC** en la estructura en árbol. El prefijo “F” quiere decir “Field”.



La conexión del bloque funcional WRREC en el árbol (en la carpeta de bloques funcionales) al bloque funcional WRREC (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá Ud. crearlas previamente en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **WRREC** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **WRREC** en el árbol.

F-Inputs	Tipo
F_Ack	BOOL
F_Done	BOOL
F_Busy	BOOL
F_Error	BOOL
F_Status	DWORD

Tabla 132: Entradas F del bloque funcional WRREC

Conecte las *F-Outputs* del bloque funcional **WRREC** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **WRREC** en el árbol.

F-Outputs	Tipo
F_Req	BOOL
F_Id	DWORD
F_Index	INT
F_Len	INT

Tabla 133: Salidas F del bloque funcional WRREC

### Para crear el bloque funcional WRREC en la estructura de árbol:

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **WRREC**.
3. Haga clic con el botón derecho en el bloque funcional **WRREC** y seleccione **Edit**.
  - Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **WRREC** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **WRREC** en el programa de usuario.

Entradas	Tipo
ID	DWORD
INDEX	INT
LEN	INT
REQ	BOOL

Tabla 134: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **WRREC** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **WRREC** en el programa de usuario.

Salidas	Tipo
ACK	BOOL
BUSY	BOOL
DONE	BOOL
ERROR	BOOL
STATUS	DWORD

Tabla 135: Variables de salida del sistema

Datos	Descripción
No hay variables predefinidas	En la ficha <i>Process Variables</i> podrá definirse la estructura de datos que se desee, debiendo ésta en todo caso ser adecuada a la estructura del registro de datos. Para la estructura del registro de datos véase el manual de uso del fabricante del Slave.

Tabla 136: Datos

**Para hacer uso del bloque funcional WRREC deben seguirse estos pasos:**

1. Aplique en el programa de usuario la dirección de Slave a la entrada *A\_ID*.
2. En el programa de usuario aplique el índice específico del Slave para el registro de datos (manual del fabricante) en la entrada *A\_Index*.
3. Aplique en el programa de usuario la longitud del registro de datos a escribir en la entrada *A\_Len*.
4. Ajuste el registro de datos en el programa de usuario tal y como se define en la ficha “Data”.
5. Ponga en el programa de usuario la entrada *A\_Req* en TRUE.



El bloque funcional reacciona a un cambio de flanco positivo en *A\_Req*.

La salida *A\_Busy* será TRUE hasta que esté escrito el registro de datos. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Done* a TRUE.

Si no pudo escribirse correctamente el registro de datos, la salida *A\_Error* cambiará a TRUE y se generará un código de error en la salida *A\_Status*.

## 6.10 Bloques funcionales auxiliares de PROFIBUS

Los bloques funcionales auxiliares sirven para parametrizar y evaluar las entradas y salidas de los bloques funcionales.

Se dispone de los siguientes bloques funcionales auxiliares:

Bloques funcionales auxiliares	Descripción de la función
ACTIVE (véase el cap. 6.10.1)	El Slave está activo o inactivo
ALARM (véase el cap. 6.10.2)	Decodificación de los datos de alarma
DEID (véase el cap. 6.10.3)	Decodificación del número de ID
ID (véase el cap. 6.10.4)	El ID de función genera un identificador a partir de cuatro bytes
NSLOT (véase el cap. 6.10.5)	Creación de números de ID consecutivos para los slots
SLOT (véase el cap. 6.10.6)	Creación de números de ID de SLOT con número de slot
STDDIAG (véase el cap. 6.10.7)	Decodificación del diagnóstico estándar de un Slave
LATCH	Se usa sólo dentro de otros bloques funcionales
PIG	Se usa sólo dentro de otros bloques funcionales
PIGII	Se usa sólo dentro de otros bloques funcionales

Tabla 137: Sinopsis de los bloques funcionales auxiliares

### 6.10.1 Bloque funcional auxiliar ACTIVE

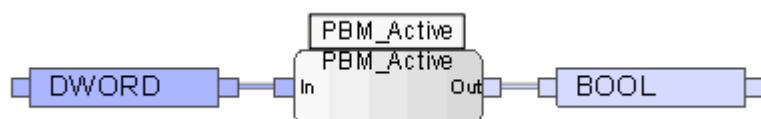


Fig. 50: Bloque funcional auxiliar ACTIVE

El bloque funcional auxiliar ACTIVE determina, a partir del diagnóstico estándar de un PROFIBUS DP Slave, si el Slave está activo o inactivo.

- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

Entradas	Descripción	Tipo
IN	Diagnóstico estándar del Slave	DWORD

Tabla 138: Entradas del bloque funcional auxiliar ACTIVE

Salidas	Descripción	Tipo
OUT	TRUE: Slave activo FALSE: Slave inactivo	BOOL

Tabla 139: Salidas del bloque funcional auxiliar ACTIVE

### 6.10.2 Bloque funcional auxiliar ALARM (decodificación de los datos de alarma)

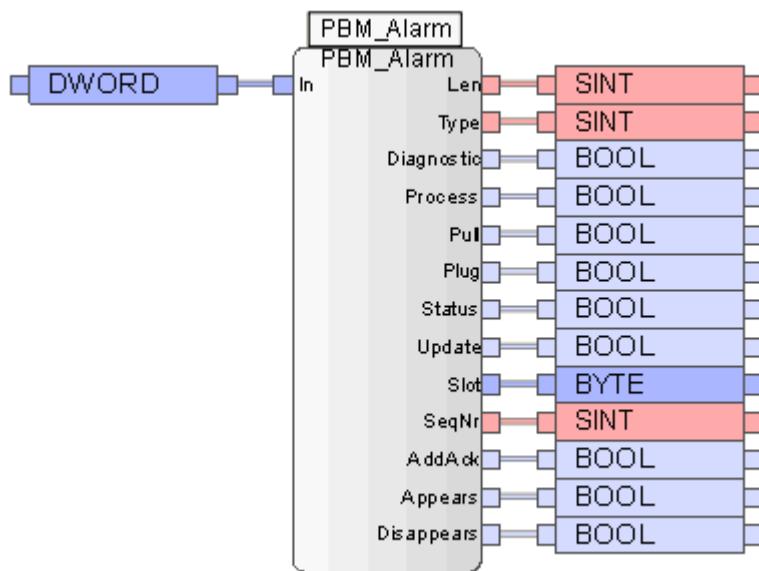


Fig. 51: Bloque funcional auxiliar ALARM

El bloque funcional auxiliar **ALARM** decodifica los datos de alarma estándar de un PROFIBUS DP Slave.

**i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

Entradas	Descripción	Tipo
IN	Alarma estándar	DWORD

Tabla 140: Entradas del bloque funcional auxiliar ALARM

Salida	Descripción	Tipo
Len	Longitud de todo el mensaje de alarma	SINT
Type	1: Alarma de diagnóstico 2: Alarma de proceso 3: Alarma “Pull” 4: Alarma “Plug” 5: Alarma de estado 6: Alarma de actualización Otros números están reservados o son específicos de cada fabricante. El significado deberá consultarse en la descripción del fabricante del dispositivo.	SINT
Diagnostic	True = Alarma de diagnóstico	BOOL
Process	True = Alarma de proceso	BOOL
Pull	True = Se retiró el módulo	BOOL
Plug	True = Se volvió a introducir el módulo	BOOL
Estado	True = Alarma de estado	BOOL
Update	True = Alarma de actualización	BOOL
Slot	Módulo desencadenante de la alarma	BYTE
SeqNr	Número secuencial de alarma	SINT

Salida	Descripción			Tipo
AddAck	TRUE significa que el Slave que ha desencadenado esa alarma, espera una confirmación adicional por parte de la aplicación. Consulte en el manual de Slave del fabricante cuál es exactamente.			BOOL
Appears Disappears	Salida	Valor	Descripción	BOOL
	Appears	TRUE	Si ambas son FALSE, no se habrá producido ningún error hasta el momento	
	Disappears	FALSE		
	Appears	TRUE	Se ha producido un error y sigue presente.	
	Disappears	FALSE		
	Appears	TRUE	Se produjo un error y justo ahora deja de estar presente.	
	Disappears	FALSE		
	Appears	TRUE	Si son ambas TRUE, el error habrá desaparecido pero el Slave seguirá perturbado.	
	Disappears	FALSE		

Tabla 141: Salidas del bloque funcional auxiliar ALARM

### 6.10.3 Bloque funcional auxiliar DEID

(decodificación del número de ID)

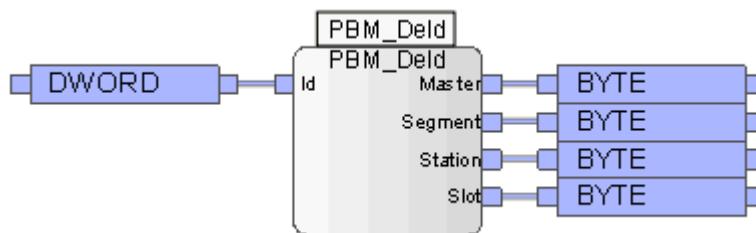


Fig. 52: Bloque funcional auxiliar DEID

El bloque funcional auxiliar **DEID** decodifica el número de ID y lo descompone en sus cuatro componentes.



Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

Entradas	Descripción	Tipo
Id	N.º de ID del Slave	DWORD

Tabla 142: Entradas del bloque funcional auxiliar DEID

Salidas	Descripción	Tipo
Master	Dirección de bus del Master	BYTE
Segment	Segment	BYTE
Station	Dirección de bus del Slave	BYTE
Slot	N.º de módulo o de slot	BYTE

Tabla 143: Salidas del bloque funcional auxiliar DEID

#### 6.10.4 Bloque funcional auxiliar ID

(generación del número de ID)

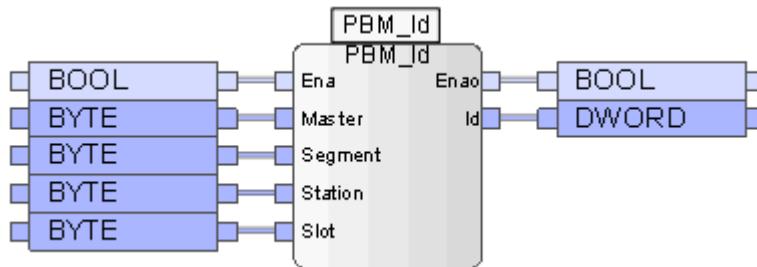


Fig. 53: Bloque funcional auxiliar ID

El bloque funcional auxiliar **ID** genera, a partir de los cuatro bytes, un identificador (número de ID) que puede ser utilizado por otros bloques funcionales auxiliares.

- 
- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).
- 

Entradas	Descripción	Tipo
Ena	No se usa	BOOL
Master	Dirección de bus	BYTE
Segment	Segment	BYTE
Station	Dirección de bus del Slave	BYTE
Slot	N.º de módulo o de slot	BYTE

Tabla 144: Entradas del bloque funcional auxiliar ID

Salidas	Descripción	Tipo
Enao	No se usa	BOOL
Id	N.º de ID del Slave	DWORD

Tabla 145: Salidas del bloque funcional auxiliar ID

### 6.10.5 Bloque funcional auxiliar NSLOT

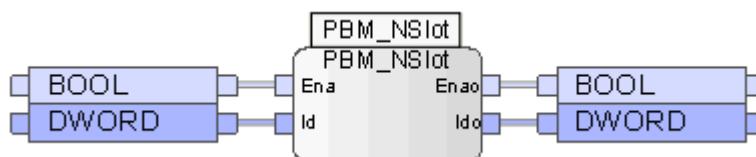


Fig. 54: Bloque funcional auxiliar NSLOT

El bloque funcional auxiliar **NSLOT** genera, a partir de un identificador, un nuevo identificador que direcciona el siguiente slot del mismo Slave. Ena deberá ser TRUE para que el bloque funcional auxiliar funcione.

Enao será TRUE cuando el resultado de la salida “Ido” sea válido.

- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

Entradas	Descripción	Tipo
Ena	Mientras TRUE esté presente, el bloque funcionará.	BOOL
Id	N.º de ID del Slave	DWORD

Tabla 146: Entradas del bloque auxiliar NSLOT

Salidas	Descripción	Tipo
Enao	TRUE = Resultado válido FALSE = No hay más números de slot	BOOL
Ido	N.º de ID del Slave	DWORD

Tabla 147: Salidas del bloque auxiliar NSLOT

### 6.10.6 Bloque funcional auxiliar SLOT

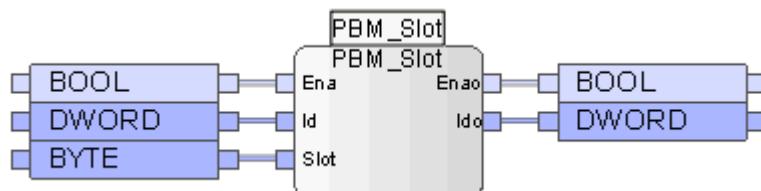


Fig. 55: Bloque funcional auxiliar SLOT

El bloque funcional auxiliar **SLOT** genera, a partir de un identificador y un número de slot, un nuevo identificador que direcciona el mismo Slave que el identificador anterior pero con el nuevo número de slot.

- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

Entradas	Descripción	Tipo
Ena	No se usa	BOOL
Id	Dirección lógica de componentes de Slave (ID de Slave y N.º de slot)	DWORD
Slot	Nuevo número de módulo o de slot	BYTE

Tabla 148: Entradas del bloque auxiliar SLOT

Salidas	Descripción	Tipo
Enao	No se usa	BOOL
Ido	N.º de ID del Slave	DWORD

Tabla 149: Salidas del bloque auxiliar SLOT

### 6.10.7 Bloque funcional auxiliar STDDIAG

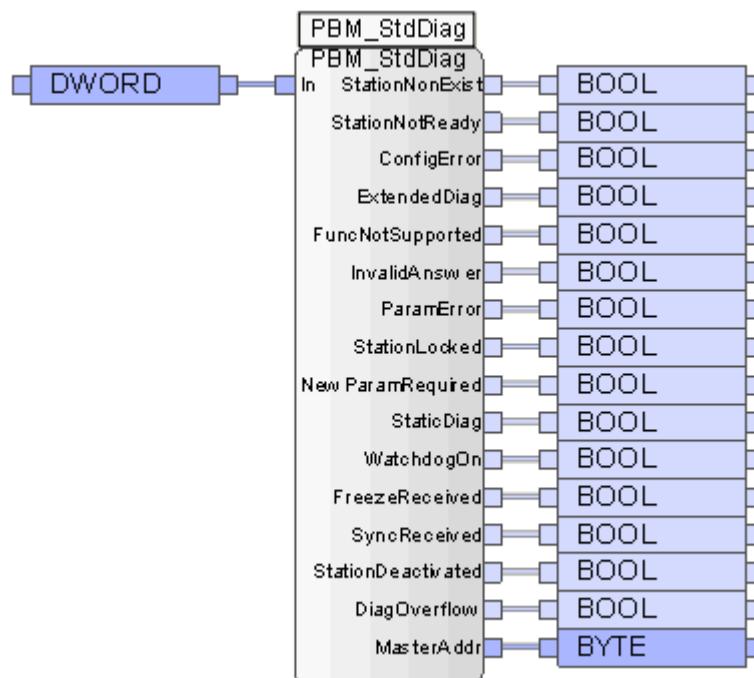


Fig. 56: Bloque funcional auxiliar STDDIAG

El bloque funcional auxiliar **STDDIAG** (Standard Diagnosis) decodifica el diagnóstico estándar de un PROFIBUS DP Slave.

Las salidas del tipo BOOL del bloque funcional **STDDIAG** serán TRUE cuando se aplique el correspondiente bit del diagnóstico estándar.

- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

Entradas	Descripción	Tipo
IN	Diagnóstico estándar del Slave	DWORD

Tabla 150: Entradas del bloque auxiliar STDDIAG

Salidas	Descripción	Tipo
StationNonExist	Slave no existe	BOOL
StationNotReady	Slave no listo	BOOL
ConfigError	Error de configuración	BOOL
ExtendedDiag	Diagnóstico ampliado	BOOL
FuncNotSupported	Función no compatible	BOOL
InvalidAnswer	Respuesta no válida del Slave	BOOL
ParamError	Error de parametrización	BOOL
StationLocked	Slave bloqueado por otro Master	BOOL
NewParamRequired	Se necesitan nuevos datos de parametrización	BOOL
StaticDiag	Diagnóstico estático	BOOL
WatchdogOn	WatchDog activado	BOOL
FreezeReceived	Se ha recibido el comando "Freeze"	BOOL
SyncReceived	Se ha recibido el comando "Sync"	BOOL
StationDeactivated	Slave desactivado	BOOL
DiagOverflow	Desborde de diagnóstico	BOOL
MasterAddr	Dirección de bus del Master	BYTE

Tabla 151: Salidas del bloque auxiliar STDDIAG

**Para leer el diagnóstico estándar del PROFIBUS DP Slave:**

1. Abra **Configuration, Resource, Protocols, PROFIBUS DP Master** en el árbol.
2. Haga clic con el botón derecho en **PROFIBUS Slave** y seleccione **Edit**.
3. Arrastre una variable global del tipo DWORD al área **Standard Diagnosis**.
4. Conecte esta variable global a la entrada del bloque funcional **STDDIAG**.

## 6.11 Códigos de error de los bloques funcionales

Si un bloque funcional no ha podido ejecutar un comando correctamente, se generará un código de error en la salida **A\_Status**. Véase el significado de los códigos de error en la siguiente tabla:

Código de error	Símbolo	Descripción
16#40800800	TEMP_NOT_AVAIL	Servicio temporalmente no disponible
16#40801000	INVALID PARA	Parámetros no válidos
16#40801100	WRONG STATE	Slave no compatible con DP-V1
16#40808000	FATAL_ERR	Error fatal del programa
16#40808100	BAD_CONFIG	Error de configuración en área de datos
16#40808200	PLC_STOPPED	Sistema de control detenido
16#4080A000	READ_ERR	Error al leer un registro
16#4080A100	WRITE_ERR	Error al escribir un registro
16#4080A200	MODULE_FAILURE	Error no especificable con mayor detalle
16#4080B000	INVALID_INDEX	Índice no válido
16#4080B100	WRITE_LENGTH	Longitud errónea al escribir
16#4080B200	INVALID_SLOT	N.º de slot no válido
16#4080B300	TYPE_CONFLICT	Tipo erróneo
16#4080B400	INVALID_AREA	Área errónea de lectura o escritura
16#4080B500	STATE_CONFLICT	Master en estado erróneo
16#4080B600	ACCESS_DENIED	Slave no activo (o similar)
16#4080B700	INVALID_RANGE	Área errónea de lectura o escritura
16#4080B800	INVALID_PARAMETER	Valor erróneo de parámetro
16#4080B900	INVALID_TYPE	Tipo erróneo de parámetro
16#4080C300	NO_RESOURCE	Slave no presente
16#4080BA00	BAD_VALUE	Valor no válido
16#4080BB00	BUS_ERROR	Error de bus
16#4080BC00	INVALID_SLAVE	ID de Slave no válido
16#4080BD00	TIMEOUT	Se ha llegado a un time-out
16#4080C000	READ_CONSTRAIN	Restricción de lectura
16#4080C100	WRITE_CONSTRAIN	Restricción de escritura
16#4080C200	BUSY	Hay un bloque de este tipo ya activo
16#4080C300	NO_RESOURCE	Slave no activo

Tabla 152: Códigos de error de bloques funcionales

## 6.12 Panel de control (PROFIBUS DP Master)

El usuario podrá comprobar y controlar la configuración del PROFIBUS DP Master en el panel de control. Además, allí se muestra la información actual del estado del Master y sus respectivos Slaves (p. ej. tiempo de ciclo, estado del bus, etc.).

**Para abrir el panel de control para supervisar el PROFIBUS DP Master:**

1. Seleccione **Hardware** y en el menú contextual elija **Online**.
2. Escriba en **System Login** sus datos de acceso para abrir la vista en línea del hardware.
3. Haga doble clic en **COM Module** y seleccione **PROFIBUS DP Master** en el árbol.

### 6.12.1 Menú contextual (PROFIBUS Master)

En el menú contextual del PROFIBUS DP Master seleccionado podrán elegirse los siguientes comandos:

**Offline:**

Desactiva el PROFIBUS DP Master seleccionado. Si el Master está desactivado, no habrá actividad.

**Stop:**

Detiene el PROFIBUS DP Master seleccionado. El PROFIBUS DP Master seguirá participando en el protocolo de Token, pero no transmitirá datos a los Slaves.

**Clear:**

Accionando el botón CLEAR, el PROFIBUS DP Master seleccionado adoptará un estado seguro e intercambiará sólo datos seguros con los Slaves. Los datos de salida enviados a los Slaves contendrán solamente ceros. Los Slaves FailSafe recibirán telegramas FailSafe sin datos. PROFIBUS DP Master ignorará los datos de entrada de los Slaves y usará en su lugar los valores iniciales del programa del usuario.

**Operate:**

Inicia el PROFIBUS DP Master seleccionado. El PROFIBUS DP Master intercambia cíclicamente datos de E/S con Slaves.

**Reset Statistics:**

Con el botón *Reset Statistics* podrá Ud. poner a cero los datos estadísticos (valores máximos, mínimos del tiempo de ciclo, etc.).

### 6.12.2 Menú contextual (PROFIBUS Slave)

En el menú contextual del PROFIBUS DP Slave seleccionado podrán elegirse los siguientes comandos:

**Activate:**

Activa el Slave seleccionado, que podrá ahora intercambiar datos con el PROFIBUS DP Master.

**Deactivate:**

Desactiva el Slave seleccionado. La comunicación finaliza.

### 6.12.3 Recuadro de lectura (PROFIBUS Master)

En el recuadro de lectura se mostrarán los siguientes valores del PROFIBUS DP Master seleccionado.

Elemento	Descripción
Name	Nombre del PROFIBUS DP Master
Master State	Indica el estado momentáneo del protocolo (véase el capítulo 6.12.4). 0 = OFFLINE 1 = STOP 2 = CLEAR 3 = OPERATE 100 = UNDEFINED
Bus State	Código de error de bus 0...6: 0 = OK 1 = Error de dirección: La dirección del Master ya existe en el bus. 2 = Fallo del bus: Fallo registrado en el bus, p. ej. bus no terminado correctamente, varios participantes transmitiendo al mismo tiempo. 3 = Error de protocolo: Paquete incorrectamente codificado recibido. 4 = Error de hardware: El hardware ha comunicado un error, p. ej. por haberse ajustado tiempos insuficientes. 5 = Error desconocido: El Master ha cambiado de estado por razones desconocidas. 6 = Reset de controlador: En caso de graves fallos del bus, el chip del controlador pasará a un estado indefinido y se reinicializará. El código de error permanecerá aplicado hasta subsanar el error de bus.
Fieldbus Interface	FB1, FB2
μP load (planned) [%]	Indicador de la tasa de trabajo del módulo COM planificada para este protocolo.
μP load (actual) [%]	Tasa de trabajo real del módulo COM para este protocolo.
Baud Rate [bps]	Velocidad de transmisión del Master. El Master podrá comunicar a todas las velocidades de transmisión especificadas como estándar. Son posibles tiempos de ciclo hasta un límite inferior de 2 ms.
Fieldbus address	Dirección de bus del Master (0...125)
PNO-IdentNo	Número de 16 bits asignado por la Organización de Usuarios "PROFIBUS DP Nutzerorganisation e.V." y que identifica inequívocamente un producto (dispositivo de campo).
Bus Error Count	Cantidad de errores de bus hasta el momento
MSI [ms]	"Min. Slave Interval" en ms, resolución de 0,1 ms
TTR [ms]	"Target Rotation Time" en ms, resolución de 0,1 ms
Last cycle time [ms]	Último tiempo de ciclo de PROFIBUS DP [ms]
Minimum Cycle Time [ms]	Mínimo tiempo de ciclo de PROFIBUS DP [ms]

Elemento	Descripción
Average Cycle Time [ms]	Promedio de tiempo de ciclo de PROFIBUS DP [ms]
Maximum Cycle Time [ms]	Máximo tiempo de ciclo de PROFIBUS DP [ms]

Tabla 153: Recuadro de lectura de PROFIBUS Master

#### 6.12.4 Estado del PROFIBUS DP Master

El estado del Master se indica en el recuadro de lectura del panel de control y puede evaluarse en el programa del usuario con la variable de estado “Master Connection State”.

Estado del Master	Estado del Master
OFFLINE	El Master está desactivado y no hay actividad en el bus.
STOP	El PROFIBUS DP Master participa en el protocolo de Token, pero no transmite datos a los Slaves.
CLEAR	El Master se halla en estado seguro e intercambia datos con los Slaves. <ul style="list-style-type: none"> <li>▪ Los datos de salida enviados a los Slaves contendrán solamente ceros.</li> <li>▪ Los Slaves FailSafe recibirán telegramas FailSafe (que no contienen datos).</li> <li>▪ Los datos de entrada de los Slaves se ignorarán y en su lugar se usarán los valores iniciales.</li> </ul>
OPERATE	El Master está en modo de trabajo e intercambia cíclicamente datos de E/S con los Slaves.
UNDEFINED	Actualización del firmware del PROFIBUS DP Master en curso.

Tabla 154: Estado del PROFIBUS DP Master

#### 6.12.5 Comportamiento del PROFIBUS DP Master

Comportamiento del PROFIBUS DP Master según el estado operativo del sistema de control.

Estado del sistema de control	Comportamiento del PROFIBUS DP Master de HIMA
STOP *)	Sistema de control en STOP, Master en estado OFFLINE.
RUN	Sistema de control en RUN, el Master intenta adoptar el estado OPERATE.
STOP	Sistema de control en STOP, el Master adopta el estado CLEAR. Si el Master ya está en STOP u OFFLINE, permanecerá en ese estado.

\*) Tras encender el sistema de control o tras cargar la configuración

Tabla 155: Comportamiento del PROFIBUS DP Master

### 6.12.6 Función del LED FBx en el PROFIBUS DP Master

El LED FBx de la interfaz de bus de campo asignada señalaiza el estado del protocolo PROFIBUS-DP. Los estados del LED FBx se representan en la siguiente tabla:

LED FBx	Descripción
Apagado	Sin configuración o configuración no válida del PROFIBUS DP Master.
Cadencia de parpadeo de 2 s	Configuración válida. PROFIBUS DP Master en estado OFFLINE o STOP.
Encendido	PROFIBUS DP Master en estado OPERATE o CLEAR si todos los Slaves activados están conectados.
Cadencia de parpadeo de 1 s	Al menos un Slave ha dejado de funcionar.

Tabla 156: LED FBx

### 6.12.7 Función del LED FAULT en el PROFIBUS DP Master (sólo HIMax)

El LED FAULT de la interfaz de bus de campo asignada señalaiza el fallo del protocolo PROFIBUS-DP. Los estados del LED FAULT se representan en la siguiente tabla:

LED FAULT	Color	Descripción
Apagado	Rojo	Protocolo PROFIBUS-DP sin fallos.
Parpadeo	Rojo	Protocolo perturbado. <ul style="list-style-type: none"> <li>▪ Al menos un Slave ha dejado de funcionar</li> <li>▪ Error de bus detectado.</li> <li>▪ Tiempo de cálculo adjudicado excedido.</li> </ul> Si durante más de 5 segundos no se produce ningún evento de error, el indicador cambiará al estado <i>Protocol not disturbed</i> .

Tabla 157: FAULT FBx

## 6.13 HIMA PROFIBUS DP Slave

En esta capítulo se describen las propiedades del PROFIBUS DP Slave de HIMA, así como las funciones de los menús de SILworX necesarios para configurar el PROFIBUS DP Slave de HIMA.

### Equipamiento y sistema necesarios

Elemento	Descripción
Sistema de control HIMA	HIMax con módulo COM HIMatrix a partir de V7 de S.Op. de CPU y V12 de S.Op. de COM
Módulo COM	En la interfaz de bus de campo serie utilizada (FB1 ó FB2), el módulo COM deberá estar equipado con un submódulo PROFIBUS DP Slave opcional de HIMA. Para la asignación de interfaces véase el capítulo 3.6.
Activación	Para la habilitación mediante el submódulo véase el capítulo 3.4.

Tabla 158: Equipamiento y sistema necesarios para PROFIBUS DP Slave de HIMA



Para HIMax, HIMA recomienda operar PROFIBUS DP mediante la interfaz de bus de campo FB1 (tasa de transmisión máxima de 12 Mbit). A través de la interfaz de bus de campo FB2 se permite una tasa de transmisión máxima de 1,5 Mbit.

### Propiedades de PROFIBUS DP Slave

Elemento	Descripción
Tipo de PROFIBUS DP Slave de HIMA	DP-V0
Velocidad de transmisión	9,6 kbit/s...12 Mbit/s
Dirección de bus	0...125
Máx. cantidad de Slaves	Para cada módulo COM podrá configurarse un PROFIBUS DP Slave de HIMA.
Volumen de datos de proceso de un PROFIBUS DP Slave de HIMA	Salida DP: máx. 192 bytes Entrada DP: máx. 240 bytes Sin embrago, en total: máx. 256 bytes
Protocolo de WatchDog	Si con el módulo COM en estado RUN se pierde la conexión al PROFIBUS DP Master, el DP Slave lo detectará una vez transcurra el tiempo de time-out de Watchdog (deberá parametrizarse en el Master). En tal caso, los datos de la salida DP (datos de entrada desde la perspectiva del recurso) se pondrán a su valor inicial y la marca <i>Data Valid</i> (variable de estado del protocolo de Slave DP) cambiará a FALSE.

Tabla 159: Propiedades de PROFIBUS DP Slave de HIMA

### 6.13.1 Creación de un PROFIBUS DP Slave de HIMA

#### Para crear un nuevo PROFIBUS DP Slave de HIMA:

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Seleccione **New, PROFIBUS DP Slave** en el menú contextual de los protocolos, para agregar un nuevo PROFIBUS DP Slave.
3. Seleccione **Edit** en el menú contextual de PROFIBUS DP Slave.
4. En la ficha **Properties** seleccione **Module e Interfaces**.

## 6.14 Funciones de menú de PROFIBUS DP Slave

### 6.14.1 Edit

El cuadro de diálogo **Edit** del PROFIBUS DP Master contiene las siguientes fichas.

#### Variables de proceso

En la ficha **Process Variables** se crean las variables de recepción y de envío.

#### Variables de entrada

Las variables que se desea que reciba este sistema de control se introducen en el área *Input Signals*.

En el área *Input Signals* pueden crearse las variables que se quiera. Sin embargo, los offsets y los tipos de variable deberán ser idénticos a los offsets y los tipos de las variables de envío del interlocutor de comunicación.

#### Variables de salida

Las variables de intercambio de datos cíclico que se desea que envíe este sistema de control se introducen en el área *Output Signals*.

En el área *Output Signals* pueden crearse las variables que se quiera. Sin embargo, los offsets y los tipos de variable deberán ser idénticos a los offsets y los tipos de las variables de recepción del interlocutor de comunicación.

#### Variables de sistema

En la ficha **System Variables** se define qué variables se desea que lea el sistema de control.

La ficha **System Variables** ofrece las siguientes variables de sistema, las cuales permiten evaluar el estado del PROFIBUS DP Slave en el programa del usuario.

Elemento	Descripción
Current baud rate	Velocidad de transmisión a la que opera actualmente el protocolo PROFIBUS-DP-Slave.
Data valid	<p>Si la variable de estado <i>Data Valid</i> es TRUE, el Slave habrá recibido datos de importación del Master. La variable de estado será FALSE si expira el tiempo de WatchDog en el Slave.</p> <p>Valor por defecto: FALSE</p> <p>Nota: Si no se activó el WatchDog del Slave por parte del Master y se corta la conexión, la variable de estado <i>Data Valid</i> conservará el valor TRUE, porque el PROFIBUS DP Slave no tiene ninguna posibilidad de detectar la pérdida de la conexión.</p> <p>¡Si usa esta variable, tenga siempre en cuenta esta circunstancia!</p>

Elemento	Descripción
Error Code	<p>Si se produce un error en el protocolo de PROFIBUS-DP-Slave, el mismo se transmitirá en esta variable. Cada vez se indicará el error actualmente producido.</p> <p>Posibles valores (hexadecimales) son:</p> <ul style="list-style-type: none"> <li>0x00: sin errores</li> <li>0xE1: parametrización errónea por parte del PROFIBUS DP Master</li> <li>0xD1: parametrización errónea por parte del PROFIBUS DP Master</li> </ul> <p>Valor por defecto: 0x00</p>
Master Address	<p>Es la dirección del PROFIBUS DP Master, el cual ha parametrizado y configurado su propio PROFIBUS DP Slave.</p> <p>Posibles valores (decimales) son:</p> <ul style="list-style-type: none"> <li>0–125: dirección del Master</li> <li>255: Slave actualmente sin asignar a un Master</li> </ul> <p>Valor por defecto: 255</p>
Protocol State	<p>Describe el estado del protocolo de PROFIBUS-DP-Slave.</p> <p>Posibles valores (hexadecimales) son:</p> <ul style="list-style-type: none"> <li>0xE1: sistema de control separado del bus o inactivo.</li> <li>0xD2: el sistema de control espera a una configuración por parte del PROFIBUS DP Master.</li> <li>0xC3: el sistema de control intercambia datos cíclicamente con el PROFIBUS DP Master.</li> </ul> <p>Valor por defecto: 0xE1</p>
Slave Address	<p>Es la dirección de PROFIBUS DP Slave del sistema de control. Esta dirección la habrá configurado previamente el usuario con el PADT.</p> <p>Posibles valores (decimales) son:</p> <ul style="list-style-type: none"> <li>0–125: dirección del Slave</li> </ul>
Watchdog Time	Tiempo de WatchDog parametrizado en el Master, en ms. Véase el capítulo 6.6.3.

Tabla 160: Variables de sistema del PROFIBUS DP Slave

### 6.14.2 Propiedades

La ficha **Properties** del PROFIBUS DP Slave de HIMA contiene los siguientes parámetros para configurar el PROFIBUS DP Slave.

Los valores predefinidos para los parámetros “*Within one cycle*” y “*Process Data Refresh Rate [ms]*” permiten un rápido intercambio de datos de PROFIBUS DP entre el módulo COM (COM) y el hardware de PROFIBUS DP Slave del sistema de control HIMax/HIMatrix.

Estos parámetros solamente deberían modificarse cuando se necesite reducir la tasa de trabajo del módulo COM y ello sea admisible para el proceso.



Se recomienda que los parámetros los modifiquen sólo programadores experimentados.

Un aumento del tiempo de actualización del hardware COM/ PROFIBUS DP implica también un aumento del tiempo real de actualización de los datos de PROFIBUS DP. Deberán examinarse los requisitos de tiempo del equipo o la instalación.

Observe asimismo el parámetro **Min. Slave Interval [ms]** (véase la ficha “Timings” en el capítulo 6.3.2), el cual define el tiempo mínimo de actualización de los datos de PROFIBUS DP entre PROFIBUS DP Master y PROFIBUS DP Slave.

Elemento	Descripción
Type	PROFIBUS DP Slave
Name	Nombre del PROFIBUS DP Slave.
Module	Selección del módulo COM en el que se ejecuta este protocolo.
Activate Max. µP Budget	Activado: Aplicar el límite µP-Budget tomado del recuadro “Max. µP-Budget in [%]”.  Desactivado: No se usará límite de µP-Budget para este protocolo.
Max. µP budget in [%]	Máximo valor de µP-Budget del módulo que se permite que tenga lugar durante la ejecución del protocolo. Rango de valores: 1...100% Valor por defecto: 30%
Behavior on CPU/COM Connection Loss	En caso de cortarse la conexión del módulo procesador al módulo de comunicación, se inicializarán las variables de entrada o se usarán sin variar en el módulo procesador, según lo que se defina en este parámetro (p. ej. si se extrae el módulo de comunicación con la comunicación en curso).  Adopt initial data      Las variables de entrada se reinicializarán a sus valores iniciales. Retain Last Value      Las variables de entrada conservarán su último valor.
Station address	Dirección de estación del Slave. Se permite que en el bus exista una sola dirección de estación del Slave. Rango de valores: 1...125 Valor por defecto: 0
Interface	Interfaz de bus de campo que se desea usar para el PROFIBUS DP Slave. Rango de valores: fb1, fb2 Valor por defecto: No

Elemento	Descripción			
Baud Rate [bps]	Velocidad de transmisión a la que opera el bus. Posibles valores:			
	Valor	Baud Rate	FB1	FB2
	9600	9,6 kbit/s	X	X
	19200	16,2 kbit/s	X	X
	45450	45,45 kbit/s	X	X
	93750	93,75 kbit/s	X	X
	187500	187,5 kbit/s	X	X
	500000	500 kbit/s	X	X
	1500000	1,5 Mbit/s	X	X
	3000000	3 Mbit/s	X	-
	6000000	6 Mbit/s	X	-
	12000000	12 Mbit/s	X	-
Process Data Refresh Rate [ms]	Tiempo de actualización, en ms, para el intercambio de datos del protocolo entre COM y el hardware de PROFIBUS DP Slave. Rango de valores: 4...1000 Valor por defecto: 10			
Force Process Data Consistency	<p>Activado: Transferencia de todos los datos del protocolo desde CPU a COM dentro de un ciclo de la CPU.</p> <p>Desactivado: Transferencia de todos los datos del protocolo desde CPU a COM repartida a lo largo de varios ciclos de la CPU, con 1100 bytes por sentido de transmisión de datos. Con ello es posible reducir también el tiempo de ciclo.</p> <p>Valor por defecto: Activado</p>			

Tabla 161: Propiedades de Slave: Ficha “General”

## 6.15 Panel de control (PROFIBUS DP Slave)

El usuario podrá comprobar y controlar la configuración del PROFIBUS DP Slave en el panel de control. Además, allí se muestra información actual del estado (p. ej. tiempo de ciclo, estado del bus, etc.) del Slave.

**Para abrir el panel de control para supervisar el PROFIBUS DP Slave:**

1. Seleccione **Hardware** y en el menú contextual elija **Online**.
2. Escriba en **System Login** sus datos de acceso para abrir la vista en línea del hardware.
3. Haga doble clic en **COM Module** y seleccione **PROFIBUS DP Slave** en el árbol.

### 6.15.1 Recuadro de lectura (PROFIBUS Slave)

En el recuadro de lectura se mostrarán los siguientes valores del PROFIBUS DP Slave seleccionado.

Elemento	Descripción
Name	Nombre del PROFIBUS DP Slave
Fieldbus Interface	Interfaz de bus de campo asignada del Slave
Protocol State	Estado de conexión 0 = Desactivado 1 = Inactivo (intentando establecer conexión), 2 = Conectado
Error State	Véase el capítulo 6.14.1
TimeOut	Tiempo de WatchDog parametrizado en el Master, en ms. Véase el capítulo 6.6.3
WatchDog Time [ms]	Se ajusta en el Master. Véase el capítulo 6.6.3.
Fieldbus address	Véase el capítulo 6.14.2.
Master Address	Dirección del PROFIBUS DP Master.
Baud Rate [bps]	Velocidad de transmisión actual. Véase el capítulo 6.14.2.
μP budget (planned) [%]	Indicador de la tasa de trabajo del módulo COM planificada para este protocolo.
μP budget (actual) [%]	Tasa de trabajo real del módulo COM para este protocolo.

Tabla 162: Recuadro de lectura (PROFIBUS Slave)

## 6.16 Función del LED FBx en el PROFIBUS DP Slave

El LED FBx de la interfaz de bus de campo asignada señala el estado del protocolo PROFIBUS-DP. Los estados del LED FBx se representan en la siguiente tabla:

LED FBx	Color	Descripción
OFF	Amarillo	¡Protocolo PROFIBUS-DP inactivo! Es decir, el sistema de control se halla en estado STOP o no se ha configurado ningún PROFIBUS DP Slave.
Cadencia de parpadeo de 2 s	Amarillo	¡No hay intercambio de datos! PROFIBUS DP Slave configurado y listo.
ON	Amarillo	Protocolo PROFIBUS-DP activo y hay intercambio de datos con PROFIBUS DP Master.

Tabla 163: LED FBx

## 6.17 Función del LED FAULT en el PROFIBUS DP Slave (sólo HIMax)

El LED FAULT de la interfaz de bus de campo asignada señala el fallo del protocolo PROFIBUS-DP. Los estados del LED FAULT se representan en la siguiente tabla:

LED FAULT	Color	Descripción
OFF	Rojo	Protocolo PROFIBUS-DP sin fallos.
Parpadeo	Rojo	Protocolo perturbado. <ul style="list-style-type: none"> <li>▪ Configuración errónea de PROFIBUS DP Master y/o PROFIBUS DP Slave.</li> <li>▪ Tiempo de cálculo adjudicado excedido.</li> </ul> Si durante más de 5 segundos no se produce ningún evento de error, el indicador cambiará al estado <i>Protocol not disturbed</i> .

Tabla 164: FAULT FBx

## 7 Modbus

El acoplamiento Modbus de los sistemas de control HIMax/HIMatrix a prácticamente cualquier sistema centralizado o de monitorización puede realizarse a través de las interfaces RS485 o a través de las interfaces Ethernet de los sistemas de control. Los sistemas HIMax/HIMatrix pueden operar como Master y también como Slave.

La funcionalidad Modbus hace más fácil anexar paneles de mando u otros sistemas de control. Modbus es un sistema acreditado en la práctica por su amplia difusión y su uso en proyectos en todo el mundo.

Modbus Master (véase el capítulo 7.1)

La redundancia del Modbus Master deberá Ud. configurarla en el programa de usuario de forma que éste supervise las vías de transmisión redundantes y asigne los datos de proceso redundantemente transmitidos a la respectiva vía.

Modbus Slave (véase el capítulo 7.3)

El Modbus Slave puede configurarse redundantemente.



Si como canal de transmisión se usan interfaces Ethernet, el sistema de control HIMax/HIMatrix y el interlocutor de comunicación deberán encontrarse en la misma subred o, en caso de usarse un enrutador, tener registrado el enrutado correspondiente.

### 7.1 Topología de bus RS485

La siguiente figura ilustra una topología de bus RS485 con componentes HIMA. Como módulos de bus se usan H 7506. La longitud completa del bus será de 1200 m como máximo. Para distancias mayores deberán usarse repetidores, p. ej. H 7505<sup>1)</sup>. Podrá usarse un total de 3 repetidores. El bus podrá abarcar asimismo 4800 m como máximo.

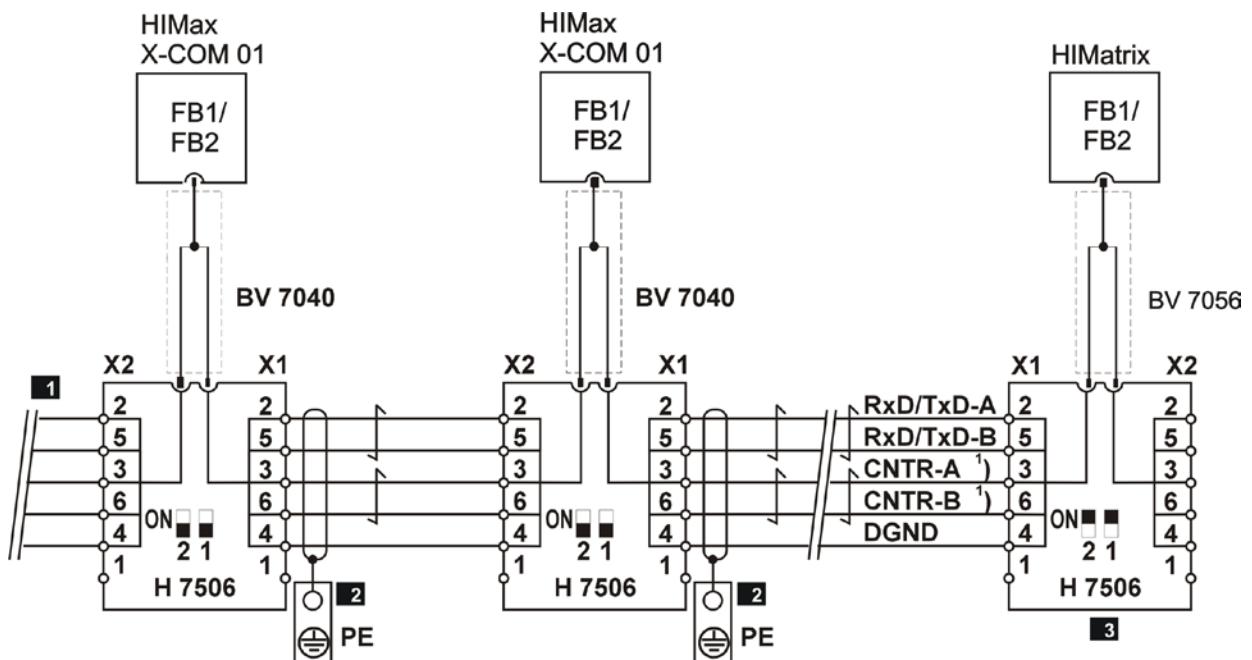


Fig. 57: Topología de bus RS485

<sup>1)</sup> Si se usan adaptadores de fibra óptica y RS485 en el bus, no se permite usar el elemento H7505 (sin cambio automático del sentido de transmisión de datos).

- 
- i** Si el bus abarca grandes distancias deberá implementarse un equipotencial.
- 

### 7.1.1 Asignación de bornes de H 7506

En la siguiente tabla se muestra la asignación de bornes del módulo de bus H 7506 de HIMA. El cable BV 7040 de HIMA conecta H 7506 a la interfaz de bus de campo FBx del sistema de control.

X1/X2	Color	Descripción
1	-	-
2	blanco	RxD/TxD-A, línea de datos
3	verde	CNTR-A, línea de control para repetidor
4	gris	DGND
5	marrón	RxD/TxD-B, línea de datos
6	amarillo	CNTR-B, línea de control para repetidor

Tabla 165: Asignación de bornes de H 7506

- 
- i** Más información sobre éste y otros componentes RS485 HIMA en la página web de HIMA.
- 

### 7.1.2 Cable de bus

Como cable de bus se recomienda usar un cable apantallado de dos hilos de par trenzado con las siguientes características:

Elemento	Descripción
Tipo de cable	LiYCY 3 x 2 x 0,25 mm <sup>2</sup>
Sección transversal de hilo	> 0,25 mm <sup>2</sup>
Impedancia característica	100...120 Ω

Tabla 166: Cable de bus

### 7.1.3 Características de transmisión por RS485

Elemento	Descripción
Topología de red	Bus lineal con terminación de bus activa en ambos extremos
Medio	Cable apantallado de dos hilos de par trenzado
Conector	Conector macho SUB-D de 9 polos. Para la asignación véase el capítulo 3.6.
Participantes de bus por cada segmento	32 participantes de bus en cada segmento sin repetidor <sup>1)</sup>
Participantes de bus en total	1 Modbus Master, 3 repetidores <sup>1)</sup> 121 Modbus Slaves
Máx. longitud de un segmento de bus	1200 m por cada segmento
Máx. longitud del bus	4800 m, 4 segmentos con 3 repetidores <sup>1)</sup>
Máx. velocidad de transmisión	HIMax: 38400 bit/s HIMatrix: 115000 bit/s

<sup>1)</sup> Por cada repetidor que se use se reducirá en 1 la máxima cantidad de participantes de bus en ese segmento. Es decir, en ese segmento podrán usarse 31 participantes de bus como máximo. Según la norma se permite un total de tres repetidores, lo que admite conectar un máximo de 121 Modbus Slaves por cada interfaz serie de un Modbus Master.

Tabla 167: Características de transmisión por RS485

## 7.2 HIMA Modbus Master

La transmisión de datos entre el Modbus Master de HIMA y los Modbus Slaves puede tener lugar tanto a través de una interfaz serie (RS485) como una TCP/UDP (Ethernet). Además, el Modbus Master de HIMA podrá usarse también como puerta de enlace (Gateway) (Modbus de TCP/UDP -> RS485).

### Equipamiento y sistema necesarios

Elemento	Descripción
Sistema de control HIMA	HIMax con módulo COM HIMatrix a partir de V7 de S.Op. de CPU y V12 de S.Op. de COM
Módulo de CPU	Las interfaces Ethernet del módulo procesador no pueden usarse para Modbus TCP.
Módulo COM	Ethernet 10/100BaseT Conexiones D-Sub FB1 y FB2 Si se usa Modbus RTU, cada módulo COM conectado a las interfaces serie de bus de campo (FB1 y/o FB2) deberá estar equipado con el submódulo opcional RS485 de HIMA. Para la asignación de interfaces véase el capítulo 3.6.
Activación	Cada una de ambas funciones Modbus Master deberá habilitarse individualmente. Véase el capítulo 3.4. Modbus Master RTU (RS 485) y Modbus Master TCP. Modbus Gateway necesitará la licencia Modbus Master RTU.

Tabla 168: Equipamiento y sistema necesarios para Modbus Master

### Propiedades de Modbus Master

Propiedad	Descripción
Modbus Master	Por cada módulo COM o por cada sistema de control HIMatrix se podrá configurar un Modbus Master. El Modbus Master podrá simultáneamente <ul style="list-style-type: none"> <li>- intercambiar datos con Slaves TCP/UDP</li> <li>- intercambiar datos con Slaves serie</li> <li>- y hacer de puerta de enlace (Gateway) de Modbus TCP a Modbus RTU.</li> </ul>
Max. number of Modbus slaves HIMax/HIMatrix	Un Modbus Master puede operar con hasta 247 Slaves. <ul style="list-style-type: none"> <li>- 121 Modbus Slaves por cada interfaz serie (FB1, FB2)</li> <li>- 64 TCP Slaves mediante conexión TCP/IP</li> <li>- 247 UDP Slaves mediante conexión UDP/IP</li> </ul> La máxima cantidad de UDP Slaves es limitada porque los Slaves deben administrarse por parte del Master.
Max. number of request telegrams	Podrán configurarse hasta 988 telegramas de requerimiento por cada Modbus Master.
Max. process data length for each request telegram	La longitud de datos de proceso en el caso de los telegramas de requerimiento específicos de HIMA es de 1100 bytes. Véase el capítulo 7.2.4.2.
Max. Size of Send Data	Véase Tabla 9 "Protocolos estándar"
Max. Size of Receive Data	<p><b>i</b> Los bytes de estado del Master y los bytes de estado de cada Slave asignado deberán restarse al máximo volumen de los datos de envío.</p>

Propiedad	Descripción				
Display format of the Modbus data	Los sistemas de control HIMax/HIMatrix utilizan el formato Big Endian. Ejemplo con datos de 32 bits (p. ej. DWORD, DINT):				
	Datos de 32 bits (hex)	0x12345678			
	Offset de memoria	0	1	2	3
	Big Endian (HIMax/HIMatrix)	12	34	56	78
	Middle Endian (H51q)	56	78	12	34
	Little Endian	78	56	34	12

Tabla 169: Propiedades de PROFIBUS DP Master

Según la norma se permite un total de tres repetidores, lo que admite un máximo de 121 Slaves por cada interfaz serie de un Master.

### 7.2.1 Ejemplo de Modbus

En este ejemplo tenemos un Modbus Master de HIMA que intercambia datos con un Modbus Slave de HIMA mediante Modbus TCP. Ambos sistemas de control se interconectan mediante las interfaces Ethernet de los módulos de comunicación.

- i** Si Modbus Master y Modbus Slave se encuentran en subredes diferentes, deberán registrarse en la tabla de enrute las correspondientes rutas definidas por el usuario.

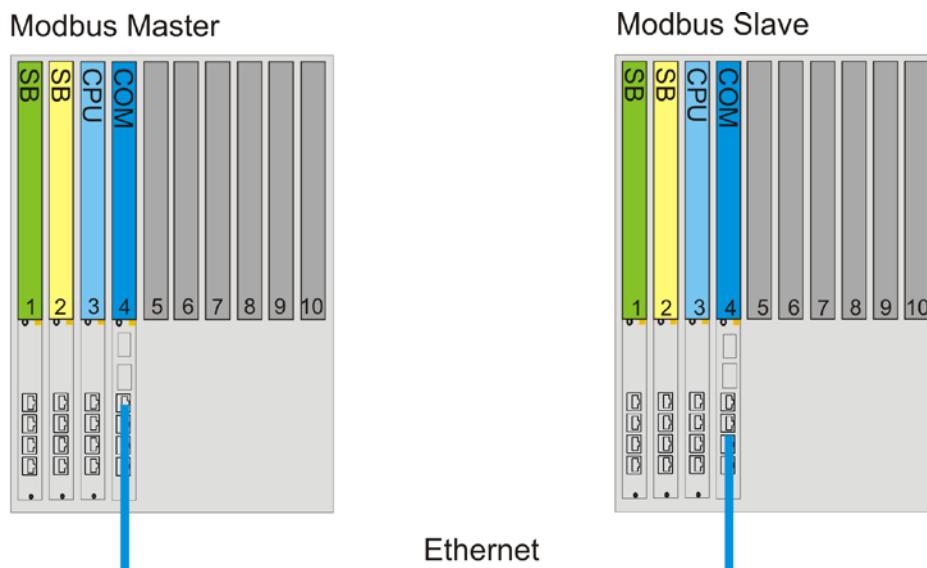


Fig. 58: Comunicación mediante Modbus TCP

Para este ejemplo, deberá Ud. crear las siguientes variables globales en SILworX:

Variables globales	Tipo
Master->Slave_BOOL_00	BOOL
Master->Slave_BOOL_01	BOOL
Master->Slave_BOOL_02	BOOL
Master->Slave_WORD_00	WORD
Master->Slave_WORD_01	WORD
Slave->Master_WORD_00	WORD
Slave->Master_WORD_01	WORD

### 7.2.1.1 Configuración del Modbus TCP Slave

**Para crear un nuevo Modbus Slave de HIMA:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Seleccione **New, Modbus Slave Set** en el menú contextual de los protocolos, para agregar un nuevo set Modbus Slave.
3. En el menú contextual del set Modbus Slave seleccione **Edit** y abra **Modbus Slave Set Properties**; conserve los valores predeterminados.
4. Seleccione la ficha **Modbus Slave** y realice los siguientes ajustes:
  - Seleccione “**COM Module**”
  - Active **Enable TCP**
  - Los demás parámetros conservarán sus valores predeterminados.

**Para configurar las variables de entrada de bit del Modbus Slave:**



En la ficha “Bit Variables” deberán introducirse las variables booleanas que el Master dirige por bits (código de función 1, 2, 5, 15).

1. Seleccione **Edit, Bit Variables** en el menú contextual del Modbus Slave.
2. En **Object Panel** seleccione las siguientes variables globales y arrástrelas con el ratón al área **Bit Inputs**.

Dirección de bit	Variable de bit	Tipo
0	Master->Slave_BOOL_00	BOOL
1	Master->Slave_BOOL_01	BOOL
2	Master->Slave_BOOL_02	BOOL

3. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Register Inputs** y seleccione **New Offsets**, para renombrar los offsets de las variables.

**Para configurar las variables de entrada de registro del Modbus Slave:**



En la ficha “Register Variables” deberán introducirse las variables que el Master dirige por registros (código de función 3, 4, 6, 16, 23).

1. Seleccione **Edit, Register Variables** en el menú contextual del Modbus Slave.
2. En **Object Panel** seleccione las siguientes variables y arrástrelas con el ratón al área **Register Inputs**.

Dirección de registro	Variables de registro	Tipo
0	Master->Slave_WORD_00	WORD
1	Master->Slave_WORD_01	WORD

3. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Register Inputs** y seleccione **New Offsets**, para renombrar los offsets de las variables.

**Para configurar las variables de salida de registro del Modbus Slave:**

1. Seleccione **Edit, Register Variables** en el menú contextual del Modbus Slave.
2. En **Object Panel** seleccione las siguientes variables y arrástrelas con el ratón al área **Register Outputs**.

Dirección de registro	Variables de registro	Tipo
0	Slave->Master_WORD_00	WORD
1	Slave->Master_WORD_01	WORD

3. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Register Outputs** y seleccione **New Offsets**, para renombrar los offsets de las variables.

**Para comprobar la configuración de Modbus TCP Slave:**

1. Abra el menú contextual del Modbus TCP Slave y seleccione **Verification**.
2. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.

### 7.2.1.2 Configuración del Modbus TCP Master

**Para crear el Modbus Master de HIMA:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Seleccione **New, Modbus Master** en el menú contextual de los protocolos para agregar un nuevo Modbus Master.
3. Seleccione **Properties, General** en el menú contextual de Modbus Master.
4. Seleccione **COM Module**.  
Los demás parámetros conservarán sus valores predeterminados.

**Para crear en Modbus Master la conexión al Modbus TCP Slave:**

1. Abra **Resource, Protocols, Modbus Master, Ethernet Slaves** en el árbol.
  2. Haga clic con el botón derecho en “**Ethernet Slaves**” y seleccione “**New**” en el menú contextual.
  3. Seleccione **TCP/UDP Slave** en la lista y confirme con **OK**.
  4. Configuración del TCP/UDP Slave en el Modbus Master:
    - Seleccione **Edit** para asignar las variables del sistema. Véase el capítulo 7.2.6.2.
    - Seleccione “**Properties**” para configurar las propiedades (véase el capítulo 7.2.6.3). En las propiedades de Slave introduzca la **IP address/dirección IP** del Slave TCP/UDP.
- Los demás parámetros conservarán sus valores predeterminados.

**Para configurar el telegrama de requerimiento de escritura de las variables de salida de bit:**

1. Haga clic con el botón derecho en **TCP/UDP Slave** y seleccione **New** en el menú contextual.
2. En la lista seleccione el telegrama de requerimiento “**Write Multiple Coils (15)**”.
3. Haga clic con el botón derecho en el telegrama de requerimiento **Write Multiple Coils (15)** y seleccione **Properties** en el menú contextual.
  - Como **start address of the write area**, escriba **0**.
4. Haga clic con el botón derecho en el telegrama de requerimiento **Write Multiple Coils (15)** y seleccione **Edit** en el menú contextual.
5. En **Object Panel** seleccione las siguientes variables y arrástrelas con el ratón a la ficha **Variables**.

Offset	Variables de bit	Tipo
0	Master->Slave_BOOL_00	BOOL
1	Master->Slave_BOOL_01	BOOL
2	Master->Slave_BOOL_02	BOOL

6. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Variables** y seleccione **New Offsets** para renombrar los offsets de las variables.

**Para configurar el telegrama de requerimiento de escritura de las variables de salida de registro:**

1. Haga clic con el botón derecho en “**TCP/UDP Slave**” y seleccione “**New**” en el menú contextual.
2. En la lista seleccione el telegrama de requerimiento **Write Multiple Coils (16)**.
3. Haga clic con el botón derecho en el telegrama de requerimiento **Write Multiple Registers (16)** y seleccione **Properties** en el menú contextual.
  - Como **start address of the write area**, escriba **0**.
4. Haga clic con el botón derecho en el telegrama de requerimiento **Write Multiple Registers (16)** y seleccione **Edit** en el menú contextual.
5. En **Object Panel** seleccione las siguientes variables y arrástrelas con el ratón a la ficha **Output Variables**.

Offset	Variables de registro	Tipo
0	Master->Slave_WORD_00	WORD
1	Master->Slave_WORD_01	WORD

6. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Variables** y seleccione **New Offsets** para renombrar los offsets de las variables.

**Para definir en Modbus Master el telegrama de requerimiento de lectura de variables de entrada:**

1. Haga clic con el botón derecho en “**TCP/UDP Slave**” y seleccione “**New**” en el menú contextual.
2. En la lista seleccione el telegrama de requerimiento “**Read Multiple Coils (03)**”.
3. Haga clic con el botón derecho en el telegrama de requerimiento **Read Holding Registers (03)** y seleccione **Properties** en el menú contextual.
  - Como **start address of the read area**, escriba **0**.
4. Haga clic con el botón derecho en el telegrama de requerimiento **Read Holding Registers (03)** y seleccione **Edit** en el menú contextual.
5. En **Object Panel** seleccione las siguientes variables y arrástrelas con el ratón a la ficha **Input Variables**.

Offset	Variables de registro	Tipo
0	Slave->Master_WORD_00	WORD
1	Slave->Master_WORD_01	WORD

6. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Input Variables** y seleccione **New Offsets** para renombrar los offsets de las variables.

**Para comprobar la configuración de Modbus TCP Master:**

1. Abra el menú contextual del Modbus TCP Master y seleccione **Verification**.

**Para comprobar la configuración de Modbus TCP Master:**

1. Abra el menú contextual del Modbus TCP Master y seleccione **Verification**.
2. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.

**Para crear el código para los sistemas de control**

1. Inicie el generador de códigos de Master y Slave Resource.
2. Asegúrese de que los códigos se hayan generado sin errores.
3. Cargue los respectivos códigos a los sistemas de control Slave y Master.

### 7.2.2 Ejemplo de direccionamiento alternativo de bit/registro

En este ejemplo se amplía la configuración del capítulo 7.2.1 en 16 variables booleanas en el área de registros. Las 16 variables booleanas se leerán con el telegrama de requerimiento **Write Multiple Coils (15)**. Véase también el capítulo 7.3.11.

**Para configurar las variables de entrada en el Modbus Slave:**

1. Seleccione **Edit, Register Variables** en el menú contextual del Modbus Slave.
2. En **Object Panel** seleccione las 16 nuevas variables booleanas y arrástrelas con el ratón al área **Register Inputs**.

Dirección de registro	Variables de registro	Tipo	
0	Master->Slave_WORD_00	WORD	
1	Master->Slave_WORD_01	WORD	
2	Master->Slave_BOOL_03 ..._18	BOOL	Agregar

3. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Register Inputs** y seleccione **New Offsets**, para renombrar los offsets de las variables.

**Para configurar en el Modbus Slave el direccionamiento alternativo de bit/registro:**

1. En el menú contextual de Master, seleccione **Edit, Offsets** y active **Use Alternative Register/Bit Addressing**.
2. En este ejemplo use los siguientes offsets para las áreas alternativas:

Register Area Offset Bits Input	1000
Register Area Offset Bits Output	1000
Bit Area Offset Register Input	8000
Bit Area Offset Register Output	8000



Para poder acceder con el telegrama de requerimiento de Modbus **Write Multiple Coils (15)** a variables booleanas en el área **Register Variables**, deberán estar reflejadas en espejo las variables en el área **Bit Variables**.

**Para configurar en Modbus Master el telegrama de requerimiento de escritura de las variables de salida (BOOL):**

1. Haga clic con el botón derecho en **TCP/UDP Slave** y seleccione **New** en el menú contextual.
2. En la lista seleccione el telegrama de requerimiento “**Write Multiple Coils (15)**”.
3. Haga clic con el botón derecho en el telegrama de requerimiento **Write Multiple Coils (15)** y seleccione **Properties** en el menú contextual.
  - Como **start address of the write area**, escriba **8032**.
4. Haga clic con el botón derecho en el telegrama de requerimiento **Write Multiple Coils (15)** y seleccione **Edit** en el menú contextual.
5. En **Object Panel** seleccione las siguientes variables y arrástrelas con el ratón a la ficha **Output Variables**.

Offset	Variables de registro reflejadas en espejo	Tipo
0 hasta 15	Master->Slave_BOOL_03..._18	BOOL

6. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Variables** y seleccione **New Offsets** para renombrar los offsets de las variables.

### 7.2.3 Funciones de menú de Modbus Master de HIMA

#### 7.2.3.1 Edit

El cuadro de diálogo **Edit** del Modbus Master contiene la siguiente ficha.

##### Variables de sistema

La ficha **System Variables** ofrece variables de sistema que permiten evaluar el estado del Modbus Master en el programa del usuario y controlar en base a ello el Modbus Master.

Elemento	Descripción
Slave Connection Error Count	Cantidad de las conexiones erróneas con Modbus Slaves que se hallen en estado activado. Aquí no se consideran los Modbus Slaves desactivados.
Modbus Master Activation Control	Sirve para detener o iniciar el Modbus Master desde el programa del usuario. 0: Activar 1: Desactivar (se comuta por flanco) (Modbus Master también podrá activarse mediante el PADT cuando se use <i>Modbus Master Activation Control = 1</i> ).
Modbus Master Bus Error	Error de bus en RS485, p. ej. error de telegrama (códigos desconocidos, etc.), error de longitud.
Modbus Master State	El estado de Modbus Master indica el estado momentáneo del protocolo.: 1: OPERATE 0: OFFLINE
Reset All Slave Errors	Con un cambio de FALSE a TRUE se resetean todos los errores de Slave y de bus.

Tabla 170: Variables de sistema Modbus Master

### 7.2.3.2 Propiedades

Con la función de menú **Properties** del menú contextual de Modbus Master se abre el cuadro de diálogo *Properties*.

Este cuadro de diálogo contiene las siguientes fichas:

#### Generalidades

En la ficha **General** se escriben el nombre y la descripción para el Modbus Master. Además, aquí se ajustarán los parámetros cuando el Modbus Master se desee usar también como puerta de enlace (Gateway) UDP y/o TCP.

Elemento	Descripción
Type	Modbus Master
Name	Nombre para el Modbus Master
Description	Descripción para el Modbus Master
Module	Selección del módulo COM en el que se ejecuta este protocolo.
Activate Max. µP-Budget	<p>Activado: Aplicar el límite µP-Budget tomado del recuadro “Max. µP-Budget in [%]”.</p> <p>Desactivado: No se usará límite de µP-Budget para este protocolo.</p>
Max. µP-Budget in [%]	Máximo valor de µP-Budget del módulo que se permite que tenga lugar durante la ejecución del protocolo.  Rango de valores: 1...100% Valor por defecto: 30%
Behavior on CPU/COM Connection Loss	En caso de cortarse la conexión del módulo procesador al módulo de comunicación, se inicializarán las variables de entrada o se usarán sin variar en el módulo procesador, según lo que se defina en este parámetro (p. ej. si se extrae el módulo de comunicación con la comunicación en curso).  Adopt initial data      Las variables de entrada se reinicializarán a sus valores iniciales. Retain Last Value      Las variables de entrada conservarán su último valor.
Enable TCP Gateway	Si se ha activado TCP Modbus Gateway, deberá configurarse como mínimo una interfaz RS485 de Modbus.
TCP-Server-Port	Estándar: 502 Podrán configurarse también otros puertos TCP. Habrá que observar la asignación de puertos en <i>Internet Corporation for Assigned Names and Numbers</i> (ICANN).
Maximum Number of TCP Connections Operating as Server	Máxima cantidad de conexiones TCP abiertas simultáneamente como servidor.  Rango de valores: 1...64 Valor por defecto: 5
Enable UDP Gateway	Si se ha activado UDP Modbus Gateway, deberá configurarse como mínimo una interfaz RS485 de Modbus
UDP-Port	Estándar: 502 Podrán configurarse también otros puertos UDP. Habrá que observar la asignación de puertos en “Internet Corporation for Assigned Names and Numbers” (ICANN).

Elemento	Descripción
Maximum length of the queue	Longitud de la cola de espera de Gateway para telegramas de requerimiento aún sin responder de otros Masters. Esto se tendrá en cuenta sólo si se tiene un Gateway activado.  Rango de valores: 1...20 Valor por defecto: 3

Tabla 171: Propiedades generales de Modbus Master

**CPU/COM**

Los valores predeterminados para los parámetros permiten un intercambio de datos de PROFIBUS DP lo más rápido posible entre el módulo COM y el módulo CPU en el sistema de control HIMax/HIMatrix.

Estos parámetros solamente deberían modificarse cuando se necesite reducir la tasa de trabajo de CPU y COM y ello sea admisible para el proceso.



Se recomienda que los parámetros los modifiquen sólo programadores experimentados. Un aumento del tiempo de actualización de COM y CPU implica también un aumento del tiempo real de actualización de los datos de Modbus. Deberán examinarse los requisitos de tiempo del equipo o la instalación.

Elemento	Descripción
Process Data Refresh Rate [ms]	Tiempo de actualización, en ms, para el intercambio de datos del protocolo entre COM y CPU.  Si para el tiempo de actualización <i>Process Data Refresh Rate [ms]</i> se ha elegido cero o un valor menor que el tiempo de ciclo del sistema de control, el intercambio de datos tendrá lugar lo más rápidamente posible.  Rango de valores: 0...(2 <sup>31</sup> - 1) Valor por defecto: 0
Force Process Data Consistency	Activado: Transferencia de todos los datos del protocolo desde CPU a COM dentro de un ciclo de la CPU.  Desactivado: Transferencia de todos los datos del protocolo desde CPU a COM repartida a lo largo de varios ciclos de la CPU, con 1100 bytes por sentido de transmisión de datos. Con ello es posible reducir también el tiempo de ciclo.  Valor por defecto: Activado

Tabla 172: Parámetros de COM/CPU

#### 7.2.4 Códigos de función Modbus del Master

Con los códigos de función Modbus (telegramas de requerimiento) dispondrá Ud. de la posibilidad de escribir y leer variables en ambos sentidos de transmisión. Podrán leerse o escribirse variables individuales o varias variables consecutivas.

**Para crear un nuevo telegrama de requerimiento para un Slave TCP/UDP:**

1. Seleccione en el árbol **Resource, Protocols, Modbus Master, Ethernet-Slaves** un **TCP/UDP Slave**.
2. Haga clic con el botón derecho en **TCP/UDP Slave** y seleccione **New** en el menú contextual.
3. En el cuadro de diálogo **New Object** seleccione un **Request Telegram**.

**Para crear un nuevo telegrama de requerimiento para un Slave Gateway:**

1. Seleccione en el árbol **Resource, Protocols, Modbus Master, Modbus Gateway** un **Gateway Slave**.
2. Haga clic con el botón derecho en **Gateway Slave** y seleccione **New** en el menú contextual.
3. En el cuadro de diálogo **New Object** seleccione un **Request Telegram**.

**Para crear un nuevo telegrama de requerimiento para un Modbus Slave RS485:**

1. Seleccione en el árbol **Resource, Protocols, Modbus Master, Serial Modbus** un **Modbus Slave**.
2. Haga clic con el botón derecho en “**Modbus Slave**” y seleccione “**New**” en el menú contextual.
3. En el cuadro de diálogo **New Object** seleccione un **Request Telegram**.

#### 7.2.4.1 Códigos de función Modbus estándar

Modbus Master de HIMA es compatible con los siguientes códigos de función estándar Modbus.

Elemento	Código	Tipo	Significado
READ COILS	01	BOOL	Lectura de varias variables (BOOL) desde el Slave.
READ DISCRETE INPUTS	02	BOOL	Lectura de varias variables (BOOL) desde el Slave.
READ HOLDING REGISTERS	03	WORD	Lectura de varias variables de cualquier tipo desde el Slave.
READ INPUT REGISTERS	04	WORD	Lectura de varias variables de cualquier tipo desde el Slave.
WRITE SINGLE COIL	05	BOOL	Escritura de una señal individual (BOOL) al Slave.
WRITE SINGLE REGISTER	06	WORD	Escritura de una señal individual (WORD) al Slave.
WRITE MULTIPLE COILS	15	BOOL	Escritura de varias variables (BOOL) al Slave.
WRITE MULTIPLE REGISTERS	16	WORD	Escritura de varias variables de cualquier tipo al Slave.
READ WRITE HOLDING REGISTERS	23	WORD	Escritura y lectura de varias variables de cualquier tipo al y desde el Slave.

Tabla 173: Códigos de función Modbus



Hallará más información sobre Modbus en la especificación *Modbus Application Protocol Specification* en [www.modbus.org](http://www.modbus.org).

#### 7.2.4.2 Códigos de función específicos de HIMA

Los códigos de función específicos de HIMA corresponden a los códigos de función estándar de Modbus. Las dos diferencias son la máxima longitud admisible de datos de proceso de 1100 bytes y el formato de cabecera de requerimiento y respuesta.

Elemento	Código	Tipo	Significado
Read Coils Extended	100 (0x64)	BOOL	Corresponde al código de función 01 Lectura de varias variables (BOOL) desde el área de importación o exportación <sup>1)</sup> del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Read Discrete Inputs Extended	101 (0x65)	BOOL	Corresponde al código de función 02 Lectura de varias variables (BOOL) desde el área de exportación del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Read Holding Registers Extended	102 (0x66)	WORD	Corresponde al código de función 03 Lectura de varias variables de cualquier tipo desde el área de importación o exportación <sup>1)</sup> del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Read Input Registers Extended	103 (0x67)	WORD	Corresponde al código de función 04 Lectura de varias variables de cualquier tipo desde el área de exportación del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Write Multiple Coils Extended	104 (0x68)	BOOL	Corresponde al código de función 15 Escritura de varias variables (BOOL) al área de importación del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Write Multiple Registers Extended	105 (0x69)	WORD	Corresponde al código de función 16 Escritura de varias variables de cualquier tipo al área de importación del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Read/Write Multiple Registers Extended	106 (0x6A)	WORD	Corresponde al código de función 23 Escritura y lectura de varias variables de cualquier tipo al y desde el área de importación o exportación del Slave. Longitud máxima de los datos de proceso: 1100 bytes (telegrama de requerimiento de Modbus Master) 1100 bytes (respuesta al Master).

### 7.2.4.3 Formato de la cabecera de requerimiento y respuesta

La cabecera de requerimiento y respuesta de los códigos de función Modbus específicos de HIMA se compone del siguiente modo:

Código	Requerimiento	Respuesta
100 (0x64)	Código de función de 1 byte 0x64 Dirección inicial de 2 bytes Cantidad de Coils 1...8800(0x2260) de 2 bytes	Código de función de 1 byte 0x64 Cantidad de 2 bytes de bytes = N Datos de Coil de N bytes (8 Coils se agrupan en un byte)
101 (0x65)	Código de función de 1 byte 0x65 Dirección inicial de 2 bytes Cantidad de entradas discretas de 2 bytes 1...8800(0x2260)	Código de función de 1 byte 0x65 Cantidad de 2 bytes de bytes = N Datos de entradas discretas de N bytes (8 entradas discretas se agrupan en un byte)
102 (0x66)	Código de función de 1 byte 0x66 Dirección inicial de 2 bytes Cantidad de registros de 2 bytes 1...550(0x226)	Código de función de 1 byte 0x66 Cantidad de 2 bytes de bytes = N Datos de registro de N bytes
103 (0x67)	Código de función de 1 byte 0x67 Dirección inicial de 2 bytes Cantidad de registros de 2 bytes 1...550(0x226)	Código de función de 1 byte 0x67 Cantidad de 2 bytes de bytes = N Datos de registro de N bytes
104 (0x68)	Código de función de 1 byte 0x68 Dirección inicial de 2 bytes Cantidad de Coils 1...8800(0x2260) de 2 bytes Cantidad de 2 bytes de bytes = N Datos de Coil de N bytes	Código de función de 1 byte 0x68 Dirección inicial de 2 bytes Cantidad de Coils 1...8800(0x2260) de 2 bytes
105 (0x69)	Código de función de 1 byte 0x69 Dirección inicial de 2 bytes Cantidad de registros de 2 bytes 1...550(0x226) Cantidad de 2 bytes de bytes = N Datos de registro de N bytes	Código de función de 1 byte 0x69 Dirección inicial de 2 bytes Cantidad de registros de 2 bytes 1...550(0x226)
106 (0x6A)	Código de función de 1 byte 0x6a Dirección inicial de lectura de 2 bytes Cantidad de registros de lectura de 2 bytes 1...550(0x226) Dirección inicial de escritura de 2 bytes Cantidad de registros de escritura de 2 bytes 1...550(0x226) Cantidad de 2 bytes de bytes de escritura = N Datos de registro de N bytes	Código de función de 1 byte 0x6a Cantidad de 2 bytes de bytes = N Datos de registro de N bytes

#### 7.2.4.4 Telegramas de requerimiento de lectura

Con los códigos de función “Read” pueden leerse variables desde el Slave.

Un telegrama de requerimiento del Modbus Master contiene, además de la función Modbus, la dirección inicial del área de lectura/escritura.

Para leer variables, el Modbus Master envía un telegrama de requerimiento de lectura *Read Request Telegram* al Modbus Slave.

El Modbus Slave responde al Modbus Master con un telegrama de respuesta que contiene las variables requeridas.

##### **Para configurar un telegrama de requerimiento de lectura:**

1. Seleccione en el árbol el telegrama de requerimiento **Request Telegram** a configurar.
2. Haga clic con el botón derecho en **Request Telegram** y seleccione **Edit** en el menú contextual.
3. En **Object Panel** seleccione una variable global que desee usar como variable de recepción Modbus y arrástrela con el ratón a una zona libre del área **Inputs Signals**.
4. Repita este paso para todas las demás variables de recepción Modbus.
5. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Inputs Signals** y seleccione **New Offsets** para renombrar los offsets de las variables.

**Se dispone de los siguientes telegramas de requerimiento de lectura *Read Request Telegrams*:**

##### **Read Coils (01) y Extended (100)**

Lectura de varias variables (BOOL) desde el Slave.

Elemento	Significado
Type	Función Modbus Read Coils
Name	Cualquier nombre inequívoco para la función Modbus
Description	Descripción de la función Modbus
Start address of the read area	0...65535

Tabla 174: Telegrama de requerimiento Read Coils

##### **Read Discrete Inputs (02) y Extended (101)**

Lectura de varias variables (BOOL) desde el Slave.

Elemento	Significado
Type	Función Modbus Read Discrete Inputs
Name	Cualquier nombre inequívoco para la función Modbus
Description	Descripción de la función Modbus
Start address of the read area	0...65535

Tabla 175: Telegrama de requerimiento Read Discrete Inputs

##### **Read Holding Registers (03) y Extended (102)**

Lectura de varias variables de cualquier tipo desde el Slave.

Elemento	Significado
Type	Función Modbus Read Holding Registers
Name	Cualquier nombre inequívoco para la función Modbus
Description	Descripción de la función Modbus
Start address of the read area	0...65535

Tabla 176: Telegrama de requerimiento Read Holding Registers

### Read Input Registers (04) y Extended (103)

Lectura de varias variables de cualquier tipo desde el Slave

Elemento	Significado
Type	Función Modbus Read Input Registers
Name	Cualquier nombre inequívoco para la función Modbus
Description	Descripción de la función Modbus
Start address of the read area	0...65535

Tabla 177: Telegrama de requerimiento Read Input Registers

#### 7.2.4.5 Telegrama de requerimiento de lectura y escritura

Para leer y escribir variables, el Modbus Master envía un telegrama de requerimiento de lectura o escritura *Read/Write Request Telegram* al Modbus Slave.

En primer lugar el Modbus Master escribe las variables de escritura definidas al área de importación del Modbus Slave.

A continuación el Modbus Master lee las variables de escritura definidas desde el área de exportación del Modbus Slave.



Las funciones de lectura y escritura son independientes una de la otra también en el telegrama *Read/Write Request Telegram*, aunque se envían conjuntamente en un telegrama de requerimiento.

No obstante, una aplicación frecuente del telegrama *Read/Write Request Telegram* es la relectura de las variables escritas del Modbus Master. Esto sirve para comprobar que las variables enviadas se hayan escrito correctamente.

#### Para configurar un telegrama de requerimiento de lectura y escritura:

1. Seleccione en el árbol el telegrama de requerimiento **Request Telegram** a configurar.
2. Haga clic con el botón derecho en **Request Telegram** y seleccione **Edit** en el menú contextual.

#### Para configurar las variables de lectura:

1. Seleccione en **Object Panel** una variable global que desee conectar a la nueva variable de recepción Modbus y arrástrela con el ratón a la columna **Global Variables** de las variables de recepción Modbus.
2. Repita este paso para todas las demás variables de recepción Modbus.
3. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Inputs Signals** y seleccione **New Offsets** para renombrar los offsets de las variables.

#### Para configurar las variables de escritura:

1. Seleccione en **Object Panel** una variable global que desee conectar a la nueva variable de envío Modbus y arrástrela con el ratón a la columna **Global Variables** de las variables de envío Modbus.
2. Repita este paso para todas las demás variables de envío Modbus.
3. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Outputs Signals** y seleccione **New Offsets** para renombrar los offsets de las variables.

### ReadWrite Holding Register (23) y Extended (106)

Escrutura y lectura de varias variables de cualquier tipo al y desde el área de importación del Slave.

Elemento	Significado
Type	Función Modbus <i>Read Write Holding Registers</i>
Name	Cualquier nombre inequívoco para la función Modbus
Description	Descripción de la función Modbus
Start address of the read area	0...65535
Start address of the write area	0...65535

Tabla 178: Register Read Write Holding

#### 7.2.4.6 Telegrama de requerimiento de escritura

Con los códigos de función “Write” se escriben variables sólo en el área de importación de un Slave.

Un telegrama de requerimiento del Modbus Master contiene, además de la función Modbus, la dirección inicial del área de lectura/escritura.

Para escribir variables, el Modbus Master envía un telegrama de requerimiento de escritura *Write Request Telegram* al Modbus Slave.

El Modbus Slave escribe las variables recibidas en su área de importación.

En el cuadro de diálogo *Variable Connections* de un *Write Request Telegram*/telegrama de requerimiento de escritura deberán insertarse las variables que el Modbus Master escribirá en el Modbus Slave.

#### Para configurar un telegrama de requerimiento de escritura:

1. Seleccione en el árbol el telegrama de requerimiento **Request Telegram** a configurar.
2. Haga clic con el botón derecho en **Request Telegram** y seleccione **Edit** en el menú contextual.
3. En **Object Panel** seleccione una variable global que desee usar como variable de envío Modbus y arrástrela con el ratón a una zona libre del área **Send Signals**.
4. Repita este paso para todas las demás variables de envío Modbus.
5. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Send Signals** y seleccione **New Offsets** para renombrar los offsets de las variables.

#### Se dispone de los siguientes telegramas de requerimiento de escritura *Write Request Telegrams*:

##### Write Multiple Coils (15) y Extended (104)

Escritura de varias variables (BOOL) al área de importación del Slave.

Elemento	Significado
Type	Función Modbus <i>Write Multiple Coils</i>
Name	Cualquier nombre inequívoco para la función Modbus
Description	Descripción de la función Modbus
Start address of the write area	0...65535

Tabla 179: Telegrama de requerimiento Write Multiple Coils

**Write Multiple Registers (16) y Extended (105)**

Escritura de varias variables de cualquier tipo al área de importación del Slave.

Elemento	Significado
Type	Función Modbus <i>Write Multiple Registers</i>
Name	Cualquier nombre inequívoco para la función Modbus
Description	Descripción de la función Modbus
Start address of the write area	0...65535

Tabla 180: Telegrama de requerimiento Write Multiple Registers

**Write Single Coil (05)**

Escritura de una variable individual (BOOL) al área de importación del Slave.

Elemento	Significado
Type	Función Modbus <i>Write Single Coil</i>
Name	Cualquier nombre inequívoco para la función Modbus
Description	Descripción de la función Modbus
Start address of the write area	0...65535

Tabla 181: Telegrama de requerimiento Write Single Coil (05)

**Write Single Register (06)**

Escritura de una variable individual (WORD) al área de importación del Slave.

Elemento	Significado
Type	Función Modbus <i>Write Single Register</i>
Name	Cualquier nombre inequívoco para la función Modbus
Description	Descripción de la función Modbus
Start address of the write area	0...65535

Tabla 182: Telegrama de requerimiento Write Single Register

### 7.2.5 Slaves Ethernet (Slaves TCP/UDP)

El Modbus Master puede comunicar con hasta 64 Slaves TCP/IP y 247 UDP/IP.

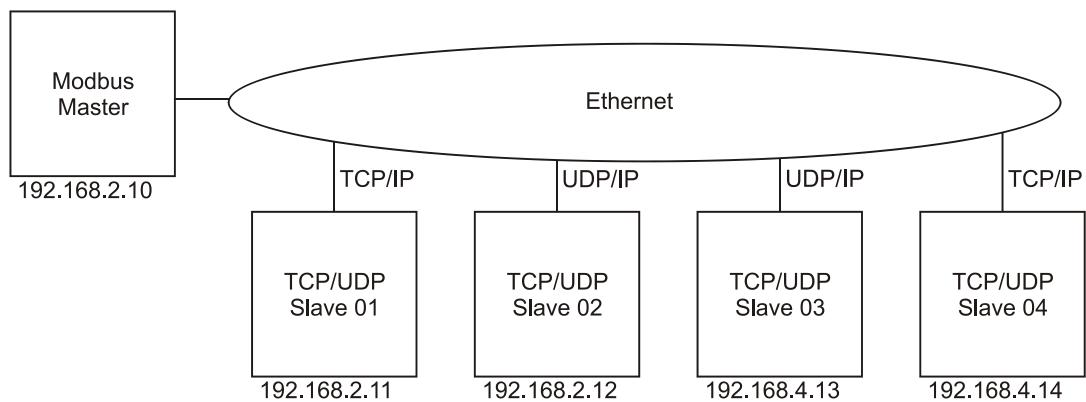


Fig. 59: Red Modbus

#### Para crear en Modbus Master una nueva conexión a un Slave TCP/UDP:

1. Abra **Resource**, **Protocols**, **Modbus Master**, **Ethernet-Slaves** en el árbol.
2. Haga clic con el botón derecho en “**Ethernet Slaves**” y seleccione “**New**” en el menú contextual.
3. Seleccione **TCP/UDP Slaves** en la lista y confirme con **OK**.
4. Para configurar el Slave TCP/UDP en el Modbus Master:  
**Edit** para asignar las variables de sistema; véase el capítulo 7.2.5.1.  
**Properties** para configurar las propiedades (véase el capítulo 7.2.5.2).



Si los Slaves TCP/UDP y el Modbus Master se encuentran en subredes diferentes, deberán registrarse en la tabla de enrutado las correspondientes rutas definidas por el usuario.

El Modbus TCP Master de HIMax/HIMatrix envía en sus telegramas al Modbus TCP Slave, además de la dirección IP, siempre una dirección de Modbus Slave (Unit Identifier), que es siempre FF (255).

### 7.2.5.1 Variables de sistema de los Slaves TCP/UDP

La ficha *System Variables* ofrece variables de sistema que permiten evaluar el estado del Slave TCP/UDP en el programa del usuario y controlarlo en base a ello.

El estado del Slave TCP/UDP puede evaluarse en el programa del usuario con las siguientes variables de estado:

Elemento	Descripción
Modbus Slave Activation Control	Sirve para desactivar o activar el Slave TCP/UDP desde el programa del usuario. 0: Activar 1: Desactivar (se conmuta por flanco) (Modbus Slave también podrá activarse también mediante el PADT cuando se use <i>Modbus Slave Activation Control = 1</i> ).
Modbus Slave Error	Código de error Los códigos de error 0x01...0x0b corresponden a los códigos de excepción de la especificación del protocolo Modbus. 0x00: Sin errores  Códigos de excepción: 0x01: Código de función no válido 0x02: Direcccionamiento no válido 0x03: Datos no válidos 0x04: (no se usa) 0x05: (no se usa) 0x06: Device Busy (sólo Gateway) 0x08: (no se usa) 0x0a: (no se usa) 0x0b: No hay respuesta desde el Slave (sólo Gateway)  Códigos específicos de HIMA: 0x10: Frame defectuoso recibido 0x11: Frame recibido con ID de transacción erróneo 0x12: Respuesta inesperada recibida 0x13: Respuesta recibida a través de una conexión errónea 0x14: Respuesta incorrecta a una orden de escritura  0xff: Time-out de Slave
Modbus Slave State	Estado de conexión del Slave TCP/UDP: 0: Desactivado 1: No conectado 2: Conectado

Tabla 183: Variables de sistema de Slaves TCP/UDP

### 7.2.5.2 Propiedades de Slaves TCP/UDP

Para configurar la conexión al Slave TCP/UDP deberán ajustarse los siguientes parámetros en el Modbus Master.

Elemento	Descripción
Type	Slave TCP/UDP
Name	Cualquier nombre inequívoco para el Slave TCP/UDP
Description	Cualquier descripción inequívoca para el Slave TCP/UDP
Master-Slave Data Exchange [ms]	Intervalo para el intercambio de datos con este Slave 1 hasta ( $2^{31} - 1$ ). Si no pudo contactarse con el Slave tras el <i>Maximal Number of Retries</i> /máximo número de reintentos, se cuadruplicará el intervalo de intercambio de datos entre <i>Master</i> y <i>Slave</i> .
TCP connection only on demand	Si se trata de un protocolo de transmisión TCP, aquí se definirá si se desea que la conexión a ese Slave se corte automáticamente tras cada intercambio de datos. TRUE: Cortar la conexión. FALSE: No cortar la conexión. Valor por defecto: FALSE
Receive Timeout [ms]	Time-out de recepción para ese Slave [ms]. Tras ese tiempo volverá a intentarse el envío.
IP Address	Dirección IP del Slave TCP/UDP
Port	Estándar: 502 Podrán configurarse también otros puertos TCP/UDP. Habrá que observar la asignación de puertos en “Internet Corporation for Assigned Names and Numbers” (ICANN).
Type of communication IP protocol	TCP o UDP Valor por defecto: TCP
Maximum Number of Resends	Máxima cantidad de repeticiones de envío cuando un Slave no responde. Podrá ajustarse cualquier cantidad de repeticiones del envío (0...65535). En TCP/IP es siempre cero y no es modificable. Se recomienda una cantidad de repeticiones entre cero y ocho.

Tabla 184: Parámetros de configuración

### 7.2.6 Modbus Gateway (TCP/UDP Gateway)

Para que el Modbus Master pueda operar como Modbus Gateway se necesitará la licencia "Modbus Master RTU". En este modo, los requerimientos del Master que la puerta Gateway reciba a través de Ethernet se reenviarán a los Slaves RS485 y/o Ethernet conectados a la puerta Gateway. Igualmente, las respuestas de los Slaves se reenviarán al Modbus Master a través de la puerta Gateway.

Mediante la interfaz serie podrán direccionarse hasta 121 Modbus Slaves serie.

El rango de direcciones de Slave es de 1 a 247. El Modbus Master 2 (Modbus Gateway) necesitará una licencia de Modbus Master, aun cuando sólo se use el Modbus Gateway.

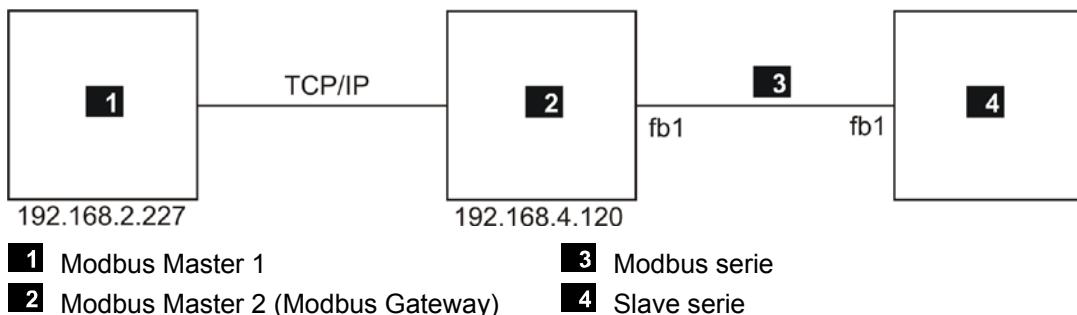


Fig. 60: Modbus Gateway



Si Modbus Gateway y Modbus Master se encuentran en subredes diferentes, deberán registrarse en la tabla de enrutado las correspondientes rutas definidas por el usuario.

#### Modbus Master 1:

##### Para crear en Modbus Master 1 la conexión al Modbus Gateway:

1. Abra **Resource, Protocols, Modbus Master** en el árbol.
2. Haga clic con el botón derecho en **Modbus Master** y seleccione **New** en el menú contextual.
3. Seleccione **Modbus Gateway** en la lista y confirme con **OK**.
4. Configuración del Modbus Gateway en el Modbus Master 1:  
**Properties** para configurar las propiedades; véase el capítulo 7.2.7.3.  
 Escriba la **IP Address/dirección IP** en las propiedades del Modbus Master 2 (Modbus Gateway).

##### Para crear en Modbus Master 1 la conexión al Gateway Slave:

En el Modbus Master 1 deberá crearse el Slave serie como Gateway Slave.

1. Abra **Resource, Protocols, Modbus Master, Modbus Gateway** en el árbol.
2. Haga clic con el botón derecho en "**Modbus Gateway**" y seleccione "**New**" en el menú contextual.
3. Seleccione **Gateway Slave** en la lista y confirme con **OK**.
4. Para configurar el Gateway Slave en el Modbus Master 1:  
**Edit** para asignar las variables de sistema; véase el capítulo 7.2.6.2.  
**Properties** para configurar las propiedades; véase el capítulo 7.2.6.3.  
 En las propiedades de Slave introduzca la **serial address/dirección serie** del Gateway Slave.

**Para definir en Modbus Master 1 las variables de entrada y salida al Slave serie:**

1. Haga clic con el botón derecho en **Gateway Slave** y seleccione **New** en el menú contextual.
2. Seleccione de la lista el telegrama de requerimiento **Request Telegram** necesario.
3. Haga clic con el botón derecho en el respectivo **Request Telegram** y seleccione **Edit** en el menú contextual. Introduzca las variables de entrada o salida en la ficha **Process Variables**.

**Modbus Master 2 (Modbus Gateway):**

En las propiedades de Modbus Master 02 deberá activarse la función Gateway. Para ello se conectarán los Gateway Slaves configurados en el Master 01 a los Slaves serie.

**Para activar la función Gateway en Modbus Master 2:**

1. Abra **Resource, Protocols, Modbus Master** en el árbol.
2. Haga clic con el botón derecho en **Modbus Master** y seleccione **Properties** en el menú contextual.
3. Active el parámetro **Enable TCP Gateway** para que Modbus Master pueda operar además como Gateway TCP.
4. Active el parámetro **Enable UDP Gateway** para que pueda operar además como Gateway UDP.

**Para configurar el Modbus serie en el Modbus Master 2:**

1. Abra **Resource, Protocols, Modbus Master** en el árbol.
2. Haga clic con el botón derecho en **Modbus Master** y seleccione **New** en el menú contextual.
3. Seleccione **Serial Modbus** en la lista y confirme con **OK**.
4. Para configurar seleccione **Modbus Serie Properties** y defina la interfaz, la velocidad de transmisión, etc.

**Para configurar en el Modbus Master 2 la conexión al Slave serie:**

1. Abra **Resource, Protocols, Modbus Master, Serial Modbus** en el árbol.
2. Haga clic con el botón derecho en **Serial Modbus** y seleccione **New** en el menú contextual.
3. Seleccione **Modbus Slave** en la lista y confirme con **OK**.
4. Para configurar seleccione **Modbus Slave Properties** e introduzca la dirección **Slave Address** y del Slave serie.

**Slave serie****Para configurar el Modbus Slave serie:**

1. Abra **Resource, Protocols, Modbus Slave** en el árbol.
2. Haga clic con el botón derecho en **Modbus Slave** y seleccione **Edit** en el menú contextual.
3. Para configurar seleccione **Modbus Slave Properties** e introduzca la dirección **Slave Address** del Slave serie.

### 7.2.6.1 Propiedades de Gateway

Modbus Master comunica con sus Modbus Slaves a través del Modbus Gateway.

Para configurar la conexión al Modbus Gateway deberán ajustarse los siguientes parámetros en el Modbus Master.

Elemento	Descripción
Type	Modbus Gateway
Name	Cualquier nombre inequívoco para la puerta Gateway
Description	Cualquier descripción inequívoca para el Slave TCP/UDP
Communication IP Protocol	TCP o UDP Valor por defecto: TCP
IP address	dirección IP de la puerta Gateway a través de la cual se desea que Modbus Master comunique con su Modbus Slave. Valor por defecto: (0.0.0.0)
Port	Valor por defecto: 502

Tabla 185: Parámetros de conexión de Modbus Gateway

### 7.2.6.2 Variables de sistema de Gateway Slave

En el editor se dispone de tres variables de estado:

Elemento	Descripción
Modbus Slave Activation Control	Sirve para desactivar o activar el Gateway Slave desde el programa del usuario. 0: Activar 1: Desactivar (se conmuta por flanco) (Modbus Slave también podrá activarse también mediante el PADT cuando se use <i>Modbus Slave Activation Control = 1</i> ).
Modbus Slave Error	Parámetros como Slave TCP/UDP; véase el capítulo 7.2.5.1.
Modbus Slave State	Estado de conexión del Gateway Slave: 0: Desactivado 1: No conectado 2: Conectado

Tabla 186: Variables de estado de Gateway Slave

### 7.2.6.3 Propiedades de Gateway Slave

Para configurar la conexión al Gateway Slave deberán ajustarse los siguientes parámetros en el Modbus Master.

Elemento	Descripción
Type	Gateway Slave
Name	Cualquier nombre inequívoco para el Gateway Slave
Description	Cualquier descripción inequívoca para el Gateway Slave
Slave Address	1...247
Parámetros restantes como Slave TCP/UDP, véase el capítulo 7.2.5.2	

Tabla 187: Parámetros de conexión de Gateway Slave

### 7.2.7 Modbus serie

Modbus Master puede comunicar con hasta 247 Slaves serie. Según la norma se permite un total de tres repetidores, lo que admite un máximo de 121 participantes de bus por cada interfaz serie de un Master.

- 
- i** Asignación de pins de las conexiones D-Sub (fb1, fb2) del módulo X-COM, véase el capítulo 3.6.
- 

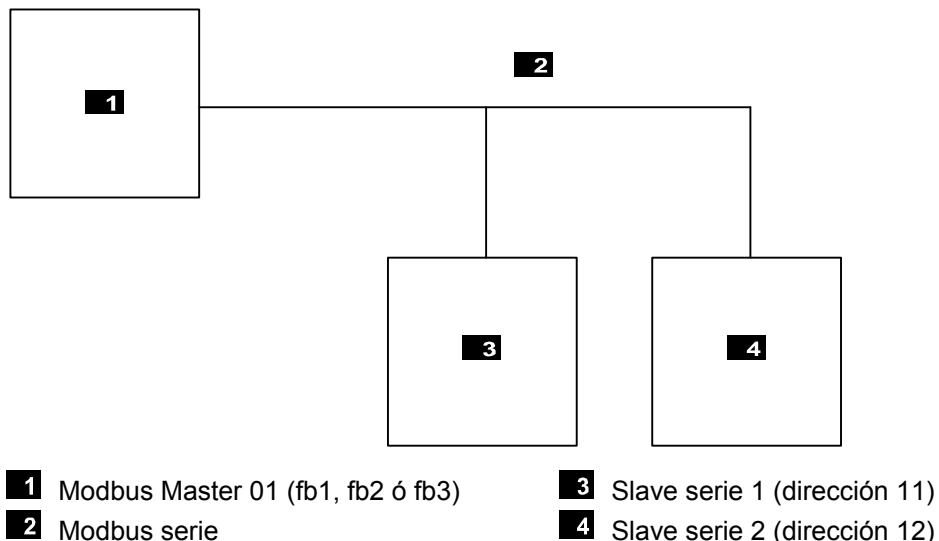


Fig. 61: Modbus serie

El Modbus Master de HIMA es compatible con la transmisión de datos en formato RTU (Remote Terminal Unit).

El frame de telegrama RTU comienza y finaliza con los caracteres idle especificados por el usuario (valor por defecto: 5 caracteres idle).

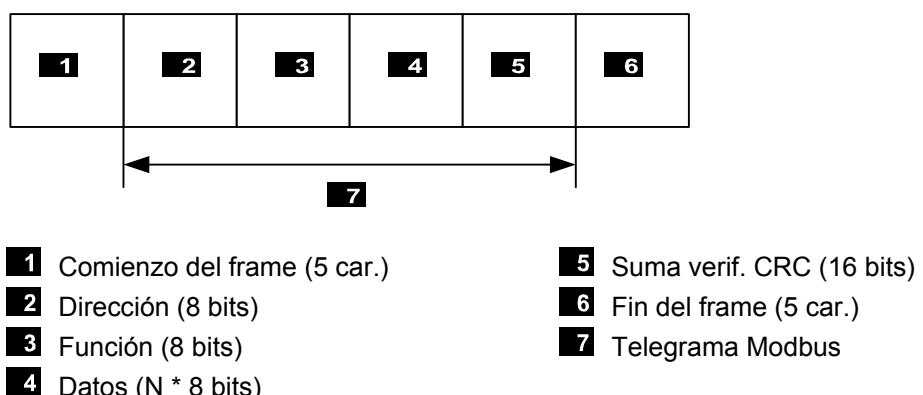


Fig. 62: Telegrama Modbus

**Para crear un Modbus serie en el Modbus Master:**

1. Abra **Resource, Protocols, Modbus Master, Serial Modbus** en el árbol.
2. Haga clic con el botón derecho en **Serial Modbus** y seleccione **New** en el menú contextual.
3. Seleccione **Modbus Slave** en la lista y confirme con **OK**.
4. Para configurar el Modbus Slave en el Modbus Master:  
**Edit** para asignar las variables de sistema; véase el capítulo 7.2.7.2.  
 Seleccione **Properties** para configurar las propiedades (véase el capítulo 7.2.7.3).

**7.2.7.1 Propiedades del Modbus serie**

Para configurar el Modbus serie deberán ajustarse los siguientes parámetros en el Modbus Master.

Elemento	Descripción
Type	Modbus serie
Name	El nombre del Modbus serie lo puede elegir el usuario
Description	Cualquier descripción inequívoca para el Modbus serie
Interface	La interfaz de bus de campo que se desea usar para el Modbus Master (fb1, fb2).
Baud rate [bps]	Velocidad de transmisión para RS485. Valores posibles: 300 bit/s 600 bit/s 1200 bit/s 2400 bit/s 4800 bit/s 9600 bit/s 19200 bit/s 38400 bit/s 57600 bit/s (máxima velocidad con HIMax a partir de V4) 62500 bit/s (sólo HIMatrix) 76800 bit/s (sólo HIMatrix) 115000 bit/s (sólo HIMatrix)
Parity	No Impar Par Valor por defecto: Par
Stop Bits	Estándar (adapta la cantidad de los bits de parada a la paridad: con paridad = 1 bit de parada; sin paridad 2 bits de parada) Un bit de parada Dos bits de parada Valor por defecto: Estándar
Number of Idle Chars	La cantidad de caracteres idle al principio y al final del frame del telegrama RTU. Rango de valores: 0...65535 Valor por defecto: 5 caracteres

Tabla 188: Parámetros de Modbus Master serie

### 7.2.7.2 Variables de sistema Modbus Slave

En el editor Edit se dispone de tres variables de estado (variables de sistema).

Elemento	Descripción
Modbus Slave Activation Control	Activación/desactivación de Modbus Slave en el programa del usuario. 0: Activar 1: Desactivar (se conmuta por flanco) (Modbus Slave también podrá activarse también mediante el PADT cuando se use <i>Modbus Slave Activation Control = 1</i> ).
Modbus Slave Error	Parámetros como Slave TCP/UDP, capítulo 7.2.5.2.
Modbus Slave State	Estado de conexión del Modbus Slave: 0: Desactivado 1: No conectado 2: Conectado

Tabla 189: Variables de sistema Modbus Slave

### 7.2.7.3 Propiedades de Modbus Slave

Para configurar la conexión a los Slaves serie deberán ajustarse los siguientes parámetros en el Modbus Master.

Elemento	Descripción
Type	Modbus Slave
Name	Nombre del Modbus Slave, a elegir por el usuario
Description	Cualquier descripción inequívoca para el Modbus Slave
Slave Address	1...247
Parámetros restantes como Slave TCP/UDP, véase el capítulo 7.2.5.2.	

Tabla 190: Parámetros de conexión de Modbus Master



El valor de “Receive Timeout” dependerá en el Modbus Slave serie de la velocidad de transmisión que se haya configurado.

A velocidades de 19200 [bit/s] o mayores, podrá usarse el valor predefinido de “Receive Timeout”. A menores velocidades que 19200 [bit/s] deberá aumentarse el valor de “Receive Timeout”.

### 7.2.8 Panel de control (Modbus Master)

El usuario podrá comprobar y controlar la configuración del Modbus Master en el panel de control. Además, allí se muestra información actual de estado del Master (p. ej. estado del Master).

**Para abrir el panel de control para supervisar el Modbus Master:**

1. Seleccione **Hardware** y en el menú contextual elija **Online**.
2. Escriba en **System Login** sus datos de acceso para abrir la vista en línea del hardware.
3. Haga doble clic en **COM Module** y seleccione **Modbus Master** en el árbol.

### 7.2.8.1 Menú contextual (Modbus Master)

En el menú contextual del Modbus Master seleccionado podrán elegirse los siguientes comandos:

#### **Offline**

Con este comando se detendrá el Modbus Master.

#### **Operate**

Con este comando se iniciará el Modbus Master.

#### **Reset statistical data**

Los datos estadísticos (p. ej. cantidad de errores de bus, tiempo de ciclo mín./máx., etc.) se pondrán a cero.

### 7.2.8.2 Recuadro de lectura (Modbus Master)

En el recuadro de lectura se mostrarán los siguientes valores del Modbus Master seleccionado.

Elemento	Descripción
Name	Nombre del Modbus Master
Master State	El estado de Modbus Master indica el estado momentáneo del protocolo.: OPERATE OFFLINE
Bus Error Count	Contador de errores del bus
Disturbed Connections	Contador de las conexiones perturbadas
µP load (planned)	Véase el capítulo 7.2.3.2
µP load (actual)	

Tabla 191: Recuadro de lectura de Modbus Master

### 7.2.9 Panel de control (Modbus Master -> Slave)

El usuario podrá comprobar y activar/desactivar la configuración del interlocutor de comunicación del Modbus Master en el panel de control. Además, allí se muestra información actual de estado (p. ej. estado del Slave, etc.) del interlocutor de comunicación.

#### **Para abrir el panel de control para supervisar la conexión de Modbus:**

1. Seleccione **Hardware** y en el menú contextual elija **Online**.
2. Escriba en **System Login** sus datos de acceso para abrir la vista en línea del hardware.
3. Haga doble clic en **COM Module** y seleccione **Modbus Master, Slave** en el árbol.

### 7.2.10 Función del LED FBx en el Modbus Master

El LED FBx de la interfaz de bus de campo asignada señaliza el estado del protocolo Modbus. Los estados del LED FBx se representan en la siguiente tabla:

LED FBx	Color	Descripción
OFF	Amarillo	¡Protocolo Modbus Master inactivo! Es decir, el sistema de control se halla en estado STOP o no se ha configurado ningún Modbus Master.
Parpadeo	Amarillo	Protocolo Modbus Master activo y hay intercambio de datos con los Modbus Slaves.

Tabla 192: LED FBx

### 7.2.11 Función del LED FAULT en el Modbus Master (sólo HIMax)

El LED FAULT de la interfaz de bus de campo asignada señaliza un fallo del protocolo Modbus. Los estados del LED FAULT se representan en la siguiente tabla:

LED FAULT	Color	Descripción
OFF	Rojo	Protocolo Modbus Master sin fallos.
Parpadeo	Rojo	<p>Los siguientes eventos dan lugar a fallos.</p> <ul style="list-style-type: none"> <li>▪ Respuesta errónea o mensaje de error recibido del Slave</li> <li>▪ Time-out de uno o más Slaves</li> <li>▪ Tiempo de cálculo adjudicado excedido</li> </ul> <p>Si durante más de 5 segundos no se produce ningún evento de error, el indicador cambiará al estado “Protocol not disturbed”.</p>

Tabla 193: FAULT FBx

### 7.3 Modbus Slave HIMA

El Modbus Slave HIMA puede atender simultáneamente a varios Modbus Master a través de la interfaz serie (RS485) y la interfaz Ethernet (TCP/UDP).

#### Equipamiento y sistema necesarios

Elemento	Descripción
Sistema de control HIMA	HIMax con módulo COM HIMatrix a partir de V7 de S.Op. de CPU y V12 de S.Op. de COM
Processor module	Las interfaces Ethernet del módulo procesador no pueden usarse para Modbus TCP.
COM module	Ethernet 10/100BaseT Conexiones D-Sub FB1 y FB2 Si se usa Modbus RTU, cada módulo COM conectado a las interfaces serie de bus de campo (FB1 y/o FB2) deberá estar equipado con el submódulo RS485 de HIMA. Para la asignación de interfaces véase el capítulo 3.6.
Activation	Cada una de ambas funciones Modbus Slave deberá habilitarse individualmente. Véase el capítulo 3.4. Modbus Slave RTU (RS485) Modbus Slave TCP Para el Modbus Slave redundante se necesitarán 2 licencias, una por cada módulo X-COM.

Tabla 194: Equipamiento y sistema necesarios para Modbus Slave HIMA

#### Modbus Slave (propiedades)

Elemento	Descripción																						
Modbus slave	Para cada módulo COM podrá configurarse un Modbus Slave.																						
Redundancy	¡Sólo para HIMax! En un sistema sistema HIMax podrá usarse un máximo de 10 pares de módulos de comunicación Modbus Slave redundantes. Mientras un par de módulos de comunicación Modbus Slave funcione redundantemente, se intercambiarán los mismos datos de entrada y salida por ambos módulos de comunicación con el Modbus Master. Véase el capítulo 7.3.3.																						
Number of Master Accesses	RTU: La técnica de transmisión por RS485 permite que sólo un Modbus Master acceda al Slave. TCP: Como máximo podrán acceder 20 Modbus Masters al Slave. UDP: Al Slave podrá acceder una cantidad ilimitada de Modbus Masters.																						
Max. Size of Send Data	Véase Tabla 9 “Protocolos estándar”																						
Max. Size of Receive Data																							
Display format of the Modbus data	Los sistemas de control HIMax/HIMatrix utilizan el formato Big Endian. Ejemplo con datos de 32 bits (p. ej. DWORD, DINT): <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Datos de 32 bits (hex)</td> <td>0x12345678</td> </tr> <tr> <td>Offset de memoria</td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>Big Endian (HIMax/HIMatrix)</td> <td>12</td> <td>34</td> <td>56</td> <td>78</td> </tr> <tr> <td>Middle Endian (H51q)</td> <td>56</td> <td>78</td> <td>12</td> <td>34</td> </tr> <tr> <td>Little Endian</td> <td>78</td> <td>56</td> <td>34</td> <td>12</td> </tr> </table>	Datos de 32 bits (hex)	0x12345678	Offset de memoria	0	1	2	3	Big Endian (HIMax/HIMatrix)	12	34	56	78	Middle Endian (H51q)	56	78	12	34	Little Endian	78	56	34	12
Datos de 32 bits (hex)	0x12345678																						
Offset de memoria	0	1	2	3																			
Big Endian (HIMax/HIMatrix)	12	34	56	78																			
Middle Endian (H51q)	56	78	12	34																			
Little Endian	78	56	34	12																			

Tabla 195: Propiedades de Modbus Slave

### 7.3.1 Configuración del Modbus TCP Slave

**Para crear un nuevo Modbus Slave de HIMA:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Seleccione **New, Modbus Slave Set** en el menú contextual de los protocolos, para agregar un nuevo set Modbus Slave.
3. En el menú contextual del set Modbus Slave seleccione **Edit** y abra **Modbus Slave Set Properties**; conserve los valores predeterminados.
4. Seleccione la ficha **Modbus Slave** y realice los siguientes ajustes:
  - Seleccione **COM Module**
  - Active **Enable TCP**
  - Los demás parámetros conservarán sus valores predeterminados.



En el capítulo 7.2.1 se describe un ejemplo de configuración de la conexión de un Modbus TCP Slave HIMA a un Modbus TCP Master HIMA.

---

### 7.3.2 Configuración del Modbus TCP Slave redundante

**Para crear un Modbus Slave HIMA redundante:**

1. Abra **Configuration, Resource, Protocols, Modbus Slave Set** en el árbol.
2. En el menú contextual del set Modbus Slave seleccione **Edit**, abra **Modbus Slave Set Properties** y realice los siguientes ajustes:
  - Active **Set Redundant Operation**.
  - La ficha **Modbus Slave Redundant** se agregará automáticamente
3. Seleccione la ficha **Modbus Slave Redundant** y realice los siguientes ajustes:
  - Seleccione **COM Module**
  - Active **Enable TCP**

Los demás parámetros conservarán sus valores predeterminados.



Las variables de envío y recepción asignadas en “Modbus Slave Set” tienen validez para ambos Modbus Slaves.

---

### 7.3.3 Reglas para el Modbus TCP Slave redundante

Para el funcionamiento redundante de módulos de comunicación Modbus Slave HIMax se recomienda una configuración de sistema redundante de HIMax. Véase el manual del sistema HI 801 141 S.

De lo contrario, ya desde la aparición de un primer error en el sistema HIMax no podrá asegurarse de que los pares de módulos de comunicación Modbus Slave se comporten coherentemente respecto a sus interlocutores externos (Modbus Master).

#### 7.3.3.1 Slots admisibles de los módulos COM Modbus Slave redundantes

Para minimizar el riesgo de colisiones de datos en el bus de sistema HIMax, deberán considerarse los segmentos de bus de sistema (1 a 3) del rack. Por tanto, los módulos de comunicación Modbus Slave redundantes deberían colocarse sólo en el mismo segmento de un rack en los siguientes slots:

Segment	Slot
1	3...6 (siempre que aquí no se provean módulos procesadores)
2	7...14
3	15...18

Tabla 196: Slots admisibles de los módulos COM Modbus Slave redundantes

#### 7.3.3.2 Módulos COM Modbus Slave redundantes en diferentes racks

No se permite usar más de dos pares de módulos de comunicación Modbus Slave redundantes cuyos módulos de comunicación Modbus Slave redundantes se encuentren en diferentes racks (0 a 15).

Los módulos de comunicación Modbus Slave redundantes deberán encontrarse entonces en racks contiguos respectivamente.

Además podrán usarse 8 pares más de módulos de comunicación Modbus Slave en el mismo sistema HIMax de acuerdo a las reglas del capítulo 7.3.3.1.

#### NOTA



**¡Posibilidad de disfunciones!**

**¡Use slots para módulos de comunicación Modbus Slave redundantes sólo de acuerdo a las reglas citadas!**

**Entre un módulo X-COM y los módulos X-CPU se permite un tiempo adicional (debido a la longitud del cable, switches, etc.) de 50 µs como máximo.**

**Recomendación: Use los módulos X-COM lo más cerca posible de los módulos X-CPU (p. ej. rack 0, rack 1)**

### 7.3.4 Funciones de menú del set Modbus Slave de HIMA

Con la función de menú “Edit” del menú contextual del set de Modbus Slave se abre el cuadro de diálogo *Modbus Slave Set Properties*. Este cuadro de diálogo contiene las siguientes fichas:

#### 7.3.4.1 Propiedades del set Modbus Slave

En la ficha *Modbus Slave Set Properties* se ajustan los siguientes parámetros para el Modbus Slave.

Elemento	Descripción				
Name	Nombre del set de Modbus Slave				
Activate Max. $\mu$ P-Budget	<p>Activado: Aplicar el límite <math>\mu</math>P-Budget tomado del recuadro <i>Max. <math>\mu</math>P-Budget in [%]</i>.</p> <p>Desactivado: No se usará límite de <math>\mu</math>P-Budget para este protocolo.</p>				
Max. $\mu$ P-Budget in [%]	Máxima carga de $\mu$ P del módulo COM que se permite que tenga lugar al ejecutar el protocolo. Rango de valores: 1...100% Valor por defecto: 30%				
Set Redundant Operation	Activado: Modo redundancia Desactivado: Modo mono Valor por defecto: Desactivado				
Max. Response Time [ms]	Período de tiempo tras la recepción de un requerimiento, dentro del cual el Modbus Slave podrá responder aún. <b>En los sistemas de control HIMatrix este valor deberá ser 0.</b> Rango de valores: 0...(2 <sup>31</sup> - 1) ms Valor por defecto: 5000 ms (0 = sin limitación)				
Area for reading function codes 1, 3, 100, 102	Este parámetro define desde qué área de datos se leerán los datos para el código de función 1, 3, 100, 102. Rango de valores: <ul style="list-style-type: none"> <li>▪ Área de importación</li> <li>▪ Área de exportación (compatible con 51q)</li> </ul>				
Area for reading function codes 23, 106	Aquí el usuario podrá definir el área del Modbus Slave desde la que se leerá el código de función 23. Import area: El Master tiene acceso de lectura y escritura al área de importación del Slave. Export area: El Master leerá desde el área de exportación del Slave y escribirá en el área de importación del Slave. Nota: La lectura y la escritura tienen lugar dentro de un ciclo de la CPU. Es decir, los datos leídos estaban disponibles del <u>último</u> ciclo de la CPU.				
COM: Values at Connection Loss to Master	En caso de cortarse la conexión del módulo de comunicación al Modbus Master, se inicializarán las variables de entrada o se reenviarán sin variar al módulo procesador, según lo que se defina en este parámetro. <table border="0" style="width: 100%;"> <tr> <td style="width: 45%;">Adopt initial data</td> <td>Las variables de entrada se reinicializarán a sus valores iniciales.</td> </tr> <tr> <td>Retain Last Value</td> <td>Las variables de entrada conservarán su último valor.</td> </tr> </table>	Adopt initial data	Las variables de entrada se reinicializarán a sus valores iniciales.	Retain Last Value	Las variables de entrada conservarán su último valor.
Adopt initial data	Las variables de entrada se reinicializarán a sus valores iniciales.				
Retain Last Value	Las variables de entrada conservarán su último valor.				

Elemento	Descripción				
CPU: Values at Connection Loss to COM	<p>En caso de cortarse la conexión del módulo procesador al módulo de comunicación, se inicializarán las variables de entrada o se usarán sin variar en el módulo procesador, según lo que se defina en este parámetro (p. ej. si se extrae el módulo de comunicación con la comunicación en curso).</p> <p><b>Si desea convertirse un proyecto creado con una versión menor que SILworX V3, este valor deberá Ud. tenerlo como “Retain Last Value” para que no se modifique la suma de verificación CRC.</b></p> <p><b>Para sistemas de control HIMatrix menores que CPU BS V8, este valor deberá ser siempre “Retain Last Value”.</b></p> <table> <tr> <td>Igual procedimiento que desde COM al Master</td> <td>Véanse los ajustes del parámetro <i>Values at Connection Loss to Master</i></td> </tr> <tr> <td>Retain Last Value</td> <td>Las variables de entrada conservarán su último valor.</td> </tr> </table> <p>Valor por defecto: Igual procedimiento que desde COM al Master</p>	Igual procedimiento que desde COM al Master	Véanse los ajustes del parámetro <i>Values at Connection Loss to Master</i>	Retain Last Value	Las variables de entrada conservarán su último valor.
Igual procedimiento que desde COM al Master	Véanse los ajustes del parámetro <i>Values at Connection Loss to Master</i>				
Retain Last Value	Las variables de entrada conservarán su último valor.				
Alternative register/bit addressing	<p>Activado: Se usará el direccionamiento alternativo</p> <p>Desactivado: No se usará el direccionamiento alternativo</p> <p>Valor por defecto: Desactivado, véase el capítulo 7.3.11</p>				
Register Area Offset Bits Input	Rango de valores: 0...65535 Valor por defecto: 0				
Register Area Offset Bits Output	Rango de valores: 0...65535 Valor por defecto: 0				
Bit Area Offset Register Input	Rango de valores: 0...65535 Valor por defecto: 0				
Bit Area Offset Register Output	Rango de valores: 0...65535 Valor por defecto: 0				
Process Data Refresh Rate [ms]	<p>Tiempo de actualización, en ms, para el intercambio de datos del protocolo entre COM y CPU.</p> <p>Si para el tiempo de actualización <i>Refresh Rate</i> se ha elegido cero o un valor menor que el tiempo de ciclo del sistema de control, el intercambio de datos tendrá lugar lo más rápidamente posible.</p> <p>Rango de valores: 0...(2<sup>31</sup> - 1)</p> <p>Valor por defecto: 0</p>				
Forzar coherencia de datos de proceso	<p>Activado: Transferencia de todos los datos del protocolo desde CPU a COM dentro de un ciclo de la CPU.</p> <p>Desactivado: Transferencia de todos los datos del protocolo desde CPU a COM repartida a lo largo de varios ciclos de la CPU, con 1100 bytes por sentido de transmisión de datos.</p> <p>Con ello es posible reducir también el tiempo de ciclo.</p> <p>Valor por defecto: Activado</p>				

Tabla 197: Ficha de propiedades del set Modbus Slave

#### 7.3.4.2 Variable de registro (ficha “Access”)

En la ficha “Register Variables” deberán introducirse las variables que el Master dirige a 16 bits (código de función 3, 4, 6, 16, 23, 102, 103, 105, 106).

#### 7.3.4.3 Variables de bit (acceso a coil o bit)

En la ficha “Bit Variables” deberán introducirse las variables que el Master dirige a 1 bit (código de función 1, 2, 5, 15, 100, 101, 104).

### 7.3.5 Asignación de variables de envío/recepción

En la ficha **Inputs** se asignan todas las variables que el Modbus Slave recibe del Modbus Master.

#### Para configurar las variables de envío del Modbus Slave:

1. Seleccione en el árbol el **Modbus Slave** que desee configurar.
2. Haga clic con el botón derecho en **Modbus Slave** y seleccione **Edit**.
3. Seleccione la ficha **Register Variables** o **Bit Variables**.
4. En *Object Panel* seleccione una **variable** y arrástrela con el ratón al área *Register Outputs*.
5. Repita este paso para todas las demás variables que desee definir como variables de envío para el Modbus Slave.
6. Haga clic con el botón derecho en el área *Register Outputs* y seleccione **New Offsets**.

#### Para configurar las variables de recepción del Modbus Slave:

1. Seleccione en el árbol el **Modbus Slave** que desee configurar.
2. Haga clic con el botón derecho del ratón en **Modbus Slave** y seleccione **Edit**.
3. Seleccione la ficha **Register Variables** o **Bit Variables**.
4. En *Object Panel* seleccione una **Variable** y arrástrela con el ratón al área *Register Inputs*.
5. Repita este paso para todas las demás variables que desee definir como variables de recepción para el Modbus Slave.
6. Haga clic con el botón derecho en el área *Register Inputs* y seleccione **New Offsets**.

### 7.3.6 Variables de sistema Modbus Slave Set

La ficha **System Variables Modbus Slave Set** ofrece las siguientes variables de sistema.

Elemento	Descripción
Redundant State	<p>Este parámetro describe el estado de redundancia del par de módulos de comunicación Modbus Slave redundantes.</p> <p>0: Módulos COM Modbus Slave redundantes activos      1: Primer módulo COM Modbus Slave no activo      2: Módulo COM Modbus Slave redundante no activo      3: Ambos módulos COM Modbus Slave no activos</p>

Tabla 198: Ficha Modbus Slave Set

### 7.3.7 Modbus Slave y Modbus Slave Redundant

La ficha **Modbus Slave** contiene las fichas “Properties” y “System Variables”.

- 
- i** La asignación de pins de las conexiones D-Sub (fb1, fb2) se describe en las hojas de características de los respectivos módulos HIMax.
-

### Propiedades

Elemento	Descripción
Module	Selección del módulo COM en el que se ejecuta este protocolo.
Master Monitoring Time [ms]	<p>Período de tiempo tras la recepción de un requerimiento, dentro del cual el Modbus Slave deberá reaccionar.</p> <p>En caso de cortarse la conexión del módulo de comunicación al Modbus Master, el parámetro X-COM: <i>Values at Connection Loss to Master</i> hará que se inicialicen las variables de entrada o se reenvíen sin variar al módulo procesador. Véase el capítulo 7.3.4.1.</p> <p>Rango de valores 1...(2<sup>31</sup> - 1) [ms]</p> <p>Valor por defecto: 0 ms (sin limitación)</p>
<b>Parámetros de la interfaz Ethernet</b>	
Enable the TCP/IP connection	<p>Activado: Conexión TCP/IP activada</p> <p>Desactivado: Conexión TCP/IP desactivada</p> <p>Valor por defecto: Desactivado</p>
TCP-Port	Valor por defecto: 502
Maximum number of TCP Connections	<p>Máxima cantidad de conexiones TCP abiertas simultáneamente como servidor.</p> <p>Rango de valores: 1...20</p> <p>Valor por defecto: 3</p>
UDP Enable	<p>Activado: Conexión UDP/IP activada</p> <p>Desactivado: Conexión UDP/IP desactivada</p> <p>Valor por defecto: Desactivado</p>
UDP Port	Valor por defecto: 502
<b>Parámetros de la interfaz serie</b>	
Name	Nombre de la interfaz serie
Interface	Selección de las interfaces de bus de campo disponibles que pueden usarse para el Modbus Slave (ninguna, fb1, fb2).
Slave Address	<p>Dirección de bus del Slave</p> <p>Rango de valores: 1...247</p>
Baud Rate [bps]	<p>Velocidad de transmisión para RS485. Valores posibles:</p> <ul style="list-style-type: none"> <li>300 bit/s</li> <li>600 bit/s</li> <li>1200 bit/s</li> <li>2400 bit/s</li> <li>4800 bit/s</li> <li>9600 bit/s</li> <li>19200 bit/s</li> <li>38400 bit/s</li> <li>57600 bit/s (máxima velocidad con HIMax a partir de V4)</li> <li>62500 bit/s (sólo HIMatrix)</li> <li>76800 bit/s (sólo HIMatrix)</li> <li>115000 bit/s (sólo HIMatrix)</li> </ul>
Parity	<p>Rango de valores:</p> <ul style="list-style-type: none"> <li>▪ No</li> <li>▪ Impar</li> <li>▪ Par</li> </ul> <p>Valor por defecto: Par</p>
Stop Bits	<p>Rango de valores:</p> <p>Estándar (adapta la cantidad de los bits de parada a la paridad: con paridad = 1 bit de parada; sin paridad 2 bits de parada)</p> <p>Un bit de parada</p> <p>Dos bits de parada</p> <p>Valor por defecto: Estándar</p>

Elemento	Descripción
Number of Idle Chars	La cantidad de caracteres idle al principio y al final del frame del telegrama RTU. Rango de valores: 1...65535 Valor por defecto: 5 caracteres

Tabla 199: Ficha “TCP and UDP Ports” para Modbus Slave HIMA

La ficha **System Variables** ofrece variables de sistema que permiten evaluar el estado del Modbus Slave en el programa del usuario y controlar en base a ello el Modbus Slave

Elemento	Descripción
Average Buffer Fill Level for Requests	Promedio de requerimientos simultáneos de Masters
Valid Master Requests	Cantidad de requerimientos válidos de Masters desde la última puesta a cero de todos los contadores o desde el encendido.
Master Requests	Cantidad total de requerimientos de Masters desde la última puesta a cero de todos los contadores o desde el encendido.
Master Monitoring Time [ms]	Período de tiempo tras la recepción de un requerimiento, dentro del cual el Modbus Slave deberá reaccionar. Véase el capítulo 7.3.7.
Master Connection State	FALSE: No conectado TRUE: Conectado
Maximum Buffer Fill Level for Requests	Máxima cantidad de requerimientos simultáneos de Masters
Reset All Counters	Con esta variable del sistema podrán ponerse a cero todos los contadores mediante el programa del usuario. Un cambio de 0 a 1 desencadenará la función de reset Valores > 1 se tratan como un 1.
Invalid Master Requests	Cantidad de requerimientos no válidos de Masters desde la última puesta a cero de todos los contadores o desde el encendido. Los requerimientos no válidos son aquellos a los que el Modbus Slave responde con un código de error al Modbus Master. Aquí no se recogen los envíos erróneos detectados y filtrados ya al nivel del controlador (driver) (Framing Errors, CRC Errors, Length Errors), sino que estos se comunican sólo mediante el diagnóstico.
Rejected Requests	Cantidad de requerimientos rechazados de Masters desde la última puesta a cero de todos los contadores o desde el encendido.
Response Timeout	Cantidad de tiempos excedidos en <i>response timeouts</i> en respuestas desde la última puesta a cero de todos los contadores o desde el encendido. <i>Response Timeout</i> es el tiempo máximo que se permite que transcurra hasta que el emisor reciba confirmación de recepción de una notificación.

Tabla 200: Ficha de variables de sistema para Modbus Slave HIMA

### 7.3.8 Códigos de función Modbus del Modbus Slave HIMA

#### 7.3.8.1 Códigos de función Modbus

Modbus Slave HIMA es compatible con los siguientes códigos de función Modbus.

Elemento	Código	Tipo	Significado
READ COILS	01	BOOL	Lectura de varias variables (BOOL) desde el área de importación o exportación <sup>1)</sup> del Slave. Longitud máxima de los datos de proceso: 251 bytes
READ DISCRETE INPUT	02	BOOL	Lectura de varias variables (BOOL) desde el área de exportación del Slave. Longitud máxima de los datos de proceso: 251 bytes
READ HOLDING REGISTER	03	WORD	Lectura de varias variables de cualquier tipo desde el área de importación o exportación <sup>1)</sup> del Slave. Longitud máxima de los datos de proceso: 250 bytes
READ INPUT REGISTER	04	WORD	Lectura de varias variables de cualquier tipo desde el área de exportación del Slave. Longitud máxima de los datos de proceso: 250 bytes
WRITE SINGLE COIL	05	BOOL	Escritura de una señal individual (BOOL) al área de importación del Slave. Longitud máxima de los datos de proceso: 1 byte
WRITE SINGLE REGISTER	06	WORD	Escritura de una señal individual (WORD) al área de importación del Slave. Longitud máxima de los datos de proceso: 2 bytes
Diagnostics	08	x	Sólo subcódigo 0: función Loopback del Slave
WRITE MULTIPLE COILS	15	BOOL	Escritura de varias variables (BOOL) al área de importación del Slave. Longitud máxima de los datos de proceso: 247 bytes
WRITE MULTIPLE REGISTER	16	WORD	Escritura de varias variables de cualquier tipo al área de importación del Slave. Longitud máxima de los datos de proceso: 246 bytes
READ WRITE MULTIPLE REGISTER	23	WORD	Escritura y lectura de varias variables de cualquier tipo al y desde el área de importación o exportación del Slave. Longitud máxima de los datos de proceso: 242 bytes (telegrama de requerimiento de Modbus Master) 250 bytes (respuesta al Master).
Read Device Identification	43	x	Indican los datos de ID del Slave al Master.

<sup>1)</sup> Área de exportación seleccionable sólo con Slaves HIMA

Tabla 201: Códigos de función Modbus de Modbus Slave HIMA

Los códigos de función 03, 04, 16 y 23 son compatibles, además de con el tipo de datos WORD (2 bytes) también con otros tipos de datos.

Para cada requerimiento deberá introducirse la dirección inicial de la primera variable a transmitir y la cantidad de bits/registros de las variables a transmitir.

**Códigos de error:**

- Si el Master envía un telegrama con código de función desconocido, el sistema de control responderá con el código de error 1 (“Invalid Code”).
- Si el telegrama del Master no concuerda con la configuración de los Modbus Slaves (p. ej. el telegrama de requerimiento no termina justo en un límite de variable), el Slave responderá con el código de error 2 (“Invalid Data”).
- Si el Master envía un telegrama con valores erróneos (p. ej. campos de longitud), el Slave responderá con el código de error 3 (“Invalid Code”).

La comunicación tendrá lugar sólo en el estado RUN del módulo COM. En todos los demás estados operativos del módulo COM no se responderá a los requerimientos del Master.

**Nota relativa a la función Modbus: Read Device Identification (43)**

El Modbus Slave HIMax comunica sus datos de ID al Master, siendo compatibles los siguientes ID de objeto:

**Basic:**

0x00 VendorName “HIMA Paul Hildebrandt GmbH”  
0x01 ProductCode “<ModulSerieNumber>”  
0x02 MajorMinorRevision “<COM Vx.y CRC>”

**Regular:**

0x03 VendorUrl “http://www.hima.com”  
0x04 ProductName “HIMax”  
0x05 ModelName “HIMax”  
0x06 UserApplicationName “-----[S.R.S]”

**Extended:**

0x80 empty “-----”  
0x81 empty “-----”  
0x82 empty “-----”  
0x83 empty “-----”  
0x84 empty “-----”  
0x85 empty “-----”  
0x86 CRC del archivo modbus.config “<0x234adcef>”  
(archivo de configuración del protocolo Modbus Slave en el sistema de archivos de la CPU.  
Para comparar con las especificaciones en SILworX en “Online/Version comparison”.

Son compatibles los siguientes códigos ReadDevice ID:

- (1) Read Basic device identification (stream access)
- (2) Read regular device identification (stream access)
- (3) Read extended device identification (stream access)
- (4) Read one specific identification object (individual access)

Hallará más información sobre Modbus en la especificación “Modbus Application Protocol Specification” en [www.modbus.org](http://www.modbus.org).

### 7.3.9 Códigos de función específicos de HIMA

Los códigos de función específicos de HIMA corresponden a los códigos de función estándar de Modbus. Las únicas diferencias son la máxima longitud admisible de datos de proceso de 1100 bytes y el formato de cabecera de requerimiento y respuesta:

Elemento	Código	Tipo	Significado
Read Coils Extended	100 (0x64)	BOOL	Corresponde al código de función 01 Lectura de varias variables (BOOL) desde el área de importación o exportación <sup>1)</sup> del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Read Discrete Inputs Extended	101 (0x65)	BOOL	Corresponde al código de función 02 Lectura de varias variables (BOOL) desde el área de exportación del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Read Holding Registers Extended	102 (0x66)	WORD	Corresponde al código de función 03 Lectura de varias variables de cualquier tipo desde el área de importación o exportación <sup>1)</sup> del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Read Input Registers Extended	103 (0x67)	WORD	Corresponde al código de función 04 Lectura de varias variables de cualquier tipo desde el área de exportación del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Write Multiple Coils Extended	104 (0x68)	BOOL	Corresponde al código de función 15 Escritura de varias variables (BOOL) al área de importación del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Write Multiple Registers Extended	105 (0x69)	WORD	Corresponde al código de función 16 Escritura de varias variables de cualquier tipo al área de importación del Slave. Longitud máxima de los datos de proceso: 1100 bytes
Read/Write Multiple Registers Extended	106 (0x6A)	WORD	Corresponde al código de función 23 Escritura y lectura de varias variables de cualquier tipo al y desde el área de importación o exportación del Slave. Longitud máxima de los datos de proceso: 1100 bytes (telegrama de requerimiento de Modbus Master). 1100 bytes (respuesta al Master).

### 7.3.9.1 Formato de la cabecera de requerimiento y respuesta

La cabecera de requerimiento y respuesta de los códigos de función Modbus específicos de HIMA se compone del siguiente modo:

Código	Requerimiento	Respuesta
100 (0x64)	Código de función de 1 byte 0x64 Dirección inicial de 2 bytes Cantidad de Coils 1...8800(0x2260) de 2 bytes	Código de función de 1 byte 0x64 Cantidad de 2 bytes de bytes = N Datos de Coil de N bytes (8 Coils se agrupan en un byte)
101 (0x65)	Código de función de 1 byte 0x65 Dirección inicial de 2 bytes Cantidad de Coils 1...8800(0x226) de 2 bytes	Código de función de 1 byte 0x65 Cantidad de 2 bytes de bytes = N Datos de Coil de N bytes (8 Coils se agrupan en un byte)
102 (0x66)	Código de función de 1 byte 0x66 Dirección inicial de 2 bytes Cantidad de registros de 2 bytes 1...550(0x226)	Código de función de 1 byte 0x66 Cantidad de 2 bytes de bytes = N Datos de registro de N bytes
103 (0x67)	Código de función de 1 byte 0x67 Dirección inicial de 2 bytes Cantidad de registros de 2 bytes 1...550(0x226)	Código de función de 1 byte 0x67 Cantidad de 2 bytes de bytes = N Datos de registro de N bytes
104 (0x68)	Código de función de 1 byte 0x68 Dirección inicial de 2 bytes Cantidad de Coils 1...8800(0x2260) de 2 bytes Cantidad de 2 bytes de bytes = N Datos de Coil de N bytes	Código de función de 1 byte 0x66 Dirección inicial de 2 bytes Cantidad de Coils 1...8800(0x2260) de 2 bytes
105 (0x69)	Código de función de 1 byte 0x69 Dirección inicial de 2 bytes Cantidad de registros de 2 bytes 1...550(0x226) Cantidad de 2 bytes de bytes = N Datos de registro de N bytes	Código de función de 1 byte 0x69 Dirección inicial de 2 bytes Cantidad de registros de 2 bytes 1...550(0x226)
106 (0x6A)	Código de función de 1 byte 0x6a Dirección inicial de lectura de 2 bytes Cantidad de registros de lectura de 2 bytes 1...550(0x226) Dirección inicial de escritura de 2 bytes Cantidad de registros de escritura de 2 bytes 1...550(0x226) Cantidad de 2 bytes de bytes de escritura = N Datos de registro de N bytes	Código de función de 1 byte 0x6a Cantidad de 2 bytes de bytes = N Datos de registro de N bytes

### 7.3.10 Direccionamiento Modbus por bits y registros

Este modo de direccionamiento corresponde al estándar de direccionamiento Modbus y reconoce sólo las dos longitudes de datos de bit (1 bit) y de registro (16 bits), con las que pueden transmitirse todos los tipos de datos admisibles.

En el Modbus Slave hay un **Register Area**/Área de Registros (entradas y salidas) y un **Bit Area**/Área de Bits (entradas y salidas). Ambas áreas están separadas entre sí y pueden recibir todos los tipos de datos admisibles. La diferencia entre ambas áreas radica en los códigos de función Modbus permitidos con los que puede accederse a estas áreas.



El direccionamiento por bits y registros no garantiza ninguna integridad de variables, es decir, con este acceso pueden leerse/escibirse las partes que se quiera de las variables. Las variables del tipo BOOL se comunican agrupadas, es decir, cada variable del tipo BOOL consta como un bit dentro de un byte.

#### 7.3.10.1 Ficha “Area”

En la ficha **Register Variables** se crean las variables del área de registros.

Más información sobre la asignación de variables de envío/recepción en el capítulo 7.3.5.



Para poder acceder con los códigos de función Modbus 1, 2, 5, 15 a las variables del área de registros, deberán reflejarse en espejo las variables al área de bits. Véase el capítulo 7.3.11.1.

A las variables del área de registros sólo podrá accederse mediante los códigos de función Modbus 3, 4, 6, 16, 23. Para ello deberá introducirse la dirección inicial de la primera variable en las propiedades del código de función.

Ejemplo: Acceso a variables del área de registro del Modbus Slave

Variables de registro	Registro.Bit	Bit
00_Register_Area_WORD	0.0	0
01_Register_Area_SINT	1.8	16
02_Register_Area_SINT	1.0	24
03_Register_Area_REAL	2.0	32
04_Register_Area_BOOL	4.8	64
05_Register_Area_BOOL	4.9	65
06_Register_Area_BOOL	4.10	66
07_Register_Area_BOOL	4.11	67
08_Register_Area_BOOL	4.12	68
09_Register_Area_BOOL	4.13	69
10_Register_Area_BOOL	4.14	70
11_Register_Area_BOOL	4.15	71
12_Register_Area_BOOL	4.0	72
13_Register_Area_BOOL	4.1	73
14_Register_Area_BOOL	4.2	74
15_Register_Area_BOOL	4.3	75
16_Register_Area_BOOL	4.4	76
17_Register_Area_BOOL	4.5	77
18_Register_Area_BOOL	4.6	78
19_Register_Area_BOOL	4.7	79

Tabla 202: Ficha de variables del área de registro del Modbus Slave

Configuración en Modbus Master HIMA del telegrama de requerimiento

Para importar al Modbus Master las variables 01\_Register\_Area\_SINT hasta 03\_Register\_Area\_REAL:

1. Haga clic con el botón derecho en **TCP/UDP Slave** y seleccione **New** en el menú contextual.
2. En la lista seleccione **Read Holding Registers (3)**.
3. Haga clic con el botón derecho en **Read Holding Registers (3)** y seleccione **Properties**.
  - Como **start address of the read area**, escriba 1.
4. Haga clic con el botón derecho en **Read Holding Registers (3)** y seleccione **Edit**.
5. En **Object Panel** seleccione las siguientes variables y arrástrelas a la ficha **Input Variables**.

Variables de registro	Offset
01_Register_Area_SINT	0
02_Register_Area_SINT	1
03_Register_Area_REAL	2

6. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Variables** y seleccione **New Offsets** para renombrar los offsets de las variables.

#### 7.3.10.2 Área de bits

En la ficha **Bit Variables** se crean las variables del área de bits. Más información sobre la asignación de variables de envío/recepción en el capítulo 7.3.5.

- 
- i** Para poder acceder con los códigos de función Modbus 3, 4, 6, 16, 23 a las variables del área de bits, deberán reflejarse en espejo las variables al área de registros. Véase el capítulo 7.3.11.2.
- 

A las variables del área de bits sólo podrá accederse mediante los códigos de función Modbus 1, 2, 5, 15. Para ello deberá introducirse la dirección inicial de la primera variable en las propiedades del código de función.

Ejemplo: Acceso a variables del área de bits del Modbus Slave

Variables de bit	Bit	Registro.Bit
00_BIT_Area_WORD	0	0.0
01_BIT_Area_SINT	16	1.8
02_BIT_Area_SINT	24	1.0
03_BIT_Area_REAL	32	2.0
04_BIT_Area_BOOL	64	4.8
05_BIT_Area_BOOL	65	4.9
06_BIT_Area_BOOL	66	4.10
07_BIT_Area_BOOL	67	4.11
08_BIT_Area_BOOL	68	4.12
09_BIT_Area_BOOL	69	4.13
10_BIT_Area_BOOL	70	4.14
11_BIT_Area_BOOL	71	4.15
12_BIT_Area_BOOL	72	4.0
13_BIT_Area_BOOL	73	4.1
14_BIT_Area_BOOL	74	4.2

Variables de bit	Bit	Registro.Bit
15_BIT_Area_BOOL	75	4.3
16_BIT_Area_BOOL	76	4.4
17_BIT_Area_BOOL	77	4.5
18_BIT_Area_BOOL	78	4.6
19_BIT_Area_BOOL	79	4.7

Tabla 203: Variables de bits del área de bits del Modbus Slave

Configuración en Modbus Master HIMA del telegrama de requerimiento

**Para importar al Modbus Master las variables 04\_BIT\_Area\_BOOL hasta 06\_Area\_BOOL:**

1. Haga clic con el botón derecho en **TCP/UDP Slave** y seleccione **New** en el menú contextual.
2. En la lista seleccione “**Read Coils**” (1).
3. Haga clic con el botón derecho en **Read Coils (1)** y seleccione **Properties** en el menú contextual.
  - Como **start address of the read area**, escriba **64**.
4. Haga clic con el botón derecho en **Read Coils (1)** y seleccione **Edit** en el menú contextual.
5. En **Object Panel** seleccione las siguientes variables y arrástrelas a la ficha **Input Variables**.

Variables de bit	Offset
04_BIT_Area_BOOL	0
05_BIT_Area_BOOL	1
06_BIT_Area_BOOL	2

6. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Variables** y seleccione **New Offsets** para renombrar los offsets de las variables.

### 7.3.11 Offsets para direccionamiento alternativo Modbus

Para poder acceder con los códigos de función Modbus (tipo registro) a variables del **Bit Area** y con los códigos de función Modbus (tipo bit) a variables del **Register Area**, deberán reflejarse en espejo las variables de un área a la otra.

En la ficha **Properties/Offsets** se introducirán los offsets de las variables reflejadas.

**Para reflejar en espejo las variables al área de bits y al área de registros:**

1. En el menú contextual del Modbus Slave seleccione **Edit, Offsets** y active **Use Alternative Register/Bit Addressing**.
  - Así se reflejarán las variables al otro área respectivamente.
2. Introduzca el offset para las variables reflejadas en el área de bits o de registros.



Las variables reflejadas en el área de bits o de registros y las variables existentes en el área de bits o de registros deberán quedar sin solaparse en lo relativo a las direcciones Modbus.

Elemento	Descripción y rango de valores
Alternative register/bit addressing	Activado: Se usará el direccionamiento alternativo Desactivado: No se usará el direccionamiento alternativo Valor por defecto: Desactivado
Register area offset/bit inputs	0...65535
Register area offset/bit outputs	0...65535
Bit area offset/register inputs	0...65535
Bit area offset/register outputs	0...65535

Tabla 204: Ficha “Offsets” para Modbus Slave HIMA

#### 7.3.11.1 Acceso a variables de registro del área de bits del Modbus Slave

Para poder acceder con los códigos de función Modbus (tipo bit) 1, 2, 5, 15 a variables de registro, deberán reflejarse en espejo las variables de registro al **Bit Area**.

En la ficha **Properties/Offsets** deberán introducirse los offsets de las variables de registro reflejadas.

Ejemplo:

Bit area offset/register inputs	8000
Bit area offset/register outputs	8000

Aquí se hallan las variables reflejadas en espejo del área de registros al área de bits a partir de la dirección de bit 8000.

Variables de registro reflejadas en espejo	Bit
00_Register_Area_WORD	8000
01_Register_Area_SINT	8016
02_Register_Area_SINT	8024
03_Register_Area_REAL	8032
04_Register_Area_BOOL	8064
05_Register_Area_BOOL	8065
06_Register_Area_BOOL	8066
07_Register_Area_BOOL	8067
08_Register_Area_BOOL	8068
09_Register_Area_BOOL	8069
10_Register_Area_BOOL	8070
11_Register_Area_BOOL	8071
12_Register_Area_BOOL	8072
13_Register_Area_BOOL	8073
14_Register_Area_BOOL	8074
15_Register_Area_BOOL	8075
16_Register_Area_BOOL	8076
17_Register_Area_BOOL	8077
18_Register_Area_BOOL	8078
19_Register_Area_BOOL	8079

Tabla 205: Variables reflejadas en espejo del área de registros al área de bits

Configuración en Modbus Master HIMA del telegrama de requerimiento

**Para importar al Modbus Master las variables 04\_Register\_Area\_BOOL hasta 06\_Register\_Area\_BOOL:**

1. Haga clic con el botón derecho en **TCP/UDP Slave** y seleccione **New** en el menú contextual.
2. En la lista seleccione **Read Coils (1)**.
3. Haga clic con el botón derecho en **Read Coils (1)** y seleccione **Properties** en el menú contextual.
  - Como **start address of the read area**, escriba **8064**.
4. Haga clic con el botón derecho en **Read Coils (1)** y seleccione **Edit** en el menú contextual.
5. En **Object Panel** seleccione las siguientes variables y arrástrelas a la ficha **Input Variables**.

Variables de registro reflejadas en espejo	Offset
04_Register_Area_BOOL	0
05_Register_Area_BOOL	1
06_Register_Area_BOOL	2

6. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Variables** y seleccione **New Offsets** para renombrar los offsets de las variables.

#### 7.3.11.2 Acceso a variables de bits del área de registros del Modbus Slave

Para poder acceder con los códigos de función Modbus (tipo registro) 3, 4, 6, 16, 23 a variables de bits, deberán reflejarse en espejo las variables de bits al **Register Area**.

En la ficha **Properties/Offsets** deberán introducirse los offsets de las variables de bits reflejadas.

Ejemplo:

Register area offset/bit inputs: 1000

Register area offset/bit outputs: 1000

Aquí se hallan las variables reflejadas del área de bits al área de registros a partir de la dirección de registro 1000.

Variables de bits reflejadas en espejo	Registro.Bit
00_BIT_Area_WORD	1000.0
01_BIT_Area_SINT	1001.8
02_BIT_Area_SINT	1001.0
03_BIT_Area_REAL	1002.0
04_BIT_Area_BOOL	1004.8
05_BIT_Area_BOOL	1004.9
06_BIT_Area_BOOL	1004.10
07_BIT_Area_BOOL	1004.11
08_BIT_Area_BOOL	1004.12
09_BIT_Area_BOOL	1004.13
10_BIT_Area_BOOL	1004.14
11_BIT_Area_BOOL	1004.15
12_BIT_Area_BOOL	1004.0
13_BIT_Area_BOOL	1004.1

Variables de bits reflejadas en espejo	Registro.Bit
14_BIT_Area_BOOL	1004.2
15_BIT_Area_BOOL	1004.3
16_BIT_Area_BOOL	1004.4
17_BIT_Area_BOOL	1004.5
18_BIT_Area_BOOL	1004.6
19_BIT_Area_BOOL	1004.7

Tabla 206: Variables reflejadas del área de bits al área de registros

Configuración en Modbus Master HIMA del telegrama de requerimiento

**Para importar al Modbus Master las variables 01\_BIT\_Area\_SINT hasta 03\_BIT\_Area\_REAL:**

1. Haga clic con el botón derecho en **TCP/UDP Slave** y seleccione **New** en el menú contextual.
2. En la lista seleccione **Read Holding Registers (3)**.
3. Haga clic con el botón derecho en **Read Holding Registers (3)** y seleccione **Properties**.
  - Como **start address of the read area**, escriba **1001**.
4. Haga clic con el botón derecho en **Read Holding Registers (3)** y seleccione **Edit**.
5. En **Object Panel** seleccione las siguientes variables y arrástrelas a la ficha **Input Variables**.

Variables de bits reflejadas en espejo	Offset
01_BIT_Area_SINT	0
02_BIT_Area_SINT	1
03_BIT_Area_REAL	2

6. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Variables** y seleccione **New Offsets** para renumerar los offsets de las variables.

### 7.3.12 Panel de control (Modbus Slave)

El usuario podrá comprobar y controlar la configuración del Modbus Slave en el panel de control. Además, allí se muestra información actual de estado del Slave (p. ej. estado del Master).

**Para abrir el panel de control para supervisar el Modbus Slave:**

1. Seleccione **Hardware** y en el menú contextual elija **Online**.
2. Escriba en **System Login** sus datos de acceso para abrir la vista en línea del hardware.
3. Haga doble clic en **COM Module** y seleccione **Modbus Slave** en el árbol.

#### 7.3.12.1 Menú contextual (Modbus Slave)

En el menú contextual del Modbus Slave seleccionado podrá elegirse el siguiente comando:

##### **Reset statistical data**

Los datos estadísticos (p. ej. tiempo de ciclo mín./máx., etc.) se pondrán a cero.

### 7.3.12.2 Recuadro de lectura (Modbus Slave)

En el recuadro de lectura se mostrarán los siguientes valores del Modbus Slave seleccionado.

Elemento	Descripción
Name	Nombre del Modbus Slave
Planned µP Budget [%]	Véase el capítulo 7.3.4
Current µP Budget [%]	
SRS of Redundant Module	SRS del módulo COM redundante
Response Time [ms]	Período de tiempo tras la recepción de un requerimiento, dentro del cual el Modbus Slave responde.

Tabla 207: Recuadro de lectura Modbus Slave

### 7.3.12.3 Recuadro de lectura (datos del Master)

En el recuadro de lectura de datos del Master se mostrarán los siguientes valores.

Elemento	Descripción
Name	Nombre de datos de Master
Requests	Cantidad total de requerimientos de Masters desde la última puesta a cero de contadores.
Valid Requests	Cantidad de requerimientos válidos de Masters desde la última puesta a cero de contadores.
Invalid Requests	Cantidad de requerimientos no válidos de Masters desde la última puesta a cero de contadores. Se cuentan sólo los requerimientos acusados por el Master. Los requerimientos erróneos recibidos con error de CRC se desechan automáticamente.
Master Timeout [ms]	Tiempo de time-out dentro del cual el Slave deberá haber recibido al menos un requerimiento de su Master. Si el Slave no recibe ningún requerimiento dentro del tiempo de time-out, el estado de <i>Master Connection Status</i> /conexión del Master cambiará a <i>not connected</i> /no conectado.
Connection State	0 = no monitoreado (Master Request Timeout igual a cero) 1 = no conectado 2 = conectado
Response Timeout	Cantidad de <i>Response Timeouts</i> desde la última puesta a cero de todos los contadores o desde el encendido. <i>Response Timeouts</i> es el tiempo máximo que se permite que transcurra hasta que el emisor reciba confirmación de recepción de una notificación.
Rejected Requests	Cantidad de requerimientos rechazados de Masters desde la última puesta a cero de todos los contadores o desde el encendido.
Maximum Buffer Fill Level for Requests	Máxima cantidad de requerimientos simultáneos de Masters
Average Buffer Fill Level for Requests	Promedio de requerimientos simultáneos de Masters

Tabla 208: Recuadro de lectura de datos del Master

### 7.3.13 Función del LED FBx en el Modbus Slave

El LED FBx de la interfaz de bus de campo asignada señaliza el estado del protocolo Modbus. Los estados del LED FBx se representan en la siguiente tabla:

LED FBx	Color	Descripción
OFF	Amarillo	Protocolo Modbus Slave inactivo! Es decir, el sistema de control se halla en estado STOP o no se ha configurado ningún Modbus Master.
Parpadeo	Amarillo	Protocolo Modbus Slave activo y hay intercambio de datos con el Modbus Master.

Tabla 209: LED FBx

### 7.3.14 Función del LED FAULT en el Modbus Slave (sólo HIMax)

El LED FAULT de la interfaz de bus de campo asignada señaliza un fallo del protocolo Modbus. Los estados del LED FAULT se representan en la siguiente tabla:

LED FAULT	Color	Descripción
OFF	Rojo	Protocolo Modbus Slave sin fallos.
Parpadeo	Rojo	Protocolo Modbus Slave perturbado. Los siguientes eventos dan lugar a fallos. <ul style="list-style-type: none"><li>▪ Código de función desconocido recibido</li><li>▪ Requerimiento recibido con direccionamiento incorrecto</li><li>▪ Tiempo de cálculo adjudicado excedido</li></ul> Si durante más de 5 segundos no se produce ningún evento de error, el indicador cambiará al estado <i>Protocol not disturbed</i> .

Tabla 210: FAULT FBx

### 7.3.15 Códigos de error de la conexión TCP/IP Modbus

Los códigos de error de la conexión TCP/IP Modbus se muestran en el cuadro de diálogo **Diagnosis**.

Código de error	Descripción
35	Operación bloqueada
48	El puerto ya está en uso
50	La red no funciona
53	Conexión cortada por el software
54	Conexión cortada por el interlocutor
55	No hay mas memoria búfer disponible
60	Se ha llegado a un time-out, conexión cerrada
61	Conexión rechazada (por el interlocutor)
65	No hay registro de enrutado al interlocutor

Tabla 211: Códigos de error de TCP/IP Modbus

## 8 Send & Receive TCP

S&R TCP es un protocolo estándar independiente del fabricante para intercambiar datos cíclica y acíclicamente, el cual no usa ningún protocolo especial aparte de TCP/IP.

Con el protocolo S&R-TCP, el sistema de control HIMax/HIMatrix es compatible prácticamente con cualquier sistema externo, tal como PCs con interfaz implementada para TCP/IP (p. ej. Winsock.dll).

S&R TCP es compatible con la interfaz SEND/RECEIVE Siemens y permite comunicar con sistemas de control Siemens mediante TCP/IP. El intercambio de datos tiene lugar mediante los bloques funcionales S7 AG\_SEND (FC5) y AG\_RECV (FC6).

### 8.1 Sistema necesario

#### Equipamiento y sistema necesarios:

Elemento	Descripción
Sistema de control	HIMax con módulo COM HIMatrix a partir de V7 de S.Op. de CPU y V12 de S.Op. de COM
Módulo de CPU	Las interfaces Ethernet del módulo procesador no pueden usarse para Send & Receive TCP.
Módulo COM	Ethernet 10/100BaseT Para cada módulo COM podrá configurarse un protocolo S&R-TCP.
Activación	Para habilitar se necesitará código de habilitación de software. Véase el capítulo 3.4.

Tabla 212: Equipamiento y sistema necesarios para S&R TCP

#### Propiedades del protocolo S&R-TCP:

Elemento	Descripción
Con función orientada a la seguridad	No
Intercambio de datos	Intercambio de datos cíclico y acíclico mediante TCP/IP.
Bloques funcionales	Los bloques funcionales S&R-TCP deberán usarse en el intercambio de datos acíclico.
Conexiones TCP	En un sistema de control podrán configurarse hasta 32 conexiones TCP, siempre y cuando no se exceda el volumen máximo de datos de envío o de recepción.
Máx. volumen de datos de recepción	Véase Tabla 9 "Protocolos estándar"
Máx. volumen de datos de recepción	<p><b>i</b> Para determinar la cantidad máxima de datos útiles, deberán restarse de la máxima cantidad de datos de envío todas las variables de estado de las conexiones TCP utilizadas y de los bloques funcionales TCP/SR. La distribución a las distintas conexiones TCP es arbitraria.</p>

Tabla 213: Propiedades de S&R TCP

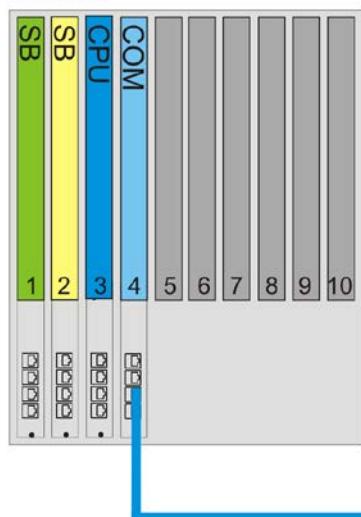
#### 8.1.1 Creación de un protocolo S&R-TCP

##### Para crear un nuevo protocolo S&R-TCP:

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. En el menú contextual de los protocolos seleccione **New, Send/Receive over TCP** para agregar un nuevo protocolo S&R-TCP.
3. En el menú contextual de Send/Receive-over-TCP-Protocol **Properties, General** seleccione **COM Module**.

## 8.2 Ejemplo: Configuración S&R TCP

HIMax



Siemens Simatic 300



Fig. 63: Conexión de HIMax y el sistema de control Siemens

En el presente ejemplo se configura el protocolo “Send/Receive over TCP” en un sistema de control HIMax. En este ejemplo se desea que HIMax comunique cíclicamente mediante S&R TCP con un sistema de control Siemens (p. ej. SIMATIC 300).

HIMax (cliente) será la estación activa, que establecerá la conexión TCP al sistema pasivo Siemens SIMATIC 300 (servidor). Sin embargo, una vez se establezca la conexión, ambos sistemas de control estarán al mismo nivel y podrán enviar o recibir en cualquier momento.

Para ensamblar HIMax y Siemens SIMATIC 300 habrá que observar lo siguiente:

Para HIMax tienen validez los requisitos indicados en la sección 8.1 (sistema necesario).

HIMax y Siemens SIMATIC 300 se interconectarán mediante sus interfaces Ethernet.

HIMax y Siemens SIMATIC 300 deberán encontrarse en la misma subred o, en caso de emplear un enrutador, disponer de los correspondientes registros de enruteado.

En este ejemplo se desea transmitir dos BYTES y una palabra WORD desde el sistema de control HIMax al sistema SIMATIC 300 de Siemens. En SIMATIC 300 las variables las recibe el bloque AG\_RECV (FC 6) y las transmite internamente al bloque AG\_SEND (FC 5). Mediante el bloque AG\_SEND (FC 5), SIMATIC 300 envía de vuelta las variables sin modificar al sistema de control HIMax.

El usuario podrá comprobar la transmisión de las variables con el editor de forzado HIMA según lo haya configurado.

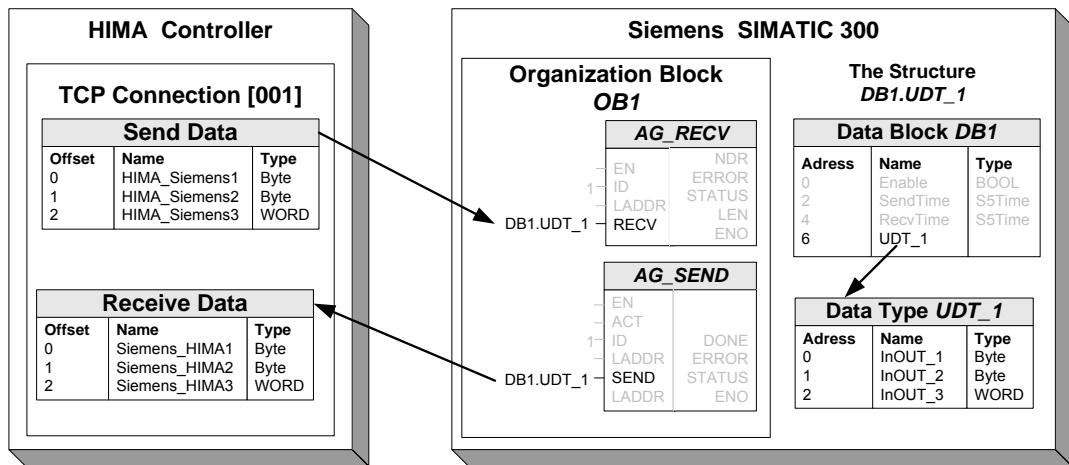


Fig. 64: Transmisión de datos entre HIMax y sistema de control Siemens

**Descripción de la configuración del sistema de control HIMax:**

Elemento	Descripción
Conexión TCP [001]	En este cuadro de diálogo se dispone de todos los parámetros necesarios para la comunicación con el interlocutor de comunicación (Siemens SIMATIC 300).
Datos de envío	Los offsets y los tipos de variables del sistema de control deberán concordar con la dirección y los tipos de variables del tipo de datos <i>UDT_1</i> de SIMATIC 300.
Datos de recepción	Los offsets y los tipos de variables del sistema de control HIMax deberán concordar con la dirección y los tipos de variables del tipo de datos <i>UDT_1</i> de SIMATIC 300.

Tabla 214: Configuración del sistema de control HIMax

**Descripción de la configuración del sistema Siemens SIMATIC 300:**

Elemento	Descripción
Bloque organizativo OB1	<i>En el bloque organizativo OB1 deberán crearse y configurarse los bloques funcionales AG_RECV (FC6) y AG_SEND (FC 5).</i>
AG_RECV (FC 6)	El bloque funcional AG_RECV (FC 6) recibe los datos recibidos desde el interlocutor de comunicación en el tipo de datos DB1.UDT_1. Las entradas ID y LADDR deberán configurarse correspondientemente para la comunicación con el interlocutor de comunicación.
AG_SEND (FC 5)	El bloque funcional AG_SEND (FC 5) transmite los datos del tipo DB1.UDT_1 al interlocutor de comunicación. Las entradas ID y LADDR deberán configurarse correspondientemente para la comunicación con el interlocutor de comunicación.
Bloque de datos DB1	El tipo de datos UDT_1 se define en el bloque de datos DB1.
Tipo de datos UDT_1	Las direcciones y los tipos de variables del sistema SIMATIC 300 deberán concordar con los offsets y los tipos del sistema de control. El tipo de datos UDT_1 recibe los datos útiles de recepción y los guarda hasta transmitirlos al interlocutor de comunicación.

Tabla 215: Configuración del sistema Siemens SIMATIC 300

## 8.2.1 Configuración S&R TCP del sistema de control SIMATIC 300 de Siemens

- i** Es posible que la siguiente secuencia de pasos para configurar el sistema de control Siemens no sea exhaustiva.

Todas las especificaciones se dan a título informativo sin garantía. Para la planificación del sistema de control Siemens tiene validez la documentación de Siemens.

### Para crear el servidor S&R-TCP en el proyecto de SIMATIC 300:

1. Inicie el administrador de SIMATIC.
2. Abra el proyecto del sistema de control SIMATIC 300 en el administrador de SIMATIC.
3. Cree y configure en este proyecto el *Industrial Ethernet* y las conexiones *MPI*.

### Para crear el tipo de datos UDT1 con las siguientes variables:

1. Abra la carpeta *Function Blocks* en el administrador de Siemens SIMATIC.
2. Abra **Add, S7 Function Block, Data Block** en el menú principal y cree un tipo de datos.
3. Asigne el nombre **UDT1** al tipo de datos
4. Asigne el nombre simbólico **UDT\_1** al tipo de datos.
5. Cree en el tipo de datos *UDT\_1* las tres variables **InOut\_x** que se ilustran en la figura.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	InOut_1	BYTE	B#16#0	
+1.0	InOut_2	BYTE	B#16#0	
+2.0	InOut_3	WORD	W#16#0	
=4.0		END_STRUCT		

Fig. 65: Lista de variables en el bloque UDT1 de Siemens

- i** Para el intercambio de datos cíclico y acíclico hay que tener en cuenta que algunos sistemas de control (p. ej. SIMATIC 300) agregan los denominados *bytes pad*. Así se garantiza que todos los tipos de datos mayores que un byte comienzan siempre por un offset par y que la longitud total de todas las variables definidas también sea par. En tales casos deberán agregarse bytes postizos (dummies) en el sistema de control HIMax (véase el capítulo 8.6 “Sistemas externos con bytes pad”).

**Para crear el bloque de datos DB1 para los bloques funcionales FC 5 y FC 6:**

1. Seleccione **Add, S7 Function Block, Data Block** en el menú principal y cree el bloque de datos.
2. Asigne el nombre **DB1** al bloque de datos.
3. Asigne el nombre simbólico **DB1** al bloque de datos.
4. Asigne al bloque de datos **DB1** el tipo de datos **UDT\_1**.
5. Parametrice en el bloque de datos **DB1** los tipos de datos tal y como se ilustra en la siguiente figura.

Address	Name	Type	Initial val	Comment
0.0		STRUCT		
+0.0	Enable	BOOL	TRUE	
+2.0	SendTime	S5TIME	S5T#100MS	
+4.0	RecvTime	S5TIME	S5T#10MS	
+6.0	UDT_1	"UDT_1"		
=10.0		END_STRUCT		

Fig. 66: Lista de variables en el bloque DB1 de Siemens

**Para crear en el editor de símbolos los siguientes símbolos**

1. Abra el cuadro de diálogo **KOP/AWL/FUP** con un doble clic en el bloque organizativo **OB1**.
2. En el menú principal abra el editor de símbolos con **Extras, Symbol Table**.
3. Complemente el *editor de símbolos* con las variables **M 1.0...MW 5** como se ilustra en la siguiente figura.

S7-Programm(1) (Symbole) -- S7_COMTEST\SIMATIC 300-				
	Status	Symbol	Adres	Data type
1		DB1	DB 1	DB 1
2		RecDone	M 1.0	BOOL
3		RecError	M 1.1	BOOL
4		SendDone	M 1.2	BOOL
5		SendError	M 1.3	BOOL
6		RecStatus	MW 1	WORD
7		RecLen	MW 3	INT
8		SendStatus	MW 5	WORD
9		Cycle Execution	OB 1	OB 1
10		UDT_1	UDT 1	UDT 1
11				

Fig. 67: Editor de símbolos de SIMATIC

**Para crear el bloque FC AG\_RECV(FC 6):**

1. Abra el cuadro de diálogo *KOP/AWL/FUP*.
2. Seleccione consecutivamente los siguientes bloques FC de la estructura de la parte izquierda del administrador Symatic:
  - 1 *OR gate*
  - 1 *S\_VIMP*
  - 1 *AG\_RECV (FC 6)*.
3. Arrastre estos *bloques funcionales* con el ratón hasta el bloque organizativo *OB1*.
4. *Conecte y configure los bloques funcionales* tal y como se ilustra en la siguiente figura.
5. Haga clic con el botón derecho en el bloque FC **AG\_RECV (FC 6)** y seleccione **Properties**.
6. Desactive **Active Connection Setup** y configure los puertos.
7. Anote el parámetro de bloque *LADDR* e introduzcalo en el esquema de funciones del bloque *AG\_RECV (FC 6)*.

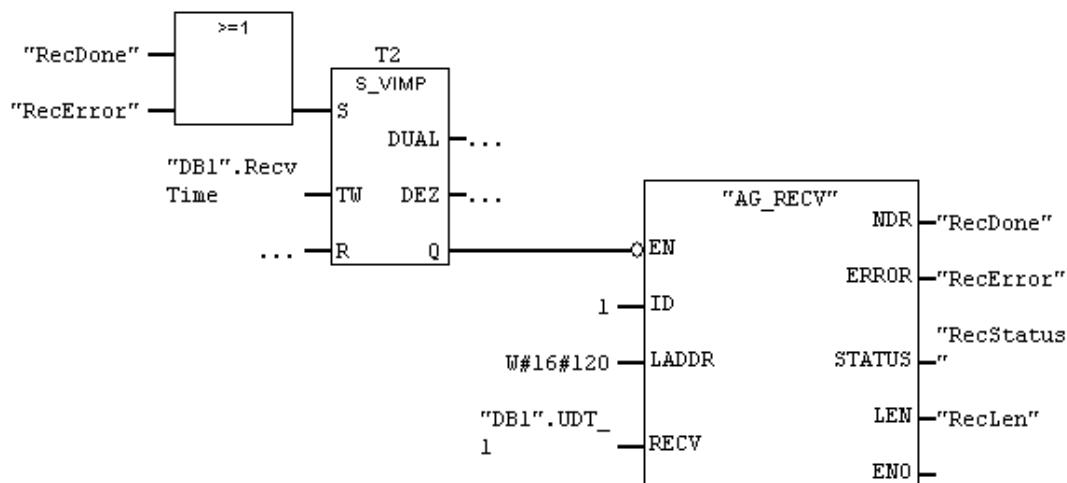


Fig. 68: Esquema de funciones para recepción

**Para crear el bloque FC AG\_SEND (FC 5):**

1. Abra el cuadro de diálogo *KOP/AWL/FUP*.
2. Seleccione consecutivamente los siguientes bloques FC de la estructura de la parte izquierda del administrador Symatic:
  - 1 *OR gate*
  - 1 *S\_VIMP*
  - 1 *AG\_SEND (FC 5)*.
3. Arrastre estos bloques funcionales con el ratón hasta el bloque organizativo *OB1*.
4. Conecte y configure los bloques funcionales tal y como se ilustra en la siguiente figura.
5. Haga clic con el botón derecho en el bloque FC **AG\_SEND (FC 5)** y seleccione **Properties**.
6. Desactive **Active Connection Setup** y configure los puertos.
7. Anote el parámetro de bloque *LADDR* e intodúzcalo en el esquema de funciones del bloque *AG\_SEND (FC 5)*.

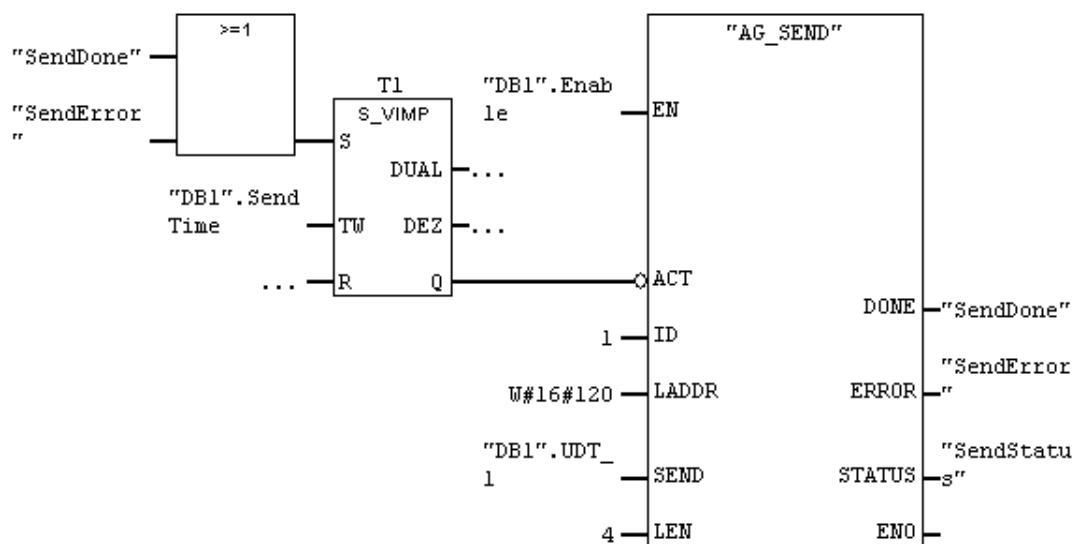


Fig. 69: Esquema de funciones para envío

**Para cargar el código al sistema de control SIMATIC 300:**

1. Inicie el **Code Generator**/generador de códigos para el programa.
2. Asegúrese de que los códigos se hayan generado sin errores.
3. Cargue el código al sistema de control SIMATIC 300.

## 8.2.2 Configuración S&R TCP del sistema de control HIMax

Para configurar los sistemas de control HIMax y para usar la utilidad de programación SILworX, recomendamos consultar el manual “Primeros pasos con SILworX”.

### Para crear las siguientes variables globales en el editor de variables:

1. Seleccione **Configuration, Global Variables** en el árbol.
2. Haga clic con el botón derecho en **Global Variables** y seleccione **Edit**.
3. Cree las variables globales como en la Tabla 216.

Nombre	Tipo
Siemens_HIMA1	Byte
Siemens_HIMA2	Byte
Siemens_HIMA3	WORD
HIMA_Siemens1	Byte
HIMA_Siemens2	Byte
HIMA_Siemens3	WORD

Tabla 216: Variables globales

### Para crear el protocolo S&R-TCP en el recurso:

1. Abra **Configuration, Resource** en el árbol.
2. Haga clic con el botón derecho en **Protocols** y seleccione **New** en el menú contextual.
3. Seleccione **Send/Receive over TCP** y escriba un nombre para el protocolo.
4. Confirme con “OK” para crear así un nuevo protocolo.
5. Haga clic con el botón derecho en **Send/Receive over TCP** y seleccione **Properties** en el menú contextual.
6. Seleccione **COM Module**. Los demás parámetros conservarán sus valores predeterminados.

### Para crear la conexión TCP:

1. Haga clic con el botón derecho en **Send/Receive over TCP** y seleccione **New** en el menú contextual.
2. Haga clic con el botón derecho en la **TCP Connection** y seleccione **Properties** en el menú contextual.
3. Configure las propiedades tal y como se ilustra en la siguiente figura.

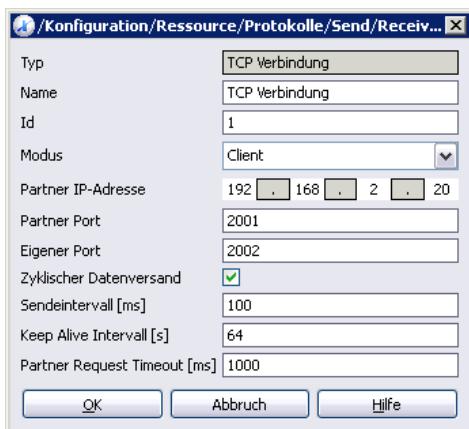


Fig. 70: Propiedades de la conexión TCP en SILworX



- Si se desea parametrizar un intercambio de datos cíclico entre dos sistemas de control, deberá activarse la opción “Cyclic data transfer” en el cuadro de diálogo “Properties” de la conexión TCP.

**Para configurar los datos de recepción del sistema de control HIMax:**

1. Haga clic con el botón derecho en **TCP Connection** y en el menú contextual seleccione **Edit**.
2. Seleccione la ficha **Process Variables**.
3. En **Object Panel** seleccione las siguientes variables globales y arrástrelas con el ratón al área **Inputs Signals**.

Variable global	Tipo
Siemens_HIMA1	Byte
Siemens_HIMA2	Byte
Siemens_HIMA3	WORD

Tabla 217: Variables para datos de recepción

4. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Register Inputs** y seleccione **New Offsets**, para renumerar los offsets de las variables.

- i** Observe que los offsets de las variables en el sistema de control HIMax deberán concordar con las direcciones de las variables en el tipo de datos *UDT\_1* de SIMATIC 300.

**Para configurar los datos de envío del sistema de control HIMax:**

1. Haga clic con el botón derecho en **TCP Connection** y en el menú contextual seleccione **Edit**.
2. Seleccione la ficha **Process Variables**.
3. En **Object Panel** seleccione las siguientes variables globales y arrástrelas con el ratón al área **Inputs Signals**.

Variable global	Tipo
HIMA_Siemens1	Byte
HIMA_Siemens2	Byte
HIMA_Siemens3	WORD

Tabla 218: Variables para datos de envío

4. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Register Inputs** y seleccione **New Offsets** para renumerar los offsets de las variables.

- i** Observe que los offsets de las variables en el sistema de control HIMax deberán concordar con las direcciones de las variables en el tipo de datos *UDT\_1* de SIMATIC 300.

## Verificación de la configuración S&amp;R TCP:

1. Abra **Configuration, Resource, Protocols, Send/Receive over TCP** en el árbol.
2. Haga clic en **Verification** en la barra de acciones y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.

## 8.3 Funciones de menú del protocolo S&R-TCP

### 8.3.1 Edit

El cuadro de diálogo **Edit** del protocolo S&R-TCP contiene la siguiente ficha.

#### Variables de sistema

Con la ficha *System variables* podrá evaluarse el estado del protocolo TCP Send Receive en el programa del usuario.

Elemento	Descripción
Active Connection Count	Variable de sistema que indica la cantidad de conexiones activas (no perturbadas).
Disturbed Connection Count	Variable de sistema que indica la cantidad de conexiones perturbadas. Por "perturbado" se entiende que la conexión TCP se ha cortado por un error o por expirarse un tiempo de time-out.
Status	Sin función

Tabla 219: Variables de sistema S&R TCP

### 8.3.2 Propiedades

El intercambio de datos a través de una conexión TCP tiene lugar cíclica o acíclicamente. Para el intercambio de datos acíclico se necesitan los bloques funcionales S&R-TCP.

Los intercambios de datos cíclico y acíclico no podrán usarse conjuntamente en una misma conexión.

#### Generalidades

Nombre	Descripción
Type	Send/Receive over TCP
Name	Nombre para el protocolo Send/Receive-over-TCP. Máximo 31 caracteres.
Module	Selección del módulo COM en el que se ejecuta este protocolo.
Activate Max. µP Budget	Activado: Aplicar el límite µP-Budget tomado del recuadro "Max. µP-Budget in [%]".  Desactivado: No se usará límite de µP-Budget para este protocolo.
Max. µP budget in [%]	Máximo valor de µP-Budget del módulo que se permite que tenga lugar durante la ejecución del protocolo.  Rango de valores: 1...100% Valor por defecto: 30%
Behavior on CPU/COM Connection Loss	En caso de cortarse la conexión del módulo procesador al módulo de comunicación, se inicializarán las variables de entrada o se usarán sin variar en el módulo procesador, según lo que se defina en este parámetro (p. ej. si se extrae el módulo de comunicación con la comunicación en curso).  Adopt initial data      Las variables de entrada se reinicializarán a sus valores iniciales. Retain Last Value      Las variables de entrada conservarán su último valor.

Tabla 220: Propiedades generales de S&R TCP

### CPU/COM

Los valores predefinidos para los parámetros permiten un intercambio de datos S&R TCP lo más rápido posible entre el módulo COM (COM) y el módulo CPU (CPU) en el sistema de control. Estos parámetros solamente deberían modificarse cuando se necesite reducir la tasa de trabajo de CPU y COM, y ello sea admisible para el proceso.



Se recomienda que los parámetros los modifiquen sólo programadores experimentados. Un aumento del tiempo de actualización de COM y CPU implica también un aumento del tiempo real de actualización de los datos en S&R TCP. Deberán examinarse los requisitos de tiempo del equipo o la instalación.

Nombre	Descripción
Process Data Refresh Rate [ms]	Tiempo de actualización, en ms, para el intercambio de datos en el protocolo S&R-TCP entre COM y CPU. Si para el tiempo de actualización “Refresh Rate” se ha elegido cero o un valor menor que el tiempo de ciclo del sistema de control, el intercambio de datos tendrá lugar lo más rápidamente posible. Rango de valores: 0...( $2^{31} - 1$ ). Valor por defecto: 0
Force Process Data Consistency	Activado: Transferencia de datos en S&R TCP de CPU a COM dentro de un ciclo de la CPU.  Desactivado: Transferencia de los datos en S&R TCP (máx. 1100 bytes por sentido de transmisión de datos) de CPU a COM repartida a lo largo de varios ciclos de la CPU.

Tabla 221: Parámetros de COM/CPU

## 8.4 Funciones de menú de la conexión TCP

### 8.4.1 Edit

Con la función de menú **Edit** se accede a la ficha “Process variables” y “System variables”.

#### Variables de proceso

##### **Input Signals/Señales de entrada**

Las variables de intercambio de datos cíclico que se desea que reciba este sistema de control se introducen en el área *Input Signals*.

En la ficha *Input Signals* pueden crearse las variables que se quiera. Sin embargo, los offsets y los tipos de variable deberán ser idénticos a los offsets y los tipos de las variables (datos de envío) del interlocutor de comunicación.

##### **Output Signals/Señales de salida**

Las variables de intercambio de datos cíclico que se desea que envíe este sistema de control se introducen en el área *Output Signals*.

En la ficha *Output Signals* pueden crearse las variables que se quiera. Sin embargo, los offsets y los tipos de variable deberán ser idénticos a los offsets y los tipos de las variables (datos de recepción) del interlocutor de comunicación.

### 8.4.2 Variables de sistema

Con las variables de la ficha *System variables* podrá evaluarse el estado de la conexión TCP en el programa del usuario.

Nombre	Descripción
Bytes received	Cantidad de bytes recibidos hasta ahora
Bytes sent	Cantidad de bytes enviados hasta ahora
Error Code	Código de error de la conexión TCP. Véase el capítulo 8.8.3.
Error Code Timestamp [ms]	Parte en milisegundos de la marca de tiempo. Momento en que se produce el error.
Error Code Timestamp [s]	Parte en segundos de la marca de tiempo. Momento en que se produce el error.
Partner Request Timeout	En el caso de la transmisión de datos cíclica: Tiempo de time-out tras el envío, dentro del cual se deberá haber recibido al menos un envío de datos por parte del interlocutor de comunicación.  0 = OFF 1... $2^{31}$ - 1 [ms]
Partner Connection State	Si no se recibe ningún envío de datos dentro del tiempo de time-out, el estado de conexión del interlocutor cambiará a <i>Not Connected</i> y se reiniciará la conexión.  0 = Sin conexión 1 = Conexión OK
Status	Estado de la conexión TCP. Véase el capítulo 8.8.5.

Tabla 222: Variables de sistema

### 8.4.3 Propiedades

El intercambio de datos a través de una conexión TCP tiene lugar cíclica o acíclicamente. Para el intercambio de datos acíclico se necesitan los bloques funcionales S&R-TCP. En el tráfico de datos cíclico no será posible operar con bloques funcionales S&R-TCP.

Nombre	Descripción
Type	Conexión TCP
Name	Cualquier nombre inequívoco para una conexión TCP. Máximo 31 caracteres.
ID	Cualquier número inequívoco de ID para cada conexión TCP. El ID se necesita también como referencia en los bloques funcionales S&R-TCP. Rango de valores 0...255 Valor por defecto: 0
Mode	<p>Servidor (valor por defecto):            Esta estación opera como servidor, es decir, en modo pasivo. Es el interlocutor de comunicación (cliente) el que tiene que establecer la conexión. Sin embargo, una vez establecida la conexión, ambas estaciones estarán al mismo nivel y podrán enviar datos en cualquier momento.            Se necesita especificar el puerto propio.</p> <p>Servidor con interlocutor definido:            Esta estación opera como servidor, es decir, en modo pasivo. Es el interlocutor de comunicación (cliente) el que tiene que establecer la conexión. Sin embargo, una vez establecida la conexión, ambas estaciones estarán al mismo nivel y podrán enviar datos en cualquier momento.            Si se introduce aquí la dirección IP y/o el puerto del interlocutor de comunicación, solamente el interlocutor de comunicación definido podrá establecer una conexión. Todas las demás estaciones se ignorarán.            Si se pone a cero alguno de los parámetros (dirección IP o puerto), no se supervisará ese parámetro.</p> <p>Cliente:            Esta estación opera como cliente, es decir, es esta estación la que establece la conexión al interlocutor de comunicación. Se necesitará especificar la dirección IP y el puerto del interlocutor de comunicación.            Opcionalmente podrá especificarse también un puerto propio.</p>
Partner IP address	Dirección IP del interlocutor de comunicación. 0.0.0.0: se permite cualquier dirección IP. Rango válido: 1.0.0.0...223.255.255.255, Excepto: 127.x.x.x Valor por defecto: 0
Partner port	Puerto del interlocutor de comunicación. 0: cualquier puerto. El S.Op. de COM rechazará los puertos reservados o ya asignados (1...1024). Rango de valores 0...65535 Valor por defecto: 0
Own Port	Puerto propio. 0: cualquier puerto. El S.Op. de COM rechazará los puertos reservados o ya asignados (1...1024). Rango de valores 0...65535 Valor por defecto: 0

Nombre	Descripción
Cyclic data transfer	Desactivado (valor por defecto): Envío cíclico de datos desactivado. El intercambio de datos a través de esta conexión TCP deberá programarse con bloques funcionales. No se permite que haya datos de E/S cíclicos definidos en esta conexión.
	Activado: El envío cíclico de datos está activado. Los datos se definen en el cuadro de diálogo "Process variables" de la conexión TCP. Deberá haber datos de recepción definidos. En esta conexión no se podrá operar con bloques funcionales.
Send Intervall [ms]	Editable sólo en caso de envío cíclico de datos. Aquí se ajusta el intervalo de envío. Rango de valores 10...2147483647 ms (valores menores se redondearán a 10 ms) Valor por defecto: 0
Keep Alive Interval [s]	El tiempo hasta que se activa el monitoreo de conexión proporcionado por TCP. Con un cero se desactiva el monitoreo de la conexión. Si dentro del intervalo "KeepAlive" elegido no se intercambian datos, se enviarán pruebas de "KeepAlive" al interlocutor de comunicación. Si sigue habiendo conexión, el interlocutor de comunicación confirmará la recepción de las pruebas "KeepAlive". Si en un tiempo > 10 intervalo KeepAlive no tiene lugar ningún intercambio de datos entre los interlocutores, se cerrará la conexión. Si tras el envío de un paquete de datos no se recibe respuesta, se repetirá el envío del paquete de datos a intervalos definidos. La conexión se cortará tras 12 repeticiones infructuosas (aprox. 7 minutos). Rango de valores 1...65535s Valor por defecto: 0 = desactivado
Partner Request Timeout [ms]	En el caso de la transmisión de datos cíclica: Tiempo de time-out tras el envío, dentro del cual se deberá haber recibido al menos un envío de datos por parte del interlocutor de comunicación. Si no se recibe ningún envío de datos dentro del tiempo de time-out, el estado de <i>Partner connection state</i> /conexión del interlocutor cambiará a <i>Not Connected</i> y se reiniciará la conexión. Tras cerrarse la conexión debido a que ha expirado un time-out o algún otro error, la parte activa volverá a establecer la conexión con un retardo de 10 x PartnerRequestTimeout o 10 segundos en caso de ser PartnerRequestTimeout = 0. La parte pasiva abrirá el puerto ya tras la mitad de ese tiempo. 0 = OFF Rango de valores 1...2 <sup>31</sup> - 1 [ms] Valor por defecto: 0

Tabla 223: Propiedades de la conexión S&amp;R TCP

## 8.5 Intercambio de datos

S&R TCP opera según el principio cliente/servidor. La conexión deberá establecerla el interlocutor de comunicación que esté configurado como cliente. Sin embargo, una vez establecida la conexión, ambos interlocutores de comunicación estarán al mismo nivel y podrán enviar datos en cualquier momento.

S&R TCP no posee un protocolo propio para asegurar los datos, sino que usa directamente el protocolo TCP/IP. Puesto que TCP envía los datos en un así llamado DataStream, deberá Ud. asegurarse de que los offsets y los tipos de las variables a intercambiar sean idénticos tanto en el receptor como en el emisor.

S&R TCP es compatible con la interfaz Siemens SEND/RECEIVE y permite el intercambio de datos cíclico con los bloques funcionales S7 Siemens AG\_SEND (FC5) y AG\_RECV (FC6) (véase el capítulo 8.2, ejemplo de configuración S&R TCP).

Además, HIMA proporciona cinco bloques funcionales S&R-TCP con los que puede controlarse y adaptarse individualmente la comunicación mediante el programa del usuario. Con los bloques funcionales S&R-TCP pueden enviarse y recibirse los protocolos transmitidos por TCP que se quiera (p. ej. Modbus).

### 8.5.1 Conexiones TCP

Para cada conexión a través de S&R TCP a un interlocutor de comunicación deberá crearse al menos una conexión TCP en el sistema de control HIMax.

En el cuadro de diálogo *Properties* de la conexión TCP deberán introducirse el número de ID de la conexión TCP y los puertos o las direcciones del sistema de control propio y del interlocutor de comunicación.

En un sistema de control HIMax pueden crearse un máximo de 32 conexiones TCP.

Las conexiones TCP creadas deberán tener diferentes números de ID y diferentes puertos/direcciones.

#### Para crear una nueva conexión TCP:

1. Abra **Configuration, Resource** en el árbol.
2. Haga clic con el botón derecho en **Protocols** y seleccione **New** en el menú contextual.
3. Seleccione **Send/Receive over TCP** y escriba un nombre para el protocolo.
4. Confirme con "OK" para crear así un nuevo protocolo.
5. Haga clic con el botón derecho en **Send/Receive over TCP** y seleccione **Properties** en el menú contextual.
6. Seleccione **COM Module**. Los demás parámetros conservarán sus valores predeterminados.

---

**SUGE-  
RENCIA** El sistema de control HIMax/HIMatrix y el sistema externo deberán encontrarse en la misma subred o, en caso de emplear un enrutador, disponer de los correspondientes registros de enrutado.

---

### 8.5.2 Intercambio de datos cíclico

Si se usa el intercambio de datos cíclico, deberá definirse un intervalo de envío en el sistema de control HIMax/HIMatrix y en el interlocutor de comunicación.

El intervalo de envío define el intervalo cíclico dentro del cual el interlocutor de comunicación emisor deberá enviar sus variables al interlocutor de comunicación receptor.

- Para garantizar un intercambio de datos continuo, debería definirse en ambos interlocutores de comunicación un intervalo de envío similar (véase el capítulo 8.5.5, control del flujo).
- La opción *Cyclic Data Transfer* deberá estar activada en la conexión TCP utilizada para el intercambio de datos cíclico.
- En una conexión TCP en la que esté activada la opción *Cyclic Data Transfer* no podrá operarse con bloques funcionales.
- Las variables a enviar y recibir se asignan en el cuadro de diálogo *Process Variables* de la conexión TCP. Deberá haber datos de recepción presentes, los datos de envío son opcionales.



Las mismas variables (mismos offsets y tipos) que en una estación de envío estén definidas como datos de envío deberán estar definidas como datos de recepción en la otra estación.

### 8.5.3 Intercambio de datos acíclico con bloques funcionales

El intercambio de datos acíclico en el sistema de control HIMax/HIMatrix lo controla el programa del usuario mediante los bloques funcionales S&R-TCP.

Así es posible usar un temporizador o un interruptor mecánico en una entrada física del sistema de control HIMax/HIMatrix para controlar el intercambio de datos.

- La opción *Cyclic Data Transfer* deberá estar desactivada en la conexión TCP utilizada.
- Solamente un bloque funcional TCP S/R podrá enviar datos cada vez.
- Las variables a enviar o recibir se asignan a en el cuadro de diálogo *Process Variables* de los bloques funcionales S&R-TCP (todos excepto *Reset*).



Las mismas variables (mismos offsets y tipos) que en una estación de envío estén definidas como datos de envío deberán estar definidas como datos de recepción en la otra estación.

### 8.5.4 Intercambio de datos cíclico y acíclico simultáneamente

A este efecto deberá configurarse una conexión TCP para datos cíclicos y una segunda para datos acíclicos. Ambas conexiones TCP deberán utilizar diferentes direcciones *partner IP addresses* y diferentes puertos *partner ports*.

No se podrá usar una sola conexión TCP para el intercambio de datos cíclico y acíclico conjuntamente.

### 8.5.5 Control del flujo

El control de flujo es parte integrante de TCP y monitorea el continuo tráfico de datos entre dos interlocutores de comunicación.

En el caso de la transmisión de datos cíclica deberá recibirse un paquete entre cada 3 y 5 paquetes enviados. De no ser así, se bloqueará el envío hasta que se reciba un nuevo paquete o el monitoreo de conexión cortará la conexión.

La cantidad (3...5) de envíos posibles sin recepción de paquetes dependerá del tamaño de los paquetes a enviar.

Cantidad = 5 para paquete pequeños < 4kB.

Cantidad = 3 para paquetes grandes  $\geq$  4kB.

- Al planificar el proyecto habrá que observar que ninguna de ambas estaciones envíe más datos que los que la otra pueda procesar de forma síncrona.
- Para el intercambio de datos cíclico deberá ajustarse en ambos interlocutores de comunicación aproximadamente el mismo intervalo de envío.

## 8.6

### Sistemas externos con bytes pad

Para el intercambio de datos cíclico y acíclico hay que tener en cuenta que algunos sistemas de control (p. ej. SIMATIC 300) agregan los denominados *bytes pad*. Así se garantiza que todos los tipos de datos que sean mayores que un byte comiencen siempre por un offset par y que la longitud total de los paquetes (en bytes) también sea par.

En el sistema de control HIMax/HIMatrix deberán insertarse bytes postizos (dummies) para los *bytes pad* en los puntos correspondientes.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	InOut_1	BYTE	B#16#0
+2.0	InOut_3	WORD	W#16#0
=4.0		END_STRUCT	

Fig. 71: Lista de variables de Siemens

En el sistema de control Siemens se inserta un *byte pad* (no visible) para que la variable *InOut\_3* comience en un offset par.

Output Signals					
F	Name	Data type	Offset	Global Variable	
1	InOut_1	BYTE	0	InOut_1	
2	Dummy	BYTE	1	Dummy	
3	InOut_3	WORD	2	InOut_3	

Fig. 72: Lista de variables de HIMax/HIMatrix

En el sistema de control HIMax/HIMatrix deberá insertarse un *byte postizo (dummy)* para que la variable *InOut\_3* tenga el mismo offset que en el sistema de control Siemens.

## 8.7 Bloques funcionales S&R-TCP

Si la transmisión de datos cíclica es demasiado poco versátil, también podrán enviarse y recibirse datos mediante los bloques funcionales S&R-TCP. Deberá desactivarse la opción *Cyclic Data Transfer* en la conexión TCP utilizada.

Con los bloques funcionales S&R-TCP, el usuario podrá adaptar óptimamente la transmisión de datos a través de TCP/IP a las necesidades de su proyecto.

Los bloques funcionales se parametrizan en el programa del usuario. Así podrán aplicarse y evaluarse las funciones (Send, Receive, Reset) del sistema de control HIMax/HIMatrix en el programa del usuario.

Los bloques funcionales S&R-TCP se necesitan sólo para el intercambio de datos acíclico. ¡Para el intercambio de datos cíclico entre servidor y cliente no se necesitan estos bloques funcionales!



La configuración de los bloques funcionales S&R-TCP se describe en el capítulo 13.1.

Se dispone de los siguientes bloques funcionales:

Bloque funcional	Descripción de la función
TCP_Reset (véase el capítulo 8.7.1)	Reinicialización de una conexión TCP
TCP_Send (véase el capítulo 8.7.2)	Envío de datos
TCP_Receive (véase el capítulo 8.7.3)	Recepción de paquetes de datos de longitud definida
TCP_ReceiveLine (véase el capítulo 8.7.4)	Recepción de una línea ASCII
TCP_ReceiveVar (véase el capítulo 8.7.5)	Recepción de paquetes de datos de longitud variable (con campo de longitud)
LATCH	Se usa sólo dentro de otros bloques funcionales
PIG	Se usa sólo dentro de otros bloques funcionales
PIGII	Se usa sólo dentro de otros bloques funcionales

Tabla 224: Bloques funcionales para conexiones S&R-TCP

### 8.7.1 TCP\_Reset

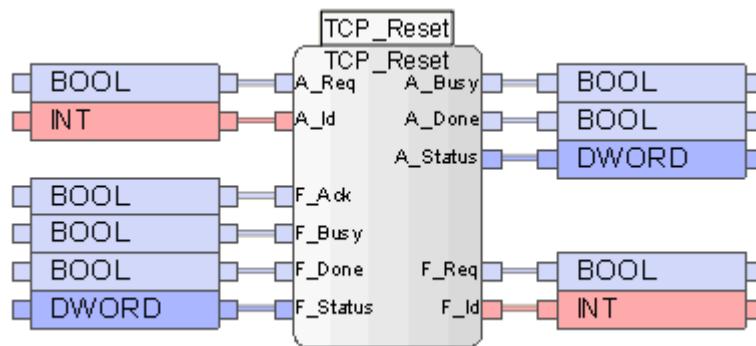


Fig. 73: Bloque funcional TCP\_Reset

Con el bloque funcional **TCP Reset** podrá restaurarse una conexión perturbada cuando un bloque funcional Send o Receive indique un error de TIMEOUT (16#8A).

- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo "A" quiere decir "Application".

Entradas A	Descripción	Tipo
A_Req	El flanko positivo da inicio al bloque	BOOL
A_Id	N. <sup>o</sup> de <i>ID</i> de la conexión TCP perturbada que se desea reinicializar.	INT

Tabla 225: Entradas A del bloque funcional TCP\_Reset

Salidas A	Descripción	Tipo
A_Busy	TRUE: La reinicialización del bloque funcional está aún por finalizar.	BOOL
A_Done	TRUE: Operación de envío finalizada sin errores	BOOL
A_Status	En la salida <i>A_Status</i> se señala el estado y el código de error del bloque funcional y de la conexión TCP.	DWORD

Tabla 226: Salidas A del bloque funcional TCP\_Reset

### Entradas y salidas del bloque funcional con el prefijo F:

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional **Reset** en la estructura en árbol. El prefijo “F” quiere decir “Field”.



La conexión del bloque funcional **RESET** en el árbol (en la carpeta de bloques funcionales) al bloque funcional **TCP\_Reset** (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá crearlas previamente el usuario en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **TCP\_Reset** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **Reset** en el árbol.

F-Inputs	Tipo
F_Ack	BOOL
F_Busy	BOOL
F_Done	BOOL
F_Status	DWORD

Tabla 227: Entradas F del bloque funcional TCP\_Reset

Conecte las *F-Outputs* del bloque funcional **TCP\_Reset** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **Reset** en el árbol.

F-Outputs	Tipo
F_Req	BOOL
F_Id	DWORD

Tabla 228: Salidas F del bloque funcional TCP\_Reset

### Para crear el bloque funcional RESET correspondiente en la estructura de árbol:

1. Abra **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **Reset**.
3. Haga clic con el botón derecho en el bloque funcional **Reset** y seleccione **Edit**  
 Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **Reset** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **TCP\_Reset** en el programa de usuario.

Entradas	Tipo
ID	DWORD
REQ	BOOL

Tabla 229: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **Reset** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **TCP\_Reset** en el programa de usuario.

Salidas	Tipo
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Tabla 230: Variables de salida del sistema

**Para hacer uso del bloque funcional TCP\_Reset deben seguirse estos pasos:**

1. Aplique en el programa del usuario en la entrada *A\_ID* el número de ID de la conexión TCP perturbada.
2. Ponga en el programa de usuario la entrada *A\_Req* en TRUE.



El bloque funcional reacciona a un cambio de flanco positivo en *A\_Req*.

---

La salida *A\_Busy* será TRUE hasta que se envíe un RESET a la conexión TCP definida. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Done* a TRUE.

### 8.7.2 TCP\_Send

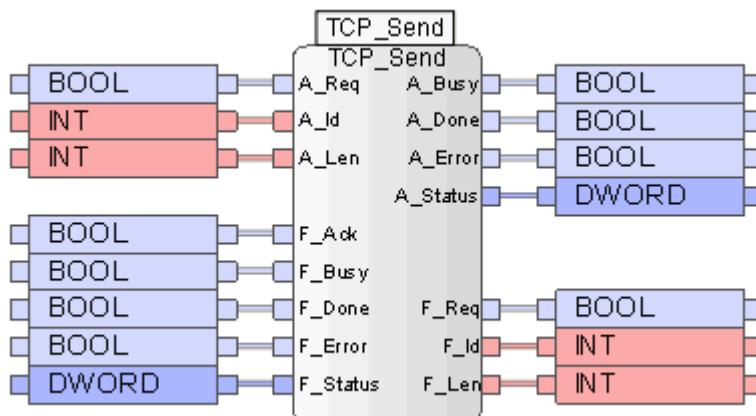


Fig. 74: Bloque funcional TCP\_Send

El bloque funcional **TCP\_Send** sirve para el envío acíclico de variables a un interlocutor de comunicación. En el interlocutor de comunicación deberá configurarse un bloque funcional (p. ej. *Receive*) con iguales variables y offsets.

- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo "A" quiere decir "Application".

Entradas A	Descripción	Tipo
A_Req	El flanco positivo da inicio al bloque.	BOOL
A_Id	N.º de ID de la conexión TCP configurada al interlocutor de comunicación al que se desea enviar los datos.	INT
A_Len	Cantidad de las variables a enviar, en bytes. A_Len debe ser mayor que cero y no terminar dentro de una variable.	INT

Tabla 231: Entradas A del bloque funcional TCP\_Send

Salidas A	Descripción	Tipo
A_Busy	TRUE: No ha finalizado aún el envío.	BOOL
A_Done	TRUE: Operación de envío finalizada sin errores	BOOL
A_Error	TRUE: Se ha producido un error FALSE: Sin errores	BOOL
A_Status	En la salida A_Status se señala el estado y el código de error del bloque funcional y de la conexión TCP.	DWORD

Tabla 232: Salidas A del bloque funcional TCP\_Send

### Entradas y salidas del bloque funcional con el prefijo F:

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional **Send** en la estructura en árbol. El prefijo "F" quiere decir "Field".



La conexión del bloque funcional **Send** en el árbol (en la carpeta de bloques funcionales) al bloque funcional **TCP\_Send** (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá crearlas previamente el usuario en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **TCP\_Send** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **Send** en el árbol.

F-Inputs	Tipo
F_Ack	BOOL
F_Busy	BOOL
F_Done	BOOL
F_Error	BOOL
F_Status	DWORD

Tabla 233: Entradas F del bloque funcional TCP\_Send

Conecte las *F-Outputs* del bloque funcional **TCP\_Send** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **Send** en el árbol.

F-Outputs	Tipo
F_Id	DWORD
F_Len	INT
F_Req	BOOL

Tabla 234: Salidas F del bloque funcional TCP\_Send

### Para crear el bloque funcional **Send** correspondiente en la estructura de árbol:

1. Abra **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **Send**.
3. Haga clic con el botón derecho en el bloque funcional **Send** y seleccione **Edit**
  - Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **Send** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **TCP\_Send** en el programa de usuario.

Entradas	Tipo
ID	DWORD
LEN	INT
REQ	BOOL

Tabla 235: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **Send** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **TCP\_Send** en el programa de usuario.

Salidas	Tipo
Ack	BOOL
Busy	BOOL
Done	BOOL
ERROR	BOOL
STATUS	DWORD

Tabla 236: Variables de salida del sistema

Datos	Descripción
Datos de envío	En la ficha <i>Process Variables</i> pueden crearse las variables que se quiera. Sin embargo, los offsets y los tipos de variable deberán ser idénticos a los offsets y los tipos de las variables del interlocutor de comunicación.

Tabla 237: Datos de envío

**Para hacer uso del bloque funcional TCP\_Send deben seguirse estos pasos:**



Las variables a enviar deberán crearse en la ficha “*Process Variables*” del cuadro de diálogo “*Send*”. Los offsets y los tipos de variable deberán ser idénticos a los offsets y los tipos de las variables del interlocutor de comunicación.

1. Aplique en el programa de usuario el ID de conexión TCP a la entrada *A\_ID*.
2. Aplique en el programa de usuario la longitud de las variables a enviar en la entrada *A\_Len*.
3. Ponga en el programa de usuario la entrada *A\_Req* en TRUE.



El bloque funcional reacciona a un cambio de flanco positivo en *A\_Req*.

La salida *A\_Busy* será TRUE hasta que se envíen las variables. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Done* a TRUE.

Si no pudo realizarse correctamente la operación de envío, la salida *A\_Error* cambiará a TRUE y se generará un código de error en la salida *A\_Status*.

### 8.7.3 TCP\_Receive

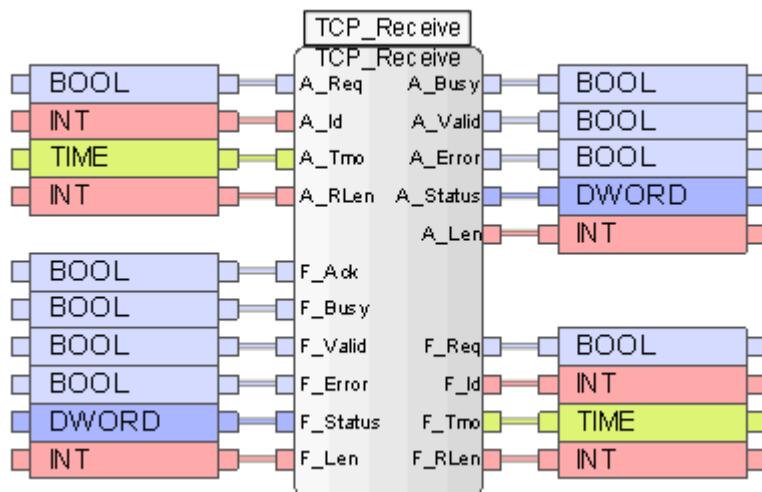


Fig. 75: Bloque funcional TCP\_Receive

El bloque funcional **TCP\_Receive** sirve para la recepción definida de variables de un interlocutor de comunicación.

En el interlocutor de comunicación deberá configurarse un bloque funcional (p. ej. **TCP\_Send**) con iguales variables y offsets.

- 
- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).
- 

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo "A" quiere decir "Application".

Entradas A	Descripción	Tipo
<b>A_Req</b>	El flanco positivo da inicio al bloque funcional.	BOOL
<b>A_Id</b>	N. <sup>o</sup> de ID de la conexión TCP configurada al interlocutor de comunicación desde el que se desea recibir los datos.	INT
<b>A_Tmo</b>	Time-out de recepción. Si dentro de este tiempo no se reciben datos, el bloque se cerrará con un mensaje de error. Si no se asigna la entrada <i>A_Tmo</i> o se define como cero, el tiempo de time-out estará desactivado.	TIME
<b>A_RLen</b>	<i>A_RLen</i> es la longitud esperada de las variables a recibir, en bytes. <i>A_RLen</i> debe ser mayor que cero y no terminar dentro de una variable.	INT

Tabla 238: Entradas A del bloque funcional TCP\_Receive

Salidas A	Descripción	Tipo
A_Busy	TRUE: La recepción de los datos está aún por finalizar.	BOOL
A_Valid	TRUE: Recepción de datos finalizada sin errores.	BOOL
A_Error	TRUE: Se ha producido un error FALSE: Sin errores	BOOL
A_Status	En la salida A_Status se señalan el estado y el código de error del bloque funcional y de la conexión TCP.	DWORD
A_Len	Cantidad de bytes recibidos.	INT

Tabla 239: Salidas A del bloque funcional TCP\_Receive

**Entradas y salidas del bloque funcional con el prefijo F:**

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional **Receive** en la estructura en árbol. El prefijo "F" quiere decir "Field".



La conexión del bloque funcional **Receive** en el árbol (en la carpeta de bloques funcionales) al bloque funcional **TCP\_Receive** (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá crearlas previamente el usuario en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **TCP\_Receive** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **Receive** en el árbol.

F-Inputs	Tipo
F_Ack	BOOL
F_Busy	BOOL
F_Valid	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Tabla 240: Entradas A del bloque funcional TCP\_Receive

Conecte las *F-Outputs* del bloque funcional **TCP\_Receive** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **Receive** en el árbol.

F-Outputs	Tipo
F_Req	BOOL
F_Id	DWORD
F_Tmo	INT
F_RLen	INT

Tabla 241: Salidas F del bloque funcional TCP\_Receive

**Para crear el bloque funcional Receive correspondiente en la estructura de árbol:**

1. Abra **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **Receive**.
3. Haga clic con el botón derecho en el bloque funcional **Receive** y seleccione **Edit**
  - Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **Receive** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **TCP\_Receive** en el programa de usuario.

Entradas	Tipo
ID	INT
REQ	BOOL
RLEN	INT
TIMEOUT	TIME

Tabla 242: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **Receive** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **TCP\_Receive** en el programa de usuario.

Salidas	Tipo
Ack	BOOL
Busy	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Tabla 243: Variables de salida del sistema

Datos	Descripción
Variables de recepción	En la ficha <i>Process Variables</i> pueden crearse las variables que se quiera. Sin embargo, los offsets y los tipos de variable deberán ser idénticos a los offsets y los tipos de las variables del interlocutor de comunicación.

Tabla 244: Variables de recepción

**Para hacer uso del bloque funcional TCP\_Receive deben seguirse estos pasos:**

- 
- i** Las variables de recepción deberán crearse en la ficha “*Process Variables*” del cuadro de diálogo “*Receive*”. Sin embargo, los offsets y los tipos de las variables de recepción deberán ser idénticos a los offsets y los tipos de las variables de envío del interlocutor de comunicación.
- 

1. Aplique en el programa del usuario en la entrada *A\_ID* el número de ID de la conexión TCP.
  2. Aplique en el programa de usuario el tiempo time-out de recepción en la entrada *A\_Tmo*.
  3. Aplique en el programa de usuario la longitud esperada de las variables de recepción en la entrada *A\_RLen*.
  4. Ponga en el programa de usuario la entrada *A\_Req* en TRUE.
- 

- i** El bloque funcional se inicia con un cambio de flanco positivo en *A\_Req*.
- 

La salida *A\_Busy* será TRUE hasta que se reciban las variables o expire el tiempo de time-out de recepción. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Valid* o *A\_Error* a TRUE.

Si las variables se reciben sin errores, la salida *A\_Valid* cambiará a TRUE. Las variables definidas en la ficha “*Data*” podrán evaluarse.

Si las variables no se reciben exentas de errores, la salida *A\_Error* cambiará a TRUE en la salida *A\_Status* se señalizará un código de error.

#### 8.7.4 TCP\_ReceiveLine

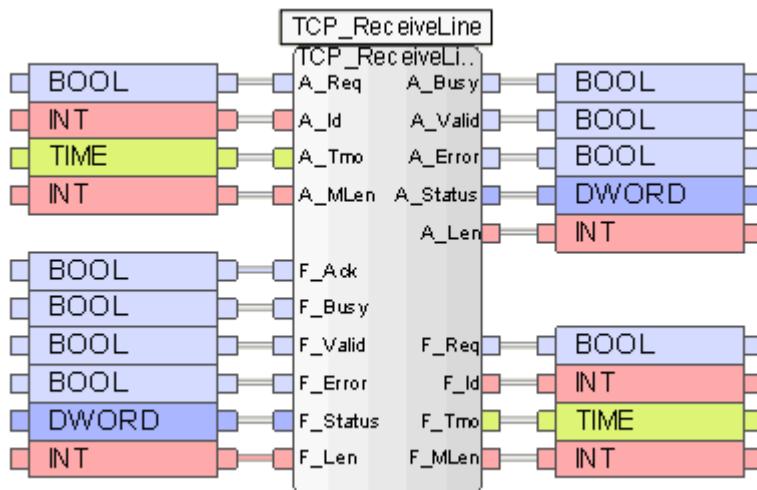


Fig. 76: Bloque funcional TCP\_ReceiveLine

El bloque funcional **TCP\_ReceiveLine** sirve para la recepción de una cadena de caracteres ASCII incluido LineFeed (16#0A) de un interlocutor de comunicación.

- i Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

#### Entradas y salidas del bloque funcional con el prefijo A:

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo “A” quiere decir “Application”.

Entradas A	Descripción	Tipo
A_Req	El flanco positivo da inicio al bloque y pone la conexión lista para recibir	BOOL
A_Id	N.º de ID de la conexión TCP configurada al interlocutor de comunicación desde el que se desea recibir los datos.	INT
A_Tmo	Time-out de recepción Si dentro de este tiempo no se reciben datos, el bloque se cerrará con un mensaje de error. Si la entrada se deja abierta o se le asigna un cero, el tiempo de time-out estará desactivado.	TIME
A_MLen	<i>A_Mlen</i> es la longitud máxima de una variable a recibir, en bytes. Las variables de recepción deberán crearse en la ficha “Data” en el bloque funcional COM. Bytes transmitidos = Min (A_MLen, longitud de línea, longitud del área de datos).	INT

Tabla 245: Entradas A del bloque funcional TCP\_ReceiveLine

Salidas A	Descripción	Tipo
A_Busy	TRUE: La recepción de los datos está aún por finalizar.	BOOL
A_Valid	TRUE: Recepción de datos finalizada sin errores.	BOOL
A_Error	TRUE: Se ha producido un error FALSE: Sin errores	BOOL
A_Status	En la salida A_Status se señala el estado y el código de error del bloque funcional y de la conexión TCP.	DWORD
A_Len	Cantidad de bytes recibidos.	INT

Tabla 246: Salidas A del bloque funcional TCP\_ReceiveLine

**Entradas y salidas del bloque funcional con el prefijo F:**

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional **ReceiveLine** en la estructura en árbol. El prefijo “F” quiere decir “Field”.



La conexión del bloque funcional **ReceiveLine** en el árbol (en la carpeta de bloques funcionales) al bloque funcional **TCP\_ReceiveLine** (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá crearlas previamente el usuario en el editor de variables.

Conecte las *F-Inputs* del bloque funcional **TCP\_ReceiveLine** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **ReceiveLine** en el árbol.

F-Inputs	Tipo
F_Ack	BOOL
F_Busy	BOOL
F_Valid	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Tabla 247: Entradas F del bloque funcional TCP\_ReceiveLine

Conecte las *F-Outputs* del bloque funcional **TCP\_ReceiveLine** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **ReceiveLine** en el árbol.

F-Outputs	Tipo
A_Req	BOOL
A_Id	INT
A_Tmo	TIME
A_MLen	INT

Tabla 248: Salidas F del bloque funcional TCP\_ReceiveLine

**Para crear el bloque funcional **ReceiveLine** correspondiente en la estructura de árbol:**

1. Abra **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **ReceiveLine**.
3. Haga clic con el botón derecho en el bloque funcional **ReceiveLine** y seleccione **Edit**  
 Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **ReceiveLine** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **TCP\_ReceiveLine** en el programa de usuario.

Entradas	Tipo
ID	INT
MLEN	INT
REQ	BOOL
TIMEOUT	TIME

Tabla 249: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **ReceiveLine** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **TCP\_ReceiveLine** en el programa de usuario.

Salidas	Tipo
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Tabla 250: Variables de salida del sistema

Datos	Descripción
Variables de recepción	En la ficha <i>Process Variables</i> deberían crearse convenientemente sólo variables del tipo BYTE. Los offsets de las variables deberán ser idénticos a los offsets de las variables del interlocutor de comunicación.

Tabla 251: Variables de recepción

**Para hacer uso del bloque funcional TCP\_ReceiveLine deben seguirse estos pasos:**

---

**i**

Las variables de recepción del tipo BYTE deberán crearse en la ficha *Process Variables* del cuadro de diálogo *ReceiveLine*. Los offsets de las variables de recepción deberán ser idénticos a los offsets de las variables de envío del interlocutor de comunicación.

---

1. Aplique en el programa del usuario en la entrada *A\_ID* el número de ID de la conexión TCP.
  2. Aplique en el programa de usuario el tiempo time-out de recepción en la entrada *A\_Tmo*.
  3. Aplique en el programa de usuario la máxima longitud de las variables a recibir en la entrada *A\_MLen*.
- 

**i**

*A\_MLen* debe ser mayor que cero y determina el tamaño del búfer de recepción, en bytes. Si el búfer de recepción se llena sin haber llegado aún al final de la línea, la operación de lectura finalizará sin mensaje de error.

En la salida *A\_Len* se proporciona la cantidad de los bytes recibidos:

Bytes recibidos = Min (*A\_MLen*, longitud de línea, longitud del área de datos).

---

4. Ponga en el programa de usuario la entrada *A\_Req* en TRUE.
- 

**i**

El bloque funcional reacciona a un cambio de flanco positivo en *A\_Req*.

---

La salida *A\_Busy* será TRUE, hasta que el búfer de recepción esté lleno o se reciba el final de línea *LineFeed* o haya expirado el tiempo de time-out de recepción. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Valid* o *A\_Error* a TRUE.

Si la línea se recibe sin errores, la salida *A\_Valid* cambiará a TRUE. Las variables definidas en la ficha "Data" podrán evaluarse.

Si la recepción de la línea no está exenta de errores, la salida *A\_Error* cambiará a TRUE y en la salida *A\_Status* se señalizará un código de error.

### 8.7.5 TCP\_ReceiveVar

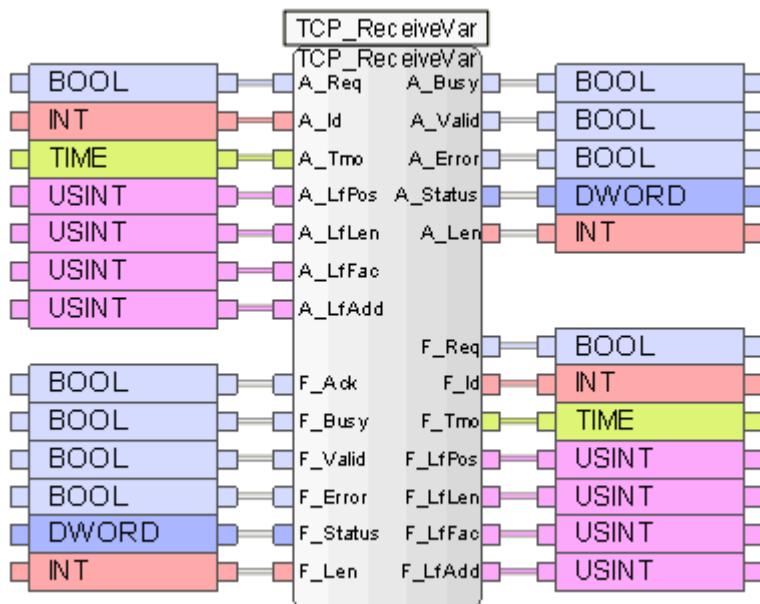


Fig. 77: Bloque funcional TCP\_ReceiveVar

Con el bloque funcional **TCP\_ReceiveVar** podrán evaluarse paquetes de datos de longitud variable que estén dotados de un campo de longitud.

- i** Para configurar, arrastre el bloque funcional con el ratón desde la biblioteca de bloques hasta el programa de usuario (véase también el capítulo 13.1).

#### Descripción de la función

Los paquetes de datos recibidos deberán tener la composición ilustrada en la figura (p. ej. protocolo Modbus). La adaptación a un formato de protocolo cualquiera se realiza mediante el ajuste de los parámetros de introducción *A\_LfPos*, *A\_LfLen*, *A\_LfFac*, *A\_LfLen*.

El paquete de datos recibido consta de un área de cabecera y un área de datos útiles. La cabecera contiene datos tales como dirección de participante, función de telegrama, campo de longitud, etc., que son necesarios para la conexión de comunicación. Para evaluar el área de datos útiles deberá separarse el área de cabecera y leerse el campo de longitud.

El tamaño del área de cabecera se introduce en el parámetro *A\_LfAdd*.

La longitud del área de datos útiles deberá leerse del campo de longitud del paquete de datos actualmente leído. La posición del campo de longitud se introduce en el parámetro *A\_LfPos*. El tamaño del campo de longitud se introduce en *LfLen*, en bytes. Si la longitud no se especifica en bytes, deberá introducirse el factor de conversión respectivo en *A\_LfFac* (p. ej. 2 para Word o 4 para Double Word).

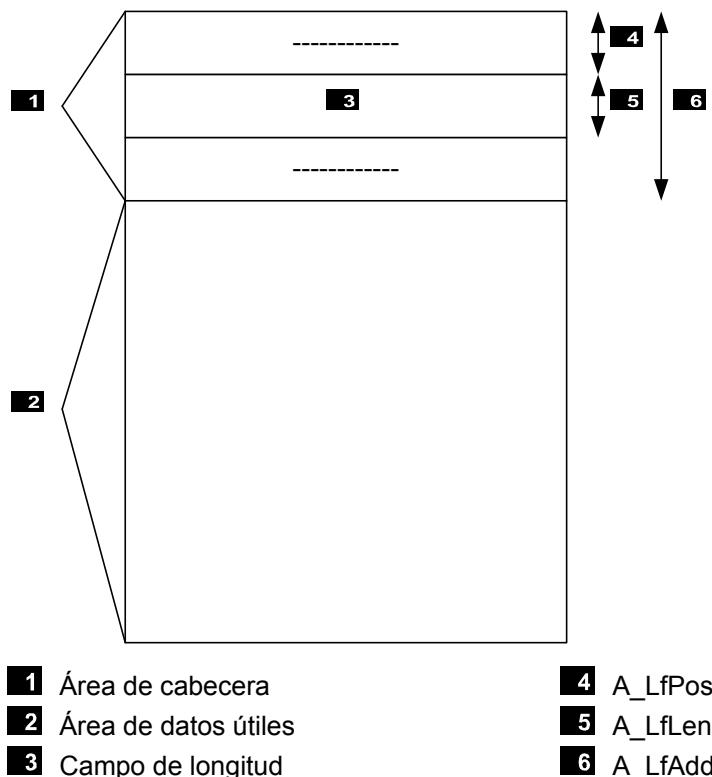


Fig. 78: Composición de los paquetes de datos

### Entradas y salidas del bloque funcional con el prefijo A

Mediante estas entradas y salidas podrá Ud. evaluar y controlar el bloque funcional con ayuda del programa de usuario. El prefijo "A" quiere decir "Application".

Entradas A	Descripción	Tipo
A_Req	Con el flanco positivo se da inicio al bloque funcional COM	BOOL
A_Id	N. <sup>o</sup> de <i>ID</i> de la conexión TCP configurada a un interlocutor de comunicación desde el que se desea recibir los datos.	DWORD
A_Tmo	Time-out de recepción Si dentro de este tiempo no se reciben datos, el bloque se cerrará con un mensaje de error. Si la entrada se deja abierta o se le asigna un cero, el tiempo de time-out estará desactivado.	INT
A_LfPos	Posición inicial del campo de longitud en el paquete de datos; la numeración empieza por cero (medida en bytes).	USINT
A_LfLen	Tamaño del campo de longitud <i>A_LfLen</i> en bytes. Se permiten 1, 2 ó 4 bytes.	USINT
A_LfFac	Factor de conversión en bytes, en caso de que el registro del campo de longitud no sea en bytes. Si la entrada se deja abierta o se le asigna un cero, se adoptará el valor 1 como valor predeterminado por defecto.	USINT
A_LfAdd	Tamaño de la cabecera, en bytes	USINT

Tabla 252: Entradas A del bloque funcional TCP\_ReceiveVar

Salidas A	Descripción	Tipo
A_Busy	TRUE: La recepción de los datos está aún por finalizar.	BOOL
A_Valid	TRUE: Recepción de datos finalizada sin errores.	BOOL
A_Error	TRUE: Se ha producido un error durante la lectura FALSE: Sin errores	BOOL
A_Status	En la salida A_Status se señala el estado y el código de error del bloque funcional y de la conexión TCP.	DWORD
A_Len	Cantidad de bytes recibidos.	INT

Tabla 253: Salidas A del bloque funcional TCP\_ReceiveVar

**Entradas y salidas del bloque funcional con el prefijo F:**

Estas entradas y salidas del bloque funcional representan la conexión al bloque funcional **ReceiveVar** en la estructura en árbol. El prefijo "F" quiere decir "Field".

- 
- i** La conexión del bloque funcional **ReceiveVar** en el árbol (en la carpeta de bloques funcionales) al bloque funcional **TCP\_ReceiveVar** (en el programa de usuario) se realiza mediante variables comunes. Éstas deberá crearlas previamente el usuario en el editor de variables.
- 

Conecte las *F-Inputs* del bloque funcional **TCP\_ReceiveVar** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las salidas del bloque funcional **ReceiveVar** en el árbol.

F-Inputs	Tipo
F_Ack	BOOL
F_Busy	BOOL
F_Valid	BOOL
F_Error	BOOL
A_Status	DWORD
A_Len	INT

Tabla 254: Entradas F del bloque funcional TCP\_ReceiveVar

Conecte las *F-Outputs* del bloque funcional **TCP\_ReceiveVar** en el programa de usuario a las mismas variables a las que más tarde se conectarán también las entradas del bloque funcional **ReceiveVar** en el árbol.

F-Outputs	Tipo
F_Req	BOOL
F_Id	INT
F_Tmo	TIME
F_LfPos	USINT
A_LfLen	USINT
A_LfFac	USINT
A_LfAdd	USINT

Tabla 255: Salidas F del bloque funcional TCP\_ReceiveVar

**Para crear el bloque funcional ReceiveVar correspondiente en la estructura de árbol:**

1. Abra **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New** en el árbol.
2. Seleccione el bloque funcional **ReceiveVar**.
3. Haga clic con el botón derecho en el bloque funcional **ReceiveVar** y seleccione **Edit**  
 Se abrirá la asignación de variables al bloque funcional.

Conecte las entradas del bloque funcional **ReceiveVar** en el árbol a las mismas variables a las que antes conectó también las *F-Outputs* del bloque funcional **TCP\_ReceiveVar** en el programa de usuario.

Entradas	Tipo
ID	INT
Lf Add	USINT
Lf Fac	USINT
Lf Len	USINT
Lf Pos	USINT
REQ	BOOL
TIMEOUT	TIME

Tabla 256: Variables de entrada del sistema

Conecte las siguientes salidas del bloque funcional **ReceiveVar** en el árbol a las mismas variables a las que antes conectó también las *F-Inputs* del bloque funcional **TCP\_ReceiveVar** en el programa de usuario.

Salidas	Tipo
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Tabla 257: Variables de salida del sistema

Datos	Descripción
Variables de recepción	En la ficha <i>Process Variables</i> pueden crearse las variables que se quiera. Sin embargo, los offsets y los tipos de variable deberán ser idénticos a los offsets y los tipos de las variables del interlocutor de comunicación.

Tabla 258: Variables de recepción

**Para hacer uso del bloque funcional TCP\_ReceiveVar deben seguirse estos pasos:**

Las variables de recepción deberán crearse en la ficha “*Process Variables*” del cuadro de diálogo “*ReceiveVar*”. Sin embargo, los offsets y los tipos de las variables de recepción deberán ser idénticos a los offsets y los tipos de las variables de envío del interlocutor de comunicación.

---

1. Aplique en el programa del usuario en la entrada *A\_ID* el número de ID de la conexión TCP.
  2. Aplique en el programa de usuario el tiempo time-out de recepción en la entrada *A\_Tmo*.
  3. En el programa del usuario aplique los parámetros *A\_LfPos*, *A\_LfLen*, *A\_LfFac* y *A\_LfAdd*.
  4. Ponga en el programa del usuario la entrada *A\_Req* en TRUE.
- 



El bloque funcional se inicia con un cambio de flanco positivo en *A\_Req*.

---

La salida *A\_Busy* será TRUE hasta que se reciban las variables o expire el tiempo de time-out de recepción. Luego cambian las salidas *A\_Busy* a FALSE y *A\_Valid* o *A\_Error* a TRUE.

Si las variables se reciben sin errores, la salida *A\_Valid* cambiará a TRUE. Las variables definidas en la ficha “*Data*” podrán evaluarse. La salida *A\_Len* contiene la cantidad de bytes que se han leído realmente.

Si las variables no se reciben exentas de errores, la salida *A\_Error* cambiará a TRUE en la salida *A\_Status* se señalizará un código de error.

## 8.8 Panel de control (Send/Receive over TCP)

El usuario podrá comprobar y controlar la configuración del protocolo Send/Receive en el panel de control. Además, allí se muestra información actual de estado del protocolo Send/Receive (p. ej. conexiones perturbadas, etc.).

### Para abrir el panel de control para supervisar el protocolo Send/Receive:

1. Seleccione **Resource** en el árbol.
2. En la **Action Bar**/barra de acciones seleccione “Online”.
3. Escriba sus datos de ingreso en **System Login** para abrir el panel de control del recurso.
4. Seleccione **Send/Receive Protocol** en el árbol del panel de control.

### 8.8.1 Recuadro de lectura de parámetros generales

En el recuadro de lectura se mostrarán los siguientes valores del protocolo Send/Receive.

Elemento	Descripción
Name	Protocolo TCP SR
μP load (planned) [%]	Véase el capítulo 7.2.3.2.
μP load (actual) [%]	
Undisturbed Connections	Cantidad de las conexiones no perturbadas
Disturbed Connections	Cantidad de las conexiones perturbadas

Tabla 259: Recuadro de lectura “S&R Protocol”

### 8.8.2 Recuadro de lectura de conexiones TCP

En este recuadro de lectura se mostrarán los siguientes valores de las conexiones TCP seleccionadas.

Elemento	Descripción
Name	Conexión TCP
Partner Timeout	Sí: Tiempo “Partner Request Timeout” expirado. No: Tiempo “Partner Request Timeout” no expirado.
Connection State	Estado actual de esta conexión 0x00: conexión OK 0x01: conexión cerrada 0x02: servidor en espera a que se establezca conexión 0x04: el cliente intenta establecer conexión 0x08: conexión bloqueada
Peer Address	Dirección IP del interlocutor de comunicación.
Peer Port	Puerto del interlocutor de comunicación.
Own Port	Puerto de este sistema de control.
WatchDog Time [ms]	Es el tiempo actual Partner Request Timeout, dentro del cual se habrá recibido al menos un envío de datos por parte del interlocutor de comunicación.
Error Code	Código de error (véase el capítulo 8.8.3).
Timestamp Error Code [ms]	Marca de tiempo del último error comunicado Rango de valores: Segundos desde 1-1-1970, en milisegundos.
Received Bytes [Bytes]	Cantidad de bytes recibidos de esta conexión TCP.
Transmitted Bytes [Bytes]	Cantidad de bytes enviados de esta conexión TCP.

Tabla 260: Recuadro de Modbus Slave

### 8.8.3 Código de error de la conexión TCP

Los códigos de error pueden leerse de la variable *Error Code*.

Por conexión configurada: “Connection State” se compone del estado de la conexión y del código de error de la última operación.

Código de error decimal	Código de error hexadecimal	Descripción
0	16#00	OK
4	16#04	Llamada del sistema cancelada
5	16#05	Error de E/S
6	16#06	Dispositivo desconocido
9	16#09	“Socket descriptor” no válido
12	16#0C	No hay más memoria
13	16#0D	Acceso denegado
14	16#0E	Dirección no válida
16	16#10	Dispositivo ocupado
22	16#16	Valor no válido (p. ej. en el campo de longitud)
23	16#17	Tabla del descriptor llena
32	16#20	Conexión interrumpida
35	16#23	Operación bloqueada
36	16#24	Operación ahora en ejecución
37	16#25	Operación ya en ejecución
38	16#27	Dirección de destino necesaria
39	16#28	Mensaje demasiado largo
40	16#29	Tipo de protocolo erróneo para socket
42	16#2A	Protocolo no disponible
43	16#2B	Protocolo no compatible
45	16#2D	Operación en socket no compatible
47	16#2F	Dirección del protocolo no compatible
48	16#30	La dirección está ya en uso
49	16#31	La dirección no puede asignarse
50	16#32	La red no funciona
53	16#35	El software ha cancelado la conexión
54	16#36	Conexión reinicializada por el interlocutor
55	16#37	No hay mas memoria búfer disponible
56	16#38	Socket ya conectado
57	16#39	Socket no conectado
58	16#3A	Socket cerrado
60	16#3C	Tiempo para la operación expirado
61	16#3D	Conexión rechazada (por el interlocutor)
65	16#41	No hay registro de enrutado al interlocutor
78	16#4E	Función no presente
254	16#FE	Se ha llegado a un time-out
255	16#FF	Conexión cerrada por el interlocutor

Tabla 261: Códigos de error de la conexión TCP

#### 8.8.4 Tabla de códigos de error adicionales de los bloques funcionales

Los códigos de error de los bloques funcionales se señalan sólo en A\_Status de los bloques funcionales S&R-TCP.

Código de error decimal	Código de error hexadecimal	Descripción
129	16#81	No hay ninguna conexión con ese identificador
130	16#82	La longitud es cero o demasiado grande
131	16#83	En esta conexión se permiten sólo datos cíclicos
132	16#84	Estado erróneo
133	16#85	Valor de time-out excesivo
134	16#86	Error interno del programa
135	16#87	Error de configuración
136	16#88	Datos transmitidos no acordes con el área de datos
137	16#89	Bloque funcional detenido
138	16#8A	Se ha llegado a un time-out o el envío está bloqueado
139	16#8B	En esta conexión hay un bloque de ese tipo ya activo

Tabla 262: Códigos de error adicionales

#### 8.8.5 Estado de conexión

Código de error hexadecimal	Descripción
16#00	Conexión OK
16#01	Conexión cerrada
16#02	Servidor en espera de que se establezca conexión
16#04	El cliente intenta establecer conexión
16#08	Conexión bloqueada

Tabla 263: Estado de conexión

#### 8.8.6 Partner Connection State

Estado del protocolo, decimal	Descripción
0	Sin conexión
1	Conexión OK

Tabla 264: Partner Connection State

## 9 Protocolo SNTP

(Simple Network Time Protocol)

Con el protocolo SNTP y mediante Ethernet se sincroniza la hora del cliente SNTP con el servidor SNTP.

Los sistemas de control HIMax/HIMatrix pueden configurarse y usarse como **SNTP server** y/o como **SNTP client**. Rige la normativa de SNTP según RFC 2030 (versión 4 de SNTP), con la limitación de que sólo es compatible el modo Unicast.

### Equipamiento y sistema necesarios:

Elemento	Descripción
Controller	HIMax con módulo COM o sólo módulo CPU HIMatrix a partir de V7 de S.Op. de CPU y V12 de S.Op. de COM
Activation	Esta función se sirve habilitada por defecto en todos los sistemas HIMax/HIMatrix.
Interface	Ethernet 10/100/1000BaseT

Tabla 265: Equipamiento y sistema necesarios para S&R TCP

### 9.1 Cliente SNTP

Para sincronizarse, el cliente SNTP usará siempre sólo el servidor SNTP de mayor prioridad.

En cada recurso podrá configurarse un cliente SNTP para la sincronización.

- 
- i** Sincronización de una E/S remota mediante un sistema de control HIMax.  
Si se configura un cliente SNTP en un sistema de control HIMax, se desactivará el servidor SNTP interno del sistema de control HIMax.  
Para seguir garantizando la sincronización de la E/S remota por parte del sistema de control HIMax, deberá configurarse un servidor SNTP en el módulo de comunicación HIMax al que esté conectado la E/S remota.
- 

#### Para crear un nuevo cliente SNTP:

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Haga clic con el botón derecho en **Protocols** y seleccione **New, SNTP Client** en el menú contextual.  
 Se agregará un nuevo cliente SNTP.
3. En el menú contextual de SNTP Client **Properties** seleccione **COM Module**.

El cuadro de diálogo del cliente SNTP contiene los siguientes parámetros.

Elemento	Descripción				
Type	Cliente SNTP				
Name	Nombre para el cliente SNTP, máximo 32 caracteres.				
Module	Selección del módulo COM o CPU en el que se ejecuta este protocolo.				
Behavior on CPU/COM Connection Loss	<p>En caso de cortarse la conexión del módulo procesador al módulo de comunicación, se inicializarán las variables de entrada o se usarán sin variar en el módulo procesador, según lo que se defina en este parámetro (p. ej. si se extrae el módulo de comunicación con la comunicación en curso).</p> <table> <tr> <td>Adopt initial data</td><td>Las variables de entrada se reinicializarán a sus valores iniciales.</td></tr> <tr> <td>Retain Last Value</td><td>Las variables de entrada conservarán su último valor.</td></tr> </table>	Adopt initial data	Las variables de entrada se reinicializarán a sus valores iniciales.	Retain Last Value	Las variables de entrada conservarán su último valor.
Adopt initial data	Las variables de entrada se reinicializarán a sus valores iniciales.				
Retain Last Value	Las variables de entrada conservarán su último valor.				
Activate Max. $\mu$ P Budget	El sistema operativo del módulo no lo tiene en cuenta. Parámetro conservado por razones de estabilidad de Reload y CRC.				
Max. $\mu$ P budget in [%]	El sistema operativo del módulo no lo tiene en cuenta. Parámetro conservado por razones de estabilidad de Reload y CRC.				
Description	Cualquier descripción inequívoca para el SNTP.				
Current SNTP Version	Indicación de la actual versión de SNTP.				
Reference Stratum	<p>El estrato de un cliente SNTP indica la exactitud de su hora local. Cuanto menor sea el estrato, más exacta será su hora local. Cero significa estrato no especificado o no disponible (no válido). El servidor SNTP actualmente utilizado de un cliente SNTP es el de mayor prioridad de aquellos a los que puede accederse.</p> <p>Si el estrato del servidor SNTP actual es menor que el del cliente SNTP, el recurso adoptará la hora del servidor SNTP actual.</p> <p>Si el estrato del servidor SNTP actual es mayor que el del cliente SNTP, el recurso no adoptará la hora del servidor SNTP actual.</p> <p>Si el estrato del servidor SNTP actual es igual que el del cliente SNTP habrá que distinguir entre dos casos posibles:</p> <ul style="list-style-type: none"> <li>▪ Si el cliente SNTP (recurso) opera exclusivamente como cliente SNTP, el recurso adoptará la hora del servidor SNTP actual.</li> <li>▪ Si el cliente SNTP (recurso) opera al mismo tiempo también como servidor SNTP, en el recurso se adoptará, por cada requerimiento del cliente SNTP, la mitad de la diferencia horaria actual respecto al servidor SNTP actual (aproximación paulatina de la hora).</li> </ul> <p>Rango de valores: 1...16 Valor por defecto: 15</p>				
Client Time Request Interval [s]	<p>Intervalo dentro del cual se produce la sincronización con el servidor SNTP actual.</p> <p>El intervalo para requerimiento del cliente SNTP deberá ser mayor que el tiempo de time-out del servidor SNTP.</p> <p>Rango de valores: 16 s...16384 s Valor por defecto: 16</p>				

Tabla 266: Propiedades del cliente SNTP

## 9.2 Cliente SNTP (información del servidor)

En la información del servidor SNTP se configura la conexión a un servidor SNTP.

Bajo un cliente SNTP pueden configurarse entre 1 y 4 informaciones de servidor SNTP.

**Para crear una nueva información de servidor SNTP:**

1. Abra **Configuration, Resource, Protocols, SNTP Client** en el árbol.
2. Haga clic con el botón derecho en **Protocols** y seleccione **New, SNTP Server Info** en el menú contextual.  
 Se agregará una nueva información de servidor SNTP.
3. En el menú contextual de SNTP Server Info **Properties** seleccione **COM Module**.

El cuadro de diálogo “SNTP Server Info” contiene los siguientes parámetros.

Elemento	Descripción
Type	Información de servidor SNTP.
Name	Nombre para la información de servidor SNTP. Máximo 31 caracteres.
Description	Descripción del servidor SNTP. Máximo 31 caracteres.
IP Address	Dirección IP del recurso o del PC en el que está configurado el servidor SNTP. Valor por defecto: 0.0.0.0
SNTP Server Priority	Prioridad con la que el cliente SNTP trata este servidor SNTP. Los servidores SNTP configurados para un cliente SNTP deberían poseer diferentes grados de prioridad. Rango de valores: 0 (prioridad más baja) hasta 4294967295 (prioridad más alta) Valor por defecto: 1
SNTP Server Timeout[s]	El tiempo de time-out ajustado en el servidor SNTP deberá ser menor que el tiempo <i>Time Request Interval</i> ajustado en el cliente SNTP. Rango de valores: 1 s...16384 s Valor por defecto: 1

Tabla 267: Propiedades de la información de servidor SNTP

## 9.3 Servidor SNTP

El servidor SNTP recibe el requerimiento de un cliente SNTP y responde comunicando una nueva hora actual al cliente SNTP.

**Para crear un nuevo servidor SNTP:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Haga clic con el botón derecho en **Protocols** y seleccione **New, SNTP Server** en el menú contextual.  
 Se agregará un nuevo servidor SNTP.
3. En el menú contextual de SNTP Server **Properties** seleccione **COM Module**.

El cuadro de diálogo del servidor SNTP contiene los siguientes parámetros.

Elemento	Descripción				
Type	Servidor SNTP				
Name	Nombre para el servidor SNTP, máximo 31 caracteres.				
Module	Selección del módulo COM o CPU en el que se ejecuta este protocolo.				
Activate Max. µP Budget	<p>Activado: Aplicar el límite µP-Budget tomado del recuadro “Max. µP-Budget in [%]”.</p> <p>Desactivado: No se usará límite de µP-Budget para este protocolo.</p>				
Max. µP budget in [%]	<p>Máxima carga de µP del módulo que se permite que tenga lugar al ejecutar el protocolo.</p> <p>Rango de valores: 1...100% Valor por defecto: 30%</p>				
Behavior on CPU/COM Connection Loss	<p>En caso de cortarse la conexión del módulo procesador al módulo de comunicación, se inicializarán las variables de entrada o se usarán sin variar en el módulo procesador, según lo que se defina en este parámetro (p. ej. si se extrae el módulo de comunicación con la comunicación en curso).</p> <table> <tr> <td>Adopt initial data</td> <td>Las variables de entrada se reinicializarán a sus valores iniciales.</td> </tr> <tr> <td>Retain Last Value</td> <td>Las variables de entrada conservarán su último valor.</td> </tr> </table>	Adopt initial data	Las variables de entrada se reinicializarán a sus valores iniciales.	Retain Last Value	Las variables de entrada conservarán su último valor.
Adopt initial data	Las variables de entrada se reinicializarán a sus valores iniciales.				
Retain Last Value	Las variables de entrada conservarán su último valor.				
Description	Descripción del SNTP				
Current SNTP Version	Indicación del SNTP actual				
Stratum of Timeserver	<p>El estrato de un servidor SNTP indica la exactitud de su hora local. Cuanto menor sea el estrato, más exacta será la hora local. Cero significa estrato no especificado o no disponible (no válido). El estrato de un servidor SNTP deberá ser menor o igual que el estrato del cliente SNTP que hace el requerimiento. De lo contrario, el cliente SNTP no adoptará la hora del servidor SNTP.</p> <p>Rango de valores: 1...15 Valor por defecto: 14</p>				

Tabla 268: Propiedades del servidor SNTP

## 10 Servidor X-OPC

El servidor X-OPC de HIMA sirve de interfaz de transmisión entre sistemas de control HIMax/HIMatrix y sistemas externos que dispongan de una interfaz OPC.

OPC significa Openess, Productivity & Collaboration y se basa en la tecnología desarrollada por Microsoft (COM/DCOM). Así podrán conectarse entre sí sistemas centrales, sistemas de monitorización y sistemas de control de diversos oferentes para el intercambio de datos. Véase también [www.opcfoundation.org](http://www.opcfoundation.org).

El servidor X-OPC de HIMA se ejecutará tras instalarlo en un PC como servicio de Windows.



Toda la configuración y el uso del servidor X-OPC se realizan en SILworX. En el panel de control de SILworX podrá Ud. cargar, iniciar y detener el servidor X-OPC igual que un sistema de control.

El servidor X-OPC es compatible con las siguientes especificaciones:

- **Data Access (DA) en las versiones 1.0, 2.05a y 3.0**

DA se usa para la transmisión de datos de proceso desde el sistema de control HIMax/HIMatrix al cliente OPC.

Todas y cada una de las variables globales del sistema de control HIMax/HIMatrix podrá Ud. transmitirlas a un cliente OPC.

- **Alarm&Event (A&E) en la versión 1.10**

A&E se usa para la transmisión de alarmas y eventos desde el sistema de control HIMax/HIMatrix al cliente OPC. Cada variable global del sistema de control HIMax/HIMatrix podrá Ud. supervisarla con el registro de eventos.

Los eventos son cambios de estado del equipo o el sistema de control y llevan una marca de tiempo.

Las alarmas son eventos que indican un incremento de peligros potenciales. Se distingue entre eventos booleanos y escalares. Véase el capítulo 10.7.

### 10.1

#### Equipamiento y sistema necesarios

Elemento	Descripción
Activation	<p>Para habilitar se necesitará código de habilitación de software. Véase el capítulo 3.4. Las siguientes licencias podrán activarse por separado:</p> <ul style="list-style-type: none"> <li>▪ Data Access (DA) Server</li> <li>▪ Alarm and Events (A&amp;E) Server</li> </ul>
PC Operating System	<p>El servidor X-OPC puede funcionar en un PC basado en x86 con los siguientes sistemas operativos:</p> <ul style="list-style-type: none"> <li>▪ Windows XP Professional (mín. Service Pack 2) (32bit)</li> <li>▪ Windows Server 2003 (32bit)</li> <li>▪ Windows Vista Ultimate (32bit)</li> <li>▪ Windows Vista Business (32bit)</li> </ul>
Requirements to the host PC	<p>Requisitos mínimos a cumplir por el PC Host:</p> <ul style="list-style-type: none"> <li>▪ Pentium 4</li> <li>▪ 1 GByte (XP) ó 2,5 GByte RAM (Vista)</li> <li>▪ La tarjeta de red deberá dimensionarse según el volumen de datos: 100 Mbit/s ó 1 Gbit/s</li> </ul> <p><b>i</b> Los requisitos mínimos serán válidos para el uso de un servidor X-OPC siempre que no se usen más aplicaciones (p. ej. SILworX, Word, etc.) en el PC Host.</p>

Tabla 269: Equipamiento y sistema necesarios para servidor X-OPC

## 10.2 Propiedades del servidor X-OPC

Elemento	Descripción
OPC server	El servidor X-OPC es compatible con las funciones <ul style="list-style-type: none"> <li>▪ OPC Data Access Custom Interface en las versiones 1.0, 2.05a y 3.0.</li> <li>▪ OPC Alarm &amp; Event Interfaces 1.10.</li> </ul>
Safety-related	El servidor X-OPC se ejecuta en un PC y no está orientado a la seguridad.
Interface	Se recomienda: Ethernet 1 Gbit/s.
Data exchange	Intercambio de datos mediante <b>safeethernet</b> .
Ethernet Network	La velocidad de la red base Ethernet deberá dimensionarse según el volumen de datos (mínimo 100 Mbit/s; recomendado 1 Gbit/s).
Global Variables	Se podrán usar variables globales del contexto de la configuración.
Permissible Types of variables	Se permiten todos los tipos de datos que pueden crearse en SILworX.
Impermissible ASCII characters	Los siguientes caracteres están reservados (p. ej. para variables globales) y no se permite usarlos: ! " # ' , . / \ :
HIMax/HIMatrix Controllers	Un servidor X-OPC admite un máximo de 255 sistemas de control HIMax./HIMatrix.
safeethernet Connection	El servidor X-OPC puede intercambiar 128 kB por cada conexión <b>safeethernet</b> .
X OPC Server	En un PC Host podrán usarse hasta 10 servidores X-OPC.
X-OPC Clients	Un servidor X-OPC admite hasta 10 clientes OPC.
Data Access Tags	Un servidor “Data Access” admite un máximo de 100 000 DA Tags. Definición: Tags: Datos proporcionados por el servidor X-OPC. Los Tags corresponden a las variables globales definidas. Items: Datos requeridos por el cliente OPC.
Alarm & Event Event Definitions	Un servidor X-OPC Alarm & Event admite un máximo de 100 000 definiciones de eventos.

Tabla 270: Propiedades del servidor X-OPC

### 10.3 Propiedades del sistema de control HIMA para la conexión X-OPC

Elemento	HIMax	HIMatrix L3	HIMatrix L2	Descripción
Volumen de datos de proceso	128 kB	128 kB	16 kB	Volumen de datos de proceso que un sistema de control HIMA puede intercambiar como máximo con un servidor X-OPC por cada conexión safeethernet.
Tamaño de fragmento	1100 bytes	1100 bytes	900 bytes	Por cada ciclo HIMax se enviará sólo un fragmento a un servidor X-OPC.
Interfaces Ethernet	10/100/ 1000BaseT	10/ 100BaseT	10/ 100BaseT	Interfaces Ethernet utilizadas aptas simultáneamente también para otros protocolos.
Máx. cantidad de eventos de sistema (CPU Event)	20 000	4000	n. a	Los eventos definidos como "CPU Event" se forman en el módulo procesador. Éste realiza la formación de eventos completamente en cada uno de sus ciclos. Así puede captarse y evaluarse el valor de cada variable global como evento.
Máx. cantidad de eventos de E/S (I/O Event)	6000	n. a	n. a	Los eventos definidos como "I/O Event" podrán formarse sólo en módulos de E/S SOE (p. ej. AI 32 02 o DI 32 04). Éste realiza la formación de eventos completamente en cada uno de sus ciclos.
Tamaño de la memoria de eventos	5000	1000 (sólo con licencia SER habilitada)	n. a	Búfer de eventos no volátil del módulo procesador HIMax. Si el búfer de eventos se llena, no se guardarán nuevos eventos hasta que un registro de eventos sea leído por al menos un servidor X-OPC A&E, con lo quedará marcado el registro como sobrescribible.
Máx. cantidad de servidores X-OPC	4	4	4	Máxima cantidad de servidores X-OPC que pueden acceder al sistema de control HIMA y leer eventos de forma simultánea desde el búfer de eventos del módulo procesador.
n. a: no aplicable				

Tabla 271: Propiedades del sistema de control HIMA para la conexión X-OPC

Rango de valores de la marca de tiempo UTC (Universal Time Coordinated):  
fracción en segundos desde 1970 en [udword]  
fracción en ms de los segundos como [udword] de 0–999  
Valor por defecto: 01/01/2000 / 00:00:00  
No dispone de cambio automático entre horario de verano e invierno.

## 10.4 Acciones necesarias en caso de modificaciones

En la siguientes tabla se exponen las acciones que deberán realizarse tras una modificación en los distintos sistemas.

Tipo de modificación	Modificaciones en		
	HIMax	HIMatrix	X-OPC
<b>DA</b>			
Agregar Tags	C+D	C+D	C+D
Nombres de Tags (cambio de nombre de VG)	C+D	C+D	C+D
Borrar Tags	C+D	C+D	C+D
Modificar fragmentos (parámetros y agregar/borrar)	C+D	C+D	C+D
<hr/>			
<b>A&amp;E</b>			
Agregar definición de eventos	C+D	n. a.	C+D
Borrar definición de eventos	C+D	n. a.	C+D
Modificar fuente de eventos	C+D	n. a.	C+D
Modificar textos de alarma	-	n. a.	C+D
Modificar grado de gravedad de alarmas	-	n. a.	C+D
Modificar parámetro "ACK Required"	-	n. a.	C+D
Modificar "Alarm Values" en eventos escalares	C+D	n. a.	C+D
Modificar parámetro "Alarm at False" en eventos booleanos	C+D	n. a.	C+D
Cambiar el nombre	C+D	n. a.	C+D
Conectar canal de E/S a la VG	C+R	n. a.	-
Conectar variables de estado a VG	C+R	n. a.	-
<hr/>			
<b>General</b>			
Modificar parámetros de safeethernet	C+D	C+D	C+D

Tabla 272: Acciones necesarias en caso de modificaciones

C: Generación de código obligatoria

R: Reload obligatorio

D: Download obligatorio

n. a: no aplicable

-: Ninguna acción necesaria

## 10.5 Forzado de variables globales en módulos de E/S



Si un módulo de E/S fuerza las variables globales vinculadas al valor de proceso, éstas no tendrán ya efecto alguno sobre las variables globales vinculadas a los parámetros -> **State- LL, -L, -N, -H, -HH**.

Esto será así aun cuando estas alarmas estén registradas en el editor Alarm&Event.

Para la prueba, estas variables globales deberán forzarse una por una.

## 10.6 Configuración de una conexión de servidor OPC

En este ejemplo configuraremos una conexión redundante de servidor X-OPC a un sistema de control HIMax.

Los servidores X-OPC proporcionan las variables de proceso y los valores de eventos del sistema de control HIMax a los clientes OPC. Los clientes OPC acceden a las variables de proceso y los valores de eventos y los representan en su interfaz de usuario.

### 10.6.1 Software necesario:

- SILworX
- Servidor X-OPC
- Cliente OPC

**i** Toda la configuración y el uso del servidor X-OPC se realizan en SILworX. En el panel de control de SILworX podrá Ud. cargar, iniciar y detener el servidor X-OPC igual que un sistema de control.

### 10.6.2 Condiciones para operar con servidor X-OPC:

- La red Ethernet debería tener una anchura de banda de al menos 100 Mbit/s (mejor 1 Gbit/s).
- Debería sincronizarse la hora de sistema del PC/servidor, p. ej. mediante SNTP.
- Cerciórese de que los registros de datos para Data Access y Alarm & Events en el sistema de control, los servidores X-OPC y los clientes OPC concuerden.

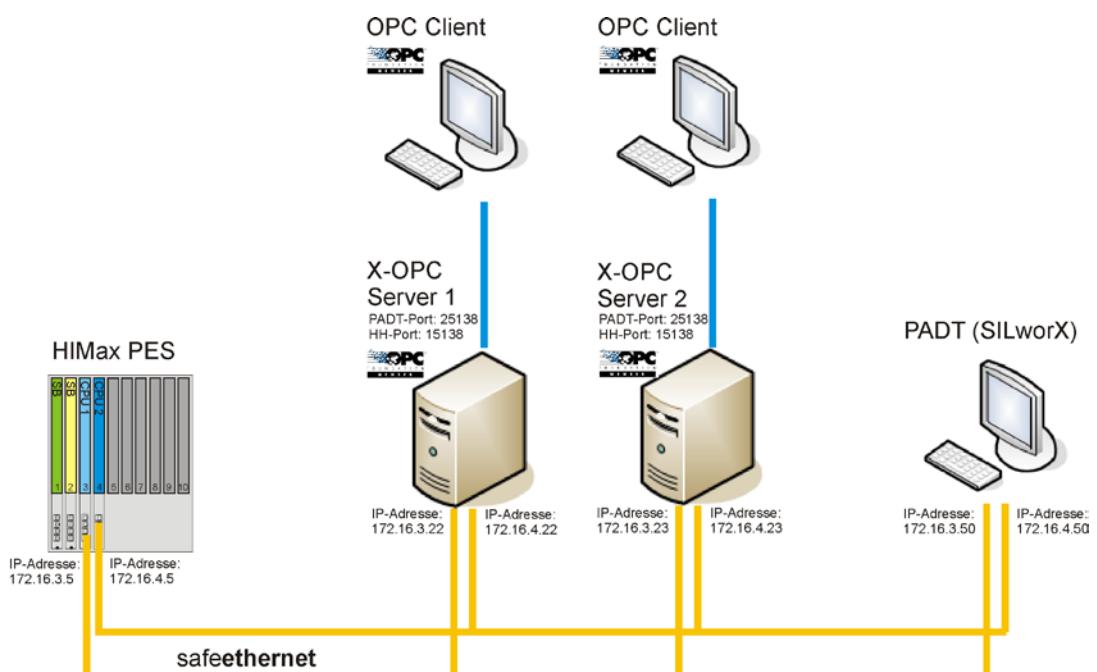


Fig. 79: Funcionamiento redundante de X-OPC

### 10.6.3 Instalación en el PC Host

El servidor X-OPC deberá instalarse en el correspondiente PC Host.

- i** Anote el ID del sistema y el número del puerto PADT. ¡Necesitará estos datos para generar la clave de la licencia!

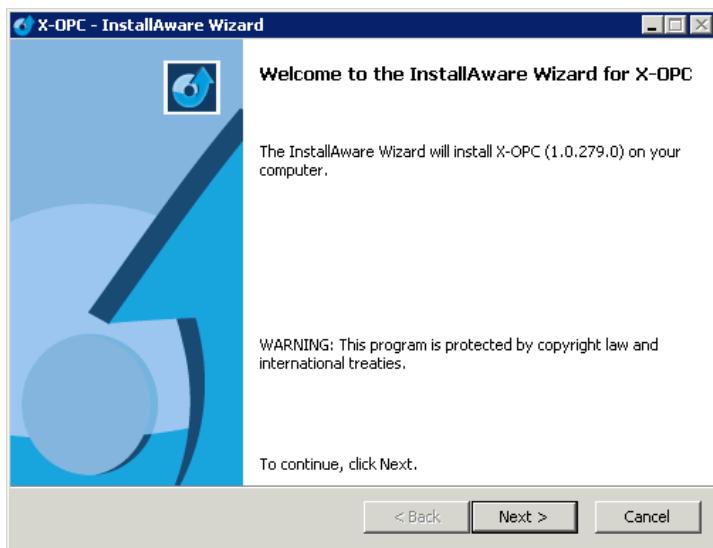


Fig. 80: Rutina de instalación del servidor X-OPC

#### Para instalar el servidor X-OPC en el primer PC Host:

Inicie el archivo X-OPC.exe en el correspondiente PC Host y siga las instrucciones de la rutina de instalación.

1. Escriba los siguientes datos para el servidor X-OPC:
  - ID del sistema: 100
  - Puerto PADT: 25138
  - Nombre de libre elección del servidor X-OPC (se indicará en el cliente OPC).
2. Para instalar el servidor X-OPC haga clic en **Next >**.

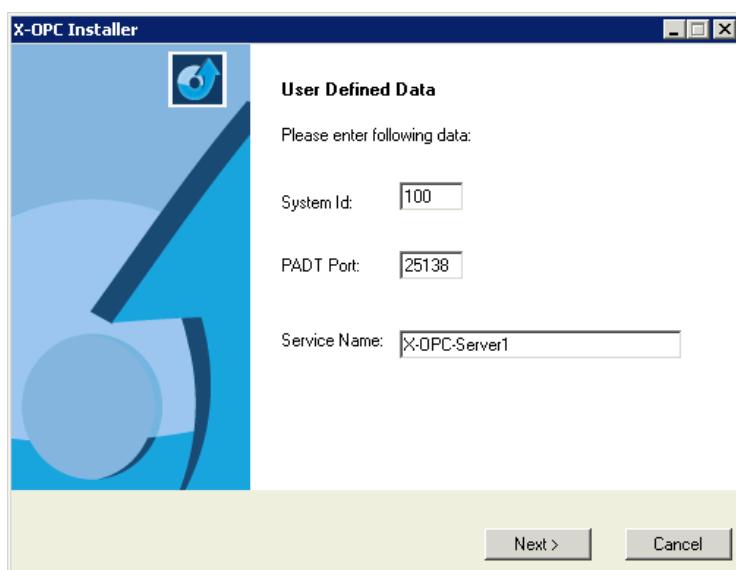


Fig. 81: Rutina de instalación del servidor X-OPC

**Para instalar el servidor X-OPC en el segundo PC Host:**

- i** Antes de instalar el segundo servidor X-OPC, averigüe los datos de ClassID del primer servidor X-OPC.

En caso de Modificar redundantemente un cliente OPC a dos servidores X-OPC, algunos clientes OPC esperan que el ClassID sea igual en ambos servidores X-OPC. Averigüe primeramente el ClassID del primer servidor X-OPC (p. ej. con ayuda del cliente OPC) y anótelo.

Inicie el archivo X-OPC.exe en el segundo PC Host y siga las instrucciones de la rutina de instalación.

1. Escriba los siguientes datos para el servidor X-OPC:

- ID del sistema: 110
- Puerto PADT: 25138
- Nombre de libre elección del servidor X-OPC (se indicará en el cliente OPC).

- i** El puerto PADT y el puerto HH del segundo servidor X-OPC podrán ser iguales a los del primero cuando los servidores X-OPC se operen en diferentes PCs.

2. Para confirmar haga clic en **Continue >**.

**Para ajustar el mismo ClassID en el segundo PC Host:**

1. Seleccione el ajuste CLSID **manual** para DA y AE.
2. Escriba el ClassID del primer servidor X-OPC en los recuadros **CLSID**.
3. Para instalar el servidor X-OPC haga clic en **Next >**.

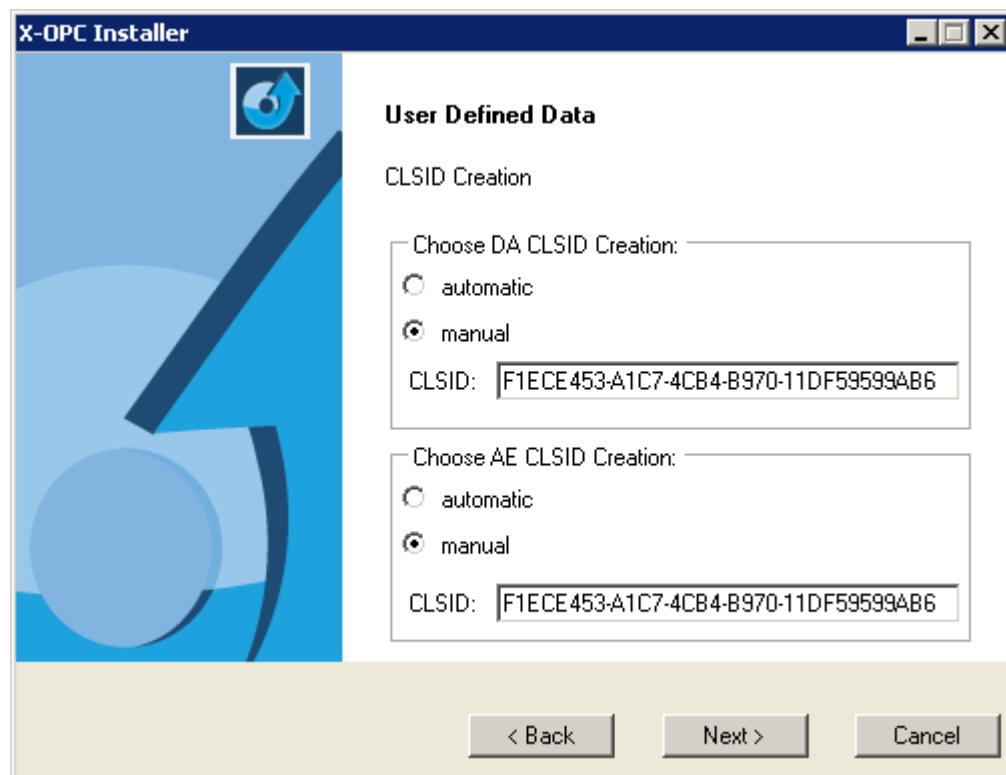


Fig. 82: Ajuste manual del ClassID del segundo servidor X-OPC

**Inicio automático de los servidores X-OPC tras un reinicio del PC:**

1. Abra en Windows **Inicio, Configuración, Panel de control, Administración, Servicios** y seleccione **X-OPC Server** en la lista.
2. Seleccione **Properties** en el menú contextual del servidor X-OPC.
3. En la ficha **General** seleccione el tipo inicial **Automatic**.

**Para comprobar si el servidor X-OPC se ejecuta en el PC:**

1. Abra el *Administrador de Tareas de Windows* y seleccione la ficha *Procesos*.
2. Compruebe si el proceso *X-OPC.exe* se está ejecutando en el PC.



Si cliente OPC y servidor OPC no se ejecutan en el mismo PC, deberá Ud. adaptar la interfaz DCOM.

Para ello deberán seguirse los pasos descritos en el manual de OPC Foundation *Using OPC via DCOM with Microsoft Windows XP Service Pack 2 Version 1.10* (véase [www.opcfoundation.org](http://www.opcfoundation.org)).

---

#### 10.6.4

## Configuración del servidor OPC en SILworX

### Para crear en SILworX un nuevo set de servidor OPC:

1. Abra **Configuration** en el árbol.
2. Haga clic con el botón derecho en **Configuration** y seleccione **New, OPC Server Set** en el menú contextual.
  - Se agregará un nuevo set de servidor OPC.
3. En el menú contextual del set de servidor OPC seleccione **Properties** y aplique los valores predeterminados.

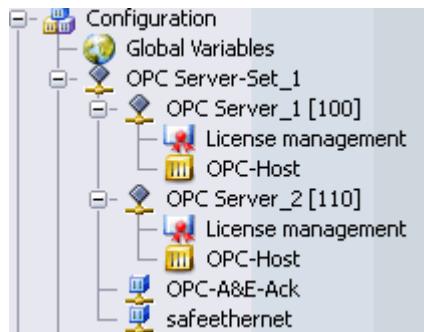


Fig. 83: Funcionamiento redundante de X-OPC

### Para configurar el primer servidor OPC en SILworX:

1. Seleccione **Configuration, OPC Server Set** en el árbol.
2. Haga clic con el botón derecho en **OPC Server Set** y seleccione **New, OPC Server** en el menú contextual.
  - Se agregará un nuevo servidor OPC.
3. Haga clic con el botón derecho en **OPC Server** y seleccione **Properties** en el menú contextual.
  - Escriba el ID [SRS] del sistema (p. ej. 100).
  - Aplique la configuración predeterminada por defecto.
3. Haga clic con el botón derecho en **OPC Host** y seleccione **Edit** en el menú contextual.
  - Se abrirá el cuadro de diálogo OPC Host para configurar las interfaces IP.
4. Haga clic con el botón derecho en una zona vacía del cuadro de diálogo OPC Host y seleccione **New IP Device** en el menú contextual.
  - Ajuste el puerto PADT (p. ej. 25138).
  - Dirección IP del PC en el que está instalado el servidor X-OPC (p. ej. 172.16.3.22).
  - Dirección IP del PC en el que está instalado el servidor X-OPC (p. ej. 172.16.4.22).
  - Márquela como interfaz predeterminada.
  - Ajuste el puerto HH (p. ej. 15138).

Configure en el mismo servidor OPC el servidor OPC redundante.

**Para configurar el segundo servidor OPC:**

1. Seleccione **Configuration, OPC Server Set** en el árbol.
2. Haga clic con el botón derecho en **OPC Server Set** y seleccione **New, OPC-Server** en el menú contextual.
  - Se agregará un nuevo servidor OPC.
3. Haga clic con el botón derecho en **OPC Server** y seleccione **Properties** en el menú contextual.
  - Escriba el ID [SRS] del sistema (p. ej. 110).
  - Aplique la configuración predeterminada por defecto.
5. Haga clic con el botón derecho en **OPC Host** y seleccione **Edit** en el menú contextual.
  - Se abrirá el cuadro de diálogo OPC Host para configurar las interfaces IP.
6. Haga clic con el botón derecho en una zona vacía del cuadro de diálogo OPC Host y seleccione **New IP Device** en el menú contextual.
  - Ajuste el puerto PADT (p. ej. 25138).
  - Dirección IP del PC en el que está instalado el servidor X-OPC (p. ej. 172.16.3.23).
  - Dirección IP del PC en el que está instalado el servidor X-OPC (p. ej. 172.16.4.23).
  - Márquela como interfaz predeterminada.
  - Ajuste el puerto HH (p. ej. 15138).



Si hay un Firewall instalado en el PC, deberá Ud. registrar como excepciones los puertos HH y TCP/UDP de PADT en la configuración del Firewall.

#### 10.6.5 Ajustes del servidor OPC en el editor de safeethernet

**Para crear la conexión safeethernet entre servidor OPC y recurso (sistema de control HIMax):**

1. En el servidor OPC abra el **safeethernet Editor**
2. En la selección de objetos haga clic en el **Resource** y arrástrelo con el ratón hasta un lugar libre del área de trabajo del editor de **safeethernet**.
3. Para Alarm & Events marque el parámetro *Activate SOE* como predeterminado.

	IF CH 1 (local)	IF CH 2 (local)	IF CH 1 (target)	IF CH 2 (target)	Profile	Sync/Async
1					Fast & Noisy	ASYNC
2	110.x.x (122.16.4.23:15138)	110.x.x (122.16.4.23:15138)	2.0.15 (172.16.4.5:6010)	2.0.15 (172.16.4.5:6010)		
3	100.x.x (172.16.3.22:15138)	100.x.x (172.16.3.22:15138)	2.0.5 (172.16.3.5:6010)	2.0.5 (172.16.3.5:6010)		

Fig. 84: Funcionamiento redundante de X-OPC



Las interfaces Ethernet del PC que se usan se relacionan en la columna **IF CH1 (local)**. Las interfaces Ethernet del sistema de control HIMax deberán seleccionarse en la columna **IF CH1 (Target)**.

Los valores predeterminados de los parámetros **safeethernet** para la comunicación del servidor X-OPC se han dimensionado para lograr una máxima disponibilidad.

Receive Timeout = 1000 ms, Response Time = 500 ms, etc.

Hallará más información de los parámetros de **safeethernet** en el capítulo 4.6.

### 10.6.6 Configuración del servidor X-OPC Data Access en SILworX

#### Para crear las definiciones de fragmentos de la conexión safeethernet:

Requisitos: El editor de safeethernet del servidor OPC deberá estar abierto.

1. Haga clic con el botón derecho en la línea del **Resource**/recurso para abrir el menú contextual del recurso.
2. Seleccione **Detail View** en el menú contextual, para abrir así la vista en detalle de la conexión safeethernet.
3. Haga clic en la ficha **Fragment Definitions**.
4. Haga clic con el botón derecho en un lugar libre del área de trabajo y seleccione **New Fragment Definition**.

En la columna “Priority” se ajusta con qué frecuencia se enviará ese fragmento en relación a los demás fragmentos (un fragmento tiene 1100 bytes).

Para las definiciones de fragmentos use primeramente el valor predeterminado por defecto (prioridad 1). Véase también el capítulo 10.6.8.

5. Haga clic en la ficha **OPC Server Set <-> Resource**.

#### Para agregar variables de recepción OPC:

Las variables de recepción OPC se envían desde el recurso al servidor OPC.

1. Abra la vista en detalle del editor de safeethernet de X-OPC y seleccione la ficha **OPC-Server Set <-> Resource**.
2. Elija en la selección de objetos una **Global Variable** y arrástrela con el ratón al área **OPC Server Set <- Resource**.
3. Haga doble clic en la columna **Fragment Name** y seleccione la **Fragment Definition** recién creada.
4. Repita este paso para las demás variables de recepción OPC.

#### Para agregar variables de envío OPC:

Las variables de envío se transmiten desde el servidor OPC al recurso.

1. Abra la vista en detalle del editor de safeethernet de X-OPC y seleccione la ficha **OPC Server Set <-> Resource**.
2. Elija en la selección de objetos una **Global Variable** y arrástrela con el ratón al área **OPC Server Set <- Resource**.
3. Haga doble clic en la columna **Fragment Name** y seleccione la **Fragment Definition** recién creada.
4. Repita este paso para las demás variables de envío OPC.



Las variables OPC de recepción y de envío deberá Ud. crearlas sólo una vez en el set de servidor OPC. Luego serán utilizadas automáticamente por el servidor X-OPC y el set de servidor OPC.

#### Generación de código y carga del recurso:

1. Seleccione **Configuration, Resource** en el árbol.
2. Haga clic en **Code Generation** en la barra de acciones y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.
4. Cargue el código generado al recurso.

**Generación de código y verificación del set servidor OPC:**

1. Seleccione **Configuration, OPC Server Set** en el árbol.
2. Haga clic en **Code Generation** en la barra de acciones y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.

**Cargue el código generado al servidor X-OPC:**

1. Haga clic con el botón derecho en **OPC Server** y seleccione en el menú contextual **Online para System Login**.
2. Escriba sus datos de ingreso:
  - Dirección IP del PC en el que está instalado el servidor X-OPC (p. ej. 172.16.3.23).
  - Nombre de usuario: Administrator
  - Contraseña: Ninguna
  - Derechos: Administrator
3. Haga clic en **Login** para abrir el panel de control.
4. En la barra de menús de SILworX haga clic en el ícono **Resource Download**.  
 Cargue el código al servidor X-OPC.
5. En la barra de menús de SILworX haga clic en el ícono **Resource Cold Start**.  
 Se iniciará el servidor X-OPC.

**Abra el cliente OPC:**

El nombre del servidor X-OPC que se lee en el cliente OPC consta de:  
**HIMA** (fabricante).**Service name** (véase el capítulo 10.6.3)-**DA** (Data Access).

Establezca conexión con el servidor X-OPC. Los datos configurados de “Data Access” deberían transmitirse ahora al cliente OPC.

### 10.6.7 Configuración del servidor X-OPC Alarm & Event en SILworX

En este ejemplo configuraremos una servidor X-OPC A&E para un sistema de control HIMax. El servidor X-OPC A&E capta los eventos mediante **safeethernet** desde el sistema de control HIMax y los proporciona al cliente OPC. El cliente OPC accede a las variables de eventos proporcionadas y las representan en su interfaz de usuario.

#### Para crear un editor de Alarm&Event:

1. Abra **Configuration, Resource** en el árbol.
2. Haga clic con el botón derecho en **Resource** y seleccione **New, Alarm & Events** en el menú contextual.
  - Se agregará el nuevo editor Alarm&Event.

#### Para crear Alarm&Event:

1. Abra **Configuration, Resource** en el árbol.
2. Haga clic con el botón derecho en **Alarm & Events** y seleccione **Edit**.
3. Seleccione la ficha **Event Definition Bool** para los eventos booleanos; véase el capítulo 10.7.1.
4. Seleccione la ficha **Event Definition Scalar** para los eventos escalares; véase el capítulo 10.7.2.
5. En la selección de objetos haga clic en la **Global Variable** y arrástrela con el ratón hasta un lugar libre del área de trabajo del editor Alarm&Event.
6. La prioridad de los eventos la definirá Ud. en el editor de **safeethernet**. Véase el capítulo 10.6.8.

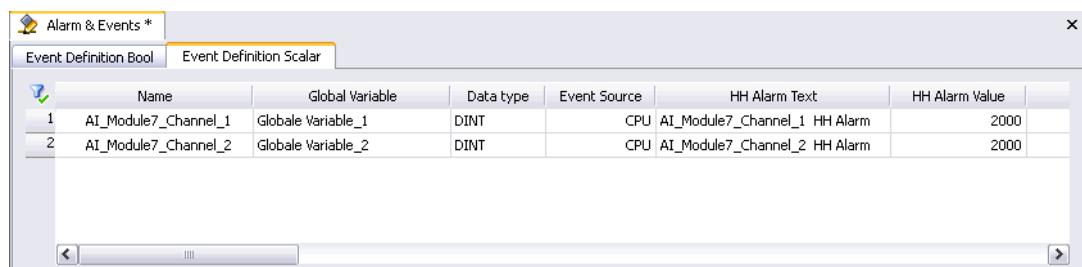


Fig. 85: Editor Alarm&Event

**Para crear la conexión de “Acknowledge” entre ambos servidores X-OPC**

**Alarm&Event:**

- i** Si se usan dos servidores X-OPC Alarm&Event de forma redundante, podrán sincronizarse los Acknowledges (acuse de alarma como atendida) de las alarmas en ambos servidores X-OPC. Para ello se creará una conexión de Acknowledge.

1. Seleccione **Configuration, OPC Server Set, New** en el árbol.
2. Haga clic con el botón derecho en **OPC Server Set** y seleccione **New, OPC-A&E-Ack** en el menú contextual.
3. Seleccione en el cuadro de diálogo OPC-A&E-Ack las siguientes conexiones IP.
  - IF CH1 (servidor OPC 1, p. ej. 172.16.3.22).
  - IF CH2 (servidor OPC 1, p. ej. 172.16.4.22).
  - IF CH1 (servidor OPC 2, p. ej. 172.16.3.23).
  - IF CH2 (servidor OPC 2, p. ej. 172.16.4.23).

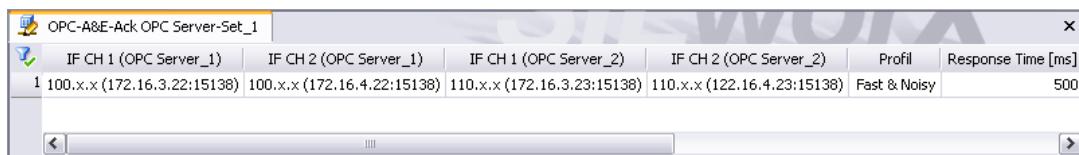


Fig. 86: Funcionamiento redundante de X-OPC

**Generación de código y carga del recurso:**

1. Seleccione **Configuration, Resource** en el árbol.
2. Haga clic en **Code Generation** en la barra de acciones y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.
4. Cargue el código generado al recurso.

**Generación de código y verificación del set servidor OPC:**

1. Seleccione **Configuration, OPC Server Set** en el árbol.
2. Haga clic en **Code Generation** en la barra de acciones y confirme con **OK**.
3. Compruebe concienzudamente las entradas del libro de registro y corrija los errores, de haberlos.

**Cargue el código generado al servidor X-OPC:**

1. Haga clic con el botón derecho en el **OPC Server** y **Online** para System Login.
2. Escriba sus datos de ingreso:
  - Dirección IP del PC en el que está instalado el servidor X-OPC (p. ej. 172.16.3.23).
  - Nombre de usuario: Administrator
  - Contraseña: Ninguna
  - Derechos: Administrator
3. Haga clic en **Login** para abrir el panel de control.
4. En la barra de menús de SILworX haga clic en el ícono **Resource Download**.
  - Cargue el código al servidor OPC.
5. En la barra de menús de SILworX haga clic en el ícono **Resource Cold Start**.
  - Se iniciará el servidor OPC.

**Abra el cliente OPC:**

El nombre del servidor X-OPC que se lee en el cliente OPC consta de:

**HIMA (fabricante).Service name** (véase el capítulo 10.6.3)-**AE (Alarm&Event)**.

Establezca conexión con el servidor X-OPC. Alarm & Events configurados deberían transmitirse ahora al cliente OPC.



Si se conecta un sistema de control a un servidor X-OPC A&E, en primer lugar deberá sincronizarse el servidor X-OPC A&E. A tal efecto, el servidor X-OPC A&E leerá de entre todas las variables definidas como eventos el estado actual y transmitirá las alarmas pendientes de atender al cliente OPC. En el cliente OPC se podrá reflejar entonces el estado del sistema de control. Los eventos se leerán sólo a partir de ese momento.

Tras sincronizar el servidor X-OPC con el sistema de control, se actualizarán todos los eventos en el cliente OPC. Los registros de eventos con una marca de tiempo anterior se sobrescribirán con los estados actualmente leídos de las variables de eventos.

---

### 10.6.8 Parametrización de los fragmentos y prioridades en SILworX

Un sistema de control HIMA puede, según tipo 128 kB ó 16 kB, transmitir por cada conexión **safeethernet** a un servidor X-OPC, pero sólo un fragmento (1100 bytes ó 900 bytes) por cada ciclo de CPU HIMA. Los datos se fragmentan, para poder enviar más datos a través de una conexión **safeethernet**. Con el parámetro “Priority” de esos fragmentos podrá Ud. definir con qué frecuencia desea que se actualicen los fragmentos.

- i Los fragmentos de prioridad **n** y los fragmentos de prioridad **m** se envían en una relación de **n** a **m**.

Para el tiempo de reacción del sistema de control al servidor X-OPC deberá considerarse además la cantidad de fragmentos y comandos (p. ej. Stop, Start) de la secuencia SOE.

$T_R = t_1 + t_2 + t_3 + t_4$ ; válido sólo si la prioridad de todos los fragmentos de datos de estado es 1

$T_R$  Worst Case Reaction Time

$t_1$  Tiempo de seguridad del PES 1

$t_2$  Number of Fragments \* Receive TMO

$t_3$  Tiempo de seguridad del servidor X-OPC

$t_4$  Retardo debido a la función SOE. Depende del volumen de eventos y el establecimiento de conexión.

Para el sentido opuesto podrá calcularse el tiempo de reacción con la misma fórmula, teniendo en cuenta que en este caso sólo se tratará de un fragmento, ya que el servidor X-OPC transmite sólo los datos escritos por los clientes OPC.

Cantidad máxima de fragmentos: 1024

Tamaño máximo de un fragmento: 1100 bytes o 900 bytes

Rango de valores de las prioridades: 1 (mayor) hasta 65 535 (menor)

Designación	Valor por defecto de la prioridad
Priority of events (Alarm&Event)	1
Priority of state values (Alarm&Event)	10
Priority of the fragment definitions (Data Access)	1

Tabla 273: Valores por defecto de las prioridades

### 10.6.8.1 Prioridades de fragmentos de eventos (Alarm&Event)

Los eventos creados por Ud. en el editor Alarm&Event se fragmentarán automáticamente y se transmitirán por fragmentos.

La prioridad de los eventos la definirá Ud. en el editor de safeethernet Editor en las columnas **Priority of Events** y **Priority of State Values**, teniendo validez esas prioridades para todos los fragmentos de Alarm&Event de esta conexión safeethernet.

#### Para ajustar las prioridades de los fragmentos de Alarm&Event

1. Abra en OPC Server Set el editor safeethernet

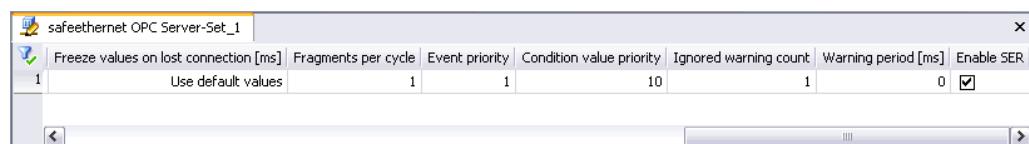


Fig. 87: Editor de safeethernet

2. Haga doble clic en **Priority of Events** para modificar la prioridad de eventos.  
A los fragmentos de eventos se les adjudicará la prioridad definida en la columna **Priority of Events** (p. ej. 1). Así se define con qué prioridad requerirá el servidor X-OPC eventos desde el sistema de control. Si en ese momento no hay eventos presentes en el sistema de control, no se transmitirá ninguno.
3. Haga doble clic en **Priority of State Values**, para modificar la prioridad de los eventos.  
A los fragmentos de valores de estado de los eventos se les adjudicará la prioridad definida en la columna **Priority of State Values** (p. ej. 10).



- Los valores de estado de los eventos se necesitarán solamente para la sincronización (p. ej. al establecer conexión) y podrán por tanto transmitirse a intervalos más largos que los eventos.

### 10.6.8.2 Prioridades de los fragmentos de “Data Access”

Para cada fragmento de “Data Access” podrá Ud. asignar variables globales y ajustar la prioridad del fragmento, la cual define la frecuencia de actualización de esa variable.

#### Para ajustar las prioridades de los fragmentos de Data Access

Requisitos: El editor de safeethernet del servidor OPC deberá estar abierto.

1. Haga clic con el botón derecho en la línea del **recurso** para abrir el menú contextual del recurso.
2. Seleccione **Detail View** en el menú contextual, para abrir así la vista en detalle de la conexión safeethernet.
3. Haga clic en la ficha **Fragment Definitions**.
4. En la columna “Priority” ajustará Ud. la prioridad de los fragmentos de Data Access.
  - Adjudique la mayor prioridad (p. ej. 1) a los fragmentos de Data Access con variables globales que deseé actualizar frecuentemente.
  - Adjudique la menor prioridad (p. ej. 10) a los fragmentos de Data Access con variables globales que deseé actualizar con poca frecuencia.

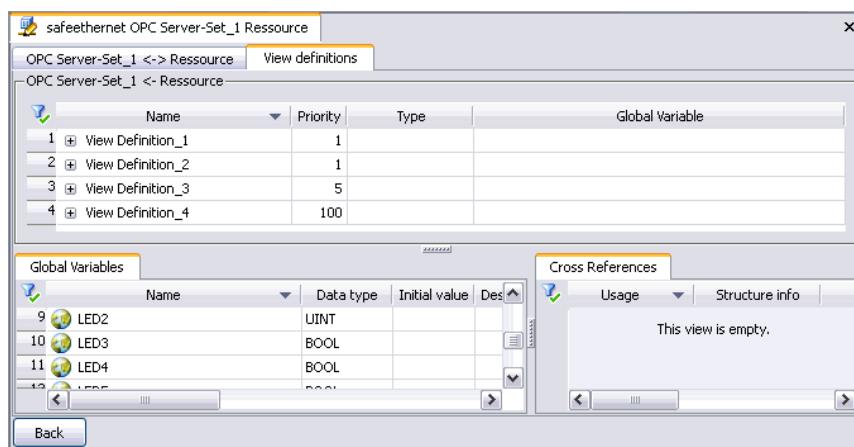


Fig. 88: Vista en detalle de la conexión safeethernet

Además, mediante las siguientes variables podrá leerse el estado y la marca de tiempo de cada fragmento de Data Access.

Nombre	Descripción									
Fragment timestamp [ms]	Fracción en milisegundos de la marca de tiempo (hora actual del sistema).									
Fragment timestamp [s]	Fracción en segundos de la marca de tiempo (hora actual del sistema).									
Fragment Status	<table border="1"> <thead> <tr> <th>Estado</th><th>Descripción</th></tr> </thead> <tbody> <tr> <td>0</td><td>CLOSED: Conexión cerrada.</td></tr> <tr> <td>1</td><td>TRY OPEN: Se intenta abrir la conexión, pero aún está sin abrir.</td></tr> <tr> <td>2</td><td>CONNECTED: Conexión establecida y se recibieron datos de fragmentos actuales (p. ej. marca de tiempo). Mientras no se reciban datos de fragmentos, “Fragment Status” seguirá siendo TRY_OPEN durante el establecimiento de conexión.</td></tr> </tbody> </table>	Estado	Descripción	0	CLOSED: Conexión cerrada.	1	TRY OPEN: Se intenta abrir la conexión, pero aún está sin abrir.	2	CONNECTED: Conexión establecida y se recibieron datos de fragmentos actuales (p. ej. marca de tiempo). Mientras no se reciban datos de fragmentos, “Fragment Status” seguirá siendo TRY_OPEN durante el establecimiento de conexión.	<p><b>i</b> El estado de conexión del editor safeethernet (véase el capítulo 4.4) cambiará a CONNECTED tan pronto como se abra la conexión. A diferencia de “Fragment Status”, aquí no deberán aún haberse intercambiado datos.</p>
Estado	Descripción									
0	CLOSED: Conexión cerrada.									
1	TRY OPEN: Se intenta abrir la conexión, pero aún está sin abrir.									
2	CONNECTED: Conexión establecida y se recibieron datos de fragmentos actuales (p. ej. marca de tiempo). Mientras no se reciban datos de fragmentos, “Fragment Status” seguirá siendo TRY_OPEN durante el establecimiento de conexión.									

Tabla 274: Estado y marca de tiempo de fragmentos de Data Access

## 10.7 Editor Alarm&Event

El editor Alarm & Event sirve para parametrizar alarmas y eventos del sistema de control HIMax.

### Para crear un editor de Alarm&Event:

1. Abra **Configuration, Resource** en el árbol.
2. Haga clic con el botón derecho en **Resource** y seleccione **New, Alarm & Events** en el menú contextual.
- Se agregará un nuevo objeto de Alarm&Events.

### Para crear Alarm&Event:

1. Abra **Configuration, Resource** en el árbol.
2. Haga clic con el botón derecho en **Alarm & Events** y seleccione **Edit**.
- 3 Seleccione la ficha **Event Definition Bool** para los eventos booleanos; véase el capítulo 10.7.1.
4. Seleccione la ficha **Event Definition Scalar** para los eventos escalares; véase el capítulo 10.7.2.
5. En la selección de objetos haga clic en la **Global Variable** y arrástrela con el ratón hasta un lugar libre del área de trabajo del editor Alarm&Event.
6. La prioridad de los eventos la definirá Ud. en el editor de safeethernet. Véase el capítulo 10.6.8.

HIMax distingue entre eventos booleanos y escalares.

### 10.7.1 Eventos booleanos

- Modificación de variables booleanas, p. ej. de entradas digitales.
- Estado de alarma y estado normal, que podrá Ud. asignar libremente a los estados de las variables

Los parámetros de los eventos **booleanos** los escribirá Ud. en el editor Alarm & Event del recurso, que contiene las siguientes columnas:

Columna	Descripción		Rango de valores
Name	Nombre de la definición de eventos		Texto, máx. 31 caracteres
Global Variable	Nombre de la variable global asignada (insertada p. ej. arrastrándola con el ratón)		
Data Type	Tipo de datos de la variable global, no puede modificarse		BOOL
Event source	CPU Event	La marca de tiempo se forma en un módulo procesador. Éste realiza la formación de eventos completamente en cada uno de sus ciclos.	CPU Event, IO Event, Auto Event
	IO Event	La marca de tiempo se forma en un módulo de E/S adecuado (p. ej. DI 32 04).	
	Auto Event	Se formará un evento de CPU y, cuando los haya, eventos de E/S de los módulos de E/S.	
	Valor por defecto:	CPU Event	
Alarm when FALSE	Activado:	La modificación de valor TRUE->FALSE de las variables globales desencadena un evento	Casilla de verificación activada, desactivada
	Desactivado:	La modificación de valor FALSE->TRUE de las variables globales desencadena un evento	
	Valor por defecto:	Desactivado	
Alarm Text	Texto que nombra el estado de alarma		Texto

Columna	Descripción	Rango de valores
Alarm Priority	Prioridad del estado de alarma Valor por defecto: 1	1...1000
Alarm Acknowledgment Required	Activado: Confirmación obligatoria del estado de fallo por parte del operador (acuse del mensaje de fallo) Desactivado: Confirmación no obligatoria del estado de fallo por parte del operador Valor por defecto: Desactivado	Casilla de verificación activada, desactivada
Return to Normal Text	Texto que nombra el estado de alarma.	Texto
Return to Normal Severity	Prioridad del estado normal.	1...1000
Return to Normal Ack Required	Confirmación obligatoria del estado normal por parte del operador (acuse del mensaje de fallo). Valor por defecto: Desactivado	Casilla de verificación activada, desactivada

Tabla 275: Parámetros para eventos booleanos

### 10.7.2 Eventos escalares

- Transiciones sobre valores límite definidos para una variable escalar, p. ej. una entrada analógica.
- Son posibles dos límites superiores y dos inferiores.

Para los valores límites regirá:

límite superior absoluto  $\geq$  límite superior  $\geq$  rango normal  $\geq$  límite inferior  $\geq$  límite inferior absoluto.

Una histéresis tendrá efecto en los siguientes casos:

- Al volver a descender por debajo de un límite superior
- Al volver a superarse un límite inferior

Especificando una histéresis se evitará Ud. una cantidad innecesariamente grande de eventos cuando la variable global fluctúe frecuentemente en torno a un valor límite.

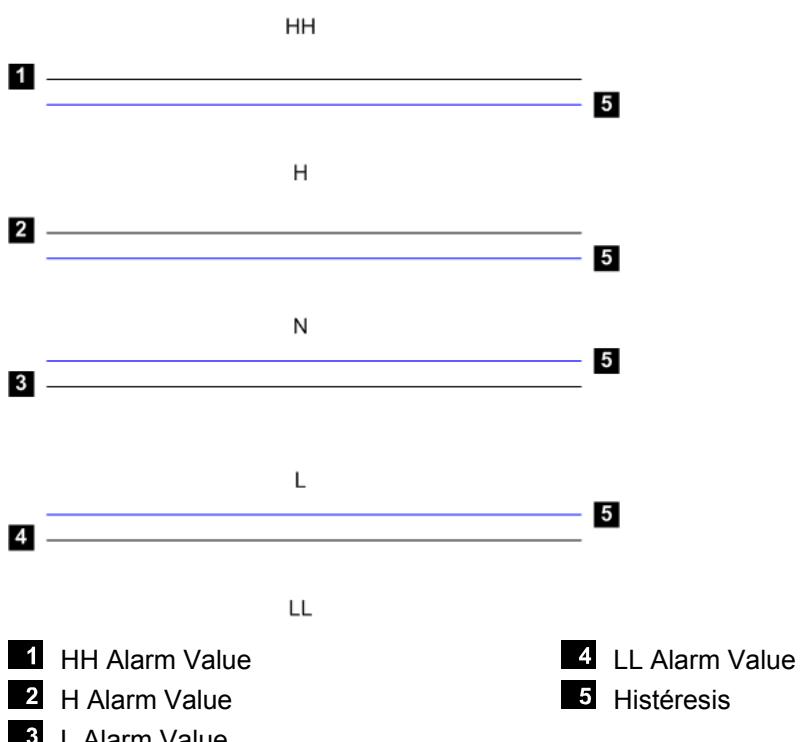


Fig. 89: Cinco rangos de un evento escalar

Los parámetros de los eventos **escalares** los escribirá Ud. en el editor Alarm & Event del recurso, que contiene las siguientes columnas:

Columna	Descripción		Rango de valores
Name	Nombre de la definición de eventos		Texto, máx. 31 caracteres
Variable global	Nombre de la variable global asignada (insertada p. ej. arrastrándola con el ratón)		
Data Type	Tipo de datos de la variable global, no puede modificarse.		Depende del tipo de las variables globales
Event Source	CPU Event	La marca de tiempo se forma en un módulo procesador. Éste realiza la formación de eventos completamente en cada uno de sus ciclos.	CPU Event, IO Event, Auto Event
	IO Event	La marca de tiempo se forma en un módulo de E/S adecuado (p. ej. AI 32 02).	
	Auto Event	Se formará un evento de CPU y, cuando los haya, eventos de E/S de los módulos de E/S.	
	Valor por defecto: CPU Event		
HH Alarm Text	Texto que nombra el estado de alarma del límite superior absoluto		Texto
HH Alarm Value	Límite superior absoluto, el cual desencadena un evento. Condición: $(\text{HH Alarm Value} - \text{Hysteresis}) > \text{H Alarm Value}$ o $\text{HH Alarm Value} = \text{H Alarm Value}$		Depende del tipo de las variables globales
HH Alarm Severity	Prioridad del límite superior absoluto, valor por defecto: 1		1...1000
HH Alarm Ack Required	Activado	El operador deberá confirmar (acusar como atendida) la transgresión del límite superior absoluto.	Casilla de verificación activada, desactivada
	Desactivado	El operador no deberá confirmar como atendida la transgresión del límite superior absoluto.	
	Valor por defecto: Desactivado		
H Alarm Text	Texto que nombra el estado de alarma de valor límite superior		Texto
H Alarm Value	Límite superior, el cual desencadena un evento. Condición: $(\text{H Alarm Value} - \text{Hysteresis}) > (\text{L Alarm Value} + \text{Hysteresis})$ o $\text{H Alarm Value} = \text{L Alarm Value}$		Depende del tipo de las variables globales
H Alarm Severity	Prioridad del valor límite superior, valor por defecto: 1		1...1000
H Alarm Ack Required	Activado	El operador deberá confirmar (acusar como atendida) la transgresión del valor límite superior.	Casilla de verificación activada, desactivada
	Desactivado	El operador no deberá confirmar como atendida la transgresión del valor límite superior.	
	Valor por defecto: Desactivado		
Return to Normal Text	Texto que nombra el estado de alarma		Texto
Return to Normal Severity	Prioridad del estado normal, valor por defecto: 1		1...1000
Return to Normal Ack Required	Confirmación obligatoria del estado normal por parte del operador (acuse del mensaje de fallo), valor por defecto: Desactivado		Casilla de verificación activada, desactivada

Columna	Descripción		Rango de valores
L Alarm Text	Texto que nombra el estado de alarma de valor límite inferior		Texto
L Alarm Value	Límite inferior, el cual desencadena un evento. Condición: $(L\ Alarm\ Value + Hysteresis) < (H\ Alarm\ Value - Hysteresis)$ o $L\ Alarm\ Value = H\ Alarm\ Value$		Depende del tipo de las variables globales
L Alarm Severity	Prioridad del valor límite inferior, valor por defecto: 1		1...1000
L Alarm Ack Required	Activado Desactivado Valor por defecto: Desactivado	El operador deberá confirmar (acusar como atendida) la transgresión del valor límite inferior. El operador no deberá confirmar como atendida la transgresión del valor límite inferior.	Casilla de verificación activada, desactivada
LL Alarm Text	Texto que nombra el estado de alarma de límite inferior absoluto		Texto
LL Alarm Value	Límite inferior absoluto, el cual desencadena un evento. Condición: $(LL\ Alarm\ Value + Hysteresis) < (L\ Alarm\ Value)$ o $LL\ Alarm\ Value = L\ Alarm\ Value$		Depende del tipo de las variables globales
LL Alarm Severity	Prioridad del límite inferior absoluto, valor por defecto: 1		1...1000
LL Alarm Ack Required	Activado Desactivado Valor por defecto: Desactivado	El operador deberá confirmar (acusar como atendida) la transgresión del límite inferior absoluto. El operador no deberá confirmar como atendida la transgresión del límite inferior absoluto.	Casilla de verificación activada, desactivada
Alarm Hysteresis	La histéresis evita que se generen numerosos eventos continuamente cuando el valor de proceso fluctúa frecuentemente en torno a un valor límite.		Depende del tipo de las variables globales

Tabla 276: Parámetros para eventos escalares

## 10.8 Parámetros de propiedades de servidor X-OPC

- i** Toda la configuración y el uso del servidor X-OPC se realizan en SILworX. En el panel de control de SILworX podrá Ud. cargar, iniciar y detener el servidor X-OPC igual que un sistema de control.

### 10.8.1 Set de Servidor OPC

El set de servidor OPC sirve como base común de parametrización para hasta dos servidores OPC.

Las propiedades del set de servidor OPC serán automáticamente idénticas para ambos servidores OPC redundantes.

**Para crear un nuevo set de servidor OPC:**

1. Abra **Configuration** en el árbol.
2. En el menú contextual de la configuración seleccione **New, OPC Server Set**, para agregar un nuevo set de servidor OPC.
3. Aplique los valores predeterminados por defecto del menú contextual **Properties** de set de servidor OPC.

El cuadro de diálogo “Properties” del set de servidor OPC contiene los siguientes parámetros.

Elemento	Descripción
Name	Nombre del Set de Servidor OPC. Máximo 31 caracteres.
Safety Time [ms]	<p>El tiempo de seguridad es el tiempo, en ms, dentro del cual el servidor X-OPC deberá reaccionar frente a un error.</p> <p>Condición:  <math>safety\ time \geq 2 \times watchdog\ time</math></p> <p>Rango de valores:  2000...400 000 ms</p> <p>Valor por defecto: 20 000 ms</p>
Watchdog Time [ms]	<p>El tiempo de WatchDog es el tiempo, en ms, del que dispone como máximo el servidor X-OPC para ejecutar un ciclo del programa. Si se sobrepasa el tiempo de WatchDog (la ejecución del ciclo del programa dura demasiado), se cerrará el servidor X-OPC.</p> <p>Condición:  <math>WDT \geq 1000\ ms\ y\ \leq 0,5 * safety\ time</math></p> <p>Rango de valores:  1000...200 000 ms</p> <p>Valor por defecto: 10 000 ms</p>

Elemento	Descripción
Main Enable	<p>El ajuste del switch OPC de habilitación principal afectará a la función de los demás switches OPC.</p> <p>Si se desactiva la habilitación principal no podrán modificarse los ajustes de los demás switches OPC durante la ejecución del programa del usuario (sistema de control en estado RUN).</p> <p>Valor por defecto: Activado</p>
AutoStart	<p>AutoStart define si las configuraciones de OPC podrán iniciarse automáticamente tras el encendido o el inicio del servidor OPC mediante Cold Start, Warm Start o no (desactivado).</p> <p>Si AutoStart está desactivado, el servidor adoptará el estado STOP/CONFIGURACIÓN VÁLIDA tras iniciarse.</p> <p>Valor por defecto: Desactivado</p>
Start Allowed	<p>Sólo con <i>Start Allowed</i> activado podrá iniciarse el servidor X-OPC desde el dispositivo programador.</p> <p>Si <i>Start Allowed</i> está desactivado, no podrá iniciarse el servidor X-OPC desde el dispositivo programador. En tal caso podrá Ud. iniciar el servidor X-OPC sólo cuando <i>AutoStart</i> esté activado y encienda o reinicie Ud. el PC Host.</p> <p>Si no están activados ni <i>AutoStart</i> ni <i>Start Allowed</i>, ya no podrá iniciarse el servidor X-OPC. Esto puede ser necesario p. ej. para trabajos de mantenimiento, para impedir la puesta en marcha accidental de una máquina o instalación.</p> <p>Valor por defecto: Activado</p>
Load Allowed	<p>Si <i>Load Allowed</i> está desactivado, no podrá cargarse ninguna (nueva) configuración OPC al sistema de control.</p> <p>Desactive <i>Load Allowed</i>, si desea impedir que se sobre escriba una configuración OPC cargada en el servidor X-OPC.</p> <p>Valor por defecto: Activado</p>
Reload Allowed	¡Todavía sin función!
Global Forcing Allowed	<p>Para poder iniciar <i>Global Forcing</i> deberá estar activado necesariamente <i>Global Forcing Allowed</i>.</p> <p><b>i</b> El editor Force podrá abrirse para ver contenidos de variables aun cuando <i>Global Forcing Allowed</i> esté desactivado.</p> <p>Valor por defecto: Desactivado</p>

Elemento	Descripción								
Global Force Timeout Reaction	<p>Si se ha elegido <i>Globale Force Timeout Reaktion, Stop Resource</i>, el servidor X-OPC adoptará el estado STOP una vez transcurra el tiempo de forzado definido. Todas las salidas del servidor X-OPC cambiarán a LOW.</p> <p>Si se ha elegido <i>Global Force Timeout Reaction Stop Forcing Only</i>, el servidor X-OPC seguirá ejecutando la configuración OPC incluso después de transcurrir el tiempo de forzado.</p> <hr/> <p><b>! Si “Forcing” está permitido, compruebe concienzudamente el ajuste de Stop en Force-Timeout. Observe a este respecto también las indicaciones del manual de seguridad.</b></p> <hr/> <p>Valor por defecto: Detener recurso</p>								
Max.Com. Time Slice ASYNC [ms]	<p>El valor <i>Max. Com. Time Slice ASYNC [ms]</i> es el tiempo, en ms, que se reserva por cada ciclo de servidor X-OPC, para poder ejecutar todas las tareas de la comunicación Peer-to-Peer.</p> <p>Valor por defecto: 500 ms</p>								
Target Cycle Time [ms]	<p>Tiempo de ciclo de consigna del servidor X-OPC</p> <p>Valor por defecto: 50 ms</p>								
Safeethernet CRC	<ul style="list-style-type: none"> <li>- Versión actual</li> <li>- SILworX V2.36</li> </ul>								
Namespace Separator	<table style="margin-left: auto; margin-right: auto;"> <tr> <td>Punto</td> <td>.</td> </tr> <tr> <td>Slash</td> <td>/</td> </tr> <tr> <td>Dos puntos</td> <td>:</td> </tr> <tr> <td>Backslash</td> <td>\</td> </tr> </table> <p>Valor por defecto: Punto</p>	Punto	.	Slash	/	Dos puntos	:	Backslash	\
Punto	.								
Slash	/								
Dos puntos	:								
Backslash	\								
Namespace Type	<p>El tipo de espaciador de nombre podrá ajustarse según los requisitos del cliente OPC:</p> <ul style="list-style-type: none"> <li>- Espaciador de nombre jerárquico</li> <li>- Espaciador de nombre plano</li> </ul> <p>Valor por defecto: Espaciador de nombre jerárquico</p>								
Changeless update	<p>Ajuste según los requisitos del cliente OPC.</p> <p>Activado: Si se ha activado <i>Changeless Update</i>, el servidor X-OPC seguirá proporcionando los mismos items al cliente OPC tras expirar OPC Group-UpdateRate.</p> <p>Desactivado: Si <i>Changeless Update</i> está desactivado, se proporcionarán sólo los valores modificados al cliente OPC (esto es lo que indica la especificación OPC).</p>								
Cycle Delay [ms]	<p>El retardo de ciclo limita la tasa de trabajo de la CPU del PC solicitable por el servidor X-OPC, para que también puedan ejecutarse otros programas.</p> <p>Rango de valores: 1...100 ms</p> <p>Valor por defecto: 5 ms</p>								

Elemento	Descripción
Short Tag Names for DA	Sólo si se ha elegido <i>Flat Namespace</i> podrá activarse este parámetro. Es una opción mediante la cual se ofrecerán datos y eventos al cliente OPC sin más contexto (ruta). Valor por defecto: Desactivado
Simple-Events for CPU I/O Events	Nunca Sólo en el inicio Siempre
Short Tag Names for A&E	Sólo si se ha elegido <i>Flat Namespace</i> podrá activarse este parámetro. Es una opción mediante la cual se ofrecerán datos y fuentes de datos al cliente OPC sin más contexto (ruta). Valor por defecto: Desactivado

Tabla 277: Propiedades

## 10.8.2 Servidor OPC

### Para crear un nuevo servidor OPC:

1. Abra **Configuration, OPC Server Set** en el árbol.
2. En el menú contextual del set de servidor OPC seleccione **New, OPC Server**, para agregar un nuevo servidor OPC.
3. En el menú contextual del servidor OPC seleccione **Properties**.

El cuadro de diálogo “Properties” del servidor OPC contiene los siguientes parámetros.

Elemento	Descripción
Name	Nombre del servidor OPC. Máximo 31 caracteres.
System ID [SRS]	Valor por defecto: 60000
Namespace Prefix	

Tabla 278: Propiedades

### Para abrir el OPC Host:

1. Abra **Configuration, OPC Server Set, OPC Server** en el árbol.
2. En el menú contextual de OPC Host seleccione **Edit**, para abrir la vista general de las interfaces IP.

El cuadro de diálogo Edit del OPC Host contiene los siguientes parámetros.

Elemento	Descripción
PADT Port	Valor por defecto: 25138
Name	Nombre del Set de Servidor OPC. Máximo 31 caracteres.
IP-Address	Dirección IP del PC Host. Valor por defecto: 192.168.0.1
Standard Interface	Deberá activarse cuando el PC Host tenga más de un puerto Ethernet. Valor por defecto: Activado
HH Port	Valor por defecto: 15138

Tabla 279: Edit

## 10.9 Desinstalación del servidor X-OPC

### Para desinstalar el servidor X-OPC:

1. Abra en Windows **Inicio, Configuración, Panel de control, Agregar o quitar programas**.
2. En la lista seleccione el servidor X-OPC que desee desinstalar y haga clic en **Remove**.
3. Siga las instrucciones de la rutina de desinstalación.

## 11 Interfaz serie síncrona

La interfaz serie síncrona (SSI) es una interfaz no orientada a la seguridad para sistemas de medición angular.

El submódulo SSI permite conectar hasta tres Absolutwertgeber en un ciclo de cadencia común, para detectar simultáneamente las posiciones X, Y y Z.

En la puesta en servicio del submódulo SSI se incluye la configuración de COM con ayuda de ComUserTask y el ajuste de la lógica del programa del usuario con ayuda de la utilidad de programación SILworX.

### 11.1 Sistema necesario

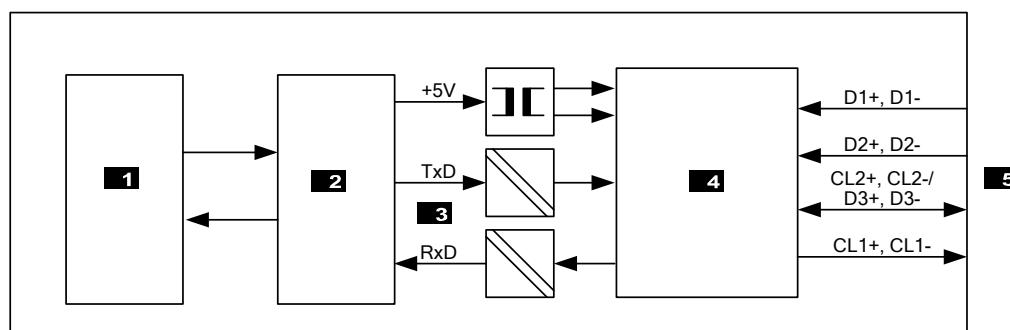
Equipamiento y sistema necesarios:

Elemento	Descripción
Controller	HIMax con módulo COM HIMatrix a partir de V7.24 de SOp de CPU y V12.30 se SOp de COM
Processor module	Las interfaces del módulo procesador no pueden usarse para SSI.
COM module	Si se usa la interfaz serie de bus de campo (FB1 o FB2), ésta deberá estar equipada con el submódulo SSI. Véase el capítulo 3.6.
Activation	Para habilitar se necesitará código de habilitación de software. Véase el capítulo 3.4.

Tabla 280: Equipamiento y sistema necesarios para ComUserTask

### 11.2 Diagrama de bloques

El submódulo SSI está galvánicamente aislado del sistema de control HIMA.



- |          |                              |          |                         |
|----------|------------------------------|----------|-------------------------|
| <b>1</b> | Módulo procesador (CPU)      | <b>4</b> | Submódulo SSI           |
| <b>2</b> | Módulo de comunicación (COM) | <b>5</b> | Interfaz SSI, FB1 o FB2 |
| <b>3</b> | Interfaz serie interna       |          |                         |

Fig. 90: Diagrama de bloques

### 11.3 Conectores hembra D-Sub de FB1 y FB2

Si se usa el submódulo SSI, regirá la asignación de pins de los conectores hembra D-Sub de FB1 y FB2, tal y como se describe en el capítulo 3.6.

### 11.4 Configuración entre submódulo SSI y COM

El intercambio de datos entre el módulo de comunicación (COM) **2** y el submódulo SSI **4** tiene lugar a través de la interfaz serie interna **3** y deberá implementarse mediante ComUserTask. Véase el capítulo 12 *ComUserTask*.

El protocolo de datos creado en ComUserTask para el intercambio de datos entre el submódulo SSI y COM deberá estar ajustado como sigue:

- Baud Rate 115,2 kBit/s
- Data Length 8 bits,
- Parity even,
- 1 Stop bit

### 11.5 Configuración de la interfaz SSI

La interfaz SSI **3** se configura mediante el byte de comando de inicio SSISTART, véase Tabla 281.

Para requerir un nuevo registro de datos de sensores (posición actual) deberá aplicarse cada vez el byte de comando de inicio SSISTART de nuevo en el programa del usuario y reenviarse al submódulo SSI **4**.

El byte de comando de inicio SSISTART a enviar tiene el siguiente formato:

Bit	Descripción
7	Auxiliary Bit Este bit define la asignación del comando de inicio.
Si el bit 7 = 1	
6...4	Es posible el siguiente ajuste para el impulso de desplazamiento SSI 000 62,5 kHz 001 125 kHz 010 250 kHz 011 500 kHz
3	Conmuta pin 3 y 8 como entrada de datos o salida de impulso. 0: D3+, D3- comutado como entrada 1: CL2+, CL2- comutado como salida
2	Salida de impulso CL1 0: activado 1: desactivado
1	No se usa
0	Salida de impulso CL2 0: activado 1: desactivado
Si el bit 7 = 0	
6...0	No se usa

Tabla 281: Formato de datos del byte de comando de inicio SSISTART

Los datos de sensores transmitidos a través de la interfaz SSI se comunican en el siguiente formato y orden a través de la interfaz serie **3** del submódulo SSI **4** a COM **2**.

En el programa del usuario podrán evaluarse estos datos de sensores para la detección de la posición X-Y-Z (nota: los bits de datos se transferirán al programa del usuario en el mismo orden en que provienen de los sensores).

N.º	Canal	Bit de datos
1	Canal 1	D47...D40
2		D39...D32
3		D31...D24
4		D23...D16
5		D15...D8
6		D7...D0
7	Canal 2	D47...D40
8		D39...D32
9		D31...D24
10		D23...D16
11		D15...D8
12		D7...D0
13	Canal 3	D47...D40
14		D39...D32
15		D31...D24
16		D23...D16
17		D15...D8
18		D7...D0

Tabla 282: Formato y orden de datos de sensores

### 11.5.1 Longitud de cable y frecuencias de reloj recomendadas

En la siguiente tabla se exponen las frecuencias recomendadas para la interfaz SSI en función de la longitud de cable de campo.

Longitud de cable/m	Frecuencia/kHz
< 25	≤ 500
< 50	< 400
< 100	< 300
< 200	< 200
< 400	< 100

Tabla 283: Frecuencias recomendadas según longitudes de cable de campo

## 11.6 Indicaciones de aplicación

El submódulo SSI está galvánicamente aislado del sistema de control HIMA. Por tanto, los sensores SSI de campo deberán tener una fuente de alimentación externa.

Con un submódulo SSI son posibles las siguientes aplicaciones:

- Conexión de 3 sensores para captación simultánea de las coordenadas X-Y-Z  
1 impulso de desplazamiento SSI (CL1+, CL1-) y 3 canales de datos (D1+, D1-, D2+, D2-, D3+, D3-) para captación simultánea de coordenadas X-Y-Z.  
Los 3 sensores se alimentan con el mismo impulso (CL1+, CL1-) y, por ende, también con la misma frecuencia de reloj.
- Conexión de 2 sensores para captación simultánea de las coordenadas X-Y  
2 impulsos de desplazamiento SSI (CL1+, CL1-, CL2+, CL2-) y 2 canales de datos (D1+, D1-, D2+, D2-) para captación simultánea de las coordenadas X-Y.  
Ambos sensores se excitan con dos impulsos separados (CL1+, CL1-, CL2+, CL2-).  
Ambos tienen la misma frecuencia de reloj.
- Conexión de 1 sensor  
1 impulso de desplazamiento SSI (CL1+, CL1- o CL2+, CL2-) y 1 canal de datos (D1+, D1- o D2+, D2-).



Es posible instalar el submódulo SSI en la Zona 2 (Directiva Europea 94/9/CE, ATEX) si se observan las condiciones especiales X.

---

## 12 ComUserTask

Además del programa lógico creado con SILworX, podrá Ud. operar con un programa C en el sistema de control.

Este programa C no seguro operará como ComUserTask libre de repercusiones sobre el módulo procesador seguro del módulo de comunicación del sistema de control.

ComUserTask tiene un ciclo propio independiente del ciclo de la CPU.

Así podrá Ud. crear las aplicaciones C que quiera e implementarlas como ComUserTask, p. ej.:

- Interfaces de comunicación para protocolos especiales (TCP, UDP, etc.).
- Función de puerta Gateway entre TCP/UDP y comunicación serie.

### 12.1 Sistema necesario

Equipamiento y sistema necesarios:

Elemento	Descripción
Controller	HIMax con módulo COM HIMatrix a partir de V7 de S.Op. de CPU y V12 de S.Op. de COM
Processor module	Las interfaces Ethernet del módulo procesador no pueden usarse para ComUserTask.
COM module	Ethernet 10/100BaseT Conexiones D-Sub FB1 y FB2, p. ej. para RS232 Si se usan las interfaces serie de bus de campo (FB1 o FB2), éstas deberán estar equipadas con un submódulo opcional de HIMA. Véase el capítulo 3.6.
Activation	Para habilitar se necesitará código de habilitación de software. Véase el capítulo 3.4.

Tabla 284: Equipamiento y sistema necesarios para ComUserTask

Propiedades de ComUserTask:

Elemento	Descripción
ComUserTask	Para cada sistema de control HIMax podrá configurarse un ComUserTask.
Safety-related	No
Data exchange	Configurable
Code and data area	Véase el capítulo 12.5.4
Stack	Stack se halla en una memoria reservada a tal fin fuera del área de datos y de código. Véase el capítulo 12.5.4

Tabla 285: Propiedades de ComUserTask

#### 12.1.1 Creación de un ComUserTask

**Para crear un nuevo ComUserTask:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. En el menú contextual de los protocolos seleccione **New, ComUserTask** para agregar un nuevo ComUserTask.
3. En el menú contextual de ComUserTask **Properties** seleccione **COM Module**. Para la primera configuración podrán aplicarse los valores predeterminados por defecto.

## 12.2 Requisitos

Para la programación de ComUserTask se dispone, además del compendio normal de comandos de C, de una biblioteca propia (véase el capítulo 12.4) con funciones definidas.

### Entorno de desarrollo

En el entorno de desarrollo se incluyen el Cygwin y el compilador GNU C, que se hallan en un CD de instalación aparte (no contenidos en SILworX) y están sujetos a las condiciones de GNU General Public License (véase [www.gnu.org](http://www.gnu.org)).

La documentación y las versiones más recientes para el entorno de desarrollo podrá Ud. descargarlas de las respectivas páginas de internet [www.cygwin.com](http://www.cygwin.com) y [www.gnu.org](http://www.gnu.org).

### Sistema de control

En los sistemas de control HIMax, ComUserTask no tiene acceso alguno a las entradas y salidas del hardware seguro. Si se tiene que acceder a las entradas y salidas del hardware, se necesitará un programa del usuario de la CPU para conectarse a las variables (ver 12.4.5).

## 12.3

### Abreviaturas

Abreviatura	Significado
CUCB	COM User Callback (las funciones CUCB son invocadas por parte de COM)
CUIT	COM User IRQ Task
CUL	COM User Library (las funciones CUL son invocadas en CUT)
CUT	ComUserTask
GNU	Proyecto GNU
IF	InterFace
FB	Interfaz FieldBus del sistema de control
FIFO	First In First Out (memoria de datos)
NVRam	Non Volatile Random Access Memory, memoria no volátil
SSI	<u>S</u> ynchron <u>S</u> erial <u>I</u> nterface

Tabla 286: Abreviaturas

## 12.4 Interfaz CUT en SILworX

La comunicación de datos de proceso de ComUserTask tiene lugar entre COM y CPU.

### ⚠ ADVERTENCIA



**El ComUserTask no habrá de usar ningún comando privilegiado del módulo COM.**  
**El código de CUT opera en COM sin repercusiones sobre la CPU. Así, la CPU segura estará protegida frente al código de CUT.**  
**No obstante, se debe observar que los errores del código de CUT perturban el funcionamiento de COM como conjunto, pudiendo también perturbar o incluso paralizar el funcionamiento del sistema de control.**  
**Ello no menoscabarán las funciones de seguridad de la CPU.**

### 12.4.1 Intervalo regular [ms]

ComUserTask recibe una invocación en un intervalo regular parametrizado [ms] en los estados RUN y STOP\_VALID\_CONFIG del sistema de control (módulo COM).

El intervalo regular [ms] se ajusta en SILworX en el cuadro *Properties* de ComUserTask.

Intervalo regular [ms]	
Rango de valores:	10...255 ms
Valor por defecto:	15 ms

Tabla 287: Intervalo regular [ms]



El tiempo de procesador COM a disposición de CUT dependerá de las demás funciones parametrizadas de COM, como p. ej. safeethernet, Modbus-TCP, etc.

Si CUT no está listo dentro del intervalo regular, se rechazarán toda invocación de reinicio de CUT hasta que se ejecute CUT en su totalidad.

### 12.4.2 Procesado previo regular

#### En el modo RUN del sistema de control:

Antes de cada invocación de CUT, COM proporcionará los datos de proceso de la CPU segura de CUT en un área de memoria definida de CUT.

#### En el modo STOP del sistema de control:

No habrá intercambio de datos de proceso desde COM a la CPU segura.

### 12.4.3 Procesado posterior regular

#### En el modo RUN del sistema de control:

Tras cada invocación de CUT, COM proporcionará los datos de proceso de CUT a la CPU segura.

#### En el modo STOP del sistema de control:

No habrá intercambio de datos de proceso desde COM a la CPU segura.

### 12.4.4 STOP\_INVALID\_CONFIG

Si COM se halla en el estado STOP\_INVALID\_CONFIG, CUT no se ejecutará.

Si COM cambia al estado STOP\_INVALID\_CONFIG y se está ejecutando CUT o CUIT, se procederá a su terminación.

### 12.4.5 Variables de la interfaz CUT (CPU<->CUT)

Parametrización de una comunicación de datos de proceso con función no orientada a la seguridad entre CPU segura y COM (CUT).

Sentido de transmisión	Tamaño máximo de los datos de proceso
COM->CPU	16375 bytes de datos (16384 bytes – 9 bytes de estado)
CPU->COM	16382 bytes de datos (16384 bytes – 2 bytes de control)

Intercambio de datos de proceso con ComUserTask

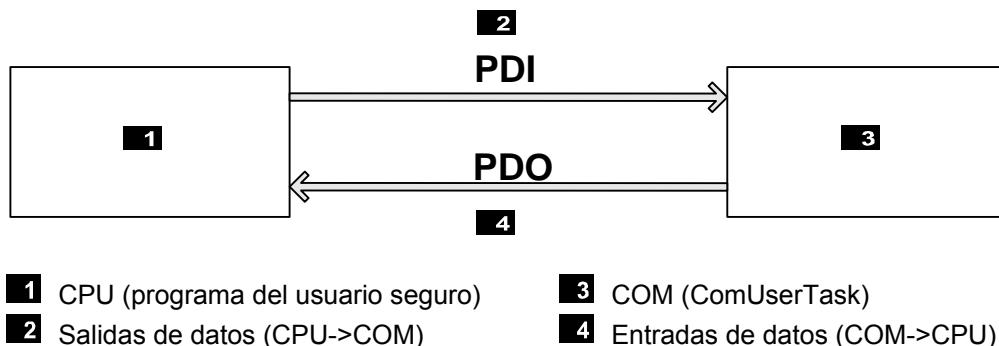


Fig. 91: Intercambio de datos de proceso entre CPU y COM (CUT)

Podrán intercambiarse todos los tipos de datos que se usen en SILworX.

La estructura de los datos deberá parametrizarse en SILworX.

El tamaño de las estructuras de datos CUT\_PDI y CUT\_PDO (en el código C compilado de CUT) deberá ser el mismo que el tamaño de la estructura de datos configurada en SILworX.

- i** Si en el código C compilado no hay estructuras de datos CUT\_PDI y CUT\_PDO, o si no tienen el mismo tamaño que la estructura de datos de proceso parametrizados en SILworX, la configuración no será válida y COM adoptará el estado STOP\_INVALID\_CONFIG.
- La comunicación de datos de proceso tiene lugar sólo en el modo RUN.

### 12.4.6 Función de menú Propiedades

Con la función de menú **Properties** del menú contextual de ComUserTask se abre el cuadro de diálogo **Properties**.

Elemento	Descripción
Type	ComUserTask
Name	Cualquier nombre inequívoco para un ComUserTask
Force Process Data Consistency	Activado: Transferencia de todos los datos del protocolo desde CPU a COM dentro de un ciclo de la CPU. Desactivado: Transferencia de todos los datos del protocolo desde CPU a COM repartida a lo largo de varios ciclos de la CPU, con 1100 bytes por sentido de transmisión de datos. Con ello es posible reducir también el tiempo de ciclo. Valor por defecto: Activado
Module	Selección del módulo COM en el que se ejecuta este protocolo.
Activate Max. $\mu$ P Budget	Activado: Aplicar el límite $\mu$ P-Budget tomado del recuadro <i>Max. <math>\mu</math>P-Budget in [%]</i> . Desactivado: No se usará límite de $\mu$ P-Budget para este protocolo.
Max. $\mu$ P budget in [%]	Máximo valor de $\mu$ P-Budget del módulo que se permite que tenga lugar durante la ejecución del protocolo. Rango de valores: 1...100% Valor por defecto: 30%
Behavior on CPU/COM Connection Loss	En caso de cortarse la conexión del módulo procesador al módulo de comunicación, se inicializarán las variables de entrada o se usarán sin variar en el módulo procesador, según lo que se defina en este parámetro (p. ej. si se extrae el módulo de comunicación con la comunicación en curso).  Adopt initial data      Las variables de entrada se reinicializarán a sus valores iniciales. Retain Last Value      Las variables de entrada conservarán su último valor.
Schedule Interval [ms]	ComUserTask es invocado en un intervalo regular [ms] parametrizado del sistema de control (módulo COM). Véase el capítulo 12.4.1. Rango de valores: 10...255 ms Valor por defecto: 15 ms
User task	

Tabla 288: Propiedades generales del dispositivo PROFINET IO

### 12.4.7 Función de menú Edit

Con la función de menú **Edit** se accede a la ficha “Process variables” y “System variables”.

#### 12.4.7.1 Variables de sistema

La ficha **System Variables** ofrece los siguientes parámetros de sistema para monitorear y controlar CUT:

Nombre	Función
Execution Time [DWORD]	Tiempo de ejecución de ComUserTask en $\mu$ s
Real schedule interval [DWORD]	Intervalo entre dos pasadas de ComUserTask, en ms
User task state control [WORD]	En la siguientes tabla se exponen las posibilidades que tiene el usuario par controlar ComUserTask con el parámetro de sistema <i>User Task State Control</i> :
Función	Descripción
DISABLED 0x8000	El programa del usuario bloquea CUT (es decir, CUT no se inicia).
AUTOSTART 0 (Default)	Tras terminar CUT, CUT se iniciará automáticamente una vez se haya subsanado el error o la perturbación.
TOGGLE_MODE_0 0x0100	Tras terminar CUT, sólo se permitirá el inicio de CUT tras aplicarse TOGGLE_MODE_1.
TOGGLE_MODE_1 0x0101	Tras terminar CUT, sólo se permitirá el inicio de CUT tras aplicarse TOGGLE_MODE_0.
State of the User Task [BYTE]	1 = RUNNING (CUT en ejecución) 0 = ERROR (CUT detenido debido a un error)

Tabla 289: Variables de sistema de ComUserTask

### 12.4.7.2 Variables de proceso

#### Señales de entrada (COM->CPU)

En la ficha **Inputs Signals** introducirá Ud. las variables que deseé que se transmitan por parte de COM (CUT) a la CPU (área de entrada de la CPU).

#### **⚠ PRECAUCIÓN**

**¡Datos no seguros de ComUserTask!**

**No se permite usar variables no seguras de ComUserTask para las funciones de seguridad del programa del usuario de la CPU.**



#### Señales de entrada de ComUserTask

Nombre	Data Type	Offset
CUT_Count	DWORD	0
Time_Stamp	DWORD	4

Tabla 290: Señales de entrada de ComUserTask

#### Registro necesario en el código C

El código C de ComUserTasks deberá contener la siguiente estructura de datos CUT\_PDO para las salidas de COM (área de entrada de la CPU):

```
/* SILworX Input Records (COM->CPU) */
uword CUT_PDO[1] __attribute__ ((section("CUT_PD_OUT_SECT"), aligned(1)));
```

El tamaño de la estructura de datos CUT\_PDO deberá equivaler al tamaño configurado de las entradas de datos en SILworX.

#### Señales de salida (CPU->COM)

En la ficha **Output Signals** introducirá Ud. las variables que deseé que se transmitan desde la CPU (área de salida de la CPU) a COM (CUT).

#### Señales de salida de ComUserTask

Nombre	Data Type	Offset
CPU_COM_1	WORD	0
CPU_COM_2	WORD	2

Tabla 291: Señales de salida de ComUserTask

#### Registro necesario en el código C

El código C de ComUserTasks deberá contener la siguiente estructura de datos CUT\_PDI para las entradas de COM (área de salida de la CPU):

```
/* SILworX Output Records (CPU->COM) */
uword CUT_PDI[1] __attribute__ ((section("CUT_PD_IN_SECT"), aligned(1)));
```

El tamaño de la estructura de datos CUT\_PDI deberá equivaler al tamaño configurado de las salidas de datos en SILworX.

## 12.5 Funciones CUT

### 12.5.1 Funciones COM-User-Callback

Las funciones COM-User-Callback tienen todas el prefijo **CUCB\_** y son invocadas directamente por parte de COM en caso de eventos.



¡Todas las funciones COM-User-Callback deberán definirse en el código C de usuario! También la función CUCB\_IrqService (no admitida HIMax y HIMatrix) deberá definirse en el código C de usuario por razones de compatibilidad.  
Prototipo de función: void CUCB\_IrqService(udword devNo) {}

Las funciones COM-User-Callback (CUCB) y COM-User-Library (CUL) comparten el mismo código y la misma memoria de datos que Stack. Estas funciones garantizan recíprocamente la coherencia de los datos conjuntamente utilizados (variables).

### 12.5.2 Funciones COM-User-Library

Todas las variables y las funciones COM-User-Library tienen el prefijo **CUL\_** y son invocadas en CUT.

Estas funciones CUL están disponibles a través del archivo de objetos **libcut.a**.

### 12.5.3 Archivos de cabecera

Ambos archivos de cabecera **cut.h** y **cut\_types.h** contienen todos los prototipos de funciones para CUL/CUCB y los correspondientes tipos de datos y constantes.

Para una notación abreviada, se definen los siguientes tipos de datos en el archivo de cabecera **cut\_types.h**:

```
typedef unsigned long      udword;
typedef unsigned short     uword;
typedef unsigned char      ubyte;
typedef signed long        dword;
typedef signed short       word;
typedef signed char        sbyte;
#ifndef HAS_BOOL
typedef unsigned char      bool; // con 0=FALSE y , de no ser así, TRUE
#endif
```

#### 12.5.4 Áreas de código y de datos y Stack para CUT

Las áreas de datos y de código son un área de memoria conjunta que comienza con el segmento de código y el segmento de datos iniciales y prosigue con los segmentos de datos. En los archivos de enlace de HIMA (**makeinc.inc.app** y **section.dld**) está definida la secuencia descrita de los segmentos y la capacidad de memoria disponible (válido para HIMax y HIMatrix).

ComUserTask reparte óptimamente la capacidad de memoria disponible entre código y datos con ayuda del archivo de enlace HIMA.

Elemento	HIMax/HIMatrix L2	HIMatrix L3
Dirección inicial	0x790000	0x800000
Longitud	448 kByte	4 MB

Stack se halla en una memoria reservada, la cual se define en el tiempo de ejecución del sistema operativo COM.

Elemento	HIMax/HIMatrix L2	HIMatrix L3
Dirección final	Dinámicamente desde la perspectiva de CUT	Dinámicamente desde la perspectiva de CUT
Longitud	64 kByte	500 kByte

#### 12.5.5 Función de inicio CUCB\_TaskLoop

La función `CUCB_TaskLoop()` es la función de inicio para ComUserTask.

La ejecución de programa de ComUserTask comienza con la invocación de esta función (véase `Intervalo regular [ms]`, capítulo 12.4.1).

##### Prototipo de función:

`void CUCB_TaskLoop(udword mode)`

##### Parámetros:

La función contiene el siguiente parámetro

Parámetro	Descripción
mode	1 = MODE_STOP equivale al modo STOP_VALID_CONFIG 2 = MODE_RUN modo de funcionamiento normal del sistema de control

Tabla 292: Parámetro

### 12.5.6 Interfaces serie RS485/RS232 IF

Las interfaces de bus de campo utilizadas deberán estar dotadas del correspondiente submódulo de bus de campo (hardware).



Para cada sistema de control HIMax se dispone de su respectiva documentación del sistema.

---

#### 12.5.6.1 CUL\_AscOpen

La función `CUL_AscOpen()` inicializa la interfaz serie introducida (`comId`) con los parámetros transferidos. Tras la invocación de la función `CUL_AscOpen()`, COM comenzará a recibir inmediatamente datos a través de esta interfaz.

Los datos recibidos se guardan en un software FIFO de 1 kbyte de tamaño por **cada** interfaz serie inicializada.

Los datos se guardarán hasta que sean leídos con la función `CUL_AscRcv()` por parte de CUT.



Si dicha lectura de los datos desde FIFO es más lenta que la recepción de nuevos datos, se desecharán los datos recién recibidos.

---

#### Prototipo de función:

```
udword CUL_AscOpen( Udword comId,  
                     Ubyte duplex,  
                     udword baudRate,  
                     ubyte parity,  
                     ubyte stopBits)
```

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
comId	Interfaz de bus de campo (RS485, RS 232) 1 = FB1 2 = FB2 3 = FB3 4 = FB4_SERVICE
duplex	0 = Full-Duplex (sólo para FB4 si RS232 lo permite) 1 = Half-Duplex
Baud Rate	1 = 1200 bits 2 = 2400 bits 3 = 4800 bits 4 = 9600 bits 5 = 19200 bits 6 = 38400 bits (máxima velocidad de transmisión de HIMax) 7 = 57600 bits (sólo HIMatrix) 8 = 115000 bits (sólo HIMatrix)
La longitud de bits de datos está ajustada de forma fija a 8 bits de datos. A estos 8 bits de datos se suman los bits parametrizados de paridad y de parada, así como un bit de inicio.	
parity	0 = NONE 1 = EVEN 2 = ODD
stopBits	1 = 1 bit 2 = 2 bits

Tabla 293: Parámetro

**Valor de retorno:**

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera cut.h.

Código de error	Descripción
CUL_OKAY	Inicialización correctamente ejecutada.
CUL_ALREADY_IN_USE	La interfaz la están usando otras funciones de COM o está ya abierta.
CUL_INVALID_PARAM	Se transmitieron parámetros o combinaciones de parámetros inadmisibles.
CUL_DEVICE_ERROR	Otros errores

Tabla 294: Valor de retorno

### 12.5.6.2 CUL\_AscClose

La función `CUL_AscClose( )` cierra la interfaz registrada en `comId`. Entonces se borrarán los datos ya recibidos pero aún no leídos con la función `CUL_AscRcv( )` en FIFO.

#### Prototipo de función:

`Udword CUL_AscClose(udword comId)`

#### Parámetros:

La función contiene el siguiente parámetro:

Parámetro	Descripción
<code>comId</code>	Interfaz de bus de campo (RS485, RS 232) 1 = FB1 2 = FB2 3 = FB3 4 = FB4_SERVICE

Tabla 295: Parámetro

#### Valor de retorno:

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera `cut.h`.

Código de error	Descripción
<code>CUL_OKAY</code>	La interfaz se cerró correctamente.
<code>CUL_NOT_OPENED</code>	La interfaz no estaba abierta (mediante CUT).
<code>CUL_INVALID_PARAM</code>	Se transmitieron parámetros o combinaciones de parámetros inadmisibles.
<code>CUL_DEVICE_ERROR</code>	Otros errores

Tabla 296: Valor de retorno

### 12.5.6.3 CUL\_AscRcv

La función `CUL_AscRcv()` ordena a COM proporcionar un volumen de datos definido desde FIFO.

Tan pronto como el volumen de datos requerido esté disponible (y CUL o el intervalo regular lo permitan), COM invocará la función `CUCB_AscRcvReady()`.

Si no hay suficientes datos en FIFO, volverá la función `CUL_AscRcv()` inmediatamente.

La orden de recepción de datos permanecerá guardada hasta que:

- la orden se haya ejecutado en su totalidad o
- se invoque la función `CUL_AscClose()` o
- se redefina mediante una nueva orden.

**i** Hasta que la orden esté ejecutada, el contenido de `*pBuf` sólo podrá modificarse mediante la función `CUCB_AscRcvReady()`.

#### Prototipo de función:

Udword `CUL_AscRcv(udword comId, CUCB_ASC_BUFFER *pBuf)`

```
typedef struct CUCB_AscBuffer {
    bool bAscState;      // for using by CUT/CUCB
    bool bError;         // for using by CUT/CUCB
    uword align;         // COM is 4 aligned, long's are higher-performance
    udword mDataIdx;    // Byte offset in aData from there on the data
                         // are located
    udword mDataMax;    // max. Byte offset: (mDataMax-mDataIdx) indicates,
    udword aData[1];     // how many bytes in aData have to be sent or
                         // received
}CUCB_ASC_BUFFER; // Start point of the data copy range
```

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
comId	Interfaz de bus de campo (RS485, RS 232) 1 = FB1 2 = FB2 3 = FB3 4 = FB4_SERVICE
pBuf	Define el volumen de datos requerido y el lugar al que se desea que se copien antes de invocar <code>CUCB_AscReady()</code> . Si ya hay suficientes datos en FIFO, durante la ejecución de <code>CUL_AscRcv()</code> se invocará <code>CUCB_AscRcvReady()</code> .

Tabla 297: Parámetro

**Valor de retorno:**

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera cut.h.

Código de error	Descripción
CUL_OKAY	Si la orden se ejecutó correctamente; de lo contrario, un código de error.
CUL_NOT_OPENED	En caso de que CUT no haya abierto la interfaz.
CUL_INVALID_PARAM	Se transmitieron parámetros o combinaciones de parámetros inadmisibles.
CUL_DEVICE_ERROR	Otros errores.

Tabla 298: Valor de retorno

**Restricciones:**

- Si el área de memoria definida por CUCB\_ASC\_BUFFER no se halla dentro del segmento de datos de CUT, se terminarán CUIT y CUT.
- Podrá requerirse un volumen de datos máximo de 1024 bytes.
- Podrá requerirse un volumen de datos mínimo de 1 byte.

#### 12.5.6.4 CUCB\_AscRcvReady

Si COM invoca la función `CUCB_AscRcvReady()`, el volumen de datos requerido estará a disposición en FIFO (datos de la interfaz serie definida en el parámetro `comId`).

Los datos fueron previamente requeridos con la función `CUL_AscRcv()`.

La invocación de la función `CUCB_AscRcvReady()` puede producirse fuera de y durante la invocación de la función `CUL_AscRcv()`. El contexto de la tarea es siempre CUT.

La función `CUCB_AscRcvReady()` podrá invocar todas las bibliotecas CUT.

Está permitido igualmente

- El aumento de `mDataMax` 0
- La nueva parametrización de `mDataIdx` y `mDataMax` por parte de los datos `*pBuf` asignados a `comId` (para relectura).

El elemento estructural de `CUCB_ASC_BUFFER.mDataIdx` tiene el valor de `CUCB_ASC_BUFFER.mDataMax`.

##### **Prototipo de función:**

`void CUCB_AscRcvReady(udword comId)`

##### **Parámetros:**

La función contiene el siguiente parámetro:

Parámetro	Descripción
<code>comId</code>	Interfaz de bus de campo (RS485, RS 232) 1 = FB1 2 = FB2 3 = FB3 4 = FB4_SERVICE

Tabla 299: Parámetro

##### **Restricciones:**

Si el área de memoria definida por `CUCB_ASC_BUFFER` no se halla dentro del segmento de datos de CUT, se terminarán CUIT y CUT.

### 12.5.6.5 CUL\_AscSend

Con la función `CUCB_AscSend` se transmite, a través de la interfaz serie comId, el volumen de datos definido por el parámetro `pBuf`.

El volumen de datos definido deberá ser  $\geq 1$  byte y  $\leq 1\text{kbyte}$ .

Tras una transmisión correcta se invocará la función `CUCB_AscSendReady()`.  
En caso de error

- no se transmitirá y
- no se invocará la función `CUCB_AscSendReady()`.

Prototipo de función:

Udword CUL\_AscSend(udword comId, CUCB\_ASC\_BUFFER \*pBuf)

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
comId	Interfaz de bus de campo (RS485, RS 232) 1 = FB1 2 = FB2 3 = FB3 4 = FB4_SERVICE
pBuf	Define el volumen de datos a enviar

Tabla 300: Parámetro

#### Valor de retorno:

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera `cut.h`.

Código de error	Descripción
CUL_OKAY	Si el envío de datos se realizó correctamente.
CUL_WOULDBLOCK	Si aún no se transmitió una notificación previamente transmitida.
CUL_NOT_OPENED	En caso de que CUT no haya abierto la interfaz.
CUL_INVALID_PARAM	Se transmitieron parámetros o combinaciones de parámetros inadmisibles.
CUL_DEVICE_ERROR	Otros errores.

Tabla 301: Valor de retorno

#### Restricciones:

Si el área de memoria definida por `CUCB_ASC_BUFFER` no se halla dentro del segmento de datos de CUT, se terminarán CUIT y CUT.

### 12.5.6.6 CUCB\_AscSendReady

Si COM invoca la función `CUCB_AscSendReady()`, estará concluido el envío de datos con la función `CUCB_AscSend()` a través de la interfaz serie.

El contexto de la tarea es siempre CUT. La función `CUCB_AscSendReady()` podrá invocar todas las bibliotecas CUT.

**Prototipo de función:**

```
void CUCB_AscSendReady(udword comId)
```

**Parámetros:**

La función contiene el siguiente parámetro:

Parámetro	Descripción
comId	Interfaz de bus de campo (RS485, RS 232) 1 = FB1 2 = FB2 3 = FB3 4 = FB4_SERVICE

Tabla 302: Parámetro

### 12.5.7 UDP/TCP-Socket-IF

Independientemente del protocolo utilizado, se dispondrá de un máximo de 8 sockets para uso simultáneo.

La conexión física tiene lugar a través de las interfaces Ethernet 10/100BaseT del sistema de control.

#### 12.5.7.1 CUL\_SocketOpenUdpBind

La función `CUL_SocketOpenUdpBind()` genera un socket del tipo UDP y vincula el socket al puerto seleccionado.

La dirección de vinculación es siempre INADDR\_ANY, es decir, se recibirán todas las notificaciones direccionalas a COM para puerto/UDP. Los sockets operarán siempre en el modo non-blocking, es decir, la función no bloqueará.

##### **Prototipo de función:**

```
dword CUL_SocketOpenUdpBind( uword port, uword *assigned_port_ptr )
```

##### **Parámetros:**

La función contiene los siguientes parámetros:

Parámetro	Descripción
port	Un número de puerto libre $\geq 0$ sin ocupar por COM. Si el parámetro port = 0, el socket se vinculará al primer puerto libre.
assigned_port_ptr	Dirección a la que se desea copiar el número de puerto vinculado, en caso de port = 0, en caso contrario a NULO.

Tabla 303: Parámetro

##### **Valor de retorno:**

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera cut.h.

Código de error	Descripción
socketNumber	Número de socket adjudicado a UDP si > 0; Códigos de error < 0.
CUL_ALREADY_BOUND	Imposible vincular a puerto para UDP.
CUL_NO_MORE_SOCKETS	No hay recursos disponibles para socket.
CUL SOCK_ERROR	Otros errores de socket.

Tabla 304: Valor de retorno

##### **Restricciones:**

Si assigned\_port\_ptr no posee CUT, se terminará CUT/CUIT.

### 12.5.7.2 CUL\_SocketOpenUdp

La función `CUL_SocketOpenUdp( )` genera un socket del tipo UDP sin vinculación a un puerto. Luego podrán sólo enviarse notificaciones por el socket, pero no recibirlas.

**Prototipo de función:**

`dword CUL_SocketOpenUdp ( void )`

**Parámetros:**

Ninguno

**Valor de retorno:**

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera `cut.h`.

Código de error	Descripción
<code>socketNumber</code>	Número de socket adjudicado a UDP si > 0 Códigos de error < 0.
<code>CUL_NO_MORE_SOCKETS</code>	No hay recursos disponibles para socket.
<code>CUL SOCK_ERROR</code>	Otros errores de socket.

Tabla 305: Valor de retorno

### 12.5.7.3 CUL\_NetMessageAlloc

La función `CULNetMessageAlloc()` adjudica memoria de mensajes para uso de

- `CUL_SocketSendTo()` en UDP y
- `CUL_SocketSend()` en TCP

En CUT podrá haber un máximo de 10 mensajes simultáneamente en uso.

#### Prototipo de función:

```
void *CUL_NetMessageAlloc(udword size, ubyte proto)
```

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
size	La capacidad de memoria necesaria, en bytes, deberá ser $\geq 1$ byte y $\leq 1400$ bytes
proto	0 = TCP 1 = UDP

Tabla 306: Parámetro

#### Retorno:

Dirección del búfer a donde se desea copiar los datos útiles a enviar. En ningún caso se permite escribir áreas de memoria fuera del área adjudicada. No se dispone de áreas para los protocolos de transmisión utilizados (Ethernet/IP/UDP o TCP).

#### Restricciones:

Si no se dispone ya de más recursos de memoria o el volumen de parámetros es excesivo o `proto > 1`, se terminarán CUT y CUIT.

#### 12.5.7.4 CUL\_SocketSendTo

La función `CUL_SocketSendTo()` transmite la notificación previamente adjudicada y completada con `CUL_NetMessageAlloc()` en forma de paquete UDP a la dirección de destino destIp/destPort.

Tras el envío, la memoria de mensajes pMsg se habilitará de nuevo automáticamente.

En cada envío deberá adjudicarse primeramente memoria de mensajes mediante la función `CULMessageAlloc()`.

##### Prototipo de función:

```
dword CUL_SocketSendTo( dword socket,
                        void *pMsg,
                        udword size,
                        udword destIp,
                        uword destPort)
```

##### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
Socket	Socket previamente generado con <code>CUL_SocketOpenUdp()</code> .
pMsg	Memoria de datos útiles UDP previamente reservada con <code>CUL_NetMessageAlloc()</code> .
Size	La capacidad de memoria, en bytes, deberá ser $\leq$ capacidad previamente adjudicada.
destIp	Dirección de destino != 0, también 0xffffffff permitido como Broadcast.
destPort	Puerto de destino != 0.

Tabla 307: Parámetro

##### Valor de retorno:

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera `cut.h`.

Código de error	Descripción
<code>CUL_OKAY</code>	Mensaje correctamente enviado.
<code>CUL_NO_ROUTE</code>	No hay enrutado registrado hasta destIp.
<code>CUL_WRONG SOCK</code>	Tipo erróneo de socket o no hay socket.
<code>CUL_SOCK_ERROR</code>	Otros errores de socket.

Tabla 308: Valor de retorno

##### Restricciones:

Si pMsg no posee mensaje en CUT o si el valor de Size para pMsg es demasiado grande, se terminará CUT/CUIT.

### 12.5.7.5 CUCB\_SocketUdpRcv

COM invocará la función CUCB\_SocketUdpRcv() cuando hay datos de socket disponibles. En Callback deberán copiarse los datos, de ser necesario, desde \*pMsg a CUT-Data. Tras el retorno la función ya no podrá acceder a \*pMsg.

#### Prototipo de función:

```
void CUCB_SocketUdpRcv( dword socket,
                        void *pMsg,
                        udword packetLength,
                        udword dataLength)
```

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
Socket	Socket previamente generado con CUL_SocketOpenUdp().
pMsg	pMsg apunta al comienzo del paquete UDP incl. cabecera de Ethernet. Mediante la cabecera de Ethernet se puede reconocer el emisor del mensaje.
packetLength	La longitud del paquete consta en packetLength, estando incluida también la longitud de la cabecera.
dataLength	La longitud de la parte de datos útiles de UDP consta en dataLength.

Tabla 309: Parámetro

### 12.5.7.6 CUL\_NetMessageFree

La función `CUL_NetMessageFree()` habilita la notificación previamente adjudicada con `CUL_NetMessageAlloc()`.

Normalmente no se necesita esta función, ya que invocando la función `CUL_SocketSendTo()` se produce la habilitación automáticamente.

**Prototipo de función:**

`void CUL_NetMessageFree(void *pMsg)`

**Parámetros:**

La función contiene el siguiente parámetro:

Parámetro	Descripción
<code>pMsg</code>	Memoria previamente reservada con <code>CUL_NetMessageAlloc()</code> .

Tabla 310: Parámetro

**Restricciones:**

Si `pMsg` no posee mensaje en CUT, se terminará CUT/CUIT.

### 12.5.7.7 CUL\_SocketOpenTcpServer

La función `CUL_SocketOpenServer()` genera un socket del tipo TCP y vincula el socket al puerto seleccionado.

La dirección de vinculación es siempre `INADDR_ANY`. Además se ordena a COM que ejecute un *Listen* en el Stream-Socket. Los sockets operarán siempre en el modo non-blocking, es decir, esta función no bloqueará.

Para más usos del socket, véanse `CUCB_SocketTryAccept()` y `CUL_SocketAccept()`.

#### **Prototipo de función:**

`dword CUL_SocketOpenTcpServer(uword port, udword backlog)`

#### **Parámetros:**

La función contiene los siguientes parámetros:

Parámetro	Descripción
<code>port</code>	Número de puerto > 0 sin ocupar por COM.
<code>backlog</code>	Máxima cantidad de conexiones en espera de establecerse para el socket.

Tabla 311: Parámetro

#### **Valor de retorno:**

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera `cut.h`.

Código de error	Descripción
<code>socketNumber</code>	Número de socket adjudicado a UDP si > 0 Códigos de error < 0.
<code>CUL_ALREADY_BOUND</code>	Imposible vincular a puerto/proto.
<code>CUL_NO_MORE_SOCKETS</code>	No hay recursos disponibles para socket.
<code>CUL SOCK_ERROR</code>	Otros errores de socket.

Tabla 312: Valor de retorno

#### **Restricciones:**

En caso correcto se usará 1 socket.

### 12.5.7.8 CUCB\_SocketTryAccept

COM invocará la función `CUL_SocketTryAccept()` en caso de haber un requerimiento de conexión TCP.

Con este requerimiento podrá luego usarse la función `CUL_SocketAccept()` para crear un socket.

#### **Prototipo de función:**

`void CUCB_SocketTryAccept(dword serverSocket).`

#### **Parámetros:**

La función contiene el siguiente parámetro:

Parámetro	Descripción
serverSocket	Socket previamente generado con <code>CUL_SocketOpenTcpServer()</code> .

Tabla 313: Parámetro

### 12.5.7.9 CUL\_SocketAccept

La función `CUL_SocketAccept()` genera un nuevo socket para el requerimiento de conexión previamente señalizado con `CUCB_SocketTryAccept()`.

#### Prototipo de función:

```
dword CUL_SocketAccept( dword serverSocket,
                        udword *pIpAddr,
                        uword *pTcpPort)
```

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
serverSocket	ServerSocket señalizado inmediatamente antes con <code>CUCB_SocketTryAccept()</code> .
pIpAddr	Dirección a la que se desea copiar la dirección IP de Peer; de lo contrario 0.
pTcpPort	Dirección a la que se desea copiar el número de puerto TCP; de lo contrario 0.

Tabla 314: Parámetro

#### Valor de retorno:

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera cut.h.

Código de error	Descripción
Socket	Si > 0, nuevo socket generado.
CUL_WRONG SOCK	Tipo erróneo de socket o no hay socket.
CUL_NO_MORE_SOCKETS	No hay más recursos de socket disponibles.
CUL_SOCK_ERROR	Otros errores de socket.

Tabla 315: Valor de retorno

#### Restricciones:

Si pIpAddr y pTcpPort no poseen CUT, se terminará CUT/CIUT.

### 12.5.7.10 CUL\_SocketOpenTcpClient

La función `CUL_SocketOpenServer()` genera un socket del tipo TCP con puerto local libre y ordena vincularlo a `destIp` y `destPort`. Los sockets operarán siempre en el modo non-blocking, es decir, la función no bloqueará. Tan pronto como se establezca la conexión, se invocará `CUCB_SocketConnected()`.

**Prototipo de función:**

`dword CUL_SocketOpenTcpClient(udword destIp, uword destPort)`

**Parámetros:**

La función contiene los siguientes parámetros:

Parámetro	Descripción
<code>destIp</code>	Dirección IP del interlocutor de comunicación.
<code>destPort</code>	Número de puerto del interlocutor de comunicación.

Tabla 316: Parámetro

**Valor de retorno:**

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera `cut.h`.

Código de error	Descripción
<code>socketNumber</code>	Si > 0; códigos de error < 0.
<code>CUL_NO_MORE_SOCKETS</code>	No hay recursos disponibles para socket.
<code>CUL_NO_ROUTE</code>	No hay enrutado registrado hasta <code>destIp</code> .
<code>CUL SOCK_ERROR</code>	Otros errores de socket.

Tabla 317: Valor de retorno

### 12.5.7.11 CUCB\_SocketConnected

COM invocará la función `CUCB_SocketConnected()` en caso de haberse establecido una conexión TCP mediante la función `CUL_SocketOpenTcpClient()`.

#### Prototipo de función:

`void CUCB_SocketConnected(dword socket, bool successfully).`

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
Socket	Socket previamente ordenado y generado con <code>CUL_SocketOpenTcpClient()</code> .
Successfully	TRUE, si la conexión llegó a establecerse; de lo contrario, FALSE.

Tabla 318: Parámetro

### 12.5.7.12 CUL\_SocketSend

La función CUL\_SocketSend() transmite la notificación previamente adjudicada y completada con CUL\_NetMessageAlloc() en forma de paquete UDP.

Tras el envío, la memoria de mensajes pMsg se habilitará de nuevo automáticamente.

En cada envío deberá adjudicarse primeramente memoria de mensajes mediante la función CULMessageAlloc( ).

#### Prototipo de función:

```
dword CUL_SocketSend(      dword socket,
                           void *pMsg,
                           udword size)
```

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
Socket	Socket previamente generado con CUL_SocketAccept( ) / CUL_SocketOpenTcpClient()
pMsg	Memoria de datos útiles TCP previamente reservada con CUL_NetMessageAlloc().
size	La capacidad de memoria, en bytes, deberá ser ≤ capacidad previamente adjudicada.

Tabla 319: Parámetro

#### Valor de retorno:

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera cut.h.

Código de error	Descripción
CUL_OKAY	Mensaje correctamente enviado.
CUL_WRONG SOCK	Tipo erróneo de socket o no hay socket.
CUL_WOULD_BLOCK	Imposible enviar el mensaje, ya que ello haría bloquear al socket.
CUL_SOCK_ERROR	Otros errores de socket.

Tabla 320: Valor de retorno

#### Restricciones:

Si pMsg no posee mensaje en CUT o si el valor de Size para pMsg es demasiado grande, se terminará CUT/CUIT.

### 12.5.7.13 CUCB\_SocketTcpRcv

COM invocará la función CUCB\_SocketTcpRcv( ) cuando hay datos útiles de socket disponibles.

Tras abandonar la función CUCB\_SocketTcpRcv( ) ya no podrá accederse a \*pMsg.

Si los datos útiles se necesitan fuera de la función CUCB\_SocketTcpRcv(), los datos útiles deberán copiarse desde \*pMsg a un área previamente creada a tal fin.



Si la conexión TCP se corta asincrónamente (tras un error o un requerimiento de la otra parte), se invocará CUCB\_SocketTcpRcv( ) con dataLength = 0.  
Con esta invocación se señalizará a CUT que debe cerrar el socket para resincronizar la comunicación.

#### Prototipo de función:

```
void CUCB_SocketTcpRcv(    dword socket,
                           void *pMsg,
                           udword dataLength)
```

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
Socket	Socket a través del cual se reciben los datos útiles.
pMsg	El parámetro pMsg apunta al comienzo de los datos útiles <b>sin</b> Ethernet-/IP-/TCP-Header.
dataLength	La longitud de los datos útiles, en bytes.

Tabla 321: Parámetro

### 12.5.7.14 CUL\_SocketClose

La función `CUL_SocketClose()` cierra un socket previamente generado.

El socket se cerrará antes de 90 segundos. La función `SocketOpen` podrá volver a ejecutarse sólo una vez se haya cerrado el socket.

#### Prototipo de función:

`dword CUL_SocketClose(dword socket)`

#### Parámetros:

La función contiene el siguiente parámetro:

Parámetro	Descripción
Socket	Socket previamente generado.

Tabla 322: Parámetro

#### Valor de retorno:

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera `cut.h`.

Código de error	Descripción
<code>CUL_OKAY</code>	Socket cerrado y un recurso de socket libre de nuevo.
<code>CUL_WRONG SOCK</code>	Socket no presente.

Tabla 323: Valor de retorno

## 12.5.8 Timer-IF

### 12.5.8.1 CUL\_GetTimeStampMS

La función `CUL_GetTimeStampMS()` comunica un tick de ms. Éste es apto para implementar un temporizador propio en CUT/CUIT. El contador se deriva del cuarzo del procesador COM y tiene por tanto su misma precisión.

#### **Prototipo de función:**

```
udword CUL_GetTimeStampMS(void)
```

### 12.5.8.2 CUL\_GetDateAndTime

La función `CUL_GetDateAndTime()` transmite al espacio de memoria comunicado `*pSec` los segundos desde el 1 de enero de 1970, 00:00 y en `*pMsec` los milisegundos correspondientes. Los valores se calibran con la CPU segura y pueden sincronizarse externamente mediante SNTP (véase el capítulo 9).

Los valores de `CUL_GetDateAndTime()` **no** deberían usarse para mediciones de tiempo, temporizadores o similares, ya que podrían ser modificados por la sincronización y/o por el usuario durante el funcionamiento.

#### **Prototipo de función:**

```
void CUL_GetDateAndTime(udword *pSec, udword *pMsec)
```

#### **Restricciones:**

Si la memoria de `pSec` o `pMsec` no está en el segmento de datos de CUT, se terminará CUT/CUIT.

### 12.5.9 Diagnóstico

La función `CUL_DiagEntry()` registra un evento en el diagnóstico COM a corto plazo y éste puede leerse mediante el PADT.

#### Prototipo de función:

```
void CUL_DiagEntry(    udword severity,  
                      udword code,  
                      udword param1,  
                      udword param2)
```

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
severity	Severity sirve para clasificar los eventos 0x45 ('E') == error, 0x57 ('W') == advertencia, 0x49 ('I') == información
code	El usuario definirá el parámetro "Code" con un número cualquiera para los eventos correspondientes. Cuando se produzca dicho evento, aparecerá el número en el diagnóstico.
param1, param2	Información complementaria sobre el evento

Tabla 324: Parámetro

## 12.6 Funciones para SEW

Las siguientes funciones son válidas sólo para sistemas de control HM30/PFF-HM31 de SEW.

### 12.6.1 COM-User-IRQ-Task

COM-User-IRQ-Task (CUT) debe compartir la memoria de datos y de código con CUIT. Tiene su propio Stack y en el Stack de CUT se definirá mediante el SOOp COM (dinámicamente desde la perspectiva de CUIT), teniendo un tamaño de 8 kbytes. Sólo hay un CUIT en COM. Inicialmente están inhabilitados los IrqServices, es decir, tras el encendido o tras cargar la configuración.

#### Restricciones:

Desde CUIT se permiten invocar sólo las funciones CUL para Semaphore-Handling.

#### 12.6.1.1 CUCB\_IrqService

COM invocará la función `CUCB_IrqService()` tras desencadenarse uno de ambos CAN IRQs posibles.

Esta función se encarga de operar la fuente IRQ `devNo` del respectivo chip CAN y debe garantizar que el chip CAN revoque de nuevo su requerimiento IRQ.

#### Prototipo de función:

```
void CUCB_IrqService(udword devNo)
```

#### Restricciones:

La administración IRQ del procesador COM la realiza el SOOp COM y no se permite que esto lo asuma `CUCB_IrqService()`.



La función `CUCB_IrqService()` deberá implementarse muy eficientemente, para minimizar latencias innecesarias de otras funciones del procesador COM. De lo contrario es posible que, a altas tasas de solicitud del procesador COM, no puedan ejecutarse ciertas funciones, con lo que se perturbará, p.ej, también la comunicación de la CPU segura.

La biblioteca CUT permite habilitar y desactivar el canal COM-IRQ al que está conectada el controlador CAN.

#### CUL\_IrqServiceEnable

La función `CUL_IrqServiceEnable()` habilita el canal COM-IRQ para el controlador CAN seleccionado `devNo`. A partir de ahora los CAN-IRQs harán que se invoque CUT-IRQ-Task.

#### Prototipo de función:

```
void CUL_IrqServiceEnable(udword devNo)
```

**Parámetros:**

La función contiene el siguiente parámetro:

Parámetro	Descripción
devNo	1 = controlador CAN A 2 = controlador CAN B

Tabla 325: Parámetro

**Restricciones:**

Si para devNo se usan valores distintos de 1 ó 2, se terminará CUIT/CUT.

**12.6.1.2 CUL\_IrqServiceDisable**

La función `CUL_IrqServiceDisable()` bloquea el canal COM-IRQ para el controlador CAN *devNo*. A partir de ahora los CAN-IRQs ya no harán que se invoque CUT-IRQ-Task. Sin embargo, las operaciones IRQ sin procesar aún en su totalidad se seguirán ejecutando.

**Prototipo de función:**

`void CUL_IrqServiceDisable(udword devNo)`

**Parámetros:**

La función contiene el siguiente parámetro:

Parámetro	Descripción
devNo	1 = controlador CAN A 2 = controlador CAN B

Tabla 326: Parámetro

**Restricciones:**

Si para *devNo* se usan valores distintos de 1 ó 2, se terminará CUIT/CUT.

**12.6.1.3 CUL\_DeviceBaseAddr**

La función `CUL_DeviceBaseAddr()` transmite la dirección base de 32 bits del controlador CAN.

**Prototipo de función:**

`void* CUL_DeviceBaseAddr(udword devNo)`

**Parámetros:**

La función contiene el siguiente parámetro:

Parámetro	Descripción
devNo	1 = controlador CAN A 2 = controlador CAN B

Tabla 327: Parámetro

**Restricciones:**

Si para *devNo* se usan valores distintos de 1 ó 2, se terminará CUIT/CUT.

## 12.6.2 NVRam-IF

CUT y CUIT podrán escribir y leer el área disponible.

La capacidad de NVRam disponible para estas funciones es de 9 kbytes.

- 
- i** COM **no** garantiza la coherencia de los datos en caso de cortarse la tensión de trabajo durante un acceso ni tampoco durante accesos simultáneos desde dos Tasks.
- 

### 12.6.2.1 CUL\_NVRamWrite

La función `CUL_NVRamWrite()` escribe datos en la memoria NVRam.

#### Prototipo de función:

`void CUL_NVRamWrite (udword offset, void *source, udword size)`

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
offset	En el área de NVRams; valores válidos son 0–9215
source	Área de memoria en el segmento de datos de CUT que se desea copiar a NVRAM.
size	Cantidad de bytes que se desea copiar

Tabla 328: Parámetro

#### Restricciones:

En caso de parámetros no válidos se terminarán CUT y CUIT, es decir en caso de:

- $\text{offset} \geq 9216$
- $\text{offset} + \text{size} > 9216$
- source fuera del segmento de datos de CUT
- source+size fuera del segmento de datos de CUT

### 12.6.2.2 CUL\_NVRamRead

La función `CUL_NVRamRead()` lee datos desde NVRam.

#### Prototipo de función:

`void CUL_NVRamRead(udword offset, void *destination, udword size).`

#### Parámetros:

La función contiene los siguientes parámetros:

Parámetro	Descripción
offset	En el área de NVRams; valores válidos son 0–9215
destination	Área de memoria en el segmento de datos de CUT que se desea copiar a NVRAM.
size	Cantidad de bytes que se desea copiar

Tabla 329: Parámetro

**Restricciones:**

En caso de parámetros no válidos se terminarán CUT y CUIT:

- offset ≥ 9216
- offset+size > 9216
- destination fuera del segmento de datos de CUT
- destination+size fuera del segmento de datos de CUT

### 12.6.3 Semaphore-IF

CUT y CUIT tienen ambos conjuntamente un semáforo para la sincronización del proceso.

Los datos de las funciones de CUCB y CUL que se usen conjuntamente con la función `CUCB_IrqService()` deberán protegerse mediante un semáforo. Así se garantiza la coherencia de los datos utilizados conjuntamente con la función `CUCB_IrqService()`.

#### 12.6.3.1 CUL\_SemaRequest

La función `CUL_SemaRequest()` requiere del semáforo de CUT/CUIT.

Si el semáforo

- está libre, la función retornará con el valor `pContext`.
- no está libre, la tarea que realiza la invocación se bloqueará hasta que el semáforo sea habilitado por otra tarea y retorne con el valor `pContext`.

El contexto que se referencia con el parámetro `pContext` lo usarán sólo las funciones CUL para la tarea que realiza la invocación y no se permite su modificación entre Request y Release.

**Prototipo de función:**

```
void CUL_SemaRequest(udword *pContext)
```

Parámetros:

La función contiene el siguiente parámetro:

Parámetro	Descripción
<code>pContext</code>	Se usa sólo dentro de la tarea que realiza la invocación de CUL. El contexto retorna mediante <code>pContext</code> y deberá volver a especificarse en la función <code>CUL_SemaRelease()</code> .

Tabla 330: Parámetro

**Restricciones:**

Si se sobrepasa la cantidad de recursiones admisibles, se terminará CUT/CUIT.

Si un semáforo bloquea CUT, no se ejecutarán más CUCB\_s excepto `CUCB_IrqService()`.

### 12.6.3.2 CUL\_SemaRelease

La función `CUL_SemaRelease()` vuelve a habilitar el semáforo que se define mediante `*pContext`.

#### Prototipo de función:

```
void CUL_SemaRelease(udword *pContext)
```

#### Parámetros:

La función contiene el siguiente parámetro:

Parámetro	Descripción
<code>pContext</code>	Con el mismo valor para <code>*pContext</code> , como se ha escrito mediante <code>CUL_SemaRequest</code> o <code>CUL_SemaTry</code> .

Tabla 331: Parámetro

#### Restricciones:

Si se realiza una invocación de Release con mayor frecuencia que Request/Try, se terminará CUT/CUIL.

### 12.6.3.3 CUL\_SemaTry

La función `CUL_SemaTry()` intenta requerir CUT/CUIT.

Si el semáforo

- está libre, la función retornará con TRUE y se asignará el semáforo.
- no está libre, la función retornará con FALSE y no se asignará el semáforo.

El udword que se haya referenciado mediante `pContext` lo usará sólo CUL para la tarea que realiza la invocación y no se permite su modificación entre Request y Release.

**Prototipo de función:**

`bool CUL_SemaTry(udword *pContext)`

**Parámetros:**

La función contiene el siguiente parámetro:

Parámetro	Descripción
<code>pContext</code>	Se usa sólo dentro de la tarea que realiza la invocación de CUL.

Tabla 332: Parámetro

**Valor de retorno:**

Retornará un código de error (udword).

Los códigos de error están definidos en el archivo de cabecera `cut.h`.

Valor de retorno	Descripción
TRUE	Pudo asignarse el semáforo.
FALSE	No pudo asignarse el semáforo.

Tabla 333: Valor de retorno

El contexto retorna mediante `pContext` y deberá volver a especificarse en la función `CUL_SemaRelease`.

**Restricciones:**

Si se sobrepasa la cantidad de recursiones admisibles, se terminará CUT/CUIT.

Si un semáforo bloquea CUT, no se ejecutarán más CUCB\_s excepto `CUCB_IrqService()`.



La funciones `CUL_SemaTry()` y `CUL_SemaRequest()` podrán invocarse también sin bloqueo cuando la tarea que realiza la invocación ya haya ocupado el semáforo, pero deberán producirse a continuación muchos `CUL_SemaRelease` hasta que el semáforo vuelva a estar libre. La recursión admite como mínimo 32 000 pasos. La necesidad de realizar pasos adicionales depende de la edición de COM.

#### 12.6.4 COM-IO-IF (sólo HM 30)

Interfaz para el uso de la IO estándar de COM. El filtrado de los canales DI no es parametrizable, sino que siempre están disponibles valores filtrados y no filtrados.

```

struct CUL_IoBufferIn {
    uword mDIRaw;           //valores DI no filtrados,
                           //con bit[0]=DI[1]...bit[11]=DI[12]
                           //bit[12]=DI[13]/DO[1]...bit[15]=DI[16]/DO[4]

    uword mDIFilter;        //valores DI filtrados,
                           //con bit[0]=DI[1]...bit[15]=DI[16]
}CUL_IO_BUFFER_IN;

struct CUL_IoBufferOut {
    uword mDO;              // valor DO 1=on, 0=off,
                           // con bit[0]=DO[1]...bit[3]=DO[4]
                           // de acuerdo con mDIDODir y bit[4]...bit[15]=0
}CUL_IO_BUFFER_OUT;

struct CUL_IoBufferCfg {
    uword mDIDODir;         // valor 1=salida, 0=entrada,
                           // con bit[0]=DI/DO[1]...bit[3]=DI/DO[4]
                           // bit[4...15] no definidos, deben ser 0
} CUL_IO_BUFFER_CFG;

```

Los canales DI/DI parametrizables son los canales 1 a 4. Los números de canal son válidos tanto para DI como para DO en las estructuras arriba citadas. Si un canal DI/DO está parametrizado como salida, su mDIRaw y mDIFilter no son válidos.

##### 12.6.4.1 CUL\_IORRead (sólo HM 30)

Lee las entradas. Las propiedades de DI o de los valores de Filtro/Raw se definen a través de la especificación HW.

##### **Prototipo de función:**

```
void CUL_IORRead(CUL_IO_BUFFER_IN *pIN)
```

##### **Restricciones:**

Si el puntero comunicado no está en posesión de CUT, se procede a la terminación de CUT/CUIT.

#### 12.6.4.2 CUL\_IOWrite (sólo HM 30)

Escribe las salidas. Antes de realizar una invocación de `CUL_IOWrite()` por primera vez, se asignan valores 0 a todas las salidas.

**Prototipo de función:**

```
void CUL_IOWrite(CUL_IO_BUFFER_OUT *pOUT)
```

**Restricciones:**

Si el puntero comunicado no está en posesión de CUT, se procede a la terminación de CUT/CUIT.

#### 12.6.4.3 CUL\_IOConfigure (sólo HM 30)

La función `CUL_IOConfigure()` configura las entradas/salidas.

Antes de que se invoque `CUL_IOConfigure()` por primera vez, los canales DI/DO se parametrizan como entradas. `CUL_IOConfigure()` puede ser invocado varias veces para configurar los canales DI/DO.

Canales DI/DO, cuya parametrización cambia

- de salida a entrada, no emiten más energía hacia fuera,
- de entrada a salida, emiten el valor previamente definido con `CUL_IOWrite()`.

Las funciones COM\_IO también pueden ser invocadas varias veces durante una invocación de CUCB.

**Prototipo de función:**

```
void CUL_IOConfigure(CUL_IO_BUFFER_CFG *pCFG)
```

**Restricciones:**

Si el puntero comunicado no está en posesión de CUT, se procede a la terminación de CUT/CUIT.

## 12.7 Instalación del entorno de desarrollo

En este capítulo se describe la instalación del entorno de desarrollo y la creación de un ComUserTask.

El entorno de desarrollo se encuentra en el CD de instalación, véase capítulo 12.2.

### 12.7.1 Instalación del entorno Cygwin

El entorno Cygwin es necesario debido a que las herramientas compiladoras GNU C sólo son ejecutables en el entorno Cygwin.

El entorno Cygwin debe instalarse en Windows 2000/XP/Vista.



Tenga en cuenta los requisitos para la instalación en el capítulo 12.2. **Desactive el antivirus** en el PC en el que se vaya a instalar Cygwin, para evitar problemas en la instalación de Cygwin.

Para instalar el entorno Cygwin, siga los siguientes pasos:

#### Inicie el programa de configuración para la instalación de Cygwin:

1. Copie el fichero de instalación de Cygwin cygwin-1.7.5-1 desde el CD de instalación a su disco duro local (p. ej., drive C:\).
2. Abra el directorio Cygwin en Windows Explorer  
C:\ cygwin-1.7.5-1.
3. Inicie la instalación de Cygwin haciendo doble clic en el archivo **setup-2.697.exe**.
4. En el cuadro de diálogo Cygwin, haga clic en el botón **Next (Weiter)**, para ejecutar el programa de instalación.

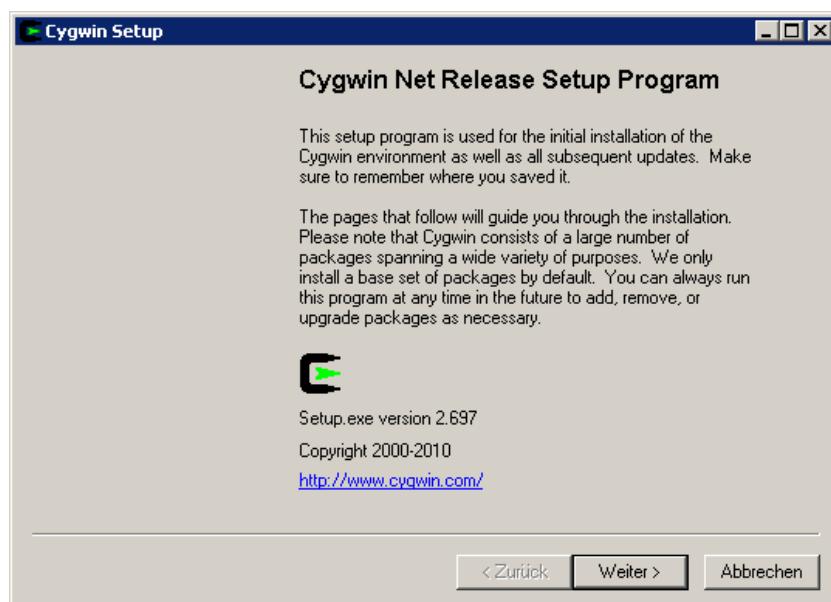


Fig. 92: Cuadro de diálogo de instalación de Cygwin *Cygwin Setup*

**El cuadro de diálogo Disable Virus Scanner aparece cuando no se ha desactivado el dispositivo.**

Ejecute este paso para desactivar el programa antivirus para la instalación e Cygwin.



Desactive el programa antivirus antes de la instalación de Cygwin, ya que, dependiendo del programa antivirus utilizado, puede ocurrir que no aparezca esta ventana a pesar de que el programa antivirus está activado.

1. Seleccione **Disable Virus scanner** para evitar problemas causados por el programa antivirus durante la instalación.
2. Haga clic en el botón **Next** para confirmar la opción.

**En el cuadro de diálogo *Choose Installation Type*, seleccione la fuente de instalación de Cygwin:**

1. Seleccione como fuente de instalación **Install from Local Directory**.
2. Haga clic en el botón **Next** para confirmar la opción.

**En el cuadro de diálogo *Choose Installation Directory*, seleccione el destino de la instalación Cygwin:**

1. Escriba el directorio en el que se instalará Cygwin.
2. Aplique todas las configuraciones predeterminadas del cuadro de diálogo.
3. Haga clic en el botón **Next** para confirmar la opción.

**En el cuadro de diálogo *Select Local Package Directory*, seleccione el fichero de instalación de Cygwin.**

1. En el recuadro *Local Package Directory*, escriba el fichero de instalación de Cygwin en el que se encuentran los archivos de instalación.
2. Haga clic en el botón **Next** para confirmar la opción.

**Seleccione en el cuadro de diálogo *Select Packages* todos los paquetes a instalar:**

1. Seleccione la opción **Curr**.
2. En el recuadro, haga clic lentamente varias veces en la opción de información junto a **All**, hasta que se muestre **Install** para una instalación completa de todos los paquetes (aprox. 1,86 GB de memoria necesarios).



Asegúrese de que detrás de cada paquete ponga **Install**.

¡Si los paquetes no se han instalado completamente, faltan las funciones más importantes para compilar después el código C!

3. Haga clic en el botón **Next** para confirmar la opción.

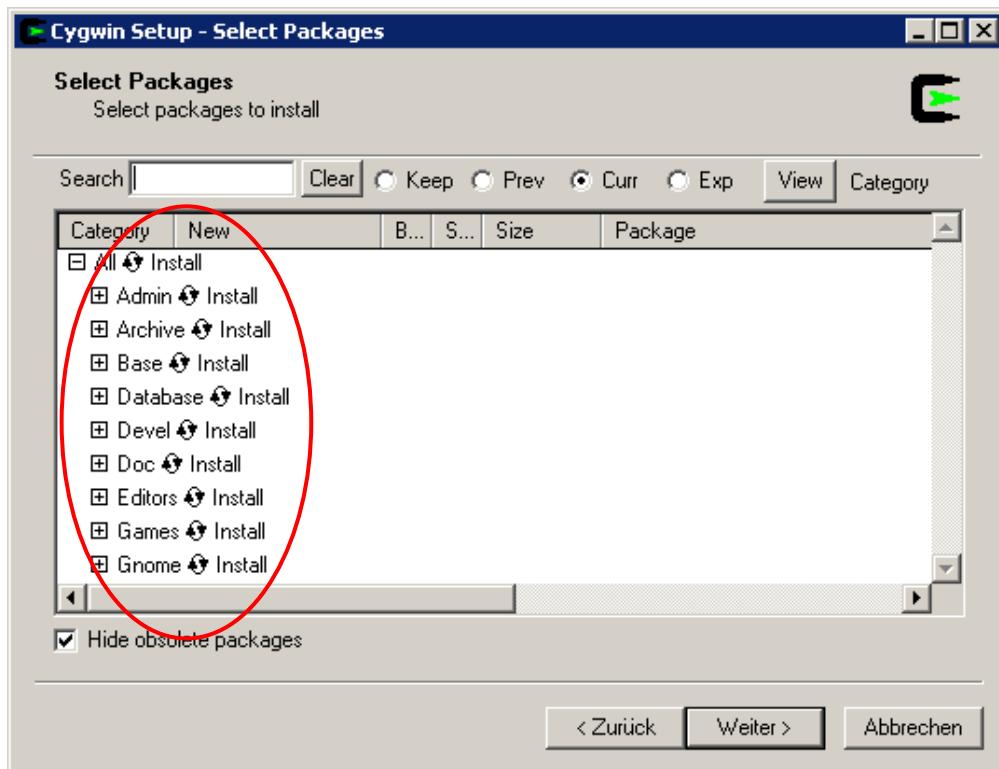


Fig. 93: Cuadro de diálogo de instalación de Cygwin Select Packages

#### Cierre la instalación de Cygwin con los siguientes registros:

1. Seleccione el registro **Start Menu**.
2. Seleccione el registro **Desktop-Icon**.
3. Haga clic en el botón **Finish** para finalizar la instalación de Cygwin.

Comandos Cygwin	Descripción
<b>cd (nombre del directorio)</b>	Cambiar directorio
<b>cd ..</b>	Cambiar a un directorio superior
<b>ls -l</b>	Mostrar todos los archivos de un directorio
<b>help</b>	Vista general de los comandos Bash Shell

Tabla 334: Comandos en Cygwin (Bash Shell)

#### 12.7.2 Instalación del compilador de GNU

##### Para instalar el compilador GNU, siga los siguientes pasos:

1. Abra en Windows Explorer el directorio del CD de instalación.
2. Haga doble clic en el archivo zip `gcc-ppc-v3.3.2_binutils-v2.15.zip`.
3. Extraiga los archivos en el directorio Cygwin (p. ej., `C:\cygwin\...`).  
El compilador GNU se descomprime en la carpeta **gcc-ppc** del directorio Cygwin.
4. Introduzca las variables de entorno en el control del sistema:

- Abra Propiedades del sistema en el menú de inicio de Windows **Settings->System Control->System**.
- Seleccione la ficha **Advanced**.
- Haga clic en el botón **Environment Variables**.
- Seleccione en el recuadro **System Variables** la variable de sistema **Path** y amplíe ésta con el registro: C:\cygwin\gcc-ppc\bin.

Desde el CD de instalación, copie la carpeta cut\_src al directorio Home.

La carpeta cut\_src contiene todos los directorios “include” y “lib” necesarios para la creación de un ComUserTask. El archivo origen suministrado cut\_cbf.c se encuentra de forma estándar en el directorio ...\\cut\_src\\cutapp\\.

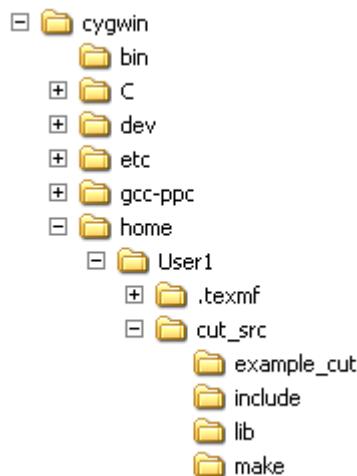


Fig. 94: Estructura en árbol de Cygwin



Si el directorio Home no se ha creado automáticamente, hágalo con Windows Explorer (p. ej. C:\\cygwin\\home\\User1).

Si se quiere crear otro directorio Home para Cygwin Bash Shell, es necesario completar el archivo batch cygwin.bat con el comando set Home.

```

@echo off
C:
chdir C:\\cygwin\\bin
set Home=C:\\User1
bash --login -i
  
```

Fig. 95: Archivo batch *Cygwin.bat*



Para generar códigos ejecutables para el programa *example\_cut* suministrado con el DVD, véase el capítulo 12.8.2.4.

## 12.8 Crear un nuevo proyecto CUT

Este capítulo explica cómo se crea un nuevo proyecto CUT y qué archivos deben adaptarse.



El proyecto CUT **example cut** se encuentra ya adaptado en el CD de CUT.

Para crear nuevos proyectos CUT adicionales, se recomienda crear un nuevo directorio para cada proyecto CUT en ...\\cut\_src\\.

### Ejemplo:

A modo de prueba se crea el directorios *example\_cut*, la fuente C se llama **example\_cut.c**, el archivo ldb generado en el directorio make se llama **example\_cut.lbd**.

**Para un nuevo ComUserTask, cree la carpeta example\_cut.**

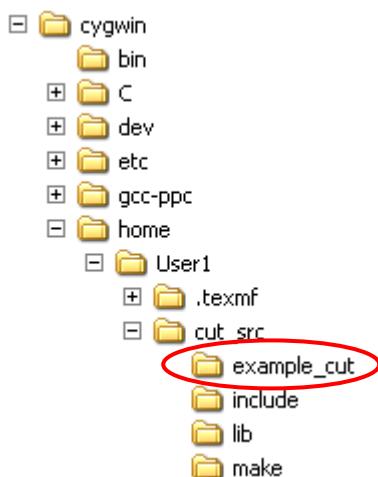


Fig. 96: Estructura en árbol de Cygwin

1. Copie los archivos
  - cutapp.c
  - cutapp.mke
  - *makefile*  
del directorio *cutapp* al directorio **example\_cut**.
2. Los nombres de los archivos tiene que cambiarse a los nombres cargables (p. ej. de **cutapp** en **example\_cut**).

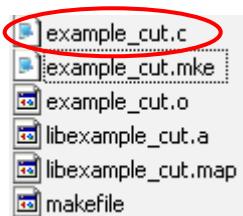


Fig. 97: Archivo de código C en la carpeta **example\_cut**

Las modificaciones en el archivo mke y makefile tienen que realizarse en un nuevo proyecto, como se describe en el próximo capítulo.

## 12.8.1 Makefiles de CUT

Configuración de los makefiles de CUT para diferentes archivos fuente y ldb

En total se deben adaptar tres makefiles, como se describe en los siguientes párrafos.

### 12.8.1.1 Makefile con la extensión “.mke”

Encontrará el archivo mke en el correspondiente directorio del código fuente p. ej. `cut_src\example_cut\example_cut.mke`.

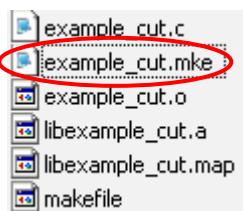


Fig. 98: Archivo mke en la carpeta example\_cut

#### Realice las siguientes modificaciones del archivo mke:

1. A la variable **module** se le debe asignar el mismo nombre cargable que al archivo .mke (p. ej., `example_cut`).
2. A la variable **c\_sources** se le pueden asignar uno o más archivos C, los cuales son necesarios para la creación del código destino (archivo cargable).

```
#####
#
# make file (DOS/NT)
# $Id: cutapp.mke 58869 2005-10-11 12:35:46Z es_fp $
#####
#
# assign name of module here (e.g. nl for NetworkLayer)
module= example_cut
#
# assign module sources here
sources=

c_sources= $(module).c
asm_sources=
```

Fig. 99: Archivo mke a partir de la línea 1

### Makefile

Encontrará el archivo makefile en el correspondiente directorio del código fuente, p. ej. `cut_src\example_cut\makefile`.

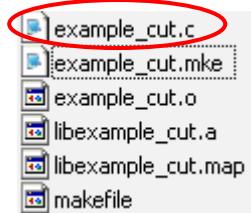


Fig. 100: makefile en la carpeta `example_cut`

#### Realice las siguientes modificaciones del archivo makefile:

1. Mueva la línea `Include` para el archivo `mke` hacia arriba y modifique el archivo `.mke` al nombre actual.
2. Amplíe la invocación de `make` para ambas variables **SUBMOD\_DIRS** y **CUT\_NAME**.

`all_objects:`

**include example\_cut.mke**

`cut:`

`$(MAKE) -C ../make elf SUBMOD_DIRS=cut_src/$(module) CUT_NAME=$(module)`

`all_objects: $(c_objects) $(asm_objects) $(objects)`

Fig. 101: makefile a partir de la línea 34

#### 12.8.1.2 Makefile con la extensión “makeinc.inc.app”

Como una modificación única para este proyecto CUT, así como para todos los que le sigan, el nombre del cargable de CUT se convierte en modificable mediante una variable make.

Encontrará el archivo `makeinc.inc.app` en el directorio `cut_src`, p. ej. `cut_src\makeinc.inc.app`.

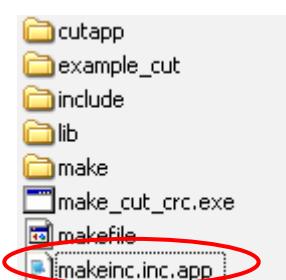


Fig. 102: Archivo `makeinc.inc.app` en la carpeta `example_cut`

**Realice las siguientes modificaciones del archivo makeinc.inc.app:**

1. Amplíe el archivo con la variable **CUT\_NAME**.

```

all : lib$(module).$(LIBEXT)
@echo 'did make for module ['lib$(module).$(LIBEXT)']'

lib$(module).$(LIBEXT) : $(objects) $(c_objects) $(asm_objects) $(libraries)

SUBMOD2_LIBS=$(foreach lib,$(SUBMOD_LIBS),../../$(lib))

CUT_NAME=cut

makeAllLibs:
$(MAKE) -C ../../cut_src cut_src

makeLoadable:
@echo; \
BGTYPE=" $(CUT_NAME)"; \
if [ ! -f $$BGTYPE.map ] ; then \
echo "Error: MAP-Datei $$BGTYPE.map existiert nicht"; \
exit 1; \
fi; \
OS_LENGTH=$$(gawk '/__OS_LENGTH/ {print substr($$1,3,8)}' $$BGTYPE.map); \
echo; \
$(OBJCOPY) --strip-all --strip-debug -O binary $$BGTYPE.elf $$BGTYPE.bin; \
echo; \
echo "Building C3-Loadable-Binary ..."; \
$(MCRC) $$BGTYPE.bin 0 $$OS_LENGTH $$OS_LENGTH $$BGTYPE.lbd; \
echo; \
$(CUT_NAME).elf: makeAllLibs $(SUBMOD2_LIBS)

elf:
@echo; test -f section.dld && $(MAKE) $(CUT_NAME).elf && $(MAKE) makeLoadable \
|| { echo "ERROR: Wrong subdir. Please invoke elf target only from make/ subdirectory." \
&& echo && false ; } ;

# end of file: makeinc.inc

```

Fig. 103: makeinc.inc.app a partir de la línea 247

## 12.8.2 Edición del código fuente C

**Siga los siguientes pasos para abrir el archivo del código fuente:**

1. Abra el directorio de proyecto *cut\_src\example\_cut* que ha creado y configurado en los pasos anteriores.
2. Abra con un editor (p. ej. Notepad) el archivo del código fuente C con la extensión **.c**.

### 12.8.2.1 Configuración de las variables de entrada y salida

**Siga los siguientes pasos para configurar las variables de entrada y salida en el archivo del código fuente:**

1. El tamaño de los datos de las variables que deben ser creadas en el registro **Data Outputs** en SILworX debe crearse en el archivo del código fuente en el vector **CUT\_PDI[X]**.
2. El tamaño de los datos de las variables que deben ser creadas en el registro **Data Inputs** en SILworX debe crearse en el archivo del código fuente en el vector **CUT\_PDO[X]**.

### 12.8.2.2 Función de inicio en CUT

La función C `void CUCB_TaskLoop (udword mode)` es la función de inicio y es invocada de forma cíclica por el programa del usuario.

### 12.8.2.3 Código de ejemplo “example\_cut.c”

El siguiente código C copia el valor de la entrada **CUT\_PDI[0]** a la salida **CUT\_PDO[0]** y devuelve el valor sin variar al programa del usuario SILworX.



El código C **example\_cut.c** se encuentra en el CD de instalación.

```
/* Example for the CUT implemetation */
#include "include/cut_types.h"
#include "include/cut.h"
#ifndef __cplusplus
extern "C" {
#endif
/*********************************************
 * SILworX Output Records (CPU->COM) *
 uword CUT_PDI[1] __attribute__ ((section("CUT_PD_IN_SECT"), aligned(1)));
 /* SILworX Input Records (COM->CPU) */
 uword CUT_PDO[1] __attribute__ ((section("CUT_PD_OUT_SECT"), aligned(1)));
/********************************************/

/* Callback function for starting the CUT */
void CUCB_TaskLoop(udword mode)
{
    if (CUT_PDI[0] > CUT_PDO[0]) /*This is executed only, if the
    {
        /*SILworX application program      */
        /*was processed.                  */
        /*The SILworX application program*/
        /*adds the value 1 to CUT_PDO[0] and */
        /*writes the result into CUT_PDI[0] */
        CUT_PDO[0] = CUT_PDI[0]; /*Copies the value from input CUT_PDI[0] */
        /*into output CUT_PDO[0] of the SPS */
        if (CUT_PDO[0] == 65535)
            (CUT_PDO[0] = 0);
    }
}
/*********************************************
```

Fig. 104: Subtítulo de imagen

```
/************************************************************************/
void CUCB_AscRcvReady(udword comId)
{
    CUL_DiagEntry(0x49, 1, comId, 0);
}
/************************************************************************/
void CUCB_AscSendReady(udword comId)
{
    CUL_DiagEntry(0x49, 2, comId, 0);
}
/************************************************************************/
void CUCB_SocketTryAccept(dword serverSocket)
{
    CUL_DiagEntry(0x49, 3, serverSocket, 0);
}
/************************************************************************/
void CUCB_SocketConnected(dword socket, bool Okay)
{
    CUL_DiagEntry(0x49, 4, socket, Okay);
}
/************************************************************************/
void CUCB_SocketTcpRcv(dword socket, void *pMsg, udword dataLength)
{
    CUL_DiagEntry(0x49, 5, socket, dataLength);
}
/************************************************************************/
void CUCB_SocketUdpRcv(dword socket, void *pMsg, udword packetLength,
                       udword dataLength)
{
    CUL_DiagEntry(0x49, 6, socket, dataLength);
}
/************************************************************************/
void CUCB_IrqService(udword devNo)
{
    CUL_DiagEntry(0x49, 7, devNo, 0);
}
/************************************************************************/

#ifndef __cplusplus
/* end extern "C" */
#endif

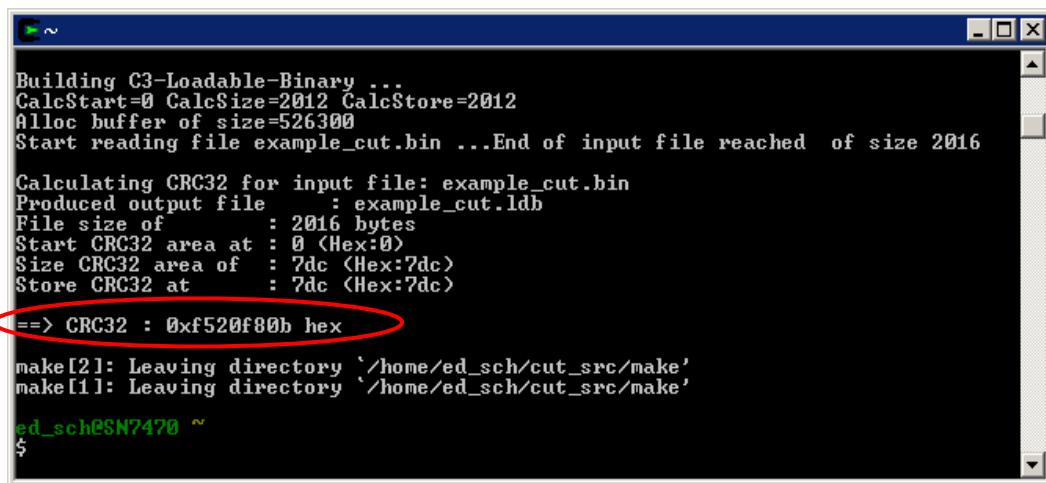
/* end of file */
```

Fig. 105: C-Code example\_cut.c

#### 12.8.2.4 Creación del código ejecutable (archivo ldb)

Siga los siguientes pasos para crear el código ejecutable (archivo ldb):

1. Inicie el **Cygwin Bash Shell**.
2. Vaya al directorio `.../cut_src/example_cut/`.
3. Inicie la generación de código con la entrada  
`make cut_himax` para HImax  
`make cut_l2` para HIMatrix L2  
`make cut_l3` para HIMatrix L3.  
El archivo binario **cut ldb** se genera automáticamente en el directorio `/cut_src/make/`.
4. Si se ha generado el CRC32, también se ha generado el código ejecutable (véase la marca roja Fig. 106).



```
Building C3-Loadable-Binary ...
CalcStart=0 CalcSize=2012 CalcStore=2012
Alloc buffer of size=526300
Start reading file example_cut.bin ...End of input file reached of size 2016

Calculating CRC32 for input file: example_cut.bin
Produced output file : example_cut.ldb
File size of : 2016 bytes
Start CRC32 area at : 0 <Hex:0>
Size CRC32 area of : 7dc <Hex:7dc>
Store CRC32 at : 7dc <Hex:7dc>

==> CRC32 : 0xf520f80b hex

make[2]: Leaving directory '/home/ed_sch/cut_src/make'
make[1]: Leaving directory '/home/ed_sch/cut_src/make'

ed_sch@SN7470 ~
$
```

Fig. 106: Cygwin Bash Shell

Este código ejecutable (archivo ldb) debe ser cargado en el proyecto como ComUserTask (véase el capítulo 12.8.3).

### 12.8.3 Integración de ComUserTask en el proyecto

Ejecute los siguientes pasos en SILworX para integrar el ComUserTask en el proyecto:

#### 12.8.3.1 Creación de ComUserTask

**Para crear un nuevo ComUserTask:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. En el menú contextual de los protocolos seleccione **New, ComUserTask** para agregar un nuevo ComUserTask.
3. En el menú contextual de ComUserTask **Properties** seleccione **COM Module**. Para la primera configuración podrán aplicarse los valores predeterminados por defecto.



Sólo puede crearse un ComUserTask por recurso.

---

#### 12.8.3.2 Carga de código de programa en el proyecto

**Para cargar un nuevo ComUserTask en el proyecto:**

1. Abra **Configuration, Resource, Protocols** en el árbol.
2. Haga clic con el botón derecho en **ComUserTask** y seleccione **Load User Task** en el menú contextual. Abra el directorio .../cut\_src/make/.
3. Seleccione el **archivo ldb** que debe ejecutarse en ComUserTask.



Es posible aceptar nuevas versiones del archivo ldb cargando de nuevo el código ejecutable (archivo ldb). Al cargar, no se comprueba que el contenido del archivo ldb se correcto. El archivo ldb se compila a continuación junto con la configuración de recursos en el proyecto y puede ser cargado en el sistema de control. Si se modifica el archivo ldb, el proyecto debe compilarse y cargarse de nuevo.

---

### Conexión de variables con CUT

El usuario puede definir una comunicación de datos de proceso con función no orientada a la seguridad entre el CPU seguro y el COM (CUT) no seguro. Al hacerlo sólo es posible intercambiar un máximo de 16 kByte de datos en cada dirección.

Cree las dos siguientes variables globales:

Variable	Tipo
COM_CPU	UINT
CPU_COM	UINT

### 12.8.3.3 Conexión de variables de proceso

#### Variables de proceso en ComUserTask:

1. Haga clic en **ComUserTask** y seleccione **Edit** en el menú contextual.
2. En el cuadro de diálogo **Edit** seleccione la ficha **Process Variables**.

#### Variables de salida (CPU->COM)

En la ficha **Output Signals**, introduzca Ud. las variables que deseé que se transmitan desde la CPU a COM.

Nombre	Tipo	Offset	Variable global
CPU_COM	UINT	0	CPU_COM

Tabla 335: Variables de salida (CPU->COM)

1. En las selección de objetos, arrastre las variables globales a enviar con el ratón hasta la ficha **Output Variables**.
2. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Output Variables**.
3. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.

#### Variables de entrada (COM->CPU)

En la ficha **Output Signals**, introduzca Ud. las variables que deseé que se transmitan desde COM a la CPU.

Nombre	Tipo	Offset	Variable global
COM_CPU	UINT	0	COM_CPU

Tabla 336: Variables de entrada (COM->CPU)

1. En la selección de objetos, arrastre las variables globales a recibir con el ratón hasta la ficha **Output Variables**.
2. Abra el menú contextual haciendo clic con el botón derecho en una zona libre del área **Input Variables**.
3. Seleccione **New Offsets** en el menú contextual, para generar los offsets de las variables.

#### Verificación de la configuración ComUserTask:

1. Seleccione **Configuration, Resource, Protocols, ComUserTask** en el árbol.
2. Haga clic con el botón de la derecha en **Verification** para verificar la configuración CUT.
3. Compruebe concienzudamente las entradas del **logbook**/libro de registro y corrija los errores, de haberlos.

### 12.8.3.4 Creación del programa del usuario SILworX

#### Para crear el programa del usuario SILworX

1. Seleccione **Edit** en el árbol **Configuration, Resource** y en el menú de contexto.
2. Desde la selección de objetos, arrastre las variables globales **COM\_CPU** y **CPU\_COM** con el ratón hasta la zona de los símbolos.
3. Cree el programa del usuario como se muestra en la siguiente imagen.

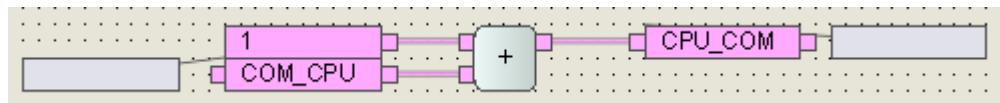


Fig. 107: Editor de programa SILworX

#### Para configurar el intervalo regular [ms]

1. Haga clic en **ComUserTask** y seleccione **Properties** en el menú contextual.
2. Introduzca en el cuadro *intervalo regular [ms]* los intervalos en los que el ComUserTask deba ser invocado.

- 
- 1 La configuración del ComUserTask debe compilarse de nuevo con el programa del usuario del recurso y transmitirse al sistema de control antes de que sea efectivo en HIMax.
- 

#### Para comprobar el ComUserTask con la prueba en línea

1. Seleccione **Configuration, Resource, Program** en el árbol.
2. Haga clic con el botón derecho en **Program** y seleccione **Online** en el menú contextual. Realice un inicio de sesión del sistema.

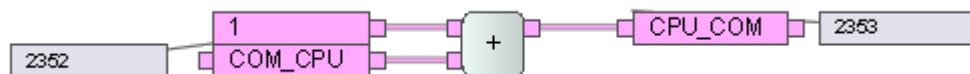


Fig. 108: Prueba en línea SILworX

#### Funciones del programa del usuario SILworX:

El programa del usuario SILworX añade a la señal **COM\_CPU** (entradas de datos) el valor 1 y envía el resultado a la señal **CPU\_COM** (salidas de datos).

En la siguiente invocación de CUT (intervalo regular [ms]), la señal **CPU\_COM** se envía a la función CUT (véase el código de ejemplo en el capítulo 12.8.2).

El ComUserTask recibe la señal **CPU\_COM** y devuelve el valor con la señal **COM\_CPU** sin variar.

#### 12.8.4 Error al cargar una configuración con CUT

**Problemas en los tiempos de ejecución (p. ej., COM-User-Task en bucle sin fin):**

**Causa de los problemas en los tiempos de ejecución:**

Si se ha programado un bucle en el código fuente CUT correspondiente que puede tener un tiempo de ejecución muy largo, esto puede producir un “atasco” en el procesador COM.

Como consecuencia de ello, ya no es posible establecer una conexión con el sistema de control ni borrar la configuración de los recursos.

**Solución:** Reinicialice el módulo de comunicación HIMax o el sistema de control HIMax:

- Reinicialice el módulo de comunicación en la vista en línea del editor de hardware mediante la función **Maintenance/Service, Module (Restart)** (reinicialización de HIMatrix a través del botón de reset, véase la hoja de características del sistema de control).
- Cree un nuevo CUT (a ser posible sin errores de tiempo de ejecución, bucle sin fin).
- Cargue el CUT (archivo ldb) en el proyecto.
- Genere el código.
- Cargue el código en el sistema de control.

## 13 Generalidades

### 13.1 Configuración de los bloques funcionales

Los protocolos del bus de campo y los bloques funcionales correspondientes funcionan en el módulo COM del sistema de control HIMax. Por ello, los bloques funcionales deben ser creados en el árbol SILworX en **Configuration, Resource, Protocols...**

Para controlar estos bloques funcionales en el módulo COM, pueden crearse bloques funcionales en el programa del usuario de SILworX (véase el capítulo 13.1.1), los cuales pueden usarse como bloques funcionales estándar.

La conexión de los bloques funcionales en el programa del usuario de SILworX con los bloques funcionales correspondientes en el árbol de SILworX se realiza a través de variables comunes. Éstas deberá crearlas previamente el usuario en el editor de variables.

#### 13.1.1 Adquisición de bibliotecas de bloques funcionales

Las bibliotecas de bloques funcionales para PROFIBUS DP y TCP Send/Receive deben agregarse al proyecto mediante la función *Restore...* (menú contextual del proyecto).

Puede obtener la biblioteca de bloques funcionales solicitándola al servicio de asistencia al cliente de HIMA.

**Tel.: +49-(0)6202-709 -185  
-259  
-261**

**E-Mail: support@hima.com**

#### 13.1.2 Configuración de los bloques funcionales en el programa del usuario

Puede copiar los bloques funcionales que necesite arrastrándolos con el ratón al programa del usuario. Configure las entradas y salidas después de la descripción de los bloques funcionales correspondientes.

##### Parte superior del bloque funcional

La parte superior del bloque funcional se corresponde con la interfaz de usuario mediante la cual aquél es controlado por el programa del usuario.

Aquí se conectan las variables que se emplean en el programa del usuario. El prefijo "A" quiere decir "Aplicación".

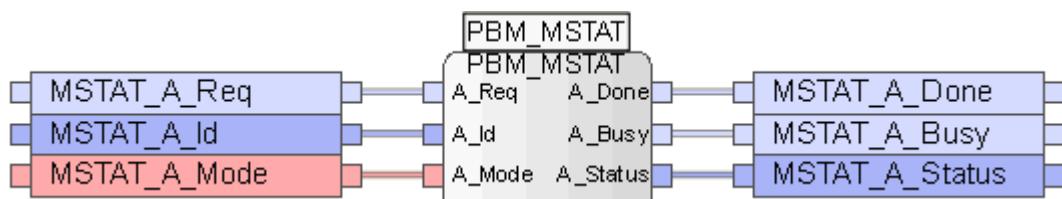


Fig. 109: Bloque funcional PNM\_MSTAT (parte superior)

### Parte inferior del bloque funcional

La parte inferior del bloque funcional establece la conexión con el bloque funcional (en el árbol de SILworX).

Aquí se conectan las variables que deben estar conectadas con el bloque funcional en el árbol SILworX. El prefijo "F" quiere decir "Field".

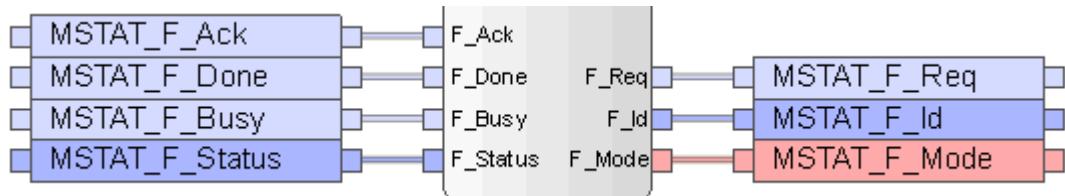


Fig. 110: Bloque funcional PNM\_MSTST (parte inferior)

#### 13.1.3 Configuración de los bloques funcionales en el árbol de SILworX

##### Para configurar los bloques funcionales en el árbol de SILworX:

1. Seleccione **Configuration, Resource, Protocols**, p. ej., **PROFIBUS Master** en el árbol.
2. Haga clic con el botón derecho del ratón en **Function Blocks** y seleccione **New**.
3. Seleccione los bloques funcionales adecuados (en el árbol de SILworX).



Fig. 111: Selección de bloques funcionales

Las entradas del bloque funcional (marcas en la columna de variables de entrada) deben estar conectadas con las mismas variables que están conectadas con las *F-Outputs* del bloque funcional en el programa del usuario.

Las salidas del bloque funcional (sin marca en la columna de variables de entrada) deben estar conectadas con las mismas variables que están conectadas con las *F-Inputs* del bloque funcional en el programa del usuario.

Systemvariablen						
F	Name	Datentyp	Transfer-Operation	Eingangsvariable	Globale Variable	
1	ACK	BOOL		5 <input checked="" type="checkbox"/>	MSTAT_F_Ack	
2	BUSY	BOOL		5 <input checked="" type="checkbox"/>	MSTAT_F_Busy	
3	DONE	BOOL		5 <input checked="" type="checkbox"/>	MSTAT_F_Done	
4	REQ	BOOL		5 <input type="checkbox"/>	MSTAT_F_Req	
5	M_ID	DWORD		5 <input type="checkbox"/>	MSTAT_F_Id	
6	STATUS	DWORD		5 <input checked="" type="checkbox"/>	MSTAT_F_Status	
7	MODE	INT		5 <input type="checkbox"/>	MSTAT_F_Mode	

Fig. 112: Variables de sistema del bloque funcional MSTAT

## 13.2 Intervalo de tiempo de comunicación máximo

El intervalo de tiempo de comunicación máximo es el tiempo asignado en milisegundos (ms) por ciclo de CPU dentro del cual el módulo procesador ejecuta las tareas de comunicación.

Si no es posible procesar todas las tareas de comunicación pendientes, la transmisión completa de los datos de comunicación se realiza a través de varios ciclos de CPU (cantidad de intervalos de comunicación > 1).

### 13.2.1 Para sistemas de control HIMax

#### Determinación del intervalo de tiempo de comunicación máximo

1. Haga funcionar el sistema HIMax a su carga máxima:  
Todos los protocolos de comunicación estarán en uso (safeethernet y protocolos estándar).
2. Abra el **Control Panel** y seleccione **Statistics** en el árbol.
3. Abra **Cyc.Async** para leer la cantidad de intervalos de comunicación.
4. Abra **Time.Async** para leer la duración de los intervalos de tiempo.



- Para el cálculo de los tiempos de reacción máximos admisibles, véase el capítulo 4.7, existe la condición de que la cantidad de los intervalos de comunicación es = 1.  
La duración del intervalo de comunicación debe ajustarse a un valor tan alto que no pueda superar el ciclo de CPU del tiempo de Watchdog predefinido cuando ha usado el intervalo de tiempo de comunicación.

## 13.3 Limitación de carga

Para cada protocolo de comunicación puede predeterminarse un presupuesto de tiempo de cálculo en % ( $\mu P$ -Budget). De esta manera es posible distribuir el tiempo de cálculo disponible entre los protocolos configurados. La suma de los presupuestos de tiempo de cálculo de todos los protocolos parametrizados de un módulo CPU o COM no debe ser mayor que 100%.

Los presupuestos de tiempo de cálculo determinados de cada protocolo de comunicación se monitorizan. Si un protocolo de comunicación ha alcanzado o sobrepasado su presupuesto de tiempo de cálculo y no hay a disposición un tiempo de cálculo adicional de reserva, el protocolo de comunicación no se ejecuta completamente.

Si todavía queda suficiente tiempo de cálculo, éste se emplea para ejecutar un protocolo de comunicación que ha alcanzado o sobrepasado su presupuesto de tiempo de cálculo. De esta manera puede ocurrir que un protocolo de comunicación reciba un presupuesto de tiempo de cálculo mayor que el que le fue asignado.

Es posible que se muestre en línea un presupuesto de tiempo de cálculo superior a 100%. Esto no es un error, el presupuesto de tiempo de cálculo superior a 100% es el tiempo de cálculo empleado adicionalmente.



- El presupuesto adicional de tiempo de cálculo no es en absoluto una garantía para un protocolo de comunicación concreto y puede ser retirado en cualquier momento por el sistema.



## Anexo

### Glosario

Término	Descripción
ARP	Address Resolution Protocol: protocolo de red para asignar direcciones de red a direcciones de hardware
AI	Analog input: entrada analógica
Connector Board	Tarjeta de conexión para módulo HIMax
COM	Módulo de comunicación
CRC	Cyclic redundancy check: suma de verificación
DI	Digital input: entrada digital
DO	Digital output: salida digital
EMC	Compatibilidad electromagnética
EN	Normas europeas
ESD	ElectroStatic Discharge: descarga electrostática
FB	Bus de campo
FBS	Lenguaje de bloques funcionales
FTA	Field Termination Assembly
FTT	Tiempo de tolerancia de errores
ICMP	Internet Control Message Protocol: protocolo de red para mensajes de estado y de error
IEC	Normas internacionales de electrotecnia
Dirección MAC	Dirección de hardware de una conexión de red (Media Access Control)
PADT	Programming and Debugging Tool (según IEC 61131-3), PC con SILworX
PE	Tierra de protección
PELV	Protective Extra Low Voltage: baja tensión funcional con separación segura
PES	Programmable Electronic System
PFD	Probability of Failure on Demand: probabilidad de un fallo al solicitar una función de seguridad
PFH	Probability of Failure per Hour: probabilidad de un fallo peligroso por hora
R	Read
ID de Rack	Identificación (número) de un rack
Sin repercusiones	Suponiendo que hay dos circuitos de entrada conectados a la misma fuente (p. ej. transmisor). Entonces un circuito de entrada se denominará "sin repercusiones", cuando no falsee las señales del otro circuito de entrada.
R/W	Read/Write
SB	Bus de sistema (módulo de bus)
SELV	Safety Extra Low Voltage: baja tensión de protección
SFF	Safe Failure Fraction: porcentaje de fallos fácilmente dominables
SIL	Safety Integrity Level (según IEC 61508)
SILworX	Software de programación para HIMax y HIMatrix
SNTP	Simple Network Time Protocol (RFC 1769)
SRS	Sistema.Rack.Slot
SW	Software
TMO	TimeOut
W	Write
WD	WatchDog
WDT	Tiempo de WatchDog

## Índice de ilustraciones

<b>Fig. 1:</b>	<b>Estructuras del sistema</b>	<b>38</b>
<b>Fig. 2:</b>	<b>Disposición para configurar una conexión redundante</b>	<b>40</b>
<b>Fig. 3:</b>	<b>Árbol del recurso</b>	<b>40</b>
<b>Fig. 4:</b>	<b>Valores de los parámetros de una conexión safeethernet redundante</b>	<b>41</b>
<b>Fig. 5:</b>	<b>Vista en detalle del editor de safeethernet</b>	<b>41</b>
<b>Fig. 6:</b>	<b>Conexión redundante entre dos sistemas de control HIMax</b>	<b>50</b>
<b>Fig. 7:</b>	<b>Conexión redundante de dos sistemas de control HIMax mediante un cable</b>	<b>51</b>
<b>Fig. 8:</b>	<b>Tiempo de reacción en caso de conectar dos sistemas de control HIMax</b>	<b>57</b>
<b>Fig. 9:</b>	<b>Tiempo de reacción de un HIMax en conjunción con un sistema de control HIMatrix</b>	<b>57</b>
<b>Fig. 10:</b>	<b>Tiempo de reacción con dos I/Os remotas y un sistema de control HIMax</b>	<b>58</b>
<b>Fig. 11:</b>	<b>Tiempo de reacción con dos sistemas de control HIMax y un sistema de control HIMatrix</b>	<b>58</b>
<b>Fig. 12:</b>	<b>Tiempo de reacción en caso de conectar dos sistemas de control HIMatrix</b>	<b>59</b>
<b>Fig. 13:</b>	<b>Tiempo de reacción con I/Os remotas</b>	<b>60</b>
<b>Fig. 14:</b>	<b>Tiempo de reacción con dos sistemas de control HIMatrix y un HIMax</b>	<b>60</b>
<b>Fig. 15:</b>	<b>Conexión safeethernet entre el recurso A1 del proyecto A y el recurso B1 del proyecto B</b>	<b>66</b>
<b>Fig. 16:</b>	<b>Variante con el proyecto A como proyecto local</b>	<b>67</b>
<b>Fig. 17:</b>	<b>Variante con el proyecto B como proyecto local</b>	<b>68</b>
<b>Fig. 18:</b>	<b>Comunicación trascendente a todo el proyecto entre SILworX y ELOP II Factory</b>	<b>69</b>
<b>Fig. 19:</b>	<b>Recurso Proxy de HIMatrix</b>	<b>70</b>
<b>Fig. 20:</b>	<b>Parámetros de una conexión safeethernet al recurso Proxy</b>	<b>71</b>
<b>Fig. 21:</b>	<b>Exportación de una conexión safeethernet</b>	<b>72</b>
<b>Fig. 22:</b>	<b>Importación de conexiones a ELOP II Factory</b>	<b>73</b>
<b>Fig. 23:</b>	<b>Editor Peer-to-Peer en ELOP II Factory</b>	<b>73</b>
<b>Fig. 24:</b>	<b>Asignación de señales de envío en ELOP II Factory</b>	<b>74</b>
<b>Fig. 25:</b>	<b>Asignación de señales de recepción en ELOP II Factory</b>	<b>74</b>
<b>Fig. 26:</b>	<b>Panel de control para supervisar la conexión safeethernet</b>	<b>75</b>
<b>Fig. 27:</b>	<b>Control del estado de consumidor/proveedor (IOxS)</b>	<b>78</b>
<b>Fig. 28:</b>	<b>Byte de estado y byte de control de PROFIsafe</b>	<b>80</b>
<b>Fig. 29:</b>	<b>Tiempo de reacción entre un F-Device y un F-Host HIMA</b>	<b>82</b>
<b>Fig. 30:</b>	<b>Tiempo de reacción con un F-Host HIMA y dos F-Devices</b>	<b>82</b>
<b>Fig. 31:</b>	<b>Controlador PROFINET IO el el árbol de HIMax</b>	<b>87</b>
<b>Fig. 32:</b>	<b>Device Access Point (DAP) para el dispositivo PROFINET IO</b>	<b>88</b>
<b>Fig. 33:</b>	<b>Estructura en árbol del controlador PROFINET IO</b>	<b>92</b>
<b>Fig. 34:</b>	<b>Comunicación mediante PROFINET/PROFIsafe</b>	<b>106</b>
<b>Fig. 35:</b>	<b>Dispositivo PROFINET IO en la estructura de árbol de HIMax</b>	<b>112</b>
<b>Fig. 36:</b>	<b>Comunicación mediante PROFIBUS DP</b>	<b>123</b>
<b>Fig. 37:</b>	<b>PROFIBUS DP Slave HIMax con módulos</b>	<b>126</b>

<b>Fig. 38:</b>	<b>Recuadro de datos de usuario</b>	129
<b>Fig. 39:</b>	<b>Cuadro de diálogo “Verification”</b>	129
<b>Fig. 40:</b>	<b>Propiedades de PROFIBUS DP Master</b>	130
<b>Fig. 41:</b>	<b>Propiedades de PROFIBUS DP Slave</b>	131
<b>Fig. 42:</b>	<b>Ciclo isócrono PROFIBUS DP</b>	140
<b>Fig. 43:</b>	<b>Cuadro de diálogo <i>Edit User Parameters</i></b>	150
<b>Fig. 44:</b>	<b>Bloque funcional MSTAT</b>	152
<b>Fig. 45:</b>	<b>Bloque funcional RALRM</b>	155
<b>Fig. 46:</b>	<b>Bloque funcional RDIAG</b>	159
<b>Fig. 47:</b>	<b>Bloque funcional RDREC</b>	163
<b>Fig. 48:</b>	<b>Bloque funcional SLACT</b>	166
<b>Fig. 49:</b>	<b>Bloque funcional WRREC</b>	169
<b>Fig. 50:</b>	<b>Bloque funcional auxiliar ACTIVE</b>	172
<b>Fig. 51:</b>	<b>Bloque funcional auxiliar ALARM</b>	173
<b>Fig. 52:</b>	<b>Bloque funcional auxiliar DEID</b>	174
<b>Fig. 53:</b>	<b>Bloque funcional auxiliar ID</b>	175
<b>Fig. 54:</b>	<b>Bloque funcional auxiliar NSLOT</b>	176
<b>Fig. 55:</b>	<b>Bloque funcional auxiliar SLOT</b>	176
<b>Fig. 56:</b>	<b>Bloque funcional auxiliar STDDIAG</b>	177
<b>Fig. 57:</b>	<b>Topología de bus RS485</b>	191
<b>Fig. 58:</b>	<b>Comunicación mediante Modbus TCP</b>	195
<b>Fig. 59:</b>	<b>Red Modbus</b>	211
<b>Fig. 60:</b>	<b>Modbus Gateway</b>	214
<b>Fig. 61:</b>	<b>Modbus serie</b>	217
<b>Fig. 62:</b>	<b>Telegrama Modbus</b>	217
<b>Fig. 63:</b>	<b>Conexión de HIMax y el sistema de control Siemens</b>	244
<b>Fig. 64:</b>	<b>Transmisión de datos entre HIMax y sistema de control Siemens</b>	245
<b>Fig. 65:</b>	<b>Lista de variables en el bloque UDT1 de Siemens</b>	246
<b>Fig. 66:</b>	<b>Lista de variables en el bloque DB1 de Siemens</b>	247
<b>Fig. 67:</b>	<b>Editor de símbolos de SIMATIC</b>	247
<b>Fig. 68:</b>	<b>Esquema de funciones para recepción</b>	248
<b>Fig. 69:</b>	<b>Esquema de funciones para envío</b>	249
<b>Fig. 70:</b>	<b>Propiedades de la conexión TCP en SILworX</b>	250
<b>Fig. 71:</b>	<b>Lista de variables de Siemens</b>	259
<b>Fig. 72:</b>	<b>Lista de variables de HIMax/HIMatrix</b>	259
<b>Fig. 73:</b>	<b>Bloque funcional TCP_Reset</b>	261
<b>Fig. 74:</b>	<b>Bloque funcional TCP_Send</b>	264
<b>Fig. 75:</b>	<b>Bloque funcional TCP_Receive</b>	267
<b>Fig. 76:</b>	<b>Bloque funcional TCP_ReceiveLine</b>	271
<b>Fig. 77:</b>	<b>Bloque funcional TCP_ReceiveVar</b>	275

<b>Fig. 78:</b>	<b>Composición de los paquetes de datos</b>	276
<b>Fig. 79:</b>	<b>Funcionamiento redundante de X-OPC</b>	291
<b>Fig. 80:</b>	<b>Rutina de instalación del servidor X-OPC</b>	292
<b>Fig. 81:</b>	<b>Rutina de instalación del servidor X-OPC</b>	292
<b>Fig. 82:</b>	<b>Ajuste manual del ClassID del segundo servidor X-OPC</b>	293
<b>Fig. 83:</b>	<b>Funcionamiento redundante de X-OPC</b>	295
<b>Fig. 84:</b>	<b>Funcionamiento redundante de X-OPC</b>	296
<b>Fig. 85:</b>	<b>Editor Alarm&amp;Event</b>	299
<b>Fig. 86:</b>	<b>Funcionamiento redundante de X-OPC</b>	300
<b>Fig. 87:</b>	<b>Editor de safeethernet</b>	303
<b>Fig. 88:</b>	<b>Vista en detalle de la conexión safeethernet</b>	304
<b>Fig. 89:</b>	<b>Cinco rangos de un evento escalar</b>	306
<b>Fig. 90:</b>	<b>Diagrama de bloques</b>	314
<b>Fig. 91:</b>	<b>Intercambio de datos de proceso entre CPU y COM (CUT)</b>	321
<b>Fig. 92:</b>	<b>Cuadro de diálogo de instalación de Cygwin <i>Cygwin Setup</i></b>	359
<b>Fig. 93:</b>	<b>Cuadro de diálogo de instalación de Cygwin <i>Select Packages</i></b>	361
<b>Fig. 94:</b>	<b>Estructura en árbol de Cygwin</b>	362
<b>Fig. 96:</b>	<b>Estructura en árbol de Cygwin</b>	363
<b>Fig. 97:</b>	<b>Archivo de código C en la carpeta <i>example_cut</i></b>	363
<b>Fig. 98:</b>	<b>Archivo mke en la carpeta <i>example_cut</i></b>	364
<b>Fig. 99:</b>	<b>Archivo mke a partir de la línea 1</b>	364
<b>Fig. 100:</b>	<b>makefile en la carpeta <i>example_cut</i></b>	365
<b>Fig. 101:</b>	<b>makefile a partir de la línea 34</b>	365
<b>Fig. 102:</b>	<b>Archivo makeinc.inc.app en la carpeta <i>example_cut</i></b>	365
<b>Fig. 103:</b>	<b>makeinc.inc.app a partir de la línea 247</b>	366
<b>Fig. 104:</b>	<b>Subtítulo de imagen</b>	367
<b>Fig. 105:</b>	<b>C-Code <i>example_cut.c</i></b>	368
<b>Fig. 106:</b>	<b>Cygwin Bash Shell</b>	369
<b>Fig. 107:</b>	<b>Editor de programa SILworX</b>	372
<b>Fig. 108:</b>	<b>Prueba en línea SILworX</b>	372
<b>Fig. 109:</b>	<b>Bloque funcional PNM_MSTST (parte superior)</b>	374
<b>Fig. 110:</b>	<b>Bloque funcional PNM_MSTST (parte inferior)</b>	375
<b>Fig. 111:</b>	<b>Selección de bloques funcionales</b>	375
<b>Fig. 112:</b>	<b>Variables de sistema del bloque funcional MSTAT</b>	376

## Índice de tablas

<b>Tabla 1:</b> Manuales vigentes adicionales	11
<b>Tabla 2:</b> Normas de compatibilidad electromagnética, clima y medio ambiente	14
<b>Tabla 3:</b> Condiciones generales	14
<b>Tabla 4:</b> Condiciones climáticas	14
<b>Tabla 5:</b> Ensayos mecánicos	15
<b>Tabla 6:</b> Ensayos de inmunidad a interferencias	15
<b>Tabla 7:</b> Ensayos de emisión de interferencias	15
<b>Tabla 8:</b> Evaluación de las características de la fuente de corriente continua	16
<b>Tabla 9:</b> Protocolos estándar	19
<b>Tabla 10:</b> Protocolos estándar disponibles	20
<b>Tabla 11:</b> Protocolos en un módulo de comunicación	20
<b>Tabla 12:</b> Protocolos en un módulo de comunicación	20
<b>Tabla 13:</b> Protocolos disponibles de HIMax/HIMatrix	22
<b>Tabla 14:</b> Propiedades de las interfaces Ethernet	23
<b>Tabla 15:</b> Parámetros de configuración	27
<b>Tabla 16:</b> Parámetros de enrutado	27
<b>Tabla 17:</b> Parámetros del switch Ethernet	28
<b>Tabla 18:</b> Ficha VLAN	28
<b>Tabla 19:</b> Valores para LLDP	29
<b>Tabla 20:</b> Submódulos de bus de campo disponibles	31
<b>Tabla 21:</b> Opciones para interfaces de bus de campo FB1 y FB2	31
<b>Tabla 22:</b> Componentes HIMax disponibles	32
<b>Tabla 23:</b> Ejemplos de números de referencia y designaciones de módulos COM	32
<b>Tabla 24:</b> Equipamiento de sistemas de control HIMatrix con submódulos de bus de campo	32
<b>Tabla 25:</b> Ejemplos de números de referencia de los sistemas de control HIMatrix	33
<b>Tabla 26:</b> Asignación de pins de las conexiones D-Sub para RS485	33
<b>Tabla 27:</b> Asignación de pins de las conexiones D-Sub para PROFIBUS DP	34
<b>Tabla 28:</b> Asignación de pins de las conexiones D-Sub para INTERBUS	34
<b>Tabla 29:</b> Asignación de pins de las conexiones D-Sub para RS232	34
<b>Tabla 30:</b> Asignación de pins de las conexiones D-Sub para RS422	35
<b>Tabla 31:</b> Asignación de pins de las conexiones D-Sub para SSI	35
<b>Tabla 32:</b> Protocolo con función orientada a la seguridad (safeethernet)	36
<b>Tabla 33:</b> Parámetros de protocolo safeethernet	45
<b>Tabla 34:</b> Ficha de variables de sistema del editor de safeethernet	48
<b>Tabla 35:</b> Interfaces Ethernet disponibles	49
<b>Tabla 36:</b> Combinaciones para conexiones safeethernet	50
<b>Tabla 37:</b> Recuadro de lectura de conexión safeethernet	76
<b>Tabla 38:</b> Sinopsis de bloques funcionales de PROFINET-IO	77

<b>Tabla 39:</b> Equipamiento y sistema necesarios para controlador PROFINET IO	86
<b>Tabla 40:</b> Propiedades del controlador PROFINET IO	86
<b>Tabla 41:</b> Ficha PROFINET-IO (propiedades)	93
<b>Tabla 42:</b> Ficha “Parameters” (propiedades)	94
<b>Tabla 43:</b> Ficha “Parameters” (propiedades)	94
<b>Tabla 44:</b> Ficha “Parameters”	95
<b>Tabla 45:</b> Ficha “System Variables”	95
<b>Tabla 46:</b> Ficha “Parameters” del módulo PROFINET IO de entrada/salida	96
<b>Tabla 47:</b> Ficha “Parameters”	97
<b>Tabla 48:</b> Ficha “System Variables”	100
<b>Tabla 49:</b> Parámetros F de entrada de submódulo (propiedades)	101
<b>Tabla 50:</b> Ficha “Parameters”	102
<b>Tabla 51:</b> Ficha “System Variables”	103
<b>Tabla 52:</b> Ficha “Parameters”	104
<b>Tabla 53:</b> Ficha “System Variables”	105
<b>Tabla 54:</b> Relación de aplicación (propiedades)	106
<b>Tabla 55:</b> Alarm CR (propiedades)	107
<b>Tabla 56:</b> Input CR (propiedades)	108
<b>Tabla 57:</b> Input CR (editar)	109
<b>Tabla 58:</b> Output CR (propiedades)	110
<b>Tabla 59:</b> Equipamiento y sistema necesarios para controlador PROFINET IO	111
<b>Tabla 60:</b> Propiedades del controlador PROFINET IO	111
<b>Tabla 61:</b> Propiedades generales del dispositivo PROFINET IO	115
<b>Tabla 62:</b> Módulos PROFINET IO	116
<b>Tabla 63:</b> Módulos PROFIsafe	117
<b>Tabla 64:</b> Propiedades generales del módulo de dispositivo	118
<b>Tabla 65:</b> Cuadro de diálogo “Edit” de submódulos	120
<b>Tabla 66:</b> Equipamiento y sistema necesarios	122
<b>Tabla 67:</b> Propiedades de PROFIBUS DP Master	122
<b>Tabla 68:</b> Salidas del PROFIBUS DP Slave de HIMA	124
<b>Tabla 69:</b> Entradas del PROFIBUS DP Slave de HIMA	124
<b>Tabla 70:</b> Variables del módulo de entrada [000] DP-Input/ELOP-Export: 2 bytes	127
<b>Tabla 71:</b> Variables del módulo de entrada [001] DP-Input/ELOP-Export: 8 bytes	127
<b>Tabla 72:</b> Variables del módulo de entrada [002] DP-Input/ELOP-Export: 1 byte	128
<b>Tabla 73:</b> Variables del módulo de salida [003] DP-Output/ELOP-Import: 2 bytes	128
<b>Tabla 74:</b> Variables del módulo de salida [004] DP-Output/ELOP-Import: 1 byte	128
<b>Tabla 75:</b> Variables de sistema del PROFIBUS DP Master	132
<b>Tabla 76:</b> Propiedades generales de PROFIBUS DP Master	133
<b>Tabla 77:</b> Ficha “Timings” en el cuadro de diálogo de propiedades de PROFIBUS DP Master	135

<b>Tabla 78:</b> Ficha “CPU/COM” en el cuadro de diálogo de propiedades de PROFIBUS DP Master	135
<b>Tabla 79:</b> Otras propiedades de PROFIBUS DP Master	136
<b>Tabla 80:</b> Valores de referencia para tiempo de rotación de Token con diversas velocidades de transmisión	137
<b>Tabla 81:</b> Tiempo de transmisión para un carácter con diversas velocidades de transmisión	138
<b>Tabla 82:</b> Elementos de cálculo del tiempo de rotación del Token	138
<b>Tabla 83:</b> Variables de sistema del PROFIBUS DP Slave	142
<b>Tabla 84:</b> Ficha “Parameters” del PROFIBUS DP Slave	143
<b>Tabla 85:</b> Ficha “Groups” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave	144
<b>Tabla 86:</b> Ficha DP-V1 en el cuadro de diálogo de propiedades de PROFIBUS DP Slave	144
<b>Tabla 87:</b> Ficha “Alarms” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave	145
<b>Tabla 88:</b> Ficha “Data” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave	145
<b>Tabla 89:</b> Ficha “Model” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave	146
<b>Tabla 90:</b> Ficha “Features” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave	146
<b>Tabla 91:</b> Ficha “Baud Rates” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave	147
<b>Tabla 92:</b> Ficha “Acyclic” en el cuadro de diálogo de propiedades de PROFIBUS DP Slave	147
<b>Tabla 93:</b> Archivo GSD del PROFIBUS DP Slave HIMax/HIMatrix	148
<b>Tabla 94:</b> Ejemplo: Bloques 1...4 del recuadro de datos de usuario	150
<b>Tabla 95:</b> Ejemplo: Bloques 1...4 del recuadro de datos de usuario	150
<b>Tabla 96:</b> Sinopsis de los bloques funcionales de PROFIBUS DP	151
<b>Tabla 97:</b> Entradas A del bloque funcional MSTAT	152
<b>Tabla 98:</b> Salidas A del bloque funcional MSTAT	152
<b>Tabla 99:</b> Entradas F del bloque funcional MSTAT	153
<b>Tabla 100:</b> Salidas F del bloque funcional MSTAT	153
<b>Tabla 101:</b> Variables de entrada del sistema	153
<b>Tabla 102:</b> Variables de salida del sistema	154
<b>Tabla 103:</b> Entradas A del bloque funcional RALRM	155
<b>Tabla 104:</b> Salidas A del bloque funcional RALRM	156
<b>Tabla 105:</b> Entradas F del bloque funcional RALRM	156
<b>Tabla 106:</b> Salidas F del bloque funcional RALRM	156
<b>Tabla 107:</b> Variables de entrada del sistema	157
<b>Tabla 108:</b> Variables de salida del sistema	157
<b>Tabla 109:</b> Datos de alarma	158

<b>Tabla 110: Entradas A del bloque funcional RALRM</b>	159
<b>Tabla 111: Salidas A del bloque funcional RALRM</b>	159
<b>Tabla 112: Entradas F del bloque funcional RDIAG</b>	160
<b>Tabla 113: Salidas F del bloque funcional RDIAG</b>	160
<b>Tabla 114: Variables de entrada del sistema</b>	160
<b>Tabla 115: Variables de salida del sistema</b>	161
<b>Tabla 116: Datos de diagnóstico</b>	161
<b>Tabla 117: Entradas A del bloque funcional RDREC</b>	163
<b>Tabla 118: Salidas A del bloque funcional RDREC</b>	163
<b>Tabla 119: Entradas F del bloque funcional RDREC</b>	164
<b>Tabla 120: Salidas F del bloque funcional RDREC</b>	164
<b>Tabla 121: Variables de entrada del sistema</b>	164
<b>Tabla 122: Variables de salida del sistema</b>	165
<b>Tabla 123: Datos</b>	165
<b>Tabla 124: Entradas A del bloque funcional SLACT</b>	166
<b>Tabla 125: Salidas A del bloque funcional SLACT</b>	166
<b>Tabla 126: Entradas F del bloque funcional SLACT</b>	167
<b>Tabla 127: Salidas F del bloque funcional SLACT</b>	167
<b>Tabla 128: Variables de entrada del sistema</b>	167
<b>Tabla 129: Variables de salida del sistema</b>	168
<b>Tabla 130: Entradas A del bloque funcional WRREC</b>	169
<b>Tabla 131: Salidas A del bloque funcional WRREC</b>	169
<b>Tabla 132: Entradas F del bloque funcional WRREC</b>	170
<b>Tabla 133: Salidas F del bloque funcional WRREC</b>	170
<b>Tabla 134: Variables de entrada del sistema</b>	170
<b>Tabla 135: Variables de salida del sistema</b>	171
<b>Tabla 136: Datos</b>	171
<b>Tabla 137: Sinopsis de los bloques funcionales auxiliares</b>	172
<b>Tabla 138: Entradas del bloque funcional auxiliar ACTIVE</b>	172
<b>Tabla 139: Salidas del bloque funcional auxiliar ACTIVE</b>	172
<b>Tabla 140: Entradas del bloque funcional auxiliar ALARM</b>	173
<b>Tabla 141: Salidas del bloque funcional auxiliar ALARM</b>	174
<b>Tabla 142: Entradas del bloque funcional auxiliar DEID</b>	174
<b>Tabla 143: Salidas del bloque funcional auxiliar DEID</b>	174
<b>Tabla 144: Entradas del bloque funcional auxiliar ID</b>	175
<b>Tabla 145: Salidas del bloque funcional auxiliar ID</b>	175
<b>Tabla 146: Entradas del bloque auxiliar NSLOT</b>	176
<b>Tabla 147: Salidas del bloque auxiliar NSLOT</b>	176
<b>Tabla 148: Entradas del bloque auxiliar SLOT</b>	177
<b>Tabla 149: Salidas del bloque auxiliar SLOT</b>	177

<b>Tabla 150: Entradas del bloque auxiliar STDDIAG</b>	177
<b>Tabla 151: Salidas del bloque auxiliar STDDIAG</b>	178
<b>Tabla 152: Códigos de error de bloques funcionales</b>	179
<b>Tabla 153: Recuadro de lectura de PROFIBUS Master</b>	182
<b>Tabla 154: Estado del PROFIBUS DP Master</b>	182
<b>Tabla 155: Comportamiento del PROFIBUS DP Master</b>	182
<b>Tabla 156: LED FBx</b>	183
<b>Tabla 157: FAULT FBx</b>	183
<b>Tabla 158: Equipamiento y sistema necesarios para PROFIBUS DP Slave de HIMA</b>	184
<b>Tabla 159: Propiedades de PROFIBUS DP Slave de HIMA</b>	184
<b>Tabla 160: Variables de sistema del PROFIBUS DP Slave</b>	186
<b>Tabla 161: Propiedades de Slave: Ficha “General”</b>	188
<b>Tabla 162: Recuadro de lectura (PROFIBUS Slave)</b>	189
<b>Tabla 163: LED FBx</b>	190
<b>Tabla 164: FAULT FBx</b>	190
<b>Tabla 165: Asignación de bornes de H 7506</b>	192
<b>Tabla 166: Cable de bus</b>	192
<b>Tabla 167: Características de transmisión por RS485</b>	193
<b>Tabla 168: Equipamiento y sistema necesarios para Modbus Master</b>	194
<b>Tabla 169: Propiedades de PROFIBUS DP Master</b>	195
<b>Tabla 170: Variables de sistema Modbus Master</b>	201
<b>Tabla 171: Propiedades generales de Modbus Master</b>	203
<b>Tabla 172: Parámetros de COM/CPU</b>	203
<b>Tabla 173: Códigos de función Modbus</b>	204
<b>Tabla 174: Telegrama de requerimiento Read Coils</b>	207
<b>Tabla 175: Telegrama de requerimiento Read Discrete Inputs</b>	207
<b>Tabla 176: Telegrama de requerimiento Read Holding Registers</b>	207
<b>Tabla 177: Telegrama de requerimiento Read Input Registers</b>	208
<b>Tabla 178: Register Read Write Holding</b>	209
<b>Tabla 179: Telegrama de requerimiento Write Multiple Coils</b>	209
<b>Tabla 180: Telegrama de requerimiento Write Multiple Registers</b>	210
<b>Tabla 181: Telegrama de requerimiento Write Single Coil (05)</b>	210
<b>Tabla 182: Telegrama de requerimiento Write Single Register</b>	210
<b>Tabla 183: Variables de sistema de Slaves TCP/UDP</b>	212
<b>Tabla 184: Parámetros de configuración</b>	213
<b>Tabla 185: Parámetros de conexión de Modbus Gateway</b>	216
<b>Tabla 186: Variables de estado de Gateway Slave</b>	216
<b>Tabla 187: Parámetros de conexión de Gateway Slave</b>	216
<b>Tabla 188: Parámetros de Modbus Master serie</b>	218
<b>Tabla 189: Variables de sistema Modbus Slave</b>	219

<b>Tabla 190: Parámetros de conexión de Modbus Master</b>	219
<b>Tabla 191: Recuadro de lectura de Modbus Master</b>	220
<b>Tabla 192: LED FBx</b>	221
<b>Tabla 193: FAULT FBx</b>	221
<b>Tabla 194: Equipamiento y sistema necesarios para Modbus Slave HIMA</b>	222
<b>Tabla 195: Propiedades de Modbus Slave</b>	222
<b>Tabla 196: Slots admisibles de los módulos COM Modbus Slave redundantes</b>	224
<b>Tabla 197: Ficha de propiedades del set Modbus Slave</b>	226
<b>Tabla 198: Ficha Modbus Slave Set</b>	227
<b>Tabla 199: Ficha “TCP and UDP Ports” para Modbus Slave HIMA</b>	229
<b>Tabla 200: Ficha de variables de sistema para Modbus Slave HIMA</b>	229
<b>Tabla 201: Códigos de función Modbus de Modbus Slave HIMA</b>	230
<b>Tabla 202: Ficha de variables del área de registro del Modbus Slave</b>	234
<b>Tabla 203: Variables de bits del área de bits del Modbus Slave</b>	236
<b>Tabla 204: Ficha “Offsets” para Modbus Slave HIMA</b>	237
<b>Tabla 205: Variables reflejadas en espejo del área de registros al área de bits</b>	237
<b>Tabla 206: Variables reflejadas del área de bits al área de registros</b>	239
<b>Tabla 207: Recuadro de lectura Modbus Slave</b>	241
<b>Tabla 208: Recuadro de lectura de datos del Master</b>	241
<b>Tabla 209: LED FBx</b>	242
<b>Tabla 210: FAULT FBx</b>	242
<b>Tabla 211: Códigos de error de TCP/IP Modbus</b>	242
<b>Tabla 212: Equipamiento y sistema necesarios para S&amp;R TCP</b>	243
<b>Tabla 213: Propiedades de S&amp;R TCP</b>	243
<b>Tabla 214: Configuración del sistema de control HIMax</b>	245
<b>Tabla 215: Configuración del sistema Siemens SIMATIC 300</b>	245
<b>Tabla 216: Variables globales</b>	250
<b>Tabla 217: Variables para datos de recepción</b>	251
<b>Tabla 218: Variables para datos de envío</b>	251
<b>Tabla 219: Variables de sistema S&amp;R TCP</b>	252
<b>Tabla 220: Propiedades generales de S&amp;R TCP</b>	252
<b>Tabla 221: Parámetros de COM/CPU</b>	253
<b>Tabla 222: Variables de sistema</b>	254
<b>Tabla 223: Propiedades de la conexión S&amp;R TCP</b>	256
<b>Tabla 224: Bloques funcionales para conexiones S&amp;R-TCP</b>	260
<b>Tabla 225: Entradas A del bloque funcional TCP_Reset</b>	261
<b>Tabla 226: Salidas A del bloque funcional TCP_Reset</b>	261
<b>Tabla 227: Entradas F del bloque funcional TCP_Reset</b>	262
<b>Tabla 228: Salidas F del bloque funcional TCP_Reset</b>	262
<b>Tabla 229: Variables de entrada del sistema</b>	262

<b>Tabla 230: Variables de salida del sistema</b>	<b>263</b>
<b>Tabla 231: Entradas A del bloque funcional TCP_Send</b>	<b>264</b>
<b>Tabla 232: Salidas A del bloque funcional TCP_Send</b>	<b>264</b>
<b>Tabla 233: Entradas F del bloque funcional TCP_Send</b>	<b>265</b>
<b>Tabla 234: Salidas F del bloque funcional TCP_Send</b>	<b>265</b>
<b>Tabla 235: Variables de entrada del sistema</b>	<b>265</b>
<b>Tabla 236: Variables de salida del sistema</b>	<b>266</b>
<b>Tabla 237: Datos de envío</b>	<b>266</b>
<b>Tabla 238: Entradas A del bloque funcional TCP_Receive</b>	<b>267</b>
<b>Tabla 239: Salidas A del bloque funcional TCP_Receive</b>	<b>268</b>
<b>Tabla 240: Entradas A del bloque funcional TCP_Receive</b>	<b>268</b>
<b>Tabla 241: Salidas F del bloque funcional TCP_Receive</b>	<b>268</b>
<b>Tabla 242: Variables de entrada del sistema</b>	<b>269</b>
<b>Tabla 243: Variables de salida del sistema</b>	<b>269</b>
<b>Tabla 244: Variables de recepción</b>	<b>269</b>
<b>Tabla 245: Entradas A del bloque funcional TCP_ReceiveLine</b>	<b>271</b>
<b>Tabla 246: Salidas A del bloque funcional TCP_ReceiveLine</b>	<b>272</b>
<b>Tabla 247: Entradas F del bloque funcional TCP_ReceiveLine</b>	<b>272</b>
<b>Tabla 248: Salidas F del bloque funcional TCP_ReceiveLine</b>	<b>272</b>
<b>Tabla 249: Variables de entrada del sistema</b>	<b>273</b>
<b>Tabla 250: Variables de salida del sistema</b>	<b>273</b>
<b>Tabla 251: Variables de recepción</b>	<b>273</b>
<b>Tabla 252: Entradas A del bloque funcional TCP_ReceiveVar</b>	<b>276</b>
<b>Tabla 253: Salidas A del bloque funcional TCP_ReceiveVar</b>	<b>277</b>
<b>Tabla 254: Entradas F del bloque funcional TCP_ReceiveVar</b>	<b>277</b>
<b>Tabla 255: Salidas F del bloque funcional TCP_ReceiveVar</b>	<b>277</b>
<b>Tabla 256: Variables de entrada del sistema</b>	<b>278</b>
<b>Tabla 257: Variables de salida del sistema</b>	<b>278</b>
<b>Tabla 258: Variables de recepción</b>	<b>278</b>
<b>Tabla 259: Recuadro de lectura “S&amp;R Protocol”</b>	<b>280</b>
<b>Tabla 260: Recuadro de Modbus Slave</b>	<b>280</b>
<b>Tabla 261: Códigos de error de la conexión TCP</b>	<b>281</b>
<b>Tabla 262: Códigos de error adicionales</b>	<b>282</b>
<b>Tabla 263: Estado de conexión</b>	<b>282</b>
<b>Tabla 264: Partner Connection State</b>	<b>282</b>
<b>Tabla 265: Equipamiento y sistema necesarios para S&amp;R TCP</b>	<b>283</b>
<b>Tabla 266: Propiedades del cliente SNTP</b>	<b>284</b>
<b>Tabla 267: Propiedades de la información de servidor SNTP</b>	<b>285</b>
<b>Tabla 268: Propiedades del servidor SNTP</b>	<b>286</b>
<b>Tabla 269: Equipamiento y sistema necesarios para servidor X-OPC</b>	<b>287</b>

<b>Tabla 270: Propiedades del servidor X-OPC</b>	288
<b>Tabla 271: Propiedades del sistema de control HIMA para la conexión X-OPC</b>	289
<b>Tabla 272: Acciones necesarias en caso de modificaciones</b>	290
<b>Tabla 273: Valores por defecto de las prioridades</b>	302
<b>Tabla 274: Estado y marca de tiempo de fragmentos de Data Access</b>	304
<b>Tabla 275: Parámetros para eventos booleanos</b>	306
<b>Tabla 276: Parámetros para eventos escalares</b>	308
<b>Tabla 277: Propiedades</b>	312
<b>Tabla 278: Propiedades</b>	313
<b>Tabla 279: Edit</b>	313
<b>Tabla 280: Equipamiento y sistema necesarios para ComUserTask</b>	314
<b>Tabla 281: Formato de datos del byte de comando de inicio SSISTART</b>	315
<b>Tabla 282: Formato y orden de datos de sensores</b>	316
<b>Tabla 283: Frecuencias recomendadas según longitudes de cable de campo</b>	316
<b>Tabla 284: Equipamiento y sistema necesarios para ComUserTask</b>	318
<b>Tabla 285: Propiedades de ComUserTask</b>	318
<b>Tabla 286: Abreviaturas</b>	319
<b>Tabla 287: Intervalo regular [ms]</b>	320
<b>Tabla 288: Propiedades generales del dispositivo PROFINET IO</b>	322
<b>Tabla 289: Variables de sistema de ComUserTask</b>	323
<b>Tabla 290: Señales de entrada de ComUserTask</b>	324
<b>Tabla 291: Señales de salida de ComUserTask</b>	324
<b>Tabla 292: Parámetro</b>	326
<b>Tabla 293: Parámetro</b>	328
<b>Tabla 294: Valor de retorno</b>	328
<b>Tabla 295: Parámetro</b>	329
<b>Tabla 296: Valor de retorno</b>	329
<b>Tabla 297: Parámetro</b>	330
<b>Tabla 298: Valor de retorno</b>	331
<b>Tabla 299: Parámetro</b>	332
<b>Tabla 300: Parámetro</b>	333
<b>Tabla 301: Valor de retorno</b>	333
<b>Tabla 302: Parámetro</b>	334
<b>Tabla 303: Parámetro</b>	335
<b>Tabla 304: Valor de retorno</b>	335
<b>Tabla 305: Valor de retorno</b>	336
<b>Tabla 306: Parámetro</b>	337
<b>Tabla 307: Parámetro</b>	338
<b>Tabla 308: Valor de retorno</b>	338
<b>Tabla 309: Parámetro</b>	339

<b>Tabla 310: Parámetro</b>	340
<b>Tabla 311: Parámetro</b>	341
<b>Tabla 312: Valor de retorno</b>	341
<b>Tabla 313: Parámetro</b>	342
<b>Tabla 314: Parámetro</b>	343
<b>Tabla 315: Valor de retorno</b>	343
<b>Tabla 316: Parámetro</b>	344
<b>Tabla 317: Valor de retorno</b>	344
<b>Tabla 318: Parámetro</b>	345
<b>Tabla 319: Parámetro</b>	346
<b>Tabla 320: Valor de retorno</b>	346
<b>Tabla 321: Parámetro</b>	347
<b>Tabla 322: Parámetro</b>	348
<b>Tabla 323: Valor de retorno</b>	348
<b>Tabla 324: Parámetro</b>	350
<b>Tabla 325: Parámetro</b>	352
<b>Tabla 326: Parámetro</b>	352
<b>Tabla 327: Parámetro</b>	352
<b>Tabla 328: Parámetro</b>	353
<b>Tabla 329: Parámetro</b>	354
<b>Tabla 330: Parámetro</b>	354
<b>Tabla 331: Parámetro</b>	355
<b>Tabla 332: Parámetro</b>	356
<b>Tabla 333: Valor de retorno</b>	356
<b>Tabla 334: Comandos en Cygwin (Bash Shell)</b>	361
<b>Tabla 335: Variables de salida (CPU-&gt;COM)</b>	371
<b>Tabla 336: Variables de entrada (COM-&gt;CPU)</b>	371

**Índice alfabético**

Activación.....	33	Nº de referencia	
Asignación de pins .....	33	HIMatrix.....	32
Condiciones de uso		HIMax.....	32
CEM .....	15	Registro .....	33
climáticas.....	14		
Fuente de alimentación .....	16		
mecánicas .....	15		
Protección contra ESD .....	16		



HI 801 195 S  
© 2010 HIMA Paul Hildebrandt GmbH  
HIMax, HIMatrix y SILworX son marcas registradas de  
HIMA Paul Hildebrandt GmbH

HIMA Paul Hildebrandt GmbH  
Albert-Bassermann-Str. 28 | 68782 Brühl | Alemania  
Teléfono +49 6202 709-0 | Telefax +49 6202 709-107  
info@hima.com | www.hima.com



SAFETY  
NONSTOP



Eine detaillierte Liste aller Niederlassungen und Vertretungen  
finden Sie unter: [www.hima.de/kontakt](http://www.hima.de/kontakt)

