

Modicon M340 con Unity Pro

Conexión serie

Manual del usuario

05/2010

Schneider Electric no asume ninguna responsabilidad ante los posibles errores que aparezcan en este documento. Si tiene alguna sugerencia para llevar a cabo mejoras o modificaciones o si ha encontrado errores en esta publicación, le rogamos que nos lo notifique.

Queda prohibido reproducir cualquier parte de este documento bajo ninguna forma o medio posible, ya sea electrónico, mecánico o fotocopia, sin autorización previa de Schneider Electric.

Deberán tenerse en cuenta todas las normas de seguridad nacionales, regionales y locales pertinentes a la hora de instalar y utilizar este producto. Por razones de seguridad y para garantizar que se siguen los consejos de la documentación del sistema, las reparaciones sólo podrá realizarlas el fabricante.

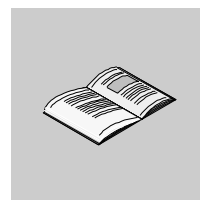
Cuando se utilicen dispositivos para aplicaciones con requisitos técnicos de seguridad, siga las instrucciones pertinentes.

Si no se utiliza el software de Schneider Electric o un software compatible con nuestros productos de hardware, pueden sufrirse daños o lesiones o provocar un funcionamiento inadecuado del dispositivo.

Si no se tiene en cuenta esta información se pueden causar daños personales o en el equipo.

© 2010 Schneider Electric. Reservados todos los derechos.

Tabla de materias



	Información de seguridad	9
	Acerca de este libro	11
Parte I	Introducción a la comunicación en modalidad de caracteres y Modbus serie	13
Capítulo 1	Introducción a la comunicación en modalidad de caracteres y Modbus serie.	15
	Introducción a la comunicación en modalidad de caracteres y Modbus serie.	15
Parte II	Instalación de hardware para las comunicaciones en modalidad de caracteres y Modbus serie.	17
Capítulo 2	Introducción a la comunicación serie	19
2.1	Conexión serie en los procesadores BMX P34 1000/2000/2010/20102/2020	20
	Presentación de las comunicaciones serie en los procesadores BMX P34 1000/2000/2010/20102/2020	20
2.2	Módulo de 2 puertos RS-485/232 BMX NOM 0200	24
	Presentación del módulo de 2 puertos BMX NOM 0200 RS-485/232 ...	25
	Equipo Modicon M340H (Endurecido)	30
	Puesta a tierra de los módulos instalados.	31
	Instalación del módulo BMX NOM 0200	32
	Consideraciones del cableado de BMX NOM 0200	34
Capítulo 3	Arquitecturas de comunicación serie.	35
3.1	Arquitecturas de comunicación serie para procesadores BMX P34 1000/2000/2010/20102/2020	36
	Polarización y terminación de la línea Modbus (RS485).	37
	Dispositivos de conexión Modbus (RS485).	39
	Conexión de equipo terminal de datos (DTE) (RS232).	42
	Conexión de equipo de terminación de circuito de datos (DCE) (RS232)	44

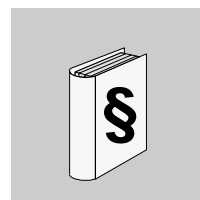
3.2	Arquitectura de comunicación serie para BMX NOM 0200	47
	Polarización y terminación de la línea Modbus (RS485)	48
	Dispositivos de conexión Modbus (RS485)	50
	Conexión de equipo terminal de datos (DTE) (RS232)	52
	Conexión de equipo de terminación de circuito de datos (DCE) (RS232)	54
3.3	Cableado	56
	Cableado	56
Parte III	Implementación de software para las comunica-	
	ciones en modalidad de caracteres y Modbus serie	61
Capítulo 4	Metodología de instalación.	63
	Introducción a la fase de instalación	63
Capítulo 5	Comunicaciones serie en los procesadores	
	BMX P34 1000/2000/2010/20102/2020.	67
5.1	Generalidades	68
	Acerca de Modbus serie	69
	Rendimiento.	70
	Acceso a los parámetros de la conexión serie	72
5.2	Configuración de la comunicación Modbus serie	76
	Pantalla de configuración de la comunicación Modbus serie	77
	Funciones Modbus accesibles.	79
	Valores predeterminados para los parámetros de comunicación Modbus serie	80
	Parámetros Modbus relacionados con la aplicación	81
	Parámetros Modbus relacionados con la transmisión.	83
	Parámetros de línea física y señales en Modbus	86
5.3	Programación de la comunicación Modbus serie	88
	Servicios admitidos por un procesador maestro de conexión Modbus . .	89
	Servicios compatibles con un procesador esclavo de conexión Modbus	97
5.4	Depuración de una comunicación Modbus serie.	99
	Pantalla de depuración de la comunicación Modbus serie	99
Capítulo 6	Comunicaciones serie en los procesadores	
	BMX P34 1000/2000/2010/20102/2020.	101
6.1	Generalidades	102
	Acerca de la comunicación en modalidad de caracteres.	103
	Rendimiento.	104
6.2	Configuración de la comunicación en modalidad de caracteres	106
	Pantalla de configuración de la comunicación en modalidad de caracteres	107
	Funciones accesibles en modalidad de caracteres	109
	Valores predeterminados para los parámetros de comunicación en modalidad de caracteres	110
	Parámetros de detección de final de mensaje en modalidad de caracteres	111
	Parámetros de transmisión en modalidad de caracteres.	113
	Parámetros de línea física y señales en modalidad de caracteres	115

6.3	Programación de la comunicación en modalidad de caracteres.	116
	Funciones de comunicación en modalidad de caracteres	116
6.4	Depuración de la comunicación en modalidad de caracteres	122
	Pantalla de depuración de la comunicación en modalidad de caracteres	122
Capítulo 7	Comunicación en modalidad Modbus serie para	
	BMX NOM 0200	125
7.1	Generalidades.	126
	Acerca de Modbus serie	127
	Rendimiento	128
	Acceso a los parámetros de la conexión serie	130
7.2	Configuración de la comunicación Modbus serie	134
	Pantalla de configuración de la comunicación Modbus serie	135
	Funciones Modbus accesibles	137
	Valores predeterminados para los parámetros de comunicación Modbus serie.	138
	Parámetros Modbus relacionados con la aplicación.	139
	Parámetros Modbus relacionados con la transmisión.	141
	Parámetros de línea física y señales en Modbus	144
	Cómo se establece la dirección del esclavo BMX NOM0200 MODBUS sin Unity Pro?	146
7.3	Programación de la comunicación Modbus serie	148
	Servicios admitidos por un módulo maestro de conexión Modbus	149
	Servicios admitidos por un módulo esclavo de conexión Modbus	157
7.4	Depuración de una comunicación Modbus serie	159
	Pantalla de depuración de la comunicación Modbus serie	159
Capítulo 8	Comunicación en modalidad de caracteres para BMX	
	NOM 0200	163
8.1	Generalidades.	164
	Acerca de la comunicación en modalidad de caracteres	164
8.2	Configuración de la comunicación en modalidad de caracteres.	165
	Pantalla de configuración de la comunicación en modalidad de caracteres	166
	Funciones accesibles en modalidad de caracteres.	168
	Valores predeterminados para los parámetros de comunicación en modalidad de caracteres.	169
	Parámetros de detección de final de mensaje en modalidad de caracteres	170
	Parámetros de transmisión en modalidad de caracteres	172
	Parámetros de línea física y señales en modalidad de caracteres.	174
8.3	Programación de la comunicación en modalidad de caracteres.	176
	Funciones de comunicación en modalidad de caracteres	176
8.4	Depuración de la comunicación en modalidad de caracteres.	183
	Pantalla de depuración de la comunicación en modalidad de caracteres	183

Capítulo 9	Diagnóstico del módulo BMX NOM 0200.....	185
9.1	Diagnóstico del módulo BMX NOM 0200	185
	Diagnóstico del módulo BMX NOM 0200	186
	Diagnóstico detallado por canal de comunicaciones.....	188
Capítulo 10	Objetos de lenguaje de la comunicación en modalidad de caracteres y Modbus	191
10.1	Objetos de lenguaje e IODDT de las comunicaciones en modalidad de caracteres y Modbus	192
	Introducción a los objetos de lenguaje para comunicaciones en modalidad de caracteres y Modbus	193
	Intercambio implícito de objetos de lenguaje asociados a la función específica de la aplicación	194
	Objetos de lenguaje de intercambio explícito asociados a la función específica de aplicaciones	195
	Gestión de intercambios y del informe con objetos explícitos	197
10.2	Objetos de lenguaje general e IODDT para todos los protocolos de comunicación.....	200
	Detalles de los objetos de intercambio implícito de IODDT de tipo T_COM_STS_GEN	201
	Detalles de los objetos de intercambio explícito de IODDT de tipo T_COM_STS_GEN	202
10.3	Objetos de lenguaje e IODDT asociados a la comunicación Modbus ..	204
	Detalles relativos a los objetos de lenguaje de intercambio explícito para una función Modbus	205
	Detalles de los objetos de intercambio implícito de IODDT de tipo T_COM_MB_BMX y T_COM_MB_BMX_CONF_EXT	206
	Detalles de los objetos de intercambio explícito de IODDT de tipo T_COM_MB_BMX y T_COM_MB_BMX_CONF_EXT	208
	Detalles de los objetos de lenguaje asociados a la modalidad Modbus de configuración	212
10.4	Objetos de lenguaje e IODDT asociados a la comunicación en modalidad de caracteres	214
	Detalles relativos a los objetos de lenguaje de intercambio explícito para la comunicación en modalidad de caracteres	215
	Detalles de los objetos de intercambio implícito de IODDT de tipo T_COM_CHAR_BMX	216
	Detalles de los objetos de intercambio explícito de IODDT de tipo T_COM_CHAR_BMX	217
	Detalles de los objetos de lenguaje asociados a la configuración en modalidad de caracteres	220
10.5	El IODDT tipo T_GEN_MOD es aplicable a todos los módulos	222
	Detalles de los objetos de lenguaje del IODDT de tipo T_GEN_MOD ..	222

Capítulo 11	Conmutación dinámica de protocolos	225
	Cambio de protocolo con los procesadores	
	BMX P34 1000/2000/2010/20102/2020	226
	Cambio de protocolo con el módulo BMX NOM 0200.	229
Parte IV	Inicio rápido: Ejemplo de implementación de conexiones serie	231
Capítulo 12	Descripción de la aplicación	233
	Descripción de la aplicación	233
Capítulo 13	Instalación de la aplicación con Unity Pro	235
13.1	Presentación de la solución utilizada	236
	Los distintos pasos del proceso utilizando Unity Pro	236
13.2	Desarrollo de la aplicación	237
	Creación del proyecto	238
	Declaración de variables	243
	Uso de módems	247
	Procedimiento de programación	249
	Estructura de programación	251
	Programación	255
Capítulo 14	Inicio de aplicaciones	265
	Ejecución de la aplicación en modalidad estándar	265
Glosario		269
Índice		279

Información de seguridad



Información importante

AVISO

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



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



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



PELIGRO

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



ADVERTENCIA

ADVERTENCIA indica una situación potencialmente peligrosa que, si no se evita, **puede provocar la** muerte o lesiones graves.

AVISO
--

AVISO indica una situación potencialmente peligrosa que, si no se evita, puede provocar lesiones leves o moderadas.

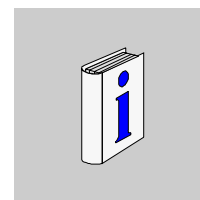
AVISO

AVISO , utilizado sin el símbolo de alerta de seguridad, indica una situación potencialmente peligrosa que, si no se evita, puede provocar daños en el equipo.
--

TENGA EN CUENTA

La instalación, manejo, puesta en servicio y mantenimiento de equipos eléctricos deberán ser realizados sólo por personal cualificado. Schneider Electric no se hace responsable de ninguna de las consecuencias del uso de este material.

Acerca de este libro



Presentación

Objeto

En este manual se describe la instalación del hardware y el software de la comunicación en modalidad de caracteres y Modbus para procesadores BMX P34 1000/2000/2010/20102/2020 y para el módulo de comunicaciones BMX NOM 0200.

Campo de aplicación

Esta documentación es válida para el software Unity Pro desde la versión 5.0.

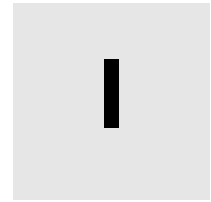
Información relativa al producto

ADVERTENCIA
FUNCIONAMIENTO INESPERADO DEL EQUIPO La aplicación de este producto requiere experiencia en el diseño y la programación de sistemas de control. Sólo las personas con dicha experiencia deberían tener permiso para programar, instalar, alterar y aplicar este producto. Siga todos los estándares y códigos de seguridad nacionales y locales. Si no se siguen estas instrucciones pueden producirse lesiones personales graves o mortales o daños en el equipo.

Comentarios del usuario

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

Introducción a la comunicación en modalidad de caracteres y Modbus serie



Introducción a la comunicación en modalidad de caracteres y Modbus serie

1

Introducción a la comunicación en modalidad de caracteres y Modbus serie

General

La conexión serie de los procesadores BMX P34 1000/2000/2010/20102/2020 y el módulo BMX NOM 0200 admite dos protocolos de comunicación:

- Modbus serie
- Modalidad de caracteres

Protocolo Modbus

Modbus es un protocolo estándar con las siguientes propiedades:

- Establece la comunicación cliente/servidor entre distintos módulos en una conexión bus o serie. El cliente se identifica mediante el módulo maestro, y los módulos esclavos representan a los servidores.
- Se basa en una modalidad de intercambio de datos que consta de peticiones y respuestas que ofrecen servicios mediante distintos códigos de función.
- Establece un medio de intercambio de tramas de aplicaciones de tipo Modbus en dos tipos de código:
 - Modalidad RTU
 - Modalidad ASCII

El procedimiento de gestión de intercambio es el siguiente:

- Un solo dispositivo puede enviar datos por el bus.
- Los intercambios se gestionan mediante el maestro. Sólo el maestro puede iniciar intercambios. Los esclavos no pueden enviar mensajes sin ser primero invitados a ello.
- En caso de un intercambio no válido, el maestro repite la petición. El maestro declara ausente al esclavo al que se hace la petición si éste no responde en una escala de tiempo dada.
- Si el esclavo no tiene en cuenta o no puede procesar la petición, envía una respuesta de excepción al maestro. En este caso, el maestro puede repetir o no la petición.

Existen dos tipos de datos posibles entre maestro y esclavos:

- El maestro envía una petición al número de esclavo específico y espera su respuesta.
- El maestro envía una petición al conjunto de esclavos sin esperar respuesta (principio de difusión general).

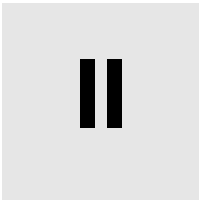
Comunicación en modalidad de caracteres

La modalidad de caracteres es una modalidad punto a punto de intercambio de datos entre dos entidades. A diferencia del protocolo Modbus, no establece ninguna comunicación de conexión serie estructurada jerárquicamente ni ofrece servicios mediante códigos de función.

La modalidad de caracteres es asíncrona. Cada elemento de información textual se envía o recibe carácter por carácter a intervalos de tiempo irregulares. El tiempo empleado en los intercambios puede determinarse a partir de las siguientes propiedades:

- Uno o dos caracteres de final de trama.
- Timeout.
- Número de caracteres.

Instalación de hardware para las comunicaciones en modalidad de caracteres y Modbus serie



En esta parte

En esta parte se ofrece una introducción a la instalación del hardware de comunicaciones en modalidad de caracteres y Modbus serie.

Contenido de esta parte

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
2	Introducción a la comunicación serie	19
3	Arquitecturas de comunicación serie	35

Introducción a la comunicación serie

2

Tema del capítulo

En este capítulo se presenta la comunicación serie en los procesadores BMX P34 1000/2000/2010/20102/2020 y en el módulo BMX NOM 0200.

En la siguiente tabla se proporciona una presentación rápida de las dos posibilidades para incrementar comunicaciones serie:

Con el puerto integrado de la CPU	Con el módulo de comunicación BMX NOM 0200
- Velocidad de transmisión limitada - Líneas serie no aisladas - Provisión de suministro eléctrico para el equipo terminal	- Mayor número de canales de comunicación disponibles - Gestión de señales RS232 específicas de modem - Mayor velocidad de transmisión - Dos líneas serie RS485 aisladas

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
2.1	Conexión serie en los procesadores BMX P34 1000/2000/2010/20102/2020	20
2.2	Módulo de 2 puertos RS-485/232 BMX NOM 0200	24

2.1 Conexión serie en los procesadores BMX P34 1000/2000/2010/20102/2020

Presentación de las comunicaciones serie en los procesadores BMX P34 1000/2000/2010/20102/2020

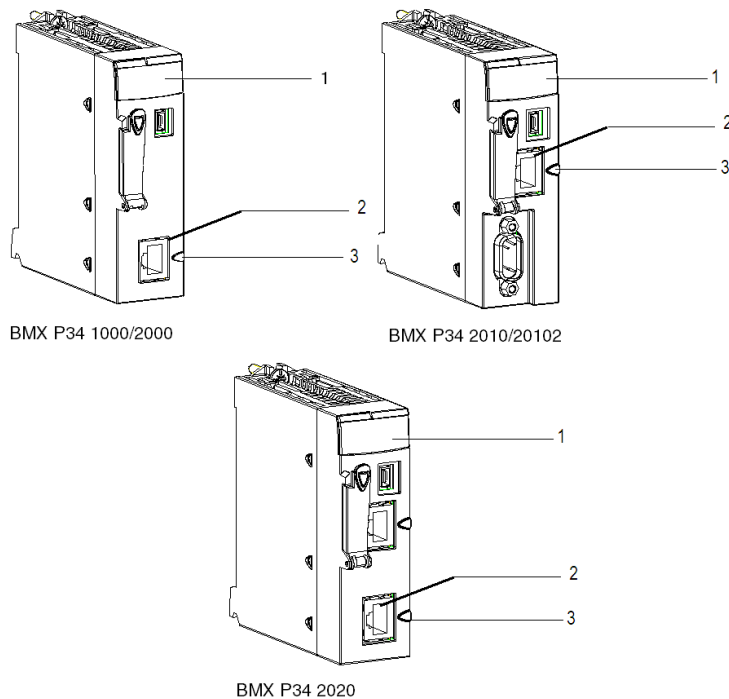
General

Los procesadores siguientes tienen un canal de comunicación integrado especializado en la comunicación serie y permiten la comunicación mediante una conexión serie:

- BMX P34 1000/2000/2020,
- BMX P34 2010/20102.

Introducción a los procesadores

En la siguiente ilustración se muestran las características físicas de los procesadores BMX P34 1000/2000/2010/20102/2020:



Estos procesadores se componen de los siguientes elementos:

Dirección	Descripción
1	LED de estado de los procesadores de la parte frontal
2	Canal integrado (canal 0) dedicado a la conexión serie.
3	Anillo de identificación del puerto serie (negro)

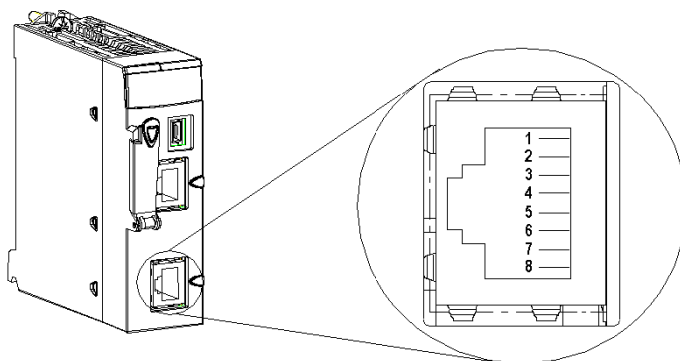
Diagnóstico visual de comunicación serie

El estado de la comunicación serie se indica mediante un indicador LED SER COM amarillo en la parte frontal de estos procesadores:

- LED intermitente: la comunicación serie está en curso.
- LED apagado: la comunicación serie no está en curso.

Introducción al puerto serie

En la siguiente ilustración se muestra el puerto serie RJ45:



El conector RJ45 tiene ocho pines. Los pines utilizados varían en función de la conexión física que se utilice.

Los pines utilizados por la conexión serie RS232 son los siguientes:

- Pin 1: señal RXD
- Pin 2: señal TXD
- Pin 3: señal RTS
- Pin 6: señal CTS
- Pin 8: puesta a tierra de la conexión serie potencial (0 V)

Los pines utilizados por la conexión serie RS485 son los siguientes:

- Pin 4: señal D1
- Pin 5: señal D0

El pin 7 se utiliza exclusivamente para suministrar alimentación a interfases hombre-máquina o a dispositivos de pequeñas dimensiones mediante el cable de la conexión serie:

- Pin 7: fuente de alimentación de la conexión serie: 5 VCC/190 mA

Características detalladas

Características de CC:

- Consumo máximo de alimentación estabilizada: 190 mA,
- Tensión mínima en el conector de la CPU para 190 mA: 4,9 V,
- Tensión máxima en el conector de la CPU para 190 mA: 5,25 V,
- Tensión máxima en el conector de la CPU sin carga: 5,5 V.

Características de CA:

- Carga del condensador: (a 5 V)
 - Máximo 1 μ F en condensador cerámico;
 - 10 μ F en tántalo
- Inicio de carga de la bomba: (a 5 V)
 - 4 x 1 μ F en condensador cerámico;
 - 2 x 10 μ F en tántalo.

NOTA: Los conectores RS232 de cuatro conductores, RS485 de dos conductores y RS485 de dos conductores con fuente de alimentación utilizan todos el mismo conector hembra RJ45. Sólo el cableado de la señal es distinto.

Características de líneas eléctricas

Las líneas RS232 y RS485 no están aisladas.

En caso de ausencia de tierra equipotencial entre los equipos conectados (cables con una longitud de 30 m como mínimo), debe utilizarse un módulo TWDXCAISO disyuntor en modalidad RS485.

La polarización de la línea RS485 está integrada en el PLC y el sistema la activa o desactiva automáticamente según la configuración elegida en la pantalla Unity Pro:

- Maestro Modbus: la polarización de líneas está activada.
- Esclavo de Modbus: la polarización de líneas está desactivada.
- Modalidad de caracteres: la polarización de líneas está desactivada.

La polarización no se ve afectada por una conmutación de protocolo dinámico. El valor de las resistencias de polarización es 560 ohmios.

En la modalidad RS232, no es necesaria la polarización.

No hay ninguna generación en la terminación de línea.

Especificaciones de los canales

El canal de estos procesadores incluye:

- Una interfase física RS485 no aislada.
- Una interfase física RS232 no aislada.
- Tipos de comunicaciones de Modbus serie (ASCII y RTU) y la modalidad de caracteres.

Las especificaciones de conexión para los dos protocolos son:

	Modbus serie/RS485	Modbus serie/RS232	Modalidad de caracteres/RS485	Modalidad de caracteres/RS232
tipo	Maestro/esclavo	Maestro/esclavo	Half Duplex	Full duplex
Flujo	19200 baudios Los parámetros pueden establecerse entre 300 baudios y 38400 baudios.	19200 baudios Los parámetros pueden establecerse entre 300 baudios y 38400 baudios.	9600 baudios Los parámetros pueden establecerse entre 300 baudios y 38400 baudios.	9600 baudios Los parámetros pueden establecerse entre 300 baudios y 38400 baudios.
Número de dispositivos	32	32	—	—
Direcciones de esclavo autorizadas	De 1 a 247	De 1 a 247	—	—
Longitud máxima del bus sin bifurcación	1000 m (15 m sin bifurcación)	15 m	1000 m (15 m con bifurcación)	15 m
Tamaño del mensaje	Modbus serie: • RTU: 256 bytes (252 bytes de datos) • ASCII: 513 bytes (2x252 bytes de datos)	Modbus serie: • RTU: 256 bytes (252 bytes de datos) • ASCII: 513 bytes (2x252 bytes de datos)	1024 bytes	1024 bytes
Utilidades	Palabras/bits de lectura. Palabras/bits de escritura. Diagnósticos.	Palabras/bits de lectura. Palabras/bits de escritura. Diagnósticos.	Enviar cadenas de caracteres. Recibir cadenas de caracteres.	Enviar cadenas de caracteres. Recibir cadenas de caracteres.

2.2 Módulo de 2 puertos RS-485/232 BMX NOM 0200

Objeto de esta sección

En esta sección se presenta la comunicación serie en el módulo BMX NOM 0200.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Presentación del módulo de 2 puertos BMX NOM 0200 RS-485/232	25
Equipo Modicon M340H (Endurecido)	30
Puesta a tierra de los módulos instalados	31
Instalación del módulo BMX NOM 0200	32
Consideraciones del cableado de BMX NOM 0200	34

Presentación del módulo de 2 puertos BMX NOM 0200 RS-485/232

General

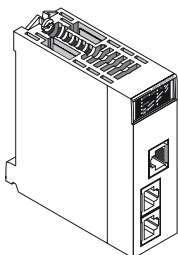
Los módulos de conexión serie BMX NOM 0200 y BMX NOM 0200H (véase página 30) son módulos en línea serie asíncrona bidireccional que permiten el uso de comunicaciones Modbus serie (maestro o esclavo) y de modalidad de caracteres.

El BMX NOM 0200 es un módulo especializado de formato simple que puede instalarse en un bastidor de la estación Modicon M340.

NOTA: En los extremos de temperatura (-25... 0 °C y 60... 70 °C) (-13...32 °F) y (140...158 °F), las características de funcionamiento del modelo BMX NOM 0200H son las mismas que las características del modelo BMX NOM 0200 dentro de su rango de temperatura (0...60 °C)(32...140 °F).

Introducción al módulo

En la siguiente ilustración se muestran las características físicas del módulo BMX NOM 0200:



Este módulo BMX NOM 0200 se compone de los elementos que se muestran en la tabla siguiente:

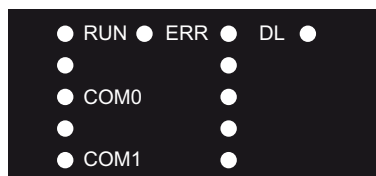
Tecla	Descripción
1	Cinco indicadores LED en la parte frontal del módulo: ● RUN y ERR muestran el estado del módulo, ● COM0 muestra el estado del tráfico en el puerto 0 o 1 (canal 0), ● COM1 muestra el estado del tráfico en el puerto 2 (canal 1), ● DL muestra el estado de descarga del firmware.
2	Canal integrado (canal 0) dedicado a la conexión serie con 2 puertos serie: RS232 (puerto 0) y RS485 (puerto 1). Nota: Sólo puede haber un puerto activo en cada momento.
3	Canal integrado (canal 1) dedicado a la conexión serie con 1 puerto serie: RS485 (puerto 2).

NOTA: En algunas modalidades de funcionamiento, los LED pueden indicar información más específica (véase página 26).

Diagnóstico visual

Hay 5 LED en el panel frontal del módulo BMX NOM 0200. Muestran información acerca del estado operativo del módulo y el estado de comunicación de la conexión serie incorporada.

Pantalla de indicadores luminosos:



- RUN = el módulo está activado y bien configurado.
- ERR = el módulo ha detectado un error y no puede funcionar correctamente.
- DL = el firmware se está descargando.
- COM0 = comunicación detectada en el puerto 0 ó 1 (canal 0).
- COM1 = comunicación detectada en el puerto 2 (canal 1).

Significado del LED:

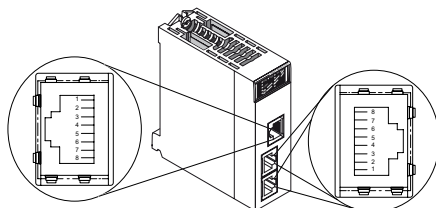
- Cada LED puede estar en uno de los estados siguientes:
 - 1 = Activo
 - 0 = Apagado
 - B = Intermitente
- Al iniciar el módulo, todos los LED se encienden y apagan, lo que permite verificar si funcionan correctamente.

RUN	ERR	COM0	COM1	DL	Diagnóstico
0	—	—	—	—	El módulo no está encendido o no está operativo.
0	B	—	—	—	El módulo no está configurado.
1	1	—	—	—	El módulo no funciona correctamente.
1	1	1	0	—	El módulo ha detectado un problema en el canal 0.
1	1	1	B	—	El módulo ha detectado un problema en el canal 0, el canal 1 está intercambiando datos.
1	1	0	1	—	El módulo ha detectado un problema en el canal 1.
1	1	B	1	—	El módulo ha detectado un problema en el canal 1, el canal 0 está intercambiando datos.

RUN	ERR	COM0	COM1	DL	Diagnóstico
1	0	B	—	—	El canal 0 está intercambiando datos.
1	0	—	B	—	El canal 1 está intercambiando datos.
B	B	—	—	0	Falta la CPU.
B	B	B	B	—	El módulo está ejecutando la autoverificación.
—	—	—	—	B	El firmware del módulo se está descargando.
—	—	—	—	1	El firmware se ha cargado; se debe restablecer el módulo.

Introducción a los puertos serie

En la siguiente ilustración se muestran los puertos serie RJ45 del módulo BMX NOM 0200:



En la siguiente tabla se muestra la asignación de pins de los puertos serie del módulo BMX NOM 0200:

Nº de pin	Canal 1/puerto 1 o 2 RS485	Canal 0/puerto 0 RS232
1	—	RXD (Recepción de datos)
2	—	TXD (Transmisión de datos)
3	—	RTS (Solicitud de envío)
4	D1 (B/B4)	DTR (Terminal de datos preparado)
5	D0 (A/A4)	DSR (Conjunto de datos preparado)
6	—	CTS (Listo para enviar)
7	—	DCD (Detección de la portadora de datos)
8	puesta a tierra de la conexión serie potencial (0 V)	puesta a tierra de la conexión serie potencial (0 V)

NOTA:

- Las líneas RS485 están aisladas. La tensión de aislamiento es de 500 V entre dos líneas serie y de hasta 500 V CA entre cada línea serie aislada y el bastidor.
- RS232 de siete conductores y RS485 de dos conductores utilizan el mismo colector hembra RJ45. Sólo el cableado de la señal es distinto.

Especificaciones de los canales

Entre los canales del módulo BMX NOM 0200 se incluyen:

- Dos interfases físicas RS485 aisladas.
- Una interfase física RS232 no aislada.
- Tipos de comunicaciones de Modbus serie (ASCII y RTU) y la modalidad de caracteres.

Las especificaciones de conexión para los dos protocolos son:

	Modbus serie/RS485	Modbus serie/RS232	Modalidad de caracteres/RS485	Modalidad de caracteres/RS232
tipo	Maestro/esclavo	Maestro/esclavo	Half Duplex	Full duplex
Flujo	19200 baudios Los parámetros pueden establecerse entre 300 baudios y 57600 baudios.	19200 baudios Los parámetros pueden establecerse entre 300 baudios y 115200 baudios.	9600 baudios Los parámetros pueden establecerse entre 300 baudios y 57600 baudios.	9600 baudios Los parámetros pueden establecerse entre 300 baudios y 115200 baudios.
Número de dispositivos	32	32	—	—
Direcciones de esclavo autorizadas	De 1 a 247	De 1 a 247	—	—
Longitud máxima del bus sin bifurcación	Consulte la tabla siguiente (15 m con bifurcación).	15 m	Consulte la tabla siguiente (15 m con bifurcación).	15 m
Tamaño del mensaje	Modbus serie: • RTU: 256 bytes (252 bytes de datos) • ASCII: 513 bytes (2x252 bytes de datos)	Modbus serie: • RTU: 256 bytes (252 bytes de datos) • ASCII: 513 bytes (2x252 bytes de datos)	1024 bytes	1024 bytes
Utilidades	Palabras/bits de lectura. Palabras/bits de escritura. Diagnósticos.	Palabras/bits de lectura. Palabras/bits de escritura. Diagnósticos.	Enviar cadenas de caracteres. Recibir cadenas de caracteres.	Enviar cadenas de caracteres. Recibir cadenas de caracteres.
Control de flujo de hardware	—	Opcionalmente mediante señales RTS/CTS.	—	Opcionalmente mediante señales RTS/CTS.

En la tabla siguiente se muestra la longitud máxima del cable RS485 que se puede utilizar según la velocidad de transmisión seleccionada:

Velocidad de transmisión seleccionada (bit/s)	Longitud (m)	Referencia del producto
300	1.000	(1)
600	1.000	(1)
1.200	1.000	(1)
2.400	1.000	(1)
9.600	1.000	(1)
19.200	600	(1)
38.400	300	(1) o (2)
57.600	200	(1) o (2)

- (1): Calibre AWG24 del cable blindado de par trenzado (TSX CSA 100, TSX CSA 200, TSX CSA 500)
- (2): Cable de categoría 5 o superior

Consumo del módulo BMX NOM 0200

En esta tabla se muestra el consumo del módulo BMX NOM 0200:

Fuente	Corriente típica	Corriente máxima	Potencia disipada típica	Potencia disipada máxima
24 V CC	80 mA	130 mA	1,92 W	3,12 W

Equipo Modicon M340H (Endurecido)

M340H

El equipo Modicon M340H (Endurecido) es una versión reforzada del equipo M340. Puede utilizarse con un mayor rango de temperatura (-25...70 °C) (-13...158 °F) y en entornos químicos severos.

El equipo M340H, cuando se encuentra dentro del rango de temperatura estándar (0...60 °C) (32...140 °F), tiene las mismas características que el equipo M340 estándar.

En los extremos de temperatura (-25... 0 °C y 60... 70 °C) (-13...32 °F) y (140...158 °F), las versiones endurecidas pueden tener valores nominales de potencia reducidos que repercuten en los cálculos de potencia para las aplicaciones Unity Pro.

Si este equipo funciona fuera del rango de temperatura de -25...70 °C (-13...158 °F), el equipo puede funcionar anormalmente.

ATENCIÓN

FUNCIONAMIENTO INESPERADO DEL EQUIPO

No utilice el equipo M340H fuera de su rango de temperatura.

Si no se siguen estas instrucciones pueden producirse lesiones personales o daños en el equipo.

El equipo endurecido posee un revestimiento homologado que se aplica a las tarjetas electrónicas. Esta protección, asociada a una instalación y un mantenimiento adecuados, permite que sea más sólido cuando funciona en entornos químicos severos.

Puesta a tierra de los módulos instalados

General

La conexión a masa de los módulos Modicon M340 resulta crucial para evitar las descargas eléctricas.

Conexión a masa de procesadores y fuentes de alimentación

⚠ PELIGRO

PELIGRO DE DESCARGA ELÉCTRICA, EXPLOSIÓN O DESTELLO DE ARCO VOLTAICO

Asegúrese de que los contactos de conexión a tierra están presentes y de que no están doblados ni deformados. Si lo están, no utilice el módulo y póngase en contacto con su representante de Schneider Electric.

Si no se siguen estas instrucciones provocará lesiones graves o incluso la muerte.

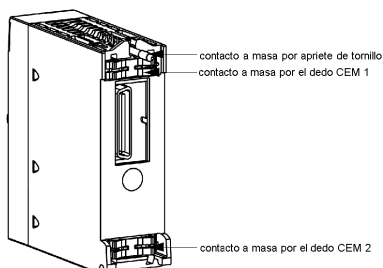
⚠ ADVERTENCIA

FUNCIONAMIENTO INESPERADO DEL EQUIPO

Apriete los tornillos de tensión de los módulos para garantizar las características del sistema. Una interrupción en el circuito podría ocasionar un comportamiento inesperado del sistema.

Si no se siguen estas instrucciones pueden producirse lesiones personales graves o mortales o daños en el equipo.

Todos los módulos M340 de Modicon están equipados con contactos de conexión a masa en su parte posterior con finalidades de toma a tierra:



Estos contactos conectan el bus de toma de tierra de los módulos al bus de toma de tierra del bastidor.

Instalación del módulo BMX NOM 0200

General

El módulo BMX NOM 0200 se instala en un bastidor de la estación Modicon M340 y no puede usar las slots necesarias para la fuente de alimentación y el procesador. La instalación debe adaptarse a las instrucciones instalación del bastidor.

El módulo BMX NOM 0200 requiere la instalación de una CPU con un sistema operativo de la versión 02.10 como mínimo. La instalación debe adaptarse a las instrucciones instalación de la CPU.

A continuación, puede conectarse un conector RJ45 al módulo en función de la red deseada.

NOTA: El módulo BMX NOM 0200 puede instalarse en un bastidor mientras la aplicación se ejecuta en el PLC.

ADVERTENCIA

FUNCIONAMIENTO INESPERADO DEL EQUIPO

La aplicación de este producto requiere experiencia en el diseño y la programación de sistemas de control. Sólo las personas que tengan experiencia están autorizadas a programar, instalar, modificar y aplicar este producto.

Siga todos los estándares y códigos de seguridad nacionales y locales.

Si no se siguen estas instrucciones pueden producirse lesiones personales graves o mortales o daños en el equipo.

Número de módulos

Puesto que el número de canales expertos gestionados por una estación PLC está relacionado con el procesador instalado, el número máximo de módulos BMX NOM 0200 en una estación sólo se basará en:

- El número de canales configurado en cada módulo BMX NOM 200 (cada canal cuenta como un canal experto),
- El tipo y la versión del procesador instalado (*véase Modicon M340 con Unity Pro, Procesadores, bastidores y módulos de fuente de alimentación, Manual de configuración*),
- El número de canales expertos ya utilizados.

Cuando se genera la aplicación, Unity Pro comprueba que no se superen las limitaciones.

Conexión/Desconexión

El módulo BMX NOM 0200 puede conectarse o desconectarse mientras está encendido. Cuando se desconecta el módulo del bastidor, se borra su memoria interna. El módulo pasa por una fase de inicialización cuando se vuelve a conectar.

De forma predeterminada, la configuración del módulo BMX NOM 0200 que es un esclavo Modbus con la dirección 248, 19200 bits/s, RTU, 8 bits, 1 parada, RS232 en el canal 0 y RS485 en el canal 1.

La dirección 248 es la dirección punto a punto a la que responde cualquier módulo esclavo BMX NOM 0200. El objetivo de esta función es encontrar cualquier módulo esclavo cuya dirección sea desconocida

Actualización de firmware

El firmware del módulo BMX NOM 0200 puede actualizarse mediante el bastidor del PLC. La actualización del firmware se define en el Manual del cargador de Unity Pro (*véase Unity Loader, un software de SoCollaborative, Manual del usuario*).

Consideraciones del cableado de BMX NOM 0200

Consideraciones operativas

ADVERTENCIA

FUNCIONAMIENTO INESPERADO DEL EQUIPO

Aunque es posible conectar o desconectar los cables del módulo BMX NOM 0200 y las CPU BMX P34 20x0 con la estación BMX XBP encendida, si lo hace puede que interrumpa la aplicación en curso.

Si no se siguen estas instrucciones pueden producirse lesiones personales graves o mortales o daños en el equipo.

Enlace

Las siguientes situaciones pueden crear una interrupción temporal en la aplicación o las comunicaciones:

- El conector RJ45 está conectado o desconectado cuando la alimentación está conectada.
- Los módulos se reinician cuando la alimentación se vuelve a conectar.

Arquitecturas de comunicación serie



Objetivo de este capítulo

En este capítulo se ofrece una introducción a las arquitecturas que utilizan la comunicación serie en los procesadores BMX P34 1000/2000/2010/20102/2020 y en el módulo BMX NOM 0200, así como los requisitos de cableado.

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
3.1	Arquitecturas de comunicación serie para procesadores BMX P34 1000/2000/2010/20102/2020	36
3.2	Arquitectura de comunicación serie para BMX NOM 0200	47
3.3	Cableado	56

3.1 Arquitecturas de comunicación serie para procesadores BMX P34 1000/2000/2010/20102/2020

Objeto de esta sección

En esta sección se ofrece una introducción a las arquitecturas que utilizan la comunicación serie en los procesadores BMX P34 1000/2000/2010/20102/2020, así como los requisitos de cableado.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Polarización y terminación de la línea Modbus (RS485)	37
Dispositivos de conexión Modbus (RS485)	39
Conexión de equipo terminal de datos (DTE) (RS232)	42
Conexión de equipo de terminación de circuito de datos (DCE) (RS232)	44

Polarización y terminación de la línea Modbus (RS485)

Descripción general

La red Modbus de múltiples puntos debe disponer de una polarización y terminación de línea.

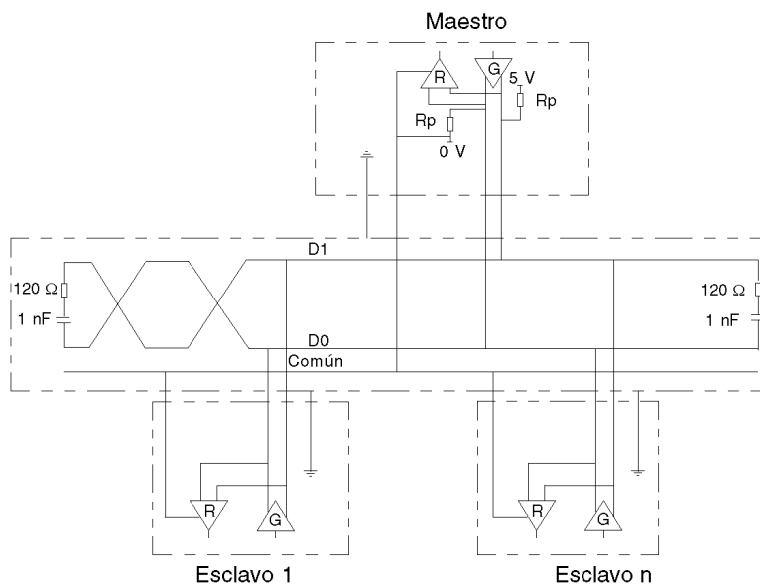
Los equipos que pueden conectarse a este bus son:

- Otros PLC como M340, Premium, Quantum, Twido o Nano
- Dispositivos de Schneider Automation como Altivar, el módulo de seguridad XPS, SEPAM, XBT o Momentum
- Otros dispositivos compatibles con el protocolo Modbus
- Modem, Concentrador

Se proporciona un ejemplo de **red Modbus multipunto** (véase página 41) incluido un procesador BMX P34 2010 en este manual.

NOTA: También puede ejecutarse una red Modbus punto a punto.

Esquema eléctrico de terminación y polarización de línea:



Terminación de línea

La terminación de línea se realiza externamente: consta de dos resistencias de 120 Ω y un condensador de 1 nF, colocados en cada extremo de la red (VW3 A8 306 RC o VW3 A8 306 DRC).

No coloque la terminación de línea en el extremo de un cable de derivación.

Polarización de línea

En una línea Modbus, es necesaria la polarización para una red RS485.

- Si la CPU M340 se utiliza como maestro, **el sistema la controla automáticamente** (véase página 22), por lo que no es necesaria la polarización externa.
- Si se utiliza la CPU M340 como un esclavo, debe implementarse la polarización mediante dos resistencias de 450 a 650 Ω (Rp) conectadas en el par equilibrado RS485:
 - una resistencia de arranque a una tensión de 5 V en el circuito D1,
 - una resistencia de tracción al circuito común en el circuito D0.

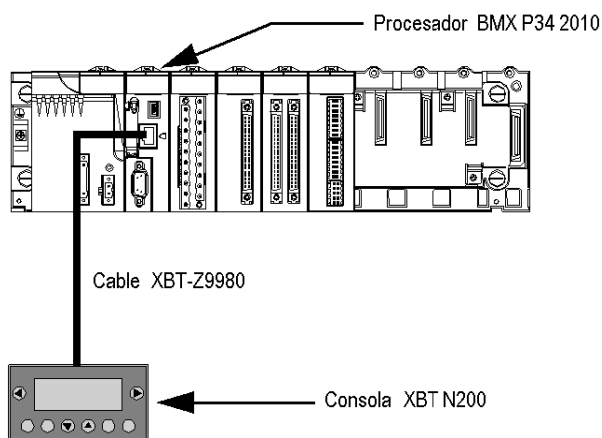
Dispositivos de conexión Modbus (RS485)

General

En las páginas siguientes se presentan dos ejemplos de conexión de dispositivos Modbus y una arquitectura de conexión serie Modbus.

Conexión de dispositivos Modbus realizadas mediante la conexión serie

En la siguiente ilustración se muestra cómo se conecta un procesador BMX P34 2010 a una consola XBT N200 alimentada mediante la conexión serie Modbus:



Los dispositivos se configuran de la manera siguiente:

- El procesador BMX P34 2010 se configura como esclavo.
- La interfase hombre-máquina XBT N200 se configura como maestro.

El cable XBT-Z9980 tiene las siguientes propiedades:

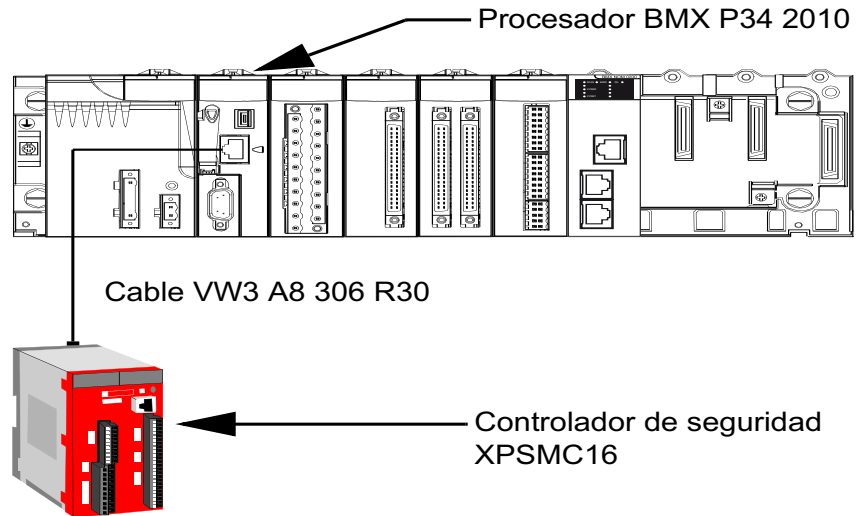
- Conexión: dos conectores macho RJ45.
- Cableado: 2 conductores para la línea física RS485 y 2 para la fuente de alimentación de la conexión serie.

Conexión de dispositivos Modbus realizadas sin la conexión serie

Esta arquitectura se compone de los siguientes elementos:

- Un procesador BMX P34 2010
- Un controlador de seguridad XPSMC16

En la siguiente ilustración se muestra cómo se conecta un procesador BMX P34 2010 a un controlador de seguridad XPSMC16:



Los dispositivos se configuran de la manera siguiente:

- El procesador BMX P34 2010 se configura como maestro.
- El controlador de seguridad XPSMC16 se configura como esclavo.

El cable VW3 A8 306 R30 tiene las siguientes propiedades:

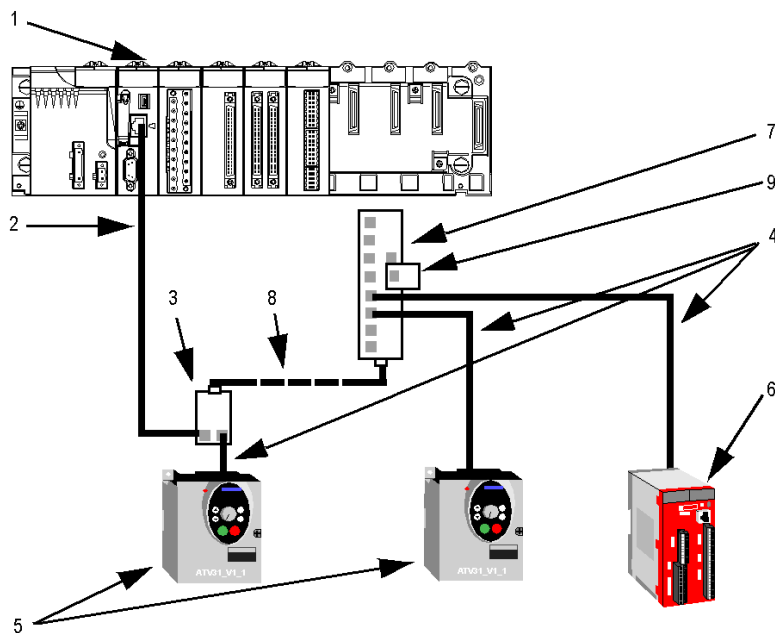
- Conexión: dos conectores macho RJ45.
- Cableado: dos conductores para la línea física RS485.

Arquitectura de conexión serie Modbus

La arquitectura de conexión serie Modbus se compone de los siguientes elementos:

- Un procesador BMX P34 2010/20102 configurado como maestro
- Un controlador de seguridad XPSMC16, configurado como esclavo
- Un bloque distribuidor TWDXCAISO con separación de potencial
- Un bloque distribuidor LU9 GC3
- Dos unidades ATV31, configuradas como esclavos

En el siguiente gráfico se representa la arquitectura de conexión serie descrita anteriormente:



- 1 Procesador BMX P34 2010
- 2 Cable XBT-Z9980
- 3 Bloque distribuidor TWDXCAISO con separación de potencial
- 4 Cable VW3 A8 306 R3
- 5 Unidad ATV31
- 6 Controlador de seguridad XPSMC16
- 7 Bloque distribuidor LU9 GC3
- 8 Cable TSXCSAx00
- 9 Cable VW3 A8 306 R

Conexión de equipo terminal de datos (DTE) (RS232)

General

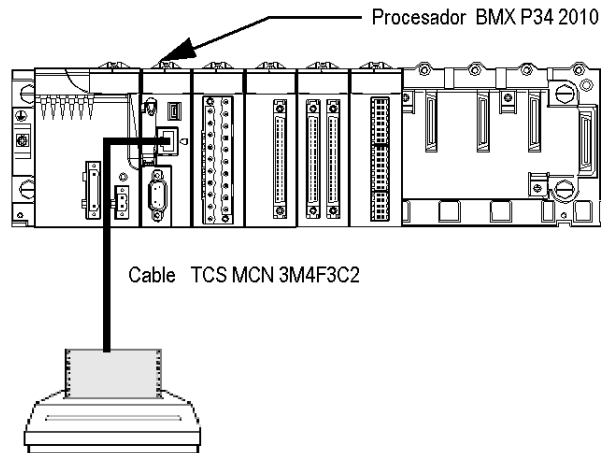
Equipo terminal de datos es el término utilizado para describir dispositivos como:

- periféricos comunes (impresora, teclado–pantalla, terminal de taller, etc.),
- periféricos especializados (lectores de código de barras, etc.) y
- PC.

Todo los equipos terminales de datos se conectan a un procesador BMX P34 1000/2000/2010/20102/2020 mediante un cable cruzado serie con la conexión física RS232.

Conexión de equipo terminal de datos

En la siguiente ilustración se muestra cómo se conecta una impresora a un procesador BMX P34 2010:



El protocolo de comunicación que se utiliza es modalidad de caracteres.

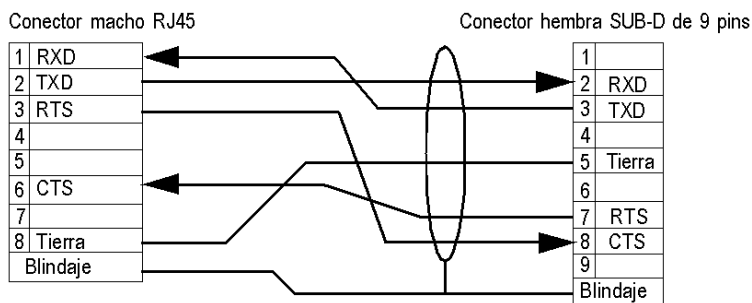
NOTA: Sólo puede conectarse un elemento de equipo terminal de datos a cada procesador BMX P34 1000/2000/2010/20102/2020.

Cable cruzado serie RS 232

El cable cruzado serie TCS MCN 3M4F3C2 tiene dos conectores:

- RJ45 macho
- hembra SUB-D de nueve pins.

En la siguiente ilustración se muestra la asignación de los pins en un cable cruzado serie TCS MCN 3M4F3C2:



Cables y accesorios de conexión

En la tabla siguiente se muestran las referencias comerciales de los cables y adaptadores que van a emplearse en función del conector serie que utilice el equipo terminal de datos:

Conector serie de equipo terminal de datos	Cableado
Conector macho SUB-D de nueve pins	Cable TCS MCN 3M4F3C2
Conector macho SUB-D de 25 pins	● Cable TCS MCN 3M4F3C2 ● Adaptador TSX CTC 07
Conector hembra SUB-D de 25 pins	● Cable TCS MCN 3M4F3C2 ● Adaptador TSX CTC 10

Conexión de equipo de terminación de circuito de datos (DCE) (RS232)

General

Equipo de terminación de circuito de datos (DCE) es el término utilizado para describir dispositivos como los módems.

Para un dispositivo de tipo DCE, los pins RTS y CTS están conectados directamente (sin cruzar).

Todo los equipos de terminación de circuito de datos se conectan a un procesador BMX P34 1000/2000/2010/20102/2020 mediante un cable directo serie con una conexión física RS232.

NOTA: Las diferencias entre las conexiones DCE y DTE estriban principalmente en los enchufes en la dirección de la señal de los pins (entrada o salida). Por ejemplo, un PC de escritorio se considera un dispositivo DTE mientras que un módem se considera un dispositivo DCE.

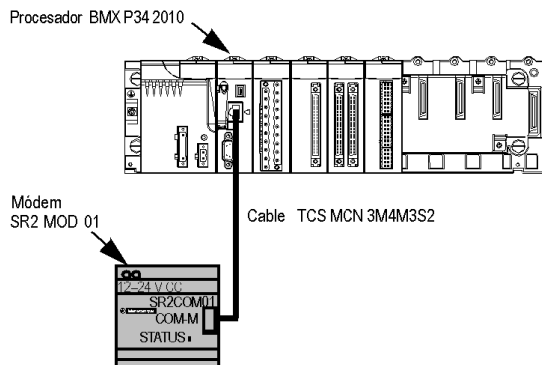
Características del módem

Las CPU de M340 funcionan con la mayoría de los módems del mercado. Para conectar un módem al puerto serie de un procesador BMX P34 1000/2000/2010/20102/2020, el módem debe tener las siguientes características:

- Admisión de 10 u 11 bits por carácter si el puerto de terminal se utiliza en el Modbus serie:
 - 7 u 8 bits de datos,
 - 1 o 2 bits de parada,
 - impar, par o sin paridad.
- Funcionamiento sin comprobación de portadora de datos.

Conexión de equipo de terminación de circuito de datos

En la siguiente ilustración se muestra cómo se conecta un módem a un procesador BMX P34 2010:



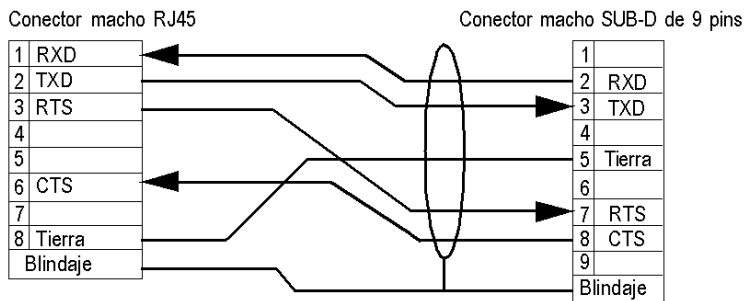
NOTA: En el Modbus serie, el tiempo de espera debe estar comprendido entre 100 y 250 ms.

Cable directo serie RS 232

El cable directo serie TCS MCN 3M4M3S2 tiene dos conectores:

- RJ45 macho
- Macho SUB-D de nueve pins

En la siguiente ilustración se muestran las ubicaciones de los pins en un cable directo serie TCS MCN 3M4M3S2:



Cables y accesorios de conexión

En la tabla siguiente se muestran las referencias comerciales de los cables y adaptadores que van a emplearse en función del conector serie que utilice el equipo de terminación de circuito de datos:

Conector serie de equipo de terminación de circuito de datos	Cableado
Conector hembra SUB-D de nueve pins	Cable TCS MCN 3M4M3S2
Conector hembra SUB-D de 25 pins	● Cable TCS MCN 3M4M3S2 ● Adaptador TSX CTC 09

3.2 Arquitectura de comunicación serie para BMX NOM 0200

Objeto de esta sección

En esta sección se ofrece una introducción a las arquitecturas que utilizan la comunicación serie en el módulo BMX NOM 0200, así como los requisitos de cableado.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Polarización y terminación de la línea Modbus (RS485)	48
Dispositivos de conexión Modbus (RS485)	50
Conexión de equipo terminal de datos (DTE) (RS232)	52
Conexión de equipo de terminación de circuito de datos (DCE) (RS232)	54

Polarización y terminación de la línea Modbus (RS485)

Descripción general

La red Modbus de múltiples puntos debe disponer de una polarización y terminación de línea.

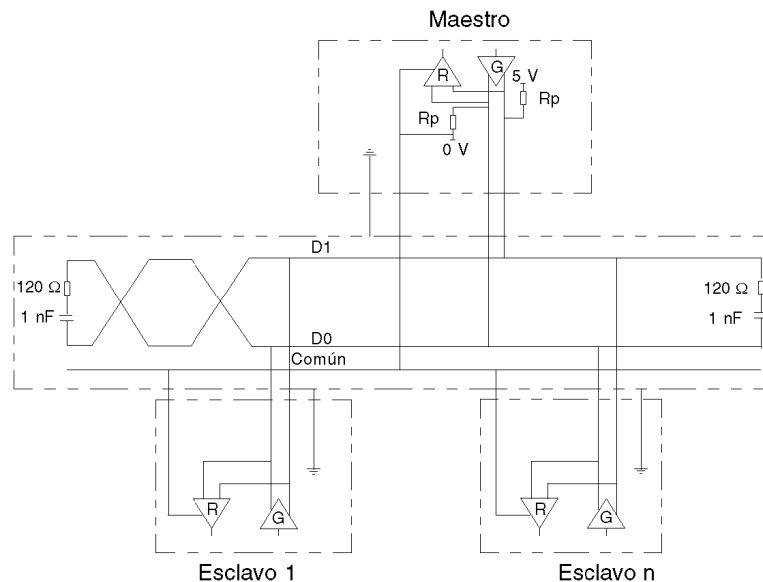
Los equipos que pueden conectarse a este bus son:

- Otros PLC como M340, Premium, Quantum, Twido o Nano
- Dispositivos de Schneider Automation como Altivar, el módulo de seguridad XPS, SEPAM, XBT o Momentum
- Otros dispositivos compatibles con el protocolo Modbus
- Modem, Concentrador

Se proporciona un ejemplo de **red Modbus multipunto** (véase página 51) incluido un módulo BMX NOM 0200 en este manual.

NOTA: También puede ejecutarse una red Modbus punto a punto.

Esquema eléctrico de terminación y polarización de línea:



Terminación de línea

La terminación de línea se realiza externamente: consta de dos resistencias de $120\ \Omega$ y un condensador de $1\ \text{nF}$, colocados en cada extremo de la red (VW3 A8 306RC o VW3 A8 306 DRC). No coloque la terminación de línea en el extremo de un cable de derivación.

Polarización de línea

En una línea Modbus, es necesaria la polarización para una red RS485.

- Si el módulo BMX NOM 0200 se utiliza como maestro, el sistema lo controla automáticamente, por lo que no es necesaria la polarización externa.
- Si se utiliza el módulo BMX NOM 0200 como un esclavo, debe implementarse la polarización mediante dos resistencias de 450 a $650\ \Omega$ (R_p) conectadas en el par equilibrado RS485:
 - una resistencia de arranque a una tensión de $5\ \text{V}$ en el circuito D1,
 - una resistencia de tracción al circuito común en el circuito D0.

NOTA:

En la modalidad de caracteres, la polarización de la línea se puede configurar en Unity Pro. Es posible elegir entre:

- polarización con baja impedancia como en las redes Modbus (el objetivo de este tipo de polarización es permitir que el maestro mantenga el estado predeterminado),
- polarización con alta impedancia (el objetivo de este tipo de polarización es permitir que cada dispositivo contribuya a mantener el estado predeterminado),
- sin polarización (si se utiliza una polarización externa).

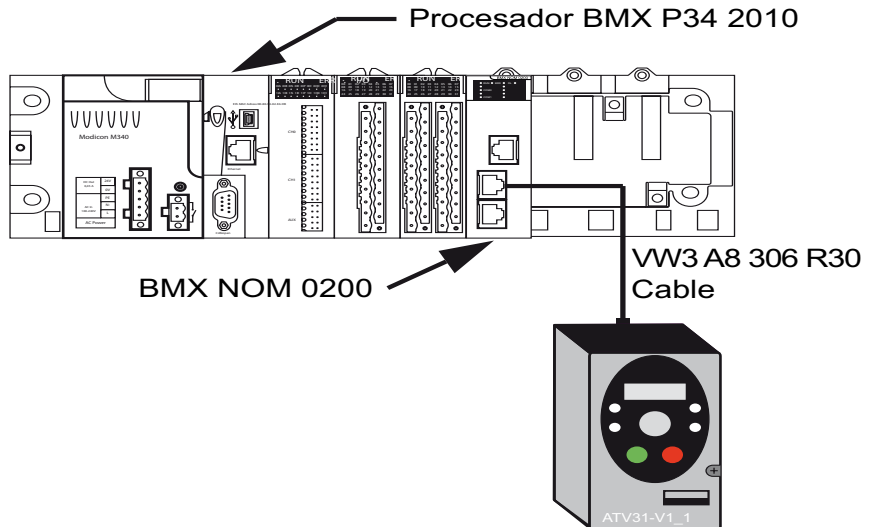
Dispositivos de conexión Modbus (RS485)

General

En las páginas siguientes se presenta un ejemplo de conexión de dispositivos Modbus y una arquitectura de conexión serie Modbus.

Conexión de dispositivos Modbus realizadas sin la conexión serie

En la siguiente figura se muestra cómo se conecta un procesador BMX NOM 0200 a una unidad ATV31:



Los dispositivos se configuran de la manera siguiente:

- Un procesador BMX P34 2010
- Un módulo BMX NOM 0200 configurado como maestro
- Una unidad ATV31, configurada como esclavo

El cable VW3 A8 306 R30 tiene las siguientes propiedades:

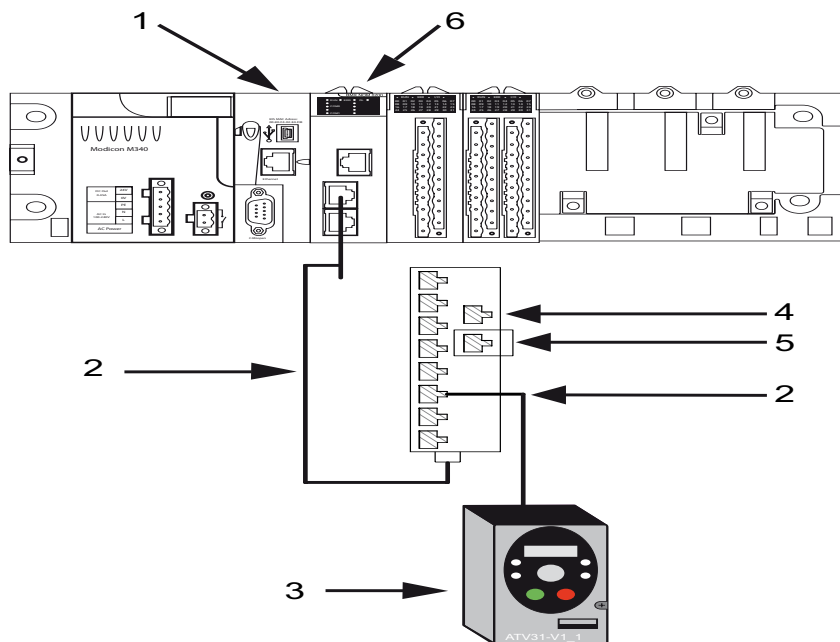
- Conexión: dos conectores macho RJ45.
- Cableado: dos conductores para la línea física RS485.

Arquitectura de conexión serie Modbus

La arquitectura de conexión serie Modbus se compone de los siguientes elementos:

- Un procesador BMX P34 2010
- Un módulo BMX NOM 0200 configurado como maestro
- Un bloque distribuidor TWDXCAISO con separación de potencial
- Un bloque distribuidor LU9 GC3
- Dos unidades ATV31, configuradas como esclavos.

En la siguiente ilustración se representa la arquitectura de conexión serie descrita anteriormente:



- 1 Procesador BMX P34 2010
- 2 Cable VW3 A8 306 R30
- 3 Unidad ATV31
- 4 Bloque distribuidor LU9 GC3
- 5 Cable VW3 A8 306 R
- 6 Módulo BMX NOM 0200

Conexión de equipo terminal de datos (DTE) (RS232)

General

Equipo terminal de datos es el término utilizado para describir dispositivos como:

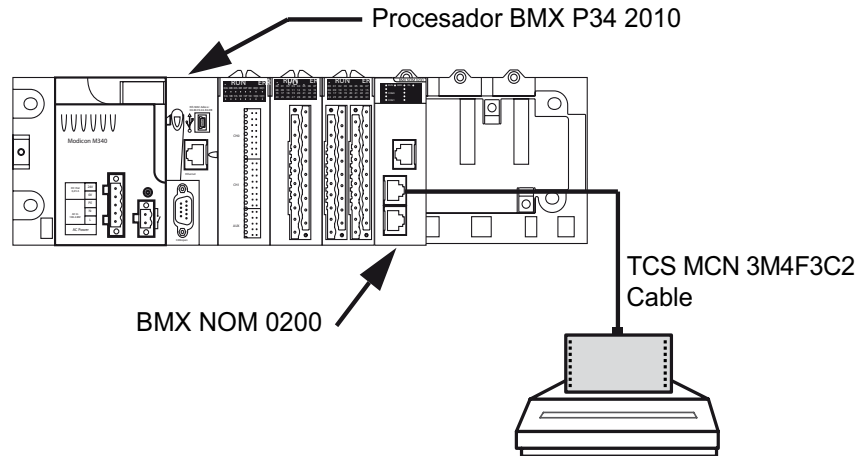
- periféricos comunes (impresora, teclado–pantalla, terminal de taller, etc.),
- periféricos especializados (lectores de código de barras, etc.),
- PC.

Para un dispositivo de tipo DTE, los pins RTS y CTS están cruzados.

Todo los equipos terminales se conectan a un módulo BMX NOM 0200 mediante un cable cruzado serie con la conexión física RS232.

Conexión de equipo terminal de datos

En la siguiente figura se muestra cómo se conecta una impresora a un módulo BMX NOM 0200:



El protocolo de comunicación que se utiliza es modalidad de caracteres.

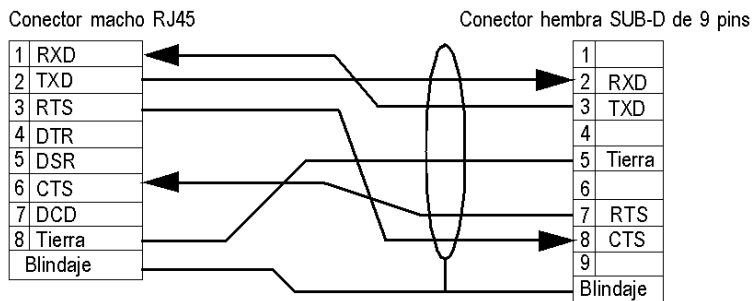
NOTA: Sólo puede conectarse un elemento de equipo terminal de datos al módulo BMX NOM 0200.

Cable cruzado serie RS 232

El cable cruzado serie TCS MCN 3M4F3C2 tiene dos conectores:

- RJ45 macho
- hembra SUB-D de nueve pins.

En la siguiente figura se muestra la asignación de los pins en un cable cruzado serie TCS MCN 3M4F3C2:



Cables y accesorios de conexión

En la tabla siguiente se muestran las referencias comerciales de los cables y adaptadores que van a emplearse en función del conector serie que utilice el equipo terminal de datos:

Conector serie de equipo terminal de datos	Cableado
Conector macho SUB-D de 9 pins	Cable TCS MCN 3M4F3C2
Conector macho SUB-D de 25 pins	● Cable TCS MCN 3M4F3C2 ● Adaptador TSX CTC 07
Conector hembra SUB-D de 25 pins	● Cable TCS MCN 3M4F3C2 ● Adaptador TSX CTC 10

Conexión de equipo de terminación de circuito de datos (DCE) (RS232)

General

Equipo de terminación de circuito de datos (DCE) es el término utilizado para describir dispositivos como los módems.

Para un dispositivo de tipo DCE, los pins RTS y CTS están conectados directamente (sin cruzar).

Todo los equipos de terminación de circuito de datos se conectan a un módulo BMX NOM 0200 mediante un cable directo serie con una conexión física RS232.

NOTA: Las diferencias entre las conexiones DCE y DTE estriban principalmente en los enchufes en la dirección de la señal de los pins (entrada o salida). Por ejemplo, un PC de escritorio se considera un dispositivo DTE mientras que un módem se considera un dispositivo DCE.

Características del módem

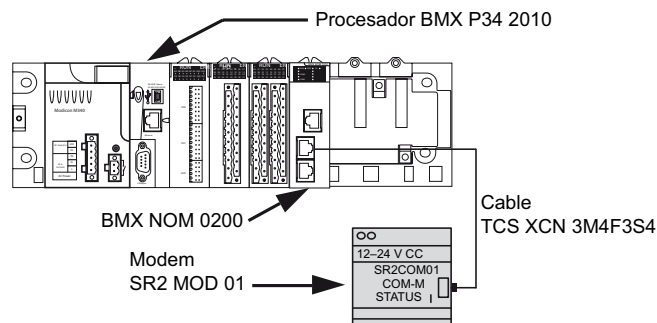
El módulo BMX NOM 0200 funciona con la mayoría de los módems del mercado. Para conectar un módem al puerto serie de un módulo BMX NOM 0200, el módem debe tener las siguientes características:

- Admisión de 10 u 11 bits por carácter si el puerto de terminal se utiliza en el Modbus serie:
 - 7 u 8 bits de datos,
 - 1 o 2 bits de parada,
 - impar, par o sin paridad.
- Funcionamiento sin comprobación de portadora de datos.

Las señales CTS, DTR, DSR y DCD pueden gestionarse mediante la aplicación.

Conexión de equipo de terminación de circuito de datos

En la siguiente figura se muestra cómo se conecta un módem a un módulo BMX NOM 0200:



La conexión del módem necesita un cable de módem específico.

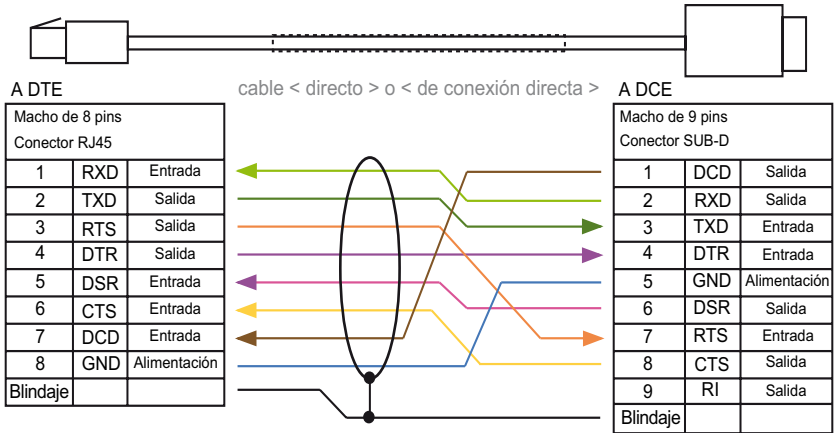
Cable directo serie RS 232

Ejemplo del cable TCS XCN 3M4F3S4:

El cable directo serie TCS XCN 3M4F3S4 tiene una versión con 8 conductores y 2 conectores:

- RJ45 macho
- Macho SUB-D de 9 pins.

En la siguiente ilustración se muestran las ubicaciones de los pins en un cable directo serie TCS XCN 3M4F3S4:



Cables y accesorios de conexión

En la tabla siguiente se muestran las referencias comerciales de los cables y adaptadores que van a emplearse en función del conector serie que utilice el equipo de terminación de circuito de datos:

Conector serie de equipo de terminación de circuito de datos	Cableado
Conector hembra SUB-D de 9 pins	● Cable TCS MCN 3M4M3S2 ● Cable TCS XCN 3M4F3S4
Conector hembra SUB-D de 25 pins	● Cable TCS MCN 3M4M3S2 ● Adaptador TSX CTC 09

3.3 Cableado

Cableado

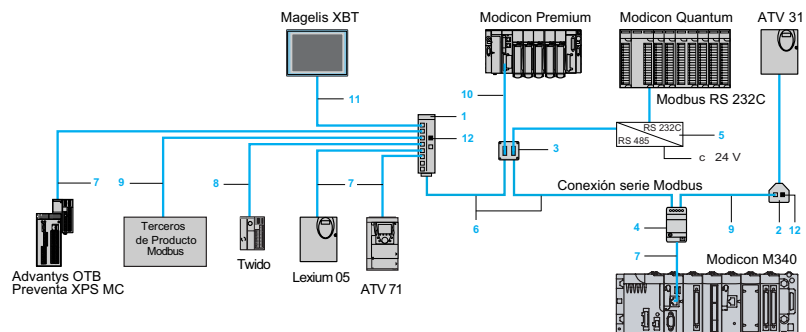
General

Se necesitan varios cables y accesorios para configurar una conexión serie en los siguientes procesadores y módulo:

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020 y el
- módulo BMX NOM 0200.

Sistema de cableado

En la figura siguiente se muestra un ejemplo de la conexión de Modbus serie de Modicon M340 y el sistema de cableado de la modalidad de caracteres. Los **cables** (véase página 57) y los **accesorios de conexión** (véase página 58) a los que se hace referencia en la figura se describen en las siguientes tablas:



Cables

En la tabla siguiente se muestran los cables disponibles que son compatibles con la comunicación serie en estos procesadores y módulo:

Referencia en la figura	Designación	Longitud	Características	Referencia del producto
6	Cable principal de par trenzado con blindaje doble RS485	100 m	Dos extremos sin revestimiento	TSX CSA 100
6	Cable principal de par trenzado con blindaje doble RS485	200 m	Dos extremos sin revestimiento	TSX CSA 200
6	Cable principal de par trenzado con blindaje doble RS485	500 m	Dos extremos sin revestimiento	TSX CSA 500
7	Cable Modbus RS485	0,3 m	Dos conectores macho RJ45	VW3 A8 306 R03
7	Cable Modbus RS485	1 m	Dos conectores macho RJ45	VW3 A8 306 R10
7	Cable Modbus RS485	3 m	Dos conectores macho RJ45	VW3 A8 306 R30
-	Cable Modbus RS485	3 m	● Un conector macho RJ45 ● Un conector macho SUB-D de 15 pins	VW3 A8 306
4	Cable Modbus RS485	0,3 m	● Un conector macho RJ45 ● Un conector mini-DIN	TWD XCA RJ003
4	Cable Modbus RS485	1 m	● Un conector macho RJ45 ● Un conector mini-DIN	TWD XCA RJ010
4	Cable Modbus RS485	3 m	● Un conector macho RJ45 ● Un conector mini-DIN	TWD XCA RJ030
5	Cable Modbus RS485	3 m	● Un conector macho RJ45 ● Un extremo sin revestimiento	VW3 A8 306 D30
9	Cable Modbus RS485	3 m	● Un conector en miniatura ● Un conector SUB-D de 15 pins	TSX SCP CM 4630
11	Cable RS485 para la pantalla y el terminal Magelis XBT	2,5 m	● Un conector macho RJ45 ● Un conector hembra SUB-D de 25 pins Nota: este cable no es compatible con el módulo BMX NOM 0200	XBT-Z938
-	Cable RS485 para dispositivos alimentados mediante la conexión serie	3 m	Dos conectores macho RJ45 Nota: este cable no es compatible con el módulo BMX NOM 0200.	XBT-Z9980

Referencia en la figura	Designación	Longitud	Características	Referencia del producto
-	Cable RS232 de cuatro conductores para equipo terminal de datos (DTE)	3 m	● Un conector macho RJ45 ● Un conector hembra SUB-D de nueve pins	TCS MCN 3M4F3C2
-	Cable RS232 de cuatro conductores para equipo de terminación de circuito de datos (DCE)	3 m	● Un conector macho RJ45 ● Un conector macho SUB-D de nueve pins	TCS MCN 3M4M3S2
-	Cable RS232 de siete conductores para equipo de terminación de circuito de datos (DCE)	3 m	● Un conector macho RJ45 ● Un conector macho SUB-D de 9 pins	TCS XCN 3M4F3S4

Accesorios de conexión

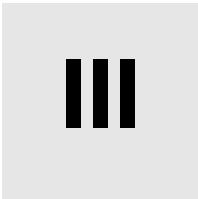
En la tabla siguiente se muestran los accesorios de conexión disponibles que son compatibles con la comunicación serie en estos procesadores y módulo:

Referencia en la figura	Designación	Características	Referencia del producto
1	Cuadro distribuidor Modbus	● Diez conectores RJ45 ● Un bloque de terminales de tornillos	LU9 GC3
2	Caja de conexiones en T	● Dos conectores RJ45 ● Cable integrado de 0,3 m con conector RJ45 en un extremo	VW3 A8 306 TF03
2	Caja de conexiones en T	● Dos conectores RJ45 ● Cable integrado de 1 m con conector RJ45 en un extremo	VW3 A8 306 TF10
-	Caja de conexiones en T pasivas	● Tres bloques de terminales de tornillos ● Adaptador de extremo de línea RC	TSX SCA 50
3	Conector suscriptor de 2 canales pasivo	● Dos conectores hembra SUB-D 15 pins ● Dos bloques de terminales de tornillos ● Adaptador de extremo de línea RC	TSX SCA 62

Referencia en la figura	Designación	Características	Referencia del producto
4	Caja de conexiones en T RS485 aisladas	● Un conector RJ45 ● Un bloque de terminales de tornillos	TWD XCA ISO
-	Caja de conexiones en T	Tres conectores RJ45	TWD XCA T3RJ
-	Adaptador Modbus/Bluetooth	● Un adaptador Bluetooth con un conector RJ45 ● Un juego de cables para PowerSuite con dos conectores RJ45 ● Un juego de cables para TwidoSuite con un conector RJ45 y un conector mini-DIN ● Un adaptador macho RJ45/SUB-D de 9 pins para unidades de velocidad ATV	TWD XCA T3RJ
5	Adaptador de línea RS232C/RS485 sin señales de módem	19,2 kbit/s	XGS Z24
12	Terminador de línea para conector RJ45	● Resistencia de 120 Ω ● Capacidad de 1 nF	VW3 A8 306 RC
-	Terminador de línea para bloque de terminales de tornillos	● Resistencia de 120 Ω ● Capacidad de 1 nF	VW3 A8 306 DRC
-	Adaptador para dispositivos no estándar	● Dos conectores macho SUB-D de 25 pins	XBT ZG999
-	Adaptador para dispositivos no estándar	● Un conector macho SUB-D de 25 pins ● Un conector macho SUB-D de nueve pins	XBT ZG909
-	Adaptador para equipo terminal de datos	● Un conector macho SUB-D de nueve pins ● Un conector hembra SUB-D de 25 pins	TSX CTC 07
-	Adaptador para equipo terminal de datos	● Un conector macho SUB-D de nueve pins ● Un conector macho SUB-D de 25 pins	TSX CTC 10
-	Adaptador para equipo de terminación de circuito de datos (DCE)	● Un conector hembra SUB-D de nueve pins ● Un conector macho SUB-D de 25 pins	TSX CTC 09

NOTA: Esta lista de cables y accesorios no es exhaustiva.

Implementación de software para las comunicaciones en modalidad de caracteres y Modbus serie



En esta parte

En esta parte se ofrece una introducción a la instalación del software de comunicaciones en modalidad de caracteres y Modbus serie mediante el software Unity Pro.

Contenido de esta parte

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
4	Metodología de instalación	63
5	Comunicaciones serie en los procesadores BMX P34 1000/2000/2010/20102/2020	67
6	Comunicaciones serie en los procesadores BMX P34 1000/2000/2010/20102/2020	101
7	Comunicación en modalidad Modbus serie para BMX NOM 0200	125
8	Comunicación en modalidad de caracteres para BMX NOM 0200	163
9	Diagnóstico del módulo BMX NOM 0200	185
10	Objetos de lenguaje de la comunicación en modalidad de caracteres y Modbus	191
11	Conmutación dinámica de protocolos	225

Introducción a la fase de instalación

Introducción

La instalación del software de los módulos específicos de la aplicación se lleva a cabo desde distintos editores de Unity Pro:

- en modalidad offline
- en modalidad online

Si no dispone de un procesador al que pueda conectarse, Unity Pro permite realizar una comprobación inicial mediante el simulador. En este caso, la instalación es diferente.

Fases de instalación al utilizar un procesador

En la siguiente tabla, se muestran las diferentes fases de instalación con el procesador:

Fase	Descripción	Modalidad
Configuración del procesador	Declaración del procesador	Offline
	Configuración del puerto serie del procesador	
Configuración del módulo (si corresponde)	Declaración del módulo	Offline
	Configuración de los canales de los módulos	
	Introducción de los parámetros de configuración	
Declaración de variables	Declaración de variables de tipo IODDT específicas del procesador/módulo y las variables del proyecto.	Offline (1)
Asociación	Asociación de variables IODDT a los canales configurados (editor de variables)	Offline (1)
Programación	Programación del proyecto	Offline (1)

Fase	Descripción	Modalidad
Generación	Generación del proyecto (análisis y edición de conexiones).	Offline
Transferir	Transfiriendo un proyecto al PLC	Online
Depuración	Depuración del proyecto desde pantallas de depuración y tablas de animación.	Online
Documentación	Creación del archivo de documentación e impresión de documentación diversa relacionada con el proyecto.	Online
Funcionamiento	Visualización de la información diversa necesaria para la supervisión del proyecto.	Online
Leyenda: (1) Estas fases también pueden llevarse a cabo online.		

Fases de instalación al utilizar un simulador

En la siguiente tabla se muestran las diferentes fases de instalación con el simulador:

Fase	Descripción	Modalidad
Configuración del procesador	Declaración del procesador	Offline
	Configuración del puerto serie del procesador	
Configuración del módulo (si corresponde)	Declaración del módulo	Offline
	Configuración de los canales de los módulos	
	Introducción de los parámetros de configuración	
Declaración de variables	Declaración de variables de tipo IODDT específicas del procesador/módulo y las variables del proyecto.	Offline (1)
Asociación	Asociación de variables IODDT a los canales configurados (editor de variables)	Offline (1)
Programación	Programación del proyecto	Offline (1)
Generación	Generación del proyecto (análisis y edición de conexiones).	Offline
Transferir	Transferencia del proyecto al simulador	Online
Simulación	Simulación del programa sin entradas/salidas.	Online
Ajuste/depuración	Depuración del proyecto desde las tablas de animación.	Online
	Modificación del programa y los parámetros de ajuste.	
Leyenda:		
(1) Estas fases también pueden llevarse a cabo online.		

Configuración del procesador y el módulo

Sólo puede accederse a los parámetros de configuración desde el software Unity Pro.

Creación de documentación técnica

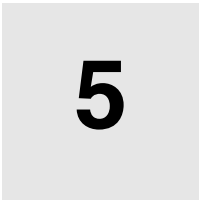
Unity Pro permite crear la **documentación técnica de un proyecto** (véase *Unity Pro, Modalidades de funcionamiento*,).

El formato general de la impresión se compone de:

- Un título: El número de pieza del módulo y su posición,
- Una sección con la identificación del módulo,
- Una sección por canal para todos los parámetros de un canal.

La impresión es coherente con la configuración: No se imprime la información atenuada no significativa.

Comunicaciones serie en los procesadores BMX P34 1000/2000/2010/20102/2 020



Objetivo de este capítulo

En este capítulo se presenta el proceso de implementación del software de comunicación Modbus serie en los procesadores BMX P34 1000/2000/2010/20102/2020

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
5.1	Generalidades	68
5.2	Configuración de la comunicación Modbus serie	76
5.3	Programación de la comunicación Modbus serie	88
5.4	Depuración de una comunicación Modbus serie	99

5.1 Generalidades

Objeto de esta sección

En esta sección se presentan las generalidades relativas a la comunicación Modbus serie y sus servicios.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Acerca de Modbus serie	69
Rendimiento	70
Acceso a los parámetros de la conexión serie	72

Acerca de Modbus serie

Introducción

La comunicación mediante Modbus permite intercambios de datos entre todos los dispositivos conectados al bus. Modbus es un protocolo que crea una estructura jerárquica (un maestro y varios esclavos).

El maestro gestiona todos los intercambios de dos maneras:

- El maestro realiza el intercambio con el esclavo y espera una respuesta.
- El maestro realiza el intercambio con el conjunto de esclavos sin esperar respuesta (difusión general).

NOTA: Procure que dos maestros (en el mismo bus) no envíen dos peticiones simultáneamente. De lo contrario, ambas se perderán y cada informe tendrá un mal resultado, que podría ser 16#0100 (la petición no se ha podido procesar) o 16#ODFF (falta el esclavo).

ADVERTENCIA

PÉRDIDA DE DATOS CRÍTICOS

Los puertos de comunicación deben utilizarse exclusivamente para transferencias de datos que no sean fundamentales.

Si no se siguen estas instrucciones pueden producirse lesiones personales graves o mortales o daños en el equipo.

Rendimiento

Presentación

Pueden utilizarse las siguientes tablas para evaluar los tiempos de intercambio de comunicación Modbus habituales en función de distintos criterios.

Los resultados mostrados corresponden al periodo de funcionamiento medio de la función `READ_VAR` en milisegundos.

Definición de tiempo de intercambio

El tiempo de intercambio es el tiempo que transcurre entre la creación de un intercambio y el final de éste. Incluye el tiempo de comunicación de conexión serie.

El intercambio se crea al efectuar la llamada a la función de comunicación.

El intercambio finaliza cuando se produce alguno de los siguientes eventos:

- Se reciben datos.
- Se produce una anomalía.
- Se agota el timeout.

Tiempo de intercambio de una palabra

En la siguiente tabla se muestran los tiempos de intercambio de una palabra de comunicación Modbus en un procesador BMX P34 2020:

Velocidad de transmisión de la comunicación en bits por segundo	Duración del ciclo en ms	Tiempos de intercambio en ms El esclavo Modbus es un BMX P34 1000 cíclico
4.800	Cíclico	68
4.800	10	72
4.800	50	100
9.600	Cíclico	35
9.600	10	40
9.600	50	50
19.200	Cíclico	20
19.200	10	27
19.200	50	50
38.400	Cíclico	13
38.400	10	20
38.400	50	50

Los tiempos de intercambio son similares en los procesadores BMX P34 2020 y BMX P34 2000/2010/20102; para el modelo BMX P34 1000, en cambio, el tiempo de intercambio es un 10% inferior.

Tiempo de intercambio de 100 palabras

En la siguiente tabla se muestran los tiempos de intercambio de 100 palabras de comunicación Modbus en un procesador BMX P34 2020:

Velocidad de transmisión de la comunicación en bits por segundo	Duración del ciclo en ms	Tiempos de intercambio en ms El esclavo Modbus es un BMX P34 1000 cíclico
4.800	Cíclico	500
4.800	10	540
4.800	50	595
9.600	Cíclico	280
9.600	10	288
9.600	50	300
19200	Cíclico	142
19.200	10	149
19.200	50	150
38.400	Cíclico	76
38.400	10	80
38.400	50	100

Los tiempos de intercambio son similares en los procesadores BMX P34 2020 y BMX P34 2000/2010/20102; para el modelo BMX P34 1000, en cambio, el tiempo de intercambio es un 10% inferior.

Precisión de la medida

Todos los tiempos de intercambio enumerados anteriormente provienen de medidas con un margen de precisión de +/-10 ms.

Acceso a los parámetros de la conexión serie

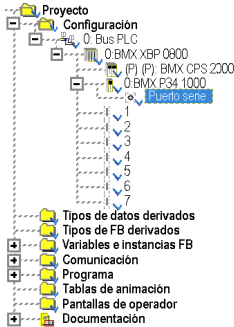
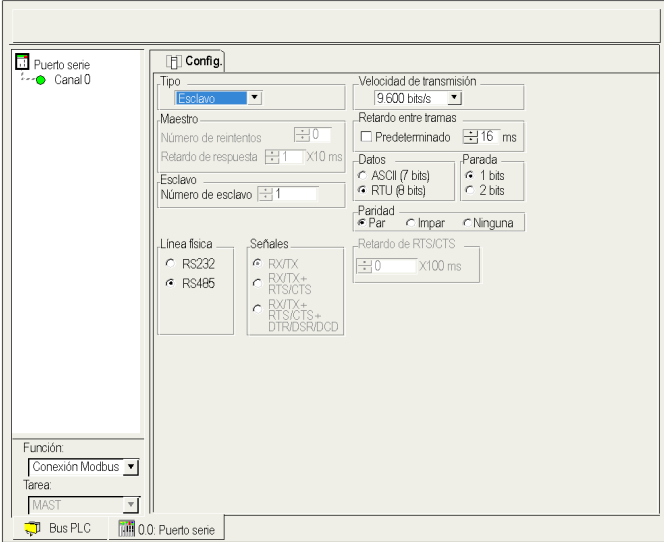
Presentación

En las siguientes páginas se describe cómo acceder a la pantalla de configuración de puertos serie para los procesadores siguientes, así como los elementos generales de las pantallas de configuración y depuración de la conexión Modbus y de modalidad de caracteres.

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020

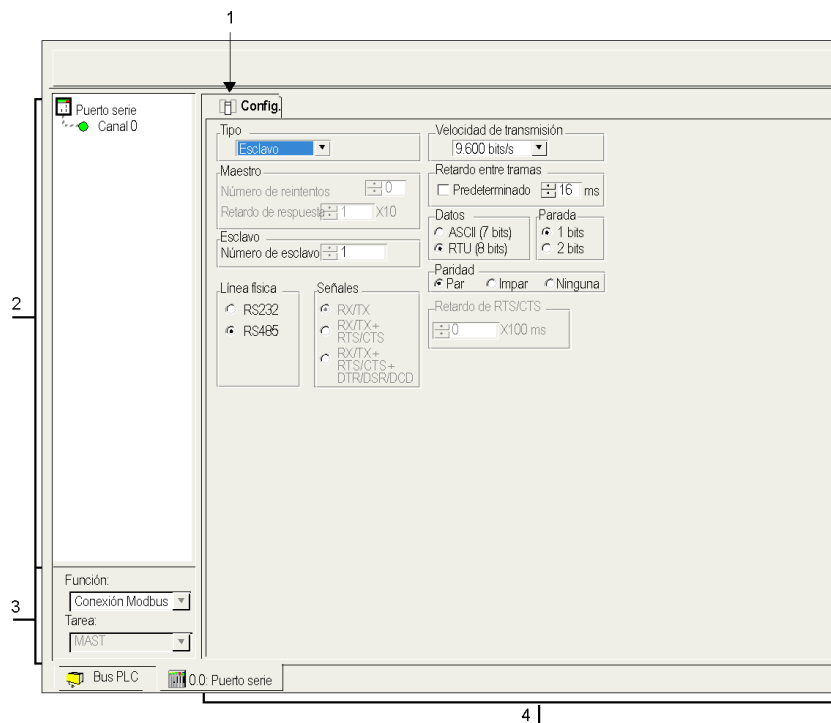
Acceso a la conexión serie

En la tabla siguiente se describe el procedimiento para acceder a la conexión serie:

Paso	Acción
1	En el explorador de proyectos, abra el siguiente directorio: <i>Station\Configuration\0: PLC bus\0: rack reference\0: processor reference\SerialPort</i>. Aparece la siguiente pantalla: 
2	Haga doble clic en el subdirectorio del puerto serie. Aparece la siguiente pantalla: 

Descripción de las pantallas Configuración y Depuración:

En la siguiente figura se muestra una pantalla de configuración de la comunicación Modbus:



Descripción

En la siguiente tabla se muestran los distintos elementos de las pantallas de configuración y depuración:

Dirección	Elemento	Función
1	Fichas	La ficha en primer plano indica la modalidad actual. Puede seleccionarse cada una de las modalidades mediante la ficha correspondiente. Las modalidades disponibles son: ● Configuración ● A la pantalla de depuración sólo puede accederse en la modalidad online.
2	Área de canal	Permite: ● Elegir entre el puerto serie y el canal 0 haciendo clic en uno de los dos. ● Mostrar las siguientes fichas al hacer clic en el puerto serie: ● "Descripción", que proporciona las características del dispositivo. ● «Objetos de E/S» (véase <i>Unity Pro, Modalidades de funcionamiento</i>), que se utiliza para presimbolizar los objetos de entrada/salida. ● Mostrar las siguientes fichas al hacer clic en el canal: ● Configuración ● Depuración ● Mostrar el nombre del canal y el símbolo definidos por el usuario mediante el editor de variables.
3	Área de parámetros generales	Permite elegir los parámetros generales asociados al canal: ● Función: las funciones disponibles son Modbus y Modalidad de caracteres. La configuración predeterminada corresponde a la función Modbus. ● Tarea: define la tarea maestra en donde se intercambiarán los objetos de intercambio implícitos del canal. Esta área aparece atenuada y, por lo tanto, no puede configurarse.
4	Área Configuración o Depuración	En la modalidad de configuración, esta área se utiliza para configurar los parámetros del canal. En la modalidad de depuración, se utiliza para depurar el canal de comunicación.

5.2 Configuración de la comunicación Modbus serie

Objeto de esta sección

En esta sección se describe el proceso de configuración del software para la comunicación Modbus serie.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Pantalla de configuración de la comunicación Modbus serie	77
Funciones Modbus accesibles	79
Valores predeterminados para los parámetros de comunicación Modbus serie	80
Parámetros Modbus relacionados con la aplicación	81
Parámetros Modbus relacionados con la transmisión	83
Parámetros de línea física y señales en Modbus	86

Pantalla de configuración de la comunicación Modbus serie

General

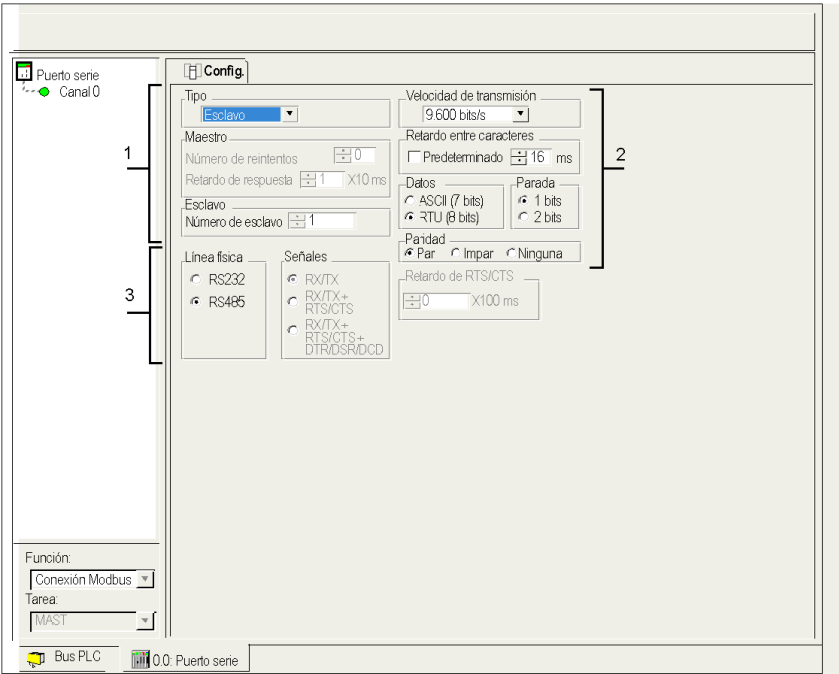
En las páginas siguientes se ofrece una introducción a la pantalla de configuración de la comunicación en para Modbus serie.

Acceso a la pantalla de configuración

Para acceder a la pantalla de configuración de la comunicación Modbus serie, abra el directorio Puerto serie del explorador de proyectos (véase *página 72*).

Ilustración

En la siguiente figura se muestra la pantalla de configuración predeterminada de la comunicación Modbus serie:



Descripción

Esta área se utiliza para configurar parámetros de canal. En la modalidad online, no se puede acceder a esta área, que estará atenuada. En la modalidad offline, puede accederse a esta área, pero es posible que no pueda accederse a algunos parámetros que estarán atenuados.

En la siguiente tabla se muestran las distintas áreas de las pantallas de configuración de la comunicación de Modbus:

Tecla	Elemento	Comentario
1	Parámetros de la aplicación (véase página 81)	Puede accederse a estos parámetros mediante tres áreas: ● Tipo ● Maestro ● Esclavo
2	Parámetros de transmisión (véase página 83)	Puede accederse a estos parámetros mediante cinco áreas: ● Velocidad de transmisión. ● Retardo entre tramas. ● Datos. ● Bits de parada. ● Paridad.
3	Parámetros de línea física y señales (véase página 86)	Puede accederse a estos parámetros mediante tres áreas: ● Línea física. ● Señales. ● Retardo de RTS/CTS.

NOTA: Al configurar la comunicación para Modbus serie el modo maestro, el área Esclavo está atenuada y no puede modificarse, y viceversa.

NOTA: En este ejemplo, las áreas "Señales" y "Retraso RTS/CTS" están atenuadas porque se ha elegido una línea física RS485.

Funciones Modbus accesibles

Presentación

La accesibilidad a funciones para la configuración de la conexión serie de los procesadores siguientes mediante Modbus serie varía en función de la conexión física que se utilice:

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020

Funciones accesibles

En la siguiente tabla se muestran las distintas funciones configurables en función del tipo de conexión serie utilizado:

Función	Conexión RS 485	Conexión RS 232
Número de reintentos del maestro	X	X
Tiempo de respuesta del maestro	X	X
Número de esclavo	X	X
Velocidad de transmisión	X	X
Retardo entre tramas	X	X
Datos	● ASCII (7 bits) ● RTU (8 bits)	● ASCII (7 bits) ● RTU (8 bits)
Parada	● 1 bit ● 2 bits	● 1 bit ● 2 bits
Paridad	● Impar ● Par ● Ninguna	● Impar ● Par ● Ninguna
Señales RX/TX	X	X
Señales RTS/CTS	-	X
Retardo de RTS/CTS	-	X

- X** Función accesible
- Función inaccesible

Valores predeterminados para los parámetros de comunicación Modbus serie

Presentación

Todos los parámetros de comunicación Modbus tienen valores predeterminados serie.

Valores predeterminados

En la siguiente tabla se muestran los valores predeterminados de los parámetros de comunicación Modbus serie:

Parámetro de configuración	Valor
Modalidad	Esclavo
Línea física	RS485
Número de esclavo	1
Retardo entre tramas	2 ms
Velocidad de transmisión	19200 bits/s
Paridad	Par
Bits de datos	RTU (8 bits)
Bits de parada	1 bit

Parámetros Modbus relacionados con la aplicación

Presentación

Una vez configurado el canal de comunicación, deben introducirse los parámetros de la aplicación.

Puede accederse a estos parámetros desde tres áreas de configuración:

- Área Tipo
- Área Maestro
- Área Esclavo

Área Tipo

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:

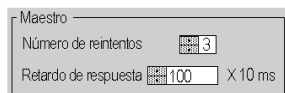


Esta área permite seleccionar el tipo de protocolo Modbus serie que va a utilizarse:

- **Maestro:** Cuando la estación afectada es el maestro.
- **Esclavo:** Cuando la estación afectada es un esclavo.

Área Maestro

Sólo puede accederse al área de configuración que se muestra a continuación cuando se selecciona «Maestro» en el área «Tipo»:



Esta área permite introducir los siguientes parámetros:

- **Número de reintentos:** número de intentos de conexión que lleva a cabo el maestro antes de declarar ausente al esclavo.
El valor predeterminado es 3.
Los valores posibles están comprendidos entre 0 y 15.
El valor 0 indica que no hay reintentos por parte del maestro.
- **Retardo de respuesta:** es el tiempo transcurrido entre la petición inicial enviada por el maestro y un intento repetido de ésta en caso de que el esclavo no responda. Se corresponde con el tiempo máximo entre el envío del último carácter de la petición enviada por el maestro y la recepción del primer carácter de la petición reenviada por el esclavo.
El valor predeterminado es 1 segundo (100*10 ms).
Los valores posibles están comprendidos entre 10 ms y 10 s.

NOTA: El retardo de respuesta del maestro debe ser por lo menos igual que el retardo de respuesta más largo de los esclavos presentes en el bus.

Área Esclavo

Sólo puede accederse al área de configuración que se muestra a continuación cuando se selecciona «Esclavo» en el área «Tipo»:



The image shows a software interface for configuring a slave device. It features a tab labeled 'Esclavo' and a text input field labeled 'Número de esclavo'. The input field contains the number '7' and has a small icon to its left, possibly representing a numeric keypad or a selection tool.

Esta área permite introducir el número de esclavo del procesador.

El valor predeterminado es 1.

Los valores posibles están comprendidos entre 1 y 247.

NOTA: En una configuración de esclavo Modbus, puede utilizarse una dirección adicional, número 248, para una comunicación serie punto a punto.

Parámetros Modbus relacionados con la transmisión

Presentación

Una vez configurado el canal de comunicación, deben introducirse los parámetros reservados de transmisión.

Puede accederse a estos parámetros desde cinco áreas:

- Área Velocidad de transmisión
- Área Retardo entre caracteres
- Área Datos
- Área Parada
- Área Paridad

Área Velocidad de transmisión

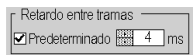
Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



Puede utilizarla para seleccionar la velocidad de transmisión del Modbus serie. La velocidad seleccionada tiene que ser coherente con los demás dispositivos. Los valores configurables son 300, 600, 1.200, 2.400, 4.800, 9.600, 19.200 y 38.400 bits por segundo.

Área Retardo entre tramas

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



Retardo entre tramas es el tiempo mínimo que separa dos tramas en la recepción. Este retardo se gestiona cuando el PLC (maestro o esclavo) recibe mensajes.

NOTA: El valor predeterminado varía en función de la velocidad de transmisión seleccionada.

NOTA: El retardo entre tramas deberá ser el valor predeterminado para ser compatible con Modbus. En caso de que un esclavo no sea conforme, el valor puede cambiarse y deberá ser idéntico para el maestro y para todos los esclavos del bus.

Área Datos

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:

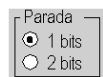


Esta zona permite introducir el tipo de codificación utilizado para comunicarse mediante el Modbus serie. Este campo se establece en función de los demás dispositivos conectados al bus. Existen dos modalidades configurables:

- Modalidad RTU:
 - Los caracteres se codificarán en 8 bits.
 - El final de la trama se detecta cuando existe un silencio de 3,5 caracteres como mínimo.
 - La integridad de la trama se comprueba mediante una palabra conocida, como la suma de control CRC, que se incluye en la trama.
- Modalidad ASCII:
 - Los caracteres se codificarán en 7 bits.
 - El inicio de la trama se detecta cuando se recibe el carácter ":".
 - El final de la trama se detecta mediante un retorno de carro y un avance de línea.
 - La integridad de la trama se comprueba mediante un byte denominado suma de control LRC, que se incluye en la trama.

Área Parada

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:

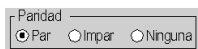


El área Parada permite introducir el número de bits de parada utilizados en la comunicación. Este campo se establece en función de los demás dispositivos. Los valores configurables son:

- 1 bit
- 2 bits

Área Paridad

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



The image shows a configuration window titled 'Paridad'. Inside, there are three radio buttons: 'Par' (which is selected), 'Impar', and 'Ninguna'.

Esta área permite determinar si se agrega un bit de paridad, así como su tipo. Este campo se establece en función de los demás dispositivos. Los valores configurables son:

- Par
- Impar
- Ninguna

Parámetros de línea física y señales en Modbus

Presentación

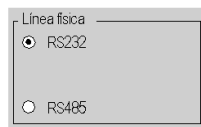
Una vez configurado el canal de comunicación, deben introducirse los parámetros de línea física y señales.

Puede accederse a estos parámetros mediante tres áreas:

- Área Línea física
- Área Señales
- Área Retardo de RTS/CTS

Área Línea física

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:

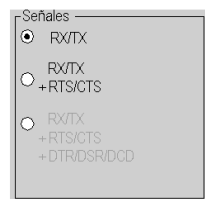


En esta área, puede elegir entre dos tipos de línea física para el puerto serie en los procesadores BMX P34 1000/2000/2010/20102/2020:

- Línea RS 232
- Línea RS 485

Área Señales

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



En esta área, puede seleccionar las señales admitidas por la línea física RS 232:

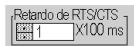
- RX/TX
- RX/TX + RTS/CTS

Si se configura RS 485, se atenúa el área completa y el valor predeterminado será RX/TX.

NOTA: Sólo hay señales RX/TX y RX/TX + RTS/CTS disponibles cuando se configura el puerto serie para los procesadores BMX P34 1000/2000/2010/20102/2020.

Área Retardo RTS/CTS

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



La zona de retardo RTS/CTS sólo está disponible cuando las casillas RS232 y RX/TX+RTS/CTS están seleccionadas. Un algoritmo de control del flujo RTS/CTS se selecciona el valor predeterminado es 0 ms. Un valor distinto de 0 permite un algoritmo de control del módem RTS/CTS.

El algoritmo de control del flujo RTS/CTS (DTE <-> DTE) es distinto del algoritmo de control del módem RTS/CTS (DTE <-> DCE) como se indica a continuación:

- El algoritmo de control del flujo RTS/CTS está relacionado con el búfer de recepción de desbordamiento (full duplex).
- El algoritmo de control del módem RTS/CTS se ocupa de los procesos de transmisión compartidos; por ejemplo. un módem de radio.

Algoritmo del control de flujo RTS/CTS

El objetivo es evitar un desbordamiento del búfer de recepción.

La señal de salida RTS de cada dispositivo se conecta a la señal de entrada CTS de otro dispositivo. El transmisor (M340) está autorizado a transmitir los datos al recibir la señal de entrada RTS (por ejemplo, otro M340) en su entrada CTS. Este algoritmo es simétrico y permite la comunicación asíncrona dúplex completo.

Algoritmo del control del módem RTS/CTS

Antes de trasmitir una solicitud, el remitente (M340) activa la señal RTS y espera a que el módem active la señal CTS. Si CTS no está activada después del retardo RTS/CTS, la solicitud se descarta.

5.3 Programación de la comunicación Modbus serie

Objeto de esta sección

En esta sección se describe el proceso de programación que se utiliza al implementar la comunicación Modbus serie.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Servicios admitidos por un procesador maestro de conexión Modbus	89
Servicios compatibles con un procesador esclavo de conexión Modbus	97

Servicios admitidos por un procesador maestro de conexión Modbus

Presentación

Cuando se utilizan como procesador maestro en una conexión Modbus, los procesadores siguientes permiten varios servicios mediante las funciones de comunicación `READ_VAR`, `WRITE_VAR` y `DATA_EXCH`.

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020

Intercambios de datos

La lectura o escritura de variables se lleva a cabo dirigiendo las siguientes peticiones al dispositivo esclavo de destino.

Estas solicitudes utilizan las funciones de comunicación `READ_VAR` y `WRITE_VAR`.

Petición Modbus	Código de función	Función de comunicación
Lectura de bits	16#01 o 16#02	<code>READ_VAR</code>
Lectura de palabras	16#03 o 16#04	<code>READ_VAR</code>
Escritura de bits	16#0F	<code>WRITE_VAR</code>
Escritura de palabras	16#10	<code>WRITE_VAR</code>

NOTA: Las utilidades de escritura se pueden enviar en modalidad Broadcast. En este caso, no se envía respuesta al emisor. A diferencia de Premium, después de enviar una solicitud de broadcast el M340 restablece el bit de actividad y se devuelve el código 16#01 (Parada del intercambio en timeout) en la segunda palabra de gestión EF.

NOTA: Los objetos que lee el PLC M340 pueden ser de tipo `%I` y `%IW`. En este caso, la función `READ_VAR` genera una solicitud Modbus: FC 0x2 o 0x4. En un PLC Quantum, permite acceder al estado de entrada o a los registros de estado de entrada.

Más generalmente, es posible enviar cualquier solicitud de Modbus a un dispositivo esclavo mediante la función de comunicación `DATA_EXCH`.

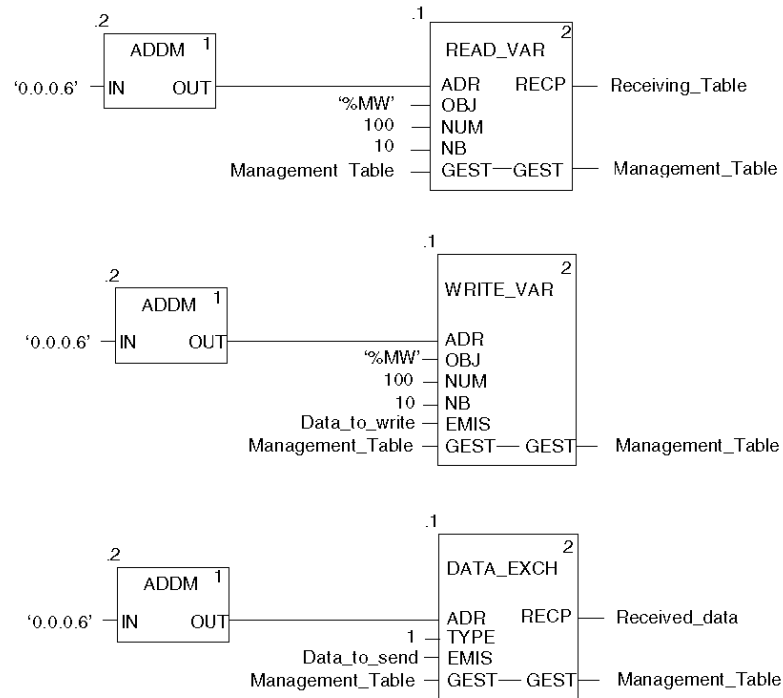
Función de comunicación `READ_VAR`, `WRITE_VAR` y `DATA_EXCH`

Se definen tres funciones específicas de comunicación para enviar y recibir datos mediante un canal de comunicación Modbus:

- `READ_VAR`: para lectura de variables.
- `WRITE_VAR`: para escritura de variables.
- `DATA_EXCH`: Para enviar solicitudes de Modbus a otro dispositivo sobre el protocolo seleccionado

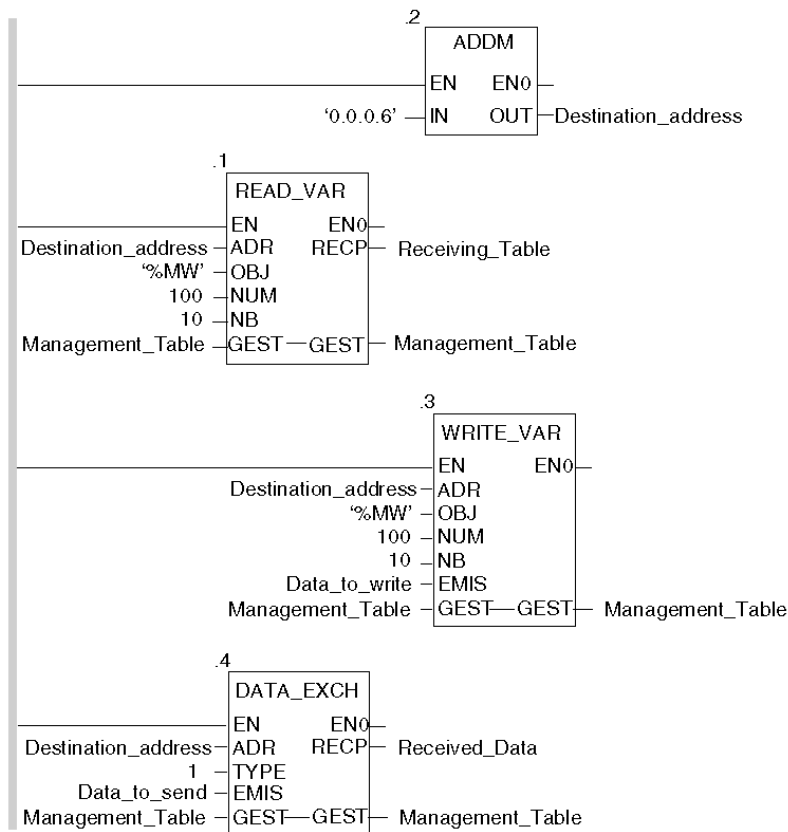
Ejemplo de programación en FBD

En el diagrama siguiente se representa un ejemplo de programación de las funciones de comunicación READ_VAR, WRITE_VAR y DATA_EXCH en lenguaje FBD:



Ejemplo de programación en Ladder

En el diagrama siguiente se representa un ejemplo de programación de las funciones de comunicación READ_VAR, WRITE_VAR y DATA_EXCH en lenguaje Ladder:



Ejemplo de programación en ST

Las siguientes líneas de código representan un ejemplo de programación de las funciones de comunicación READ_VAR, WRITE_VAR y DATA_EXCH en lenguaje ST:

```
READ_VAR(ADDM('0.0.0.6'), 'MW', 100, 10, Management_Table,  
Receiving_Table);  
  
WRITE_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, Data_to_write,  
Management_Table);  
  
DATA_EXCH(ADDM('0.0.0.6'), 1, Data_to_send, Management_Table,  
Received_data);
```

Cancelación de un intercambio

Un intercambio ejecutado por las funciones `READ_VAR`, `WRITE_VAR` y `DATA_EXCH` se puede cancelar con una de las dos formas de programación; a continuación, se presentan ambas en lenguaje ST:

- Utilización de la función `CANCEL`:

```
IF (%MW40.0) THEN
    %MW200:=SHR(%MW40,8);
    CANCEL(%MW200,%MW185);
END_IF;
```

`%MW40` corresponde al parámetro `GEST` (tabla de gestión). `%MW40.0` corresponde al bit de actividad de la función `READ_VAR` y se establece en 1 cuando la función de comunicación está activa. Si el bit se establece en 1, el programa lleva a cabo las instrucciones siguientes:

- Desplaza los bits `%MW40` un byte (8 bits) hacia la derecha y carga el byte correspondiente al número de intercambio de comunicación en la palabra `%MW200`.
- Cancela el intercambio cuyo número de intercambio está contenido en la palabra `%MW200` que utiliza la función `CANCEL`.

- Utilización del bit de cancelación de la función de comunicación:

```
IF (%MW40.0) THEN
    SET(%MW40.1);
    READ_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, %MW40:4,
    %MW10:10);
END_IF;
```

`%MW40` corresponde al parámetro `GEST` (tabla de gestión). `%MW40.0` corresponde al bit de actividad de la función `READ_VAR` y se establece en 1 cuando la función de comunicación está activa. Si este bit se ha establecido en 1, el programa establece el bit `%MW40.1`, el bit de cancelación de la función, en 1. Esto detiene la comunicación de la función `READ_VAR`.

NOTA: Al utilizar el bit de cancelación de la función de comunicación que está en la palabra de gestión de intercambios de la función (`%MW40` en este ejemplo), se debe llamar a la función (`READ_VAR` en este ejemplo) con el fin de activar la cancelación del intercambio.

NOTA: Al utilizar el bit de cancelación de la función de comunicación, es posible cancelar una comunicación desde una tabla de animación. Para ello, basta con establecer el bit de cancelación de la función en 1 (`%MW40.1` en este ejemplo) y, a continuación, iniciar de nuevo la función de comunicación.

NOTA: Este ejemplo de programación afecta a la función `READ_VAR`, pero es igualmente aplicable a las funciones `WRITE_VAR` y `DATA_EXCH`.

NOTA: La función `CANCEL` utiliza una palabra de informe en la función `CANCEL` (`%MW185` en este ejemplo).

Descripción de los parámetros de la función ADDM

En la tabla siguiente se describen los distintos parámetros de la función ADDM:

Parámetro	Tipo	Descripción
IN	STRING	Dirección de dispositivo en una conexión bus o serie. La sintaxis de la dirección es del tipo 'r.m.c.node'. La dirección se compone de los siguientes parámetros: ● r: Número de bastidor del procesador, siempre = 0. ● m: Número de ranura del procesador del bastidor, siempre = 0 ● c: Número de canal, siempre = 0, ya que la conexión serie de un procesador es siempre el canal 0 ● nodo: Número de esclavo al que se envía la solicitud.
OUT	ARRAY [0..7] OF INT	Matriz que representa la dirección de un dispositivo. Este parámetro se puede utilizar como parámetro de entrada para varias funciones de comunicación.

Descripción de los parámetros de la función `READ_VAR`

En la tabla siguiente se describen los distintos parámetros de la función `READ_VAR`:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección de la entidad de destino proporcionada por el parámetro OUT de la función <code>ADDMM</code> .
OBJ	STRING	Tipo de objeto que se va a leer. Los tipos disponibles son: ● %M: bit interno ● %MW: palabra interna ● %I: bit de entrada externa ● %IW: palabra de entrada externa
NUM	DINT	Dirección del primer objeto que se va a leer.
NB	INT	Número de objetos consecutivos que se van a leer.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Número de intercambio ● Byte de menor valor: Bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Informe de operación ● Byte de menor valor: Informe de comunicación ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.
RECP	ARRAY [n..m] OF INT	Tabla de palabras que contiene el valor de los objetos leídos.

Descripción de los parámetros de la función `WRITE_VAR`

En la tabla siguiente se describen los distintos parámetros de la función `WRITE_VAR`:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección de la entidad de destino proporcionada por el parámetro OUT de la función <code>ADDMM</code> .
OBJ	STRING	Tipo de objeto que se va a escribir. Los tipos disponibles son: ● <code>%M</code>: bit interno ● <code>%MW</code>: palabra interna Nota: <code>WRITE_VAR</code> no puede utilizarse para las variables <code>%I</code> y <code>%IW</code> .
NUM	DINT	Dirección del primer objeto que se va a escribir.
NB	INT	Número de objetos consecutivos que se van a escribir.
EMIS	ARRAY [n..m] OF INT	Tabla de palabras que contienen el valor de los objetos que van a escribirse.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Número de intercambio ● Byte de menor valor: Bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Informe de operación ● Byte de menor valor: Informe de comunicación ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.

Descripción de los parámetros de la función DATA_EXCH

En la tabla siguiente se describen los distintos parámetros de la función DATA_EXCH:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección de la entidad de destino proporcionada por el parámetro OUT de la función ADDM.
TYPE	INT	Para PLC Modicon M340, el único valor posible es el siguiente: 1 : transmisión de una matriz EMIS ; a continuación el PLC espera la recepción de una matriz RECP .
EMIS	ARRAY [n..m] OF INT	Tabla de enteros que se enviará al dispositivo destino de la solicitud. Nota: Para que la función se ejecute correctamente, es obligatorio que la longitud de los datos que se van a enviar (en bytes) se asignen a la cuarta palabra de la tabla de gestión antes de iniciar la función.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Número de intercambio ● Byte de menor valor: Bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Informe de operación ● Byte de menor valor: Informe de comunicación ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.
RECP	ARRAY [n..m] OF INT	Tabla de enteros que contiene los datos recibidos. Nota: El sistema escribe automáticamente tamaño de los datos recibidos (en bytes) en la cuarta palabra de la tabla de gestión.

Servicios compatibles con un procesador esclavo de conexión Modbus

Presentación

Cuando se utilizan como un procesador esclavo en una conexión Modbus, los procesadores siguientes admiten varios servicios:

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020

Intercambios de datos

Un procesador esclavo gestiona las siguientes peticiones:

Petición Modbus	Código de función	Objeto del PLC
Lectura de n bits de salida	16#01	%M
Lectura de n palabras de salida	16#03	%MW
Escritura de n bits de salida	16#0F	%M
Escritura de n palabras de salida	16#10	%MW

Diagnóstico y mantenimiento

La información de diagnóstico y mantenimiento accesible desde una conexión Modbus se enumera a continuación:

Designación	Código de función/código de subfunción
Eco	16#08 / 16#00
Lectura de los registros de diagnóstico del PLC	16#08 / 16#02
Restablecimiento en 0 de los registros y los contadores de diagnóstico del PLC	16#08/16#0A
Lectura del número de mensajes en el bus	16#08/16#0B
Lectura del número de errores de comunicación detectados en el bus	16#08/16#0C
Lectura del número de errores de excepción detectados en el bus	16#08/16#0D
Lectura del número de mensajes recibidos del esclavo	16#08/16#0E
Lectura del número de faltas de respuestas del esclavo	16#08/16#0F
Lectura del número de confirmaciones negativas del esclavo	16#08 / 16#10
Lectura del número de respuestas de excepción del esclavo	16#08 / 16#11
Lectura del número de caracteres de desborde en el bus	16#08 / 16#12
Lectura del contador de eventos	16#0B
Lectura de evento de conexión	16#0C
Identificación de lectura	16#11
Identificación del dispositivo de lectura	16#2B/16#0E

5.4 Depuración de una comunicación Modbus serie

Pantalla de depuración de la comunicación Modbus serie

General

A la pantalla de depuración de la comunicación Modbus serie sólo puede accederse en la modalidad online.

Acceso a la pantalla de depuración

En la siguiente tabla se describe el procedimiento para acceder a la pantalla de depuración para la comunicación Modbus serie:

Paso	Acción
1	Acceder a la pantalla de configuración de la comunicación Modbus serie. (véase página 77)
2	En la pantalla que aparece, seleccionar la ficha "Depuración".

Descripción de la pantalla Depuración

La pantalla de depuración se divide en dos áreas:

- Área Tipo
- Área Contadores

Área Tipo

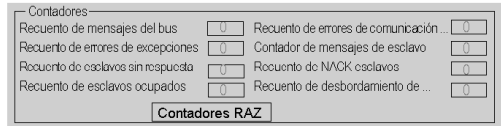
Esta área tiene un aspecto similar al siguiente:



Indica el tipo de función Modbus configurada (en este caso, Maestro).

Área Contadores

Esta área tiene un aspecto similar al siguiente:



Esta área muestra los distintos contadores de depuración.

El botón Resetea contadores restablece a cero todos los contadores de la modalidad de depuración.

Funcionamiento del contador

Los contadores de depuración de la comunicación Modbus serie son los siguientes:

- Contador de mensajes del bus: este contador indica el número de mensajes que el procesador ha detectado en la conexión serie. No incluye mensajes con un resultado negativo de suma de control CRC.
- Contador de errores de comunicación del bus: este contador indica el número de resultados negativos de suma de control CRC que ha contado el procesador. Si se detecta un error de caracteres (error de paridad, desborde), o el mensaje es inferior a tres bytes de longitud, el sistema que recibe los datos no puede realizar la suma de control CRC. En estos casos, el contador se incrementa en consecuencia.
- Contador de errores de excepción de esclavo: este contador indica el número de errores de excepción de Modbus detectados por el procesador.
- Contador de mensajes de esclavo: este contador indica el número de mensajes recibidos y procesados por la conexión Modbus.
- Contador de esclavos sin respuesta: este contador indica el número de mensajes enviados por el sistema remoto de los que no ha recibido respuesta (ni una respuesta normal ni una respuesta de excepción). Incluye además el número de mensajes recibidos en modalidad Broadcast.
- Contador de confirmaciones de esclavo negativas: este contador indica el número de mensajes enviados al sistema remoto para los que ha devuelto una confirmación negativa.
- Contador de esclavos ocupados: este contador indica el número de mensajes enviados al sistema remoto para los que ha devuelto un mensaje de excepción de esclavo ocupado.
- Contador de desbordamiento de caracteres del bus: este contador indica el número de mensajes enviados al procesador que no pueden adquirirse debido al desbordamiento de caracteres del bus. El desbordamiento ha sido provocado por uno de estos motivos:
 - Datos de tipo carácter que se transmiten por el puerto serie más rápidamente de lo que pueden almacenarse.
 - Pérdida de datos debida a una anomalía del hardware.

NOTA: Para todos los contadores, la cuenta comienza en el reinicio, la operación de borrado de contadores o el arranque del procesador más reciente.

**Comunicaciones serie en los
procesadores
BMX P34 1000/2000/2010/20102/2
020**

6

Objeto de esta sección

En este capítulo se presenta el proceso de implementación del software de comunicación en modalidad de caracteres en los procesadores BMX P34 1000/2000/2010/20102/2020.

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
6.1	Generalidades	102
6.2	Configuración de la comunicación en modalidad de caracteres	106
6.3	Programación de la comunicación en modalidad de caracteres	116
6.4	Depuración de la comunicación en modalidad de caracteres	122

6.1 Generalidades

Objeto de esta sección

En esta sección se ofrece una descripción general de las generalidades relativas a la comunicación en modalidad de caracteres y sus servicios.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Acerca de la comunicación en modalidad de caracteres	103
Rendimiento	104

Acerca de la comunicación en modalidad de caracteres

Introducción

La comunicación en modalidad de caracteres permite llevar a cabo funciones de diálogo y comunicación entre los PLC y los siguientes dispositivos:

- periféricos normales (impresora, teclado–pantalla, terminal de taller, etc.),
- periféricos especializados (lectores de código de barras, etc.),
- calculadoras (comprobación, gestión de producción, etc.),
- dispositivos heterogéneos (comandos numéricos, controladores de velocidad variable, etc.),
- módem externo.

ADVERTENCIA

PÉRDIDA DE DATOS CRÍTICOS

Los puertos de comunicación deben utilizarse exclusivamente para transferencias de datos que no sean fundamentales.

Si no se siguen estas instrucciones pueden producirse lesiones personales graves o mortales o daños en el equipo.

Rendimiento

Presentación

En las siguientes tablas se describen los tiempos de intercambio habituales en la modalidad de caracteres.

Los resultados mostrados corresponden al periodo de funcionamiento medio de la función `PRINT_CHAR` en milisegundos.

Definición de tiempo de intercambio

El tiempo de intercambio es el tiempo que transcurre entre la creación de un intercambio y el final de éste. Incluye el tiempo de comunicación de conexión serie.

El intercambio se crea al efectuar la llamada a la función de comunicación.

El intercambio finaliza cuando se produce alguno de los siguientes eventos:

- Recepción de datos
- Una anomalía
- Se agota el timeout

Tiempos de intercambio de 80 caracteres

En la siguiente tabla se muestran los tiempos de intercambio para la transmisión de 80 caracteres en modalidad de caracteres en un procesador BMX P34 2020:

Velocidad de transmisión de la comunicación en bits por segundo	Duración del ciclo en ms	Tiempos de intercambio en ms
1.200	10	805
1.200	20	820
1.200	50	850
1.200	100	900
1.200	255	980
4.800	10	210
4.800	20	220
4.800	50	250
4.800	100	300
4.800	255	425
9.600	10	110
9.600	20	115
9.600	50	145
9.600	100	200
9.600	255	305

19.200	10	55
19.200	20	60
19.200	50	95
19.200	100	100
19.200	255	250

Los tiempos de intercambio del procesador BMX P34 2000/2010/20102 son similares a los del procesador BMX P34 2020. Los tiempos de intercambio del procesador BMX P34 1000 son un 10% más bajos.

Precisión de la medida

Todos los tiempos de intercambio enumerados anteriormente provienen de medidas con un margen de precisión de +/-10 ms.

6.2 Configuración de la comunicación en modalidad de caracteres

Objeto de esta sección

En esta sección se describe el proceso de configuración que se utiliza al implementar la comunicación en modalidad de caracteres.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Pantalla de configuración de la comunicación en modalidad de caracteres	107
Funciones accesibles en modalidad de caracteres	109
Valores predeterminados para los parámetros de comunicación en modalidad de caracteres	110
Parámetros de detección de final de mensaje en modalidad de caracteres	111
Parámetros de transmisión en modalidad de caracteres	113
Parámetros de línea física y señales en modalidad de caracteres	115

Pantalla de configuración de la comunicación en modalidad de caracteres

General

En las páginas siguientes se ofrece una introducción a la pantalla de configuración de la comunicación en modalidad de caracteres.

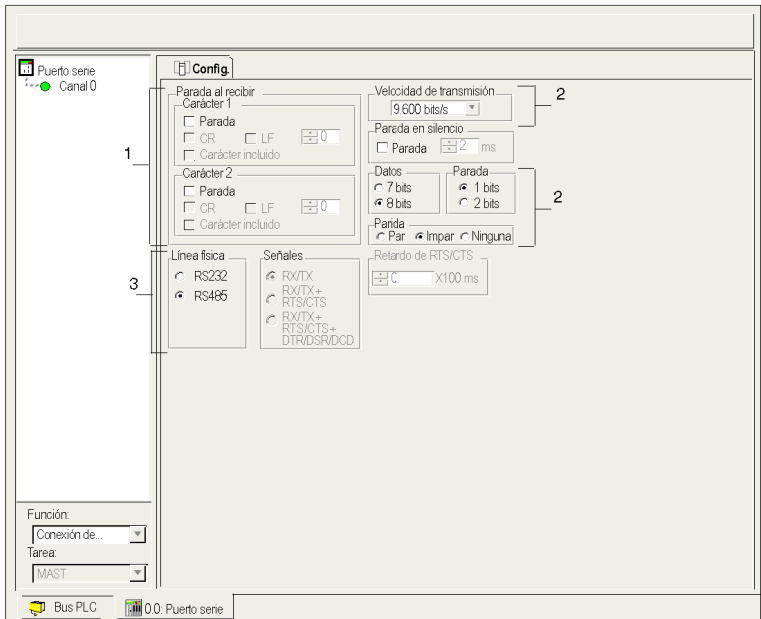
Acceso a la pantalla Configuración

En la siguiente tabla se describe el procedimiento para acceder a la pantalla de configuración para la comunicación en modalidad de caracteres:

Paso	Acción
1	Abra el subdirectorio Puerto serie del explorador de proyectos (véase página 72).
2	En la pantalla que aparece, seleccione la función CONEXIÓN MODALIDAD DE CARACTERES.

Ilustración

En la siguiente figura se muestra la pantalla de configuración predeterminada de la comunicación en modalidad de caracteres:



Descripción

Esta área se utiliza para configurar parámetros de canal. En la modalidad online, no se puede acceder a esta área, que estará atenuada. En la modalidad offline, puede accederse a esta área, pero es posible que no pueda accederse a algunos parámetros que, por tanto, estarán atenuados.

En la siguiente tabla se muestran las distintas áreas de las pantallas de configuración de la comunicación en modalidad de caracteres:

Tecla	Elemento	Comentario
1	Parámetros de detección de final de mensaje (véase página 111)	Puede accederse a estos parámetros mediante dos áreas: ● Parada al recibir. ● Parada en silencio.
2	Parámetros de transmisión (véase página 113)	Puede accederse a estos parámetros mediante cuatro áreas: ● Velocidad de transmisión. ● Datos. ● Bits de parada. ● Paridad.
3	Parámetros de línea física y señales (véase página 115)	Puede accederse a estos parámetros mediante tres áreas: ● Línea física. ● Señales. ● Retardo de RTS/CTS.

NOTA: En este ejemplo, las áreas "Señales" y "Retraso RTS/CTS" están atenuadas porque se ha elegido una línea física RS485.

Funciones accesibles en modalidad de caracteres

Presentación

La accesibilidad a funciones para la configuración de la conexión serie de los procesadores siguientes mediante el protocolo de modalidad de caracteres varía en función de la conexión física que se utilice:

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010/20102
- BMX P34 2020

Funciones accesibles

En la siguiente tabla se muestran las distintas funciones configurables en función del tipo de conexión serie utilizado:

Función	Conexión RS 485	Conexión RS 232
Velocidad de transmisión	X	X
Datos	● 7 bits ● 8 bits	● 7 bits ● 8 bits
Parada	● 1 bit ● 2 bits	● 1 bit ● 2 bits
Paridad	● Impar ● Par ● Ninguna	● Impar ● Par ● Ninguna
Parada al recibir	X	X
Parada en silencio	X	X
Señales RX/TX	X	X
Señales RTS/CTS	-	X
Retardo de RTS/CTS	-	X

- X** Función accesible
- Función inaccesible

Valores predeterminados para los parámetros de comunicación en modalidad de caracteres

Presentación

Todos los parámetros de comunicación en modalidad de caracteres tienen valores predeterminados.

Valores predeterminados

En la siguiente tabla se muestran los valores predeterminados de los parámetros de comunicación en modalidad de caracteres:

Parámetro de configuración	Valor
Línea física	RS 485
Velocidad de transmisión	9.600 bits/s
Paridad	Impar
Bits de datos	8 bits
Bits de parada	1 bit

Parámetros de detección de final de mensaje en modalidad de caracteres

Presentación

Una vez configurado el canal de comunicación, deben introducirse los parámetros de detección de final de mensaje.

Puede accederse a estos parámetros mediante dos áreas:

- Área Parada al recibir: condición de parada al recibir un carácter especial.
- Área Parada en silencio: condición de parada por silencio.

Condiciones de uso

Al seleccionar Parada en silencio se anula la selección de Parada al recibir y viceversa.

Área Parada al recibir

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:

Parada al recibir

Carácter 1

☒ Parada

☐ CR ☒ LF Datos 10

☐ Caracteres incluidos

Carácter 2

☒ Parada

☒ CR ☐ LF Datos 13

☐ Caracteres incluidos

Una petición de recepción puede terminarse a partir de la recepción de un carácter determinado.

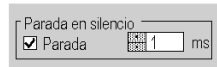
Al marcar la opción Parada, es posible configurar Parada al recibir para que se active mediante un carácter de final de mensaje determinado:

- CR: permite detectar el final del mensaje por un retorno de carro.
- LF: permite detectar el final del mensaje por un retorno de línea.
- Campo de entrada Datos: permite identificar un carácter de final de mensaje distinto de los caracteres CR o LF utilizando un valor decimal:
 - Entre 0 y 255 si los datos se codifican en 8 bits
 - Entre 0 y 127 si los datos se codifican en 7 bits
- Carácter incluido: permite incluir un carácter de final de mensaje en la tabla de recepción de la aplicación del PLC.

Pueden configurarse dos caracteres de fin de recepción. En la ventana siguiente, el fin de recepción de un mensaje se detecta mediante un carácter LF o CR.

Área Parada en silencio:

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:

A screenshot of a configuration window titled 'Parada en silencio'. It contains a checked checkbox labeled 'Parada' and a text input field with the value '1' and the unit 'ms'.

Esta área permite detectar el final de un mensaje en la recepción mediante la ausencia de caracteres de final de mensaje durante un tiempo determinado.

Parada en silencio se valida al marcar la casilla Detener. La duración del silencio (expresada en milisegundos) se establece mediante el campo de entrada de datos.

NOTA: Los valores disponibles están comprendidos entre 1 ms y 10000 ms y varían en función de la velocidad de transmisión seleccionada.

Parámetros de transmisión en modalidad de caracteres

Presentación

Una vez configurado el canal de comunicación, deben introducirse los parámetros reservados de transmisión.

Puede accederse a estos parámetros mediante cuatro áreas:

- Área Velocidad de transmisión
- Área Datos
- Área Parada
- Área Paridad

Área Velocidad de transmisión

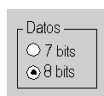
Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



Puede utilizar esta área para seleccionar la velocidad de transmisión del protocolo de modalidad de caracteres. La velocidad seleccionada tiene que ser coherente con los demás dispositivos. Los valores configurables son 300, 600, 1.200, 2.400, 4.800, 9.600, 19.200 y 38.400 bits por segundo.

Área Datos

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



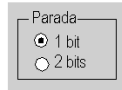
En esta área, puede especificarse el tamaño de los datos que se intercambian en la conexión. Los valores disponibles son:

- 7 bits
- 8 bits

Se recomienda ajustar el número de bits de datos en función de los dispositivos remotos.

Área Parada

Esta área tiene un aspecto similar al siguiente:



El área Parada permite introducir el número de bits de parada utilizados en la comunicación. Se recomienda ajustar el número de bits de parada en función de los dispositivos remotos.

Los valores configurables son:

- 1 bit
- 2 bits

Área Paridad

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



Esta área permite determinar si se agrega un bit de paridad, así como su tipo. Se recomienda ajustar la paridad en función de los dispositivos remotos. Los valores configurables son:

- Par
- Impar
- Ninguna

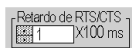
Parámetros de línea física y señales en modalidad de caracteres

Presentación

Una vez configurado el canal de comunicación, deben introducirse los parámetros de línea física y señales. Estos parámetros son idénticos a los parámetros de línea física y señales de la comunicación Modbus (*véase página 86*).

Área Retardo RTS/CTS

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



La zona de retardo RTS/CTS sólo está disponible cuando las casillas RS232 y RX/TX+RTS/CTS están seleccionadas.

Un algoritmo de control del flujo RTS/CTS está seleccionado: Antes de transmitir una cadena de caracteres, el sistema espera a que la señal CTS (Clear To Send) se active. Esta área permite introducir el tiempo de espera máximo entre las dos señales. Si el tiempo se termina, la petición no se envía al bus. Los valores configurables están comprendidos entre 0 s y 10 s.

NOTA: El valor predeterminado es 0 ms.

NOTA: Un valor de 0 s indica que no se ha gestionado el retardo entre las dos señales.

Algoritmo del control de flujo RTS/CTS

El objetivo es evitar un desbordamiento del búfer de recepción.

La señal de salida RTS de cada dispositivo se conecta a la señal de entrada CTS de otro dispositivo. El transmisor (M340) está autorizado a transmitir los datos al recibir la señal de entrada RTS (por ejemplo, otro M340) en su entrada CTS. Este algoritmo es simétrico y permite la comunicación asíncrona dúplex completo.

6.3 Programación de la comunicación en modalidad de caracteres

Funciones de comunicación en modalidad de caracteres

Funciones disponibles

Se definen dos funciones específicas de comunicación para enviar y recibir datos mediante un canal de comunicación en modalidad de caracteres:

- `PRINT_CHAR`: enviar una cadena de caracteres de un máximo de 1.024 bytes.
- `INPUT_CHAR`: leer una cadena de caracteres de un máximo de 1.024 bytes.

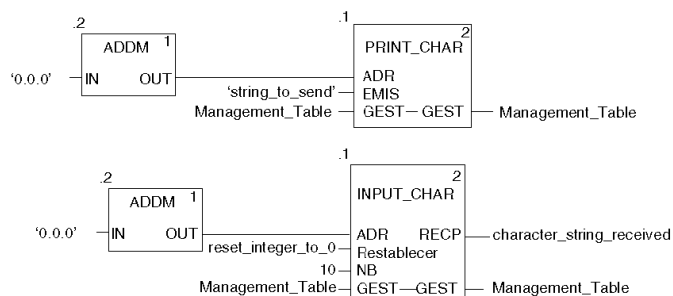
El puerto serie del PLC Modicon M340 es dúplex completo, por lo que es posible enviar una función `PRINT_CHAR` aunque se haya enviado una función `INPUT_CHAR` y esté todavía pendiente.

NOTA: Para la función `INPUT_CHAR`, se necesita un timeout configurado si el canal está configurado sin parada en silencio, para confirmar el bit de actividad de la función. Para la función `PRINT_CHAR` es recomendable, pero no necesario, configurar un timeout.

NOTA: Al contrario del NOM0200 en el enlace RS485, la CPU guarda el ECO de los datos transmitidos en el mismo búfer que los datos recibidos. Por lo tanto, es obligatorio borrar el búfer de la CPU después de cada `PRINT_CHAR` o antes de que un usuario envíe datos al canal. De lo contrario, los datos recibidos de un `INPUT_CHAR` o `INPUT_BYTE` no serán los esperados. Para borrar el búfer de la CPU, puede realizar un `INPUT_CHAR` con el búfer de Reset activado y cancelar esta función elemental antes del timeout.

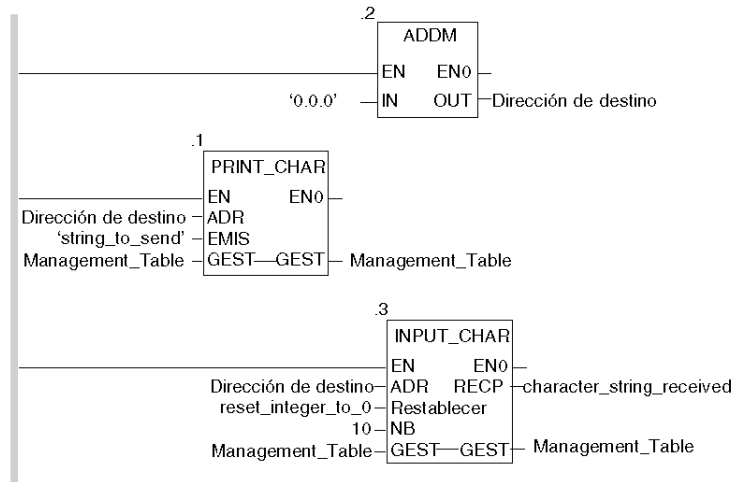
Ejemplo de programación en FBD

En el siguiente diagrama se representa un ejemplo de programación de las funciones de comunicación `PRINT_CHAR` e `INPUT_CHAR` en lenguaje FBD:



Ejemplo de programación en Ladder

En el siguiente diagrama se representa un ejemplo de programación de las funciones de comunicación PRINT_CHAR e INPUT_CHAR en lenguaje Ladder:



Ejemplo de programación en ST

Las siguientes líneas de código representan un ejemplo de programación de las funciones de comunicación PRINT_CHAR e INPUT_CHAR en lenguaje ST:

```
PRINT_CHAR(ADDM('0.0.0'), 'string_to_send',  
Management_Table);  
  
INPUT_CHAR(ADDM('0.0.0'), reset_integer_to_0, 10,  
Management_Table, character_string_received);
```

Característica de la función INPUT_CHAR

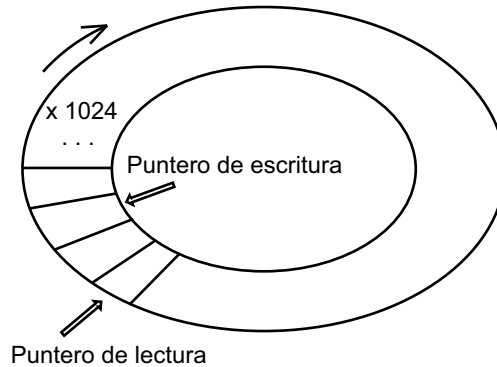
Si el parámetro de entrada Restablecer está establecido en 1, primero se restablece el búfer y, a continuación, el procesador espera la recepción de datos. Se recomienda utilizar esta función para iniciar correctamente una recepción eliminando los datos antiguos que puedan permanecer en el búfer.

Programa interno de la CPU

Los datos recibidos se almacenan en un búfer cíclico de 1024 bits: Una vez que se ha llenado completamente el búfer, el bit 1025 recibido sobrescribe al primer bit, etc. Se restablece cada bit del búfer leído mediante la función INPUT CHAR.

Dos punteros independientes permiten el acceso para leer y escribir los datos.

En la siguiente figura se representa este mecanismo:



Cancelación de un intercambio

Existen dos maneras de programar que habilitan la cancelación de un intercambio ejecutado mediante las funciones PRINT_CHAR y INPUT_CHAR. Estas dos funciones se presentan a continuación en lenguaje ST:

- Utilización de la función CANCEL:

```
IF (%MW40.0) THEN
    %MW200 := SHR (%MW40, 8;)
    CANCEL (%MW200, %MW185);
```

```
END_IF;
```

%MW40 corresponde al parámetro GEST (tabla de gestión). %MW40.0

corresponde al bit de actividad de la función PRINT_CHAR y se establece en 1 cuando la función de comunicación está activa. Si el bit se establece en 1, el programa lleva a cabo las instrucciones siguientes:

- Desplaza los bits %MW40 un byte (8 bits) hacia la derecha y carga el byte correspondiente al número de intercambio de comunicación en la palabra %MW200.
- Cancela el intercambio cuyo número de intercambio está contenido en la palabra %MW200 que utiliza la función CANCEL.

- Utilización del bit de cancelación de la función de comunicación:

```
IF (%MW40.0) THEN
  SET (%MW40.1);
  PRINT_CHAR (ADDM('0.0.0'), 'string_to_send', %MW40:4);
END_IF;
```

%MW40 corresponde al parámetro GEST (tabla de gestión). %MW40.0 corresponde al bit de actividad de la función PRINT_CHAR y se establece en 1 cuando la función de comunicación está activa. Si este bit se ha establecido en 1, el programa establece el bit %MW40.1, el bit de cancelación de la función, en 1. Esto detiene la comunicación de la función PRINT_CHAR.

NOTA: Al utilizar el bit de cancelación de la función de comunicación, debe llamarse a la función para habilitar el bit de cancelación contenido en la palabra de gestión de intercambios de la función (%MW40 en este ejemplo).

NOTA: Al utilizar el bit de cancelación de la función de comunicación, es posible cancelar una comunicación desde una tabla de animación. Para ello, basta con establecer el bit de cancelación de la función en 1 (%MW40.1 en este ejemplo).

NOTA: Este ejemplo de programación afecta a la función PRINT_CHAR, pero es igualmente aplicable a la función INPUT_CHAR.

NOTA: La función CANCEL utiliza una palabra de informe en la función CANCEL (%MW185 en este ejemplo).

Descripción de los parámetros de la función ADDM

En la tabla siguiente se describen los distintos parámetros de la función ADDM:

Parámetro	Tipo	Descripción
IN	STRING	Dirección de dispositivo en una conexión bus o serie. La sintaxis de la dirección es del tipo 'r.m.c.node'. La dirección se compone de los siguientes parámetros: ● r: número de bastidor del sistema de destino, siempre = 0. ● m: número de slot del sistema de destino del bastidor, siempre = 0. ● c: número de canal, siempre = 0, ya que la conexión serie de un sistema remoto es siempre el canal 0. ● nodo: campo opcional que puede ser SYS o estar vacío.
OUT	ARRAY [0..7] OF INT	Tabla que muestra la dirección de un dispositivo. Este parámetro se puede utilizar como parámetro de entrada para varias funciones de comunicación.

Descripción de los parámetros de la función PRINT_CHAR

En la tabla siguiente se describen los distintos parámetros de la función PRINT_CHAR:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección del canal del modo de caracteres que reciben mensajes proporcionada por el parámetro OUT de la función ADDM.
EMIS	STRING	Cadena de caracteres para enviar.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: número de intercambio ● Byte de menor valor: bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: informe de funcionamiento. ● Byte de menor valor: informe de comunicación. ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el usuario, que define la duración del intercambio. ● Si la longitud de este parámetro se establece en 0, el sistema envía toda la cadena. ● Si la longitud de este parámetro es superior a la longitud de la cadena, se devuelve el error 16#0A (Tamaño del búfer de envío insuficiente) en la segunda palabra de gestión y no se envía ningún carácter.

Descripción de los parámetros de la función INPUT_CHAR

En la tabla siguiente se describen los distintos parámetros de la función INPUT_CHAR:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección del canal del modo de caracteres que reciben mensajes proporcionada por el parámetro OUT de la función ADDM.
Restablecer	INT	Este parámetro puede adoptar dos valores: ● Valor 1: restablece en 0 la memoria de recepción del módulo. ● Valor 0: no restablece en 0 la memoria de recepción del módulo.
NB	INT	Longitud de la cadena de caracteres que se va a recibir.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: número de intercambio ● Byte de menor valor: bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: informe de funcionamiento. ● Byte de menor valor: informe de comunicación. ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.
RECP	STRING	Cadena de caracteres recibida. Esta cadena se guarda en una cadena de caracteres.

6.4 Depuración de la comunicación en modalidad de caracteres

Pantalla de depuración de la comunicación en modalidad de caracteres

General

A la pantalla de depuración en modalidad de caracteres sólo puede accederse en la modalidad online.

Acceso a la pantalla de depuración

En la tabla siguiente se describe el procedimiento para acceder a la pantalla de depuración para la comunicación en modalidad de caracteres:

Paso	Acción
1	Acceder a la pantalla de configuración de la comunicación en modalidad de caracteres. (véase página 107)
2	En la pantalla que aparece, seleccionar la ficha "Depuración".

Descripción de la pantalla Depuración

La pantalla de depuración se compone de un área Error y un área Señales.

Área Error

El área Error tiene un aspecto similar al siguiente:

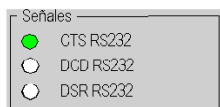
Esta área indica el número de interrupciones de comunicación computados por el procesador:

- **Al transmitir:** corresponde al número de interrupciones durante la transmisión (imagen de la palabra %MW4).
- **Al recibir:** corresponde al número de interrupciones durante la recepción (imagen de la palabra %MW5).

El botón Resetear contadores restablece a cero ambos contadores.

Área Señales

El área Señales tiene un aspecto similar al siguiente:



Esta área indica la actividad de las señales:

- **CTS RS232:** muestra la actividad de la señal CTS.
- **DCD RS232:** no está gestionada por el procesador (ausencia de actividad en este indicador LED).
- **DSR RS232:** no está gestionada por el procesador (ausencia de actividad en este indicador LED).

Comunicación en modalidad Modbus serie para BMX NOM 0200



Objetivo de este capítulo

En este capítulo se presenta el proceso de implementación del software de comunicación en modalidad Modbus serie para BMX NOM 0200.

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
7.1	Generalidades	126
7.2	Configuración de la comunicación Modbus serie	134
7.3	Programación de la comunicación Modbus serie	148
7.4	Depuración de una comunicación Modbus serie	159

7.1 Generalidades

Objeto de esta sección

En esta sección se presentan las generalidades relativas a la comunicación Modbus serie y sus servicios.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Acerca de Modbus serie	127
Rendimiento	128
Acceso a los parámetros de la conexión serie	130

Acerca de Modbus serie

Introducción

La comunicación mediante Modbus permite intercambios de datos entre todos los dispositivos conectados al bus. Modbus es un protocolo que crea una estructura jerárquica (un maestro y varios esclavos).

El maestro gestiona todos los intercambios de dos maneras:

- El maestro realiza el intercambio con el esclavo y espera una respuesta.
- El maestro realiza el intercambio con el conjunto de esclavos sin esperar respuesta (difusión general).

NOTA: Procure que dos maestros (en el mismo bus) no envíen dos peticiones simultáneamente. De lo contrario, ambas se perderán y cada informe tendrá un mal resultado, que podría ser 16#0100 (la petición no se ha podido procesar) o 16#ODFF (falta el esclavo).



ADVERTENCIA

PÉRDIDA DE DATOS CRÍTICOS

Los puertos de comunicación deben utilizarse exclusivamente para transferencias de datos que no sean fundamentales.

Si no se siguen estas instrucciones pueden producirse lesiones personales graves o mortales o daños en el equipo.

Rendimiento

Presentación

Pueden utilizarse las siguientes tablas para evaluar los tiempos de intercambio de comunicación Modbus habituales en función de distintos criterios.

Los resultados mostrados corresponden al periodo de funcionamiento medio de la función `READ_VAR` en milisegundos.

Definición de tiempo de intercambio

El tiempo de intercambio es el tiempo que transcurre entre la creación de un intercambio y el final de éste. Incluye el tiempo de comunicación de conexión serie.

El intercambio se crea al efectuar la llamada a la función de comunicación.

El intercambio finaliza cuando se produce alguno de los siguientes eventos:

- Se reciben datos.
- Se produce una anomalía.
- Se agota el timeout.

Tiempo de intercambio de una palabra

En la siguiente tabla se muestran los tiempos de intercambio de una palabra de comunicación Modbus en un módulo BMX NOM 0200:

Velocidad de transmisión de la comunicación en bits por segundo	Duración del ciclo en ms	Tiempos de intercambio en ms El esclavo Modbus es un BMX P34 1000 cíclico
4.800	Cíclico	65
4.800	10	68
4.800	50	100
9.600	Cíclico	38
9.600	10	47
9.600	50	50
19.200	Cíclico	29
19.200	10	38
19.200	50	50
38.400	Cíclico	24
38.400	10	30
38.400	50	50
57.600	Cíclico	17
57.600	10	20

Velocidad de transmisión de la comunicación en bits por segundo	Duración del ciclo en ms	Tiempos de intercambio en ms El esclavo Modbus es un BMX P34 1000 cíclico
57.600	50	50
115.200	Cíclico	17
115.200	10	20
115.200	50	50

Tiempo de intercambio de 100 palabras

En la siguiente tabla se muestran los tiempos de intercambio de 100 palabras de comunicación Modbus en un procesador BMX NOM 0200:

Velocidad de transmisión de la comunicación en bits por segundo	Duración del ciclo en ms	Tiempos de intercambio en ms El esclavo Modbus es un BMX P34 1000 cíclico
4.800	Cíclico	560
4.800	10	560
4.800	50	600
9.600	Cíclico	286
9.600	10	295
9.600	50	300
19.200	Cíclico	152
19.200	10	160
19.200	50	200
38.400	Cíclico	86
38.400	10	90
38.400	50	100
57.600	Cíclico	56
57.600	10	60
57.600	50	100
115.200	Cíclico	36
115.200	10	40
115.200	50	50

Precisión de la medida

Todos los tiempos de intercambio enumerados anteriormente provienen de medidas con un margen de precisión de +/-10 ms.

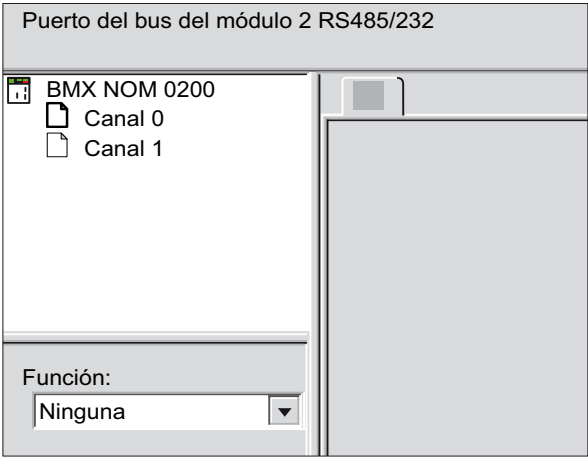
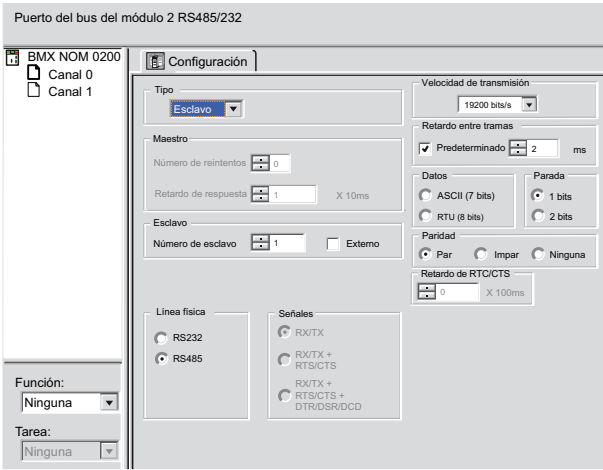
Acceso a los parámetros de la conexión serie

Introducción

En las siguientes páginas se describe cómo acceder a la pantalla de configuración de puertos serie para el módulo BMX NOM 0200, así como los elementos generales de las pantallas de configuración y depuración de la conexión Modbus y de modalidad de caracteres.

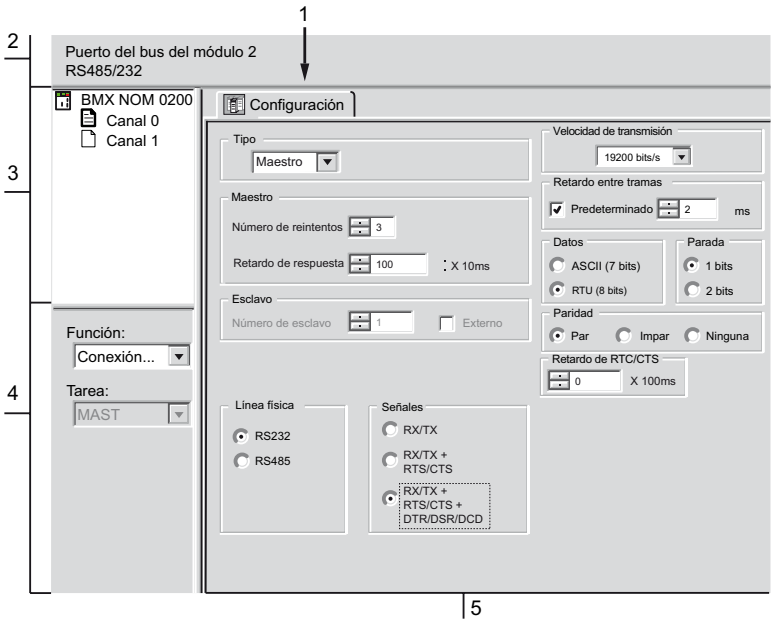
Acceso a la conexión serie

En la siguiente tabla se describe el procedimiento para acceder a la conexión serie de un módulo BMX NOM 0200:

Paso	Acción
1	Abrir el editor de configuración de hardware.
2	Haga doble clic el módulo BMX NOM 0200.
3	Seleccione el canal que se configurará (canal 0 o canal 1). Resultado con el canal 0 seleccionado: 
4	Seleccione la función conexión Modbus . Resultado con el canal 0 seleccionado: 

Descripción de las pantallas Configuración y Depuración:

En la siguiente figura se muestra la pantalla de configuración de la comunicación en Modbus serie en el canal 0:



Descripción

En la siguiente tabla se muestran los distintos elementos de las pantallas de configuración y depuración:

Tecla	Elemento	Función
1	Fichas	La ficha en primer plano indica la modalidad actual (en este caso, Configuración). Cada modalidad puede seleccionarse con la ficha correspondiente. Las modalidades disponibles son: ● Configuración ● Depuración (accesible únicamente en modalidad online) ● Diagnóstico (accesible únicamente en modalidad online)
2	Área de módulo	Muestra la referencia del módulo y el estado de los indicadores LED del módulo en la modalidad online.
3	Área de canal	Permite: ● Mostrar las siguientes fichas haciendo clic en BMX NOM 0200: ● "Descripción general", que proporciona las características del dispositivo. ● «Objetos de E/S» (véase <i>Unity Pro, Modalidades de funcionamiento</i>,), que se utiliza para presimbolizar los objetos de entrada/salida. ● "Fallo", que muestra los fallos del dispositivo detectados (en modalidad online). ● Mostrar las siguientes fichas al hacer clic en Canal 0 o Canal 1: ● "Configuración" ● "Depuración" ● "Fallo" ● Mostrar el nombre del canal y el símbolo definidos por el usuario (mediante el editor de variables).
4	Área de parámetros generales	Permite elegir los parámetros generales asociados al canal: ● Función: las funciones disponibles son "Ninguno", "Conexión Modbus" y "Conexión en modalidad de caracteres". De forma predeterminada, está configurada la función "Ninguno". ● Tarea: define la tarea maestra en donde se intercambiarán los objetos de intercambio implícitos del canal. Esta área aparece atenuada y no puede configurarse.
5	Área Configuración o Depuración o Fallo	En la modalidad de configuración, esta área se utiliza para configurar los parámetros del canal. En la modalidad de depuración, se utiliza para depurar el canal de comunicación. En el modo de diagnóstico, se utiliza para mostrar los errores actuales detectados en el módulo o en el canal.

7.2 Configuración de la comunicación Modbus serie

Objeto de esta sección

En esta sección se describe el proceso de configuración del software para la comunicación Modbus serie.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Pantalla de configuración de la comunicación Modbus serie	135
Funciones Modbus accesibles	137
Valores predeterminados para los parámetros de comunicación Modbus serie	138
Parámetros Modbus relacionados con la aplicación	139
Parámetros Modbus relacionados con la transmisión	141
Parámetros de línea física y señales en Modbus	144
Cómo se establece la dirección del esclavo BMX NOM0200 MODBUS sin Unity Pro?	146

Pantalla de configuración de la comunicación Modbus serie

General

En las páginas siguientes se ofrece una introducción a la pantalla de configuración de la comunicación en para Modbus serie.

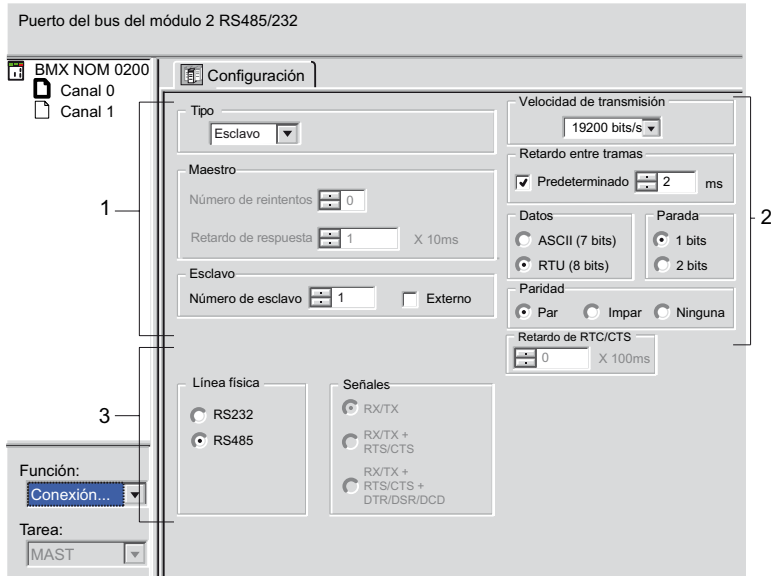
Acceso a la pantalla de configuración

En la siguiente tabla se describe el procedimiento para acceder a la pantalla de configuración para la comunicación Modbus serie:

Paso	Acción
1	Abra el subdirectorio BMX NOM 0200 del explorador de proyectos (véase página 130).
2	En la pantalla que aparece, seleccione el canal que se configurará y la función de conexión Modbus.

Ilustración

En la siguiente figura se muestra la pantalla de configuración predeterminada de la comunicación en Modbus serie en el canal 0:



Descripción

Esta área se utiliza para configurar parámetros de canal. En la modalidad online, no se puede acceder a esta área, que estará atenuada. En la modalidad offline, puede accederse a esta área, pero es posible que no pueda accederse a algunos parámetros que estarán atenuados.

En la siguiente tabla se muestran las distintas áreas de las pantallas de configuración de la comunicación de Modbus:

Tecla	Elemento	Comentario
1	Parámetros de la aplicación (véase página 139)	Puede accederse a estos parámetros mediante tres áreas: ● Tipo ● Maestro ● Esclavo
2	Parámetros de transmisión (véase página 141)	Puede accederse a estos parámetros mediante cinco áreas: ● Velocidad de transmisión. ● Retardo entre tramas. ● Datos. ● Bits de parada. ● Paridad.
3	Parámetros de línea física y señales (véase página 144)	Puede accederse a estos parámetros mediante tres áreas: ● Línea física. ● Señales. ● Retardo de RTS/CTS.

NOTA: Al configurar la comunicación para Modbus serie el modo maestro, el área Esclavo está atenuada y no puede modificarse, y viceversa.

NOTA: En este ejemplo, las áreas "Señales" y "Retraso RTS/CTS" están atenuadas porque se ha elegido una línea física RS485.

Funciones Modbus accesibles

Presentación

La accesibilidad a funciones para la configuración de la conexión serie de un módulo BMX NOM 0200 mediante Modbus serie varía en función de la conexión física que se utilice.

Funciones accesibles

En la siguiente tabla se muestran las distintas funciones configurables en función del tipo de conexión serie utilizado:

Función	Conexión RS485 (en el canal 0 o el canal 1)	Conexión RS232 (en el canal 0)
Número de reintentos del maestro	X	X
Retardo de respuesta de maestro	X	X
Número de esclavo	X	X
Velocidad de transmisión	X	X
Retardo entre tramas	X	X
Datos	● ASCII (7 bits) ● RTU (8 bits)	● ASCII (7 bits) ● RTU (8 bits)
Parada	● 1 bit ● 2 bits	● 1 bit ● 2 bits
Paridad	● Impar ● Par ● Ninguna	● Impar ● Par ● Ninguna
Señales RX/TX	X	X
Señales RTS/CTS	-	X
Retardo de RTS/CTS	-	X
Señales DTR/DSR/DCD	-	X
Polarización	-	-

- X** Función accesible
- Función inaccesible

Valores predeterminados para los parámetros de comunicación Modbus serie

Presentación

Todos los parámetros de comunicación Modbus tienen valores predeterminados serie.

Valores predeterminados

En la siguiente tabla se muestran los valores predeterminados de los parámetros de comunicación Modbus serie en el canal 0 y en el canal 1 del módulo BMX NOM 0200:

Parámetro de configuración	Valor
Modalidad	Esclavo
Línea física	RS485
Número de esclavo	1
Retardo entre tramas	2 ms
Velocidad de transmisión	19200 bits/s
Paridad	Par
Bits de datos	RTU (8 bits)
Bits de parada	1 bit

Parámetros Modbus relacionados con la aplicación

Presentación

Una vez configurado el canal de comunicación, deben introducirse los parámetros de la aplicación.

Puede accederse a estos parámetros desde tres áreas de configuración:

- Área Tipo
- Área Maestro
- Área Esclavo

Área Tipo

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:

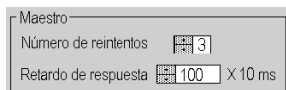


Esta área permite seleccionar la función que se configurará para el módulo en la comunicación Modbus serie:

- **Maestro:** Cuando el módulo es el maestro.
- **Esclavo:** Cuando el módulo es un esclavo.

Área Maestro

Sólo puede accederse al área de configuración que se muestra a continuación cuando se selecciona «Maestro» en el área «Tipo»:



Maestro

Número de reintentos: 3

Retardo de respuesta: 100 X 10 ms

Esta área permite introducir los siguientes parámetros:

- **Número de reintentos:** número de intentos de conexión que lleva a cabo el maestro antes de declarar ausente al esclavo.
El valor predeterminado es 3.
Los valores posibles están comprendidos entre 0 y 15.
El valor 0 indica que no hay reintentos por parte del maestro.
- **Retardo de respuesta:** es el tiempo transcurrido entre la petición inicial enviada por el maestro y un intento repetido de ésta en caso de que el esclavo no responda. Se corresponde con el tiempo máximo entre el envío del último carácter de la petición enviada por el maestro y la recepción del primer carácter de la petición reenviada por el esclavo.
El valor predeterminado es 1 segundo (100*10 ms).
Los valores posibles están comprendidos entre 10 ms y 10 s.

NOTA: El retardo de respuesta del maestro debe ser por lo menos igual que el retardo de respuesta más largo de los esclavos presentes en el bus.

Área Esclavo

Sólo puede accederse al área de configuración que se muestra a continuación cuando se selecciona «Esclavo» en el área «Tipo»:



Esclavo

Número de esclavo: 98

☐ Externo

Esta área permite introducir el número de esclavo del procesador:

El valor predeterminado es 1.

Los valores posibles están comprendidos entre 1 y 247.

La selección de **Externa** atenúa el campo **Número de esclavo** y hace que el módulo utilice el valor de la dirección del esclavo guardada (*véase página 146*) en su memoria FLASH interna.

NOTA: Si la dirección almacenada en la memoria FLASH no está dentro de la dirección del rango MODBUS, se utilizará la dirección predeterminada del esclavo: 248.

Cuando se actualiza el firmware del módulo, la dirección predeterminada del esclavo almacenada en la memoria FLASH se establece en 248. Es necesario utilizar un nuevo comando para reinicializar la dirección FLASH.

Parámetros Modbus relacionados con la transmisión

Presentación

Una vez configurado el canal de comunicación, deben introducirse los parámetros reservados de transmisión.

Puede accederse a estos parámetros desde cinco áreas:

- Área Velocidad de transmisión
- Área Retardo entre caracteres
- Área Datos
- Área Parada
- Área Paridad

Área Velocidad de transmisión

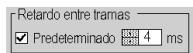
Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



Puede utilizarla para seleccionar la velocidad de transmisión de la conexión Modbus serie. La velocidad seleccionada tiene que ser coherente con los demás dispositivos. Los valores configurables son 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 y 115200 (sólo en el canal 0 en el modo RS232) bits por segundo.

Área Retardo entre tramas

Sólo puede accederse al área de configuración que se muestra a continuación en el modo RTU (está atenuada en el modo ASCII).



Retardo entre tramas es el tiempo mínimo que separa dos tramas en la recepción. Este retardo se gestiona cuando el BMX NOM 0200 (maestro o esclavo) recibe mensajes.

NOTA: El valor predeterminado varía en función de la velocidad de transmisión seleccionada.

NOTA: El retardo entre tramas deberá ser el valor predeterminado para ser compatible con Modbus. En caso de que un esclavo no sea conforme, el valor puede cambiarse y deberá ser idéntico para el maestro y para todos los esclavos del bus.

Área Datos

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



Esta zona permite introducir el tipo de codificación utilizado para comunicarse mediante la conexión Modbus serie. Este campo se establece en función de los demás dispositivos conectados al bus. Existen dos modalidades configurables:

- Modalidad RTU:
 - Los caracteres se codificarán en 8 bits.
 - El final de la trama se detecta cuando existe un silencio de 3,5 caracteres como mínimo.
 - La integridad de la trama se comprueba mediante una palabra conocida, como la suma de control CRC, que se incluye en la trama.
- Modalidad ASCII:
 - Los caracteres se codificarán en 7 bits.
 - El inicio de la trama se detecta cuando se recibe el carácter ":".
 - El final de la trama se detecta mediante un retorno de carro y un avance de línea.
 - La integridad de la trama se comprueba mediante un byte denominado suma de control LRC, que se incluye en la trama.

Área Parada

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



El área Parada permite introducir el número de bits de parada utilizados en la comunicación. Este campo se establece en función de los demás dispositivos. Los valores configurables son:

- 1 bit
- 2 bits

Área Paridad

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



A screenshot of a configuration window titled 'Paridad'. It contains three radio buttons: 'Par' (which is selected), 'Impar', and 'Ninguna'.

Esta área permite determinar si se agrega un bit de paridad, así como su tipo. Este campo se establece en función de los demás dispositivos. Los valores configurables son:

- Par
- Impar
- Ninguna

Parámetros de línea física y señales en Modbus

Introducción

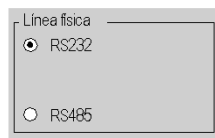
Una vez configurado el canal de comunicación, deben introducirse los parámetros de línea física y señales.

Puede accederse a estos parámetros mediante tres áreas:

- Área Línea física
- Área Señales
- Área Retardo de RTS/CTS

Área Línea física

Sólo puede accederse al área de configuración que se muestra a continuación en el canal 0 (está atenuada y configurada para RS485 en el canal 1).

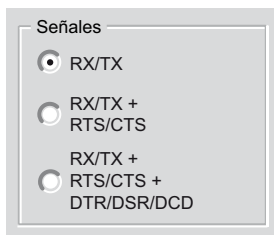


En esta área, puede elegir entre dos tipos de línea física para el puerto serie en el módulo BMX NOM 0200:

- Línea RS232
- Línea RS485

Área Señales

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



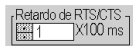
En esta área, puede seleccionar las señales admitidas por la línea física RS232:

- RX/TX
- RX/TX + RTS/CTS (señales de gestión del flujo de hardware).
- RX/TX + RTS/CTS + DTR/DSR/DCD (señales de modem)

Si se configura RS485, se atenúa el área completa y el valor predeterminado será RX/TX.

Área Retardo RTS/CTS

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



La zona de retardo RTS/CTS sólo está disponible cuando las casillas RS232 y RX/TX+RTS/CTS o RX/TX+RTS/CTS+DTR/DSR/DCD están seleccionadas. Se realiza un control del flujo de hardware de RTS/CTS.

El algoritmo de control del flujo de hardware RTS/CTS está diseñado para prevenir el búfer de recepción de desbordamiento (full duplex).

El retardo de RTS/CTS corresponde al retardo de timeout entre el aumento de RTS y el aumento de CTS. Un valor de retardo de RTS/CTS distinto de 0 también corresponde al tiempo de espera máximo entre cada transmisión de caracteres después del aumento de las señales de RTS y CTS. Si el valor se establece en 0, el UART puede quedarse atascado en un estado de espera durante un tiempo infinito hasta que CTS aumente, por lo que sólo se utiliza el valor 0 en casos concretos, como los bucles de la señal RTS en la señal CTS para comprobar que todas las conexiones funcionen correctamente.

NOTA: El valor predeterminado es 0 ms.

Cómo se establece la dirección del esclavo BMX NOM0200 MODBUS sin Unity Pro?

Condiciones y requisitos previos

La dirección FLASH se puede actualizar desde cualquier modalidad, pero sólo se tiene en cuenta cuando se realiza una modalidad de funcionamiento.

En la siguiente lista se indican las condiciones y los requisitos previos para establecer la dirección BMX NOM0200 MODBUS sin Unity Pro:

- Para utilizar la dirección FLASH, el módulo debe estar configurado:
 - En protocolo esclavo MODBUS con la casilla de verificación **FUENTE**.
 - En el protocolo maestro MODBUS o en modalidad **CHAR** y conectado al protocolo esclavo MODBUS.

Los comandos de aplicación actualizan la dirección del esclavo MODBUS en el FLASH.

En la siguiente tabla se muestran las operaciones para que los comandos de aplicación actualicen la dirección de esclavo MODBUS en el FLASH:

Paso	Acción
1	Almacenar la dirección de esclavo en %MWr.m.c.25.
2	Establecer el bit %MWr.m.c.24.7.
3	Enviar WRITE_CMD al canal del módulo.
4	Verificar el final del comando (%MWr.m.c.0.1 cae) y aceptar el comando (si %MWr.m.c.1.1 está en cero, significa que no hay errores) => FLASH se ha actualizado.
5	Realice uno de las siguientes modalidades de funcionamiento en el canal para tener en cuenta la nueva dirección: ● Descarga de aplicaciones ● Inicio en frío ● Inicio en caliente ● Intercambio en caliente ● Conmutación de protocolo (A ESCLAVO)
6	Lleve a cabo un READ_STS en el canal para verificar la dirección de esclavo en el byte %MWr.m.c.3 de mayor valor.

NOTA: En el mismo comando se pueden incrustar varias secuencias. Si no se puede ejecutar una de las secuencias, se rechazará el comando entero y no se ejecutará ninguna secuencia.

Actualice la dirección de esclavo MODBUS en el FLASH a través de la línea serie

En la siguiente tabla se muestran las operaciones para actualizar la dirección de esclavo MODBUS en el FLASH a través de la línea serie:

Paso	Acción
1	Configurar el equipo MAESTRO con el mismo parámetro de línea serie que el canal del módulo.
2	Conectar el MAESTRO al módulo punto a punto.
3	Enviar la petición 0x11 a la dirección punto a punto: número de canal 0xF8 0x11 0x01 (0 ó 1) ID de esclavo (0 . . 0xF8)
4	Verificar que la respuesta sea correcta => se actualiza el FLASH.
5	Realizar una modalidad de funcionamiento en el canal para aplicar la modificación en el paso 4.
6	Enviar una petición 0x11 para verificar la nueva dirección de esclavo: ID de esclavo 0x11 0x01

NOTA: Para evitar dañar este componente, procure no modificar el FLASH con regularidad (100.000 ciclos de escritura como máximo).

7.3

Programación de la comunicación Modbus serie

Objeto de esta sección

En esta sección se describe el proceso de programación que se utiliza al implementar la comunicación Modbus serie.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Servicios admitidos por un módulo maestro de conexión Modbus	149
Servicios admitidos por un módulo esclavo de conexión Modbus	157

Servicios admitidos por un módulo maestro de conexión Modbus

Presentación

Cuando se utiliza como maestro en una conexión Modbus, un módulo BMX NOM 0200 permite varios servicios mediante las funciones de comunicación `READ_VAR`, `WRITE_VAR` y `DATA_EXCH`.

Intercambios de datos

La lectura o escritura de variables se lleva a cabo dirigiendo las siguientes peticiones al dispositivo esclavo de destino.

Estas solicitudes utilizan las funciones de comunicación `READ_VAR` y `WRITE_VAR`.

Petición Modbus	Código de función	Función de comunicación
Lectura de bits	16#01 o 16#02	<code>READ_VAR</code>
Lectura de palabras	16#03 o 16#04	<code>READ_VAR</code>
Escritura de bits	16#0F	<code>WRITE_VAR</code>
Escritura de palabras	16#10	<code>WRITE_VAR</code>

Más generalmente, es posible enviar cualquier solicitud de Modbus a un dispositivo esclavo mediante la función de comunicación `DATA_EXCH`.

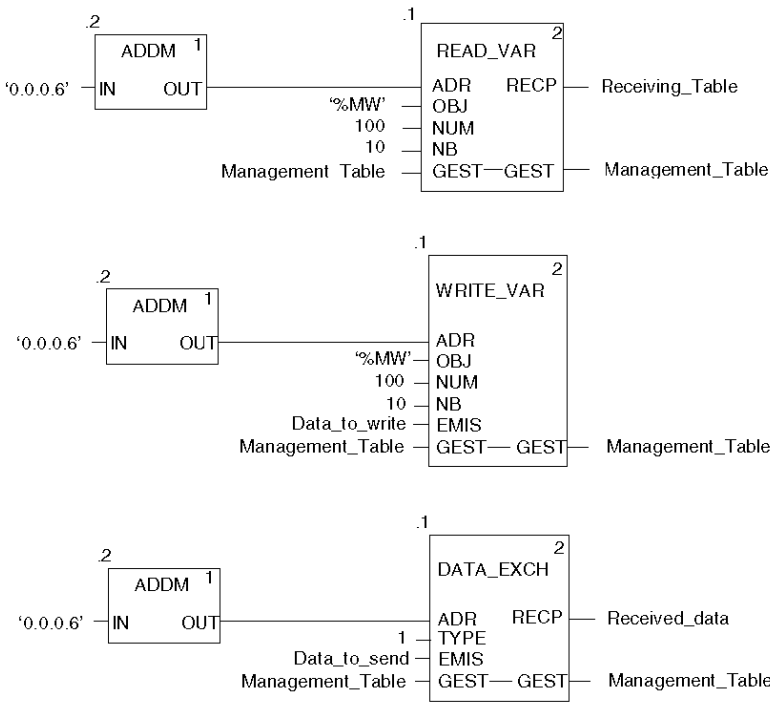
Función de comunicación `READ_VAR`, `WRITE_VAR` y `DATA_EXCH`

Se definen tres funciones específicas de comunicación para enviar y recibir datos mediante un canal de comunicación Modbus:

- `READ_VAR`: para lectura de variables.
- `WRITE_VAR`: para escritura de variables.
- `DATA_EXCH`: Para enviar solicitudes de Modbus a otro dispositivo sobre el protocolo seleccionado

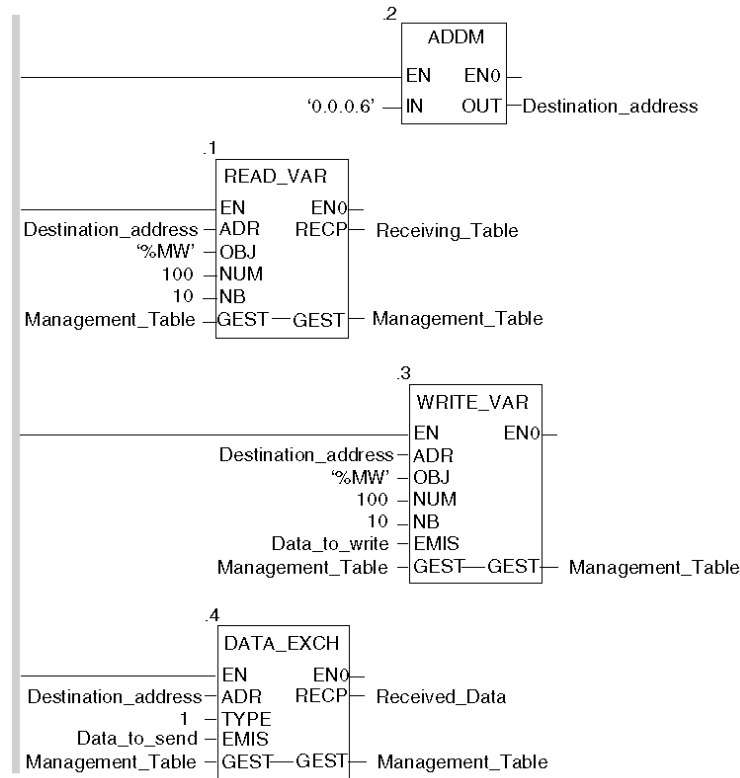
Ejemplo de programación en FBD

En el diagrama siguiente se representa un ejemplo de programación de las funciones de comunicación READ_VAR, WRITE_VAR y DATA_EXCH en lenguaje FBD:



Ejemplo de programación en Ladder

En el diagrama siguiente se representa un ejemplo de programación de las funciones de comunicación READ_VAR, WRITE_VAR y DATA_EXCH en lenguaje Ladder:



Ejemplo de programación en ST

Las siguientes líneas de código representan un ejemplo de programación de las funciones de comunicación READ_VAR, WRITE_VAR y DATA_EXCH en lenguaje ST:

```
READ_VAR(ADDM('0.0.0.6'), 'MW', 100, 10, Management_Table,  
Receiving_Table);  
  
WRITE_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, Data_to_write,  
Management_Table);  
  
DATA_EXCH(ADDM('0.0.0.6'), 1, Data_to_send, Management_Table,  
Received_data);
```

Cancelación de un intercambio

Un intercambio ejecutado por las funciones `READ_VAR`, `WRITE_VAR` y `DATA_EXCH` se puede cancelar con una de las dos formas de programación; a continuación, se presentan ambas en lenguaje ST:

- Utilización de la función `CANCEL`:

```
IF (%MW40.0) THEN
    %MW200:=SHR(%MW40,8);
    CANCEL(%MW200,%MW185);
END_IF;
```

`%MW40` corresponde al parámetro `GEST` (tabla de gestión). `%MW40.0` corresponde al bit de actividad de la función `READ_VAR` y se establece en 1 cuando la función de comunicación está activa. Si el bit se establece en 1, el programa lleva a cabo las instrucciones siguientes:

- Desplaza los bits `%MW40` un byte (8 bits) hacia la derecha y carga el byte correspondiente al número de intercambio de comunicación en la palabra `%MW200`.
- Cancela el intercambio cuyo número de intercambio está contenido en la palabra `%MW200` que utiliza la función `CANCEL`.

- Utilización del bit de cancelación de la función de comunicación:

```
IF (%MW40.0) THEN
    SET(%MW40.1);
    READ_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, %MW40:4,
    %MW10:10);
END_IF;
```

`%MW40` corresponde al parámetro `GEST` (tabla de gestión). `%MW40.0` corresponde al bit de actividad de la función `READ_VAR` y se establece en 1 cuando la función de comunicación está activa. Si este bit se ha establecido en 1, el programa establece el bit `%MW40.1`, el bit de cancelación de la función, en 1. Esto detiene la comunicación de la función `READ_VAR`.

NOTA: Al utilizar el bit de cancelación de la función de comunicación que está en la palabra de gestión de intercambios de la función (`%MW40` en este ejemplo), se debe llamar a la función (`READ_VAR` en este ejemplo) con el fin de activar la cancelación del intercambio.

NOTA: Al utilizar el bit de cancelación de la función de comunicación, es posible cancelar una comunicación desde una tabla de animación. Para ello, basta con establecer el bit de cancelación de la función en 1 (`%MW40.1` en este ejemplo) y, a continuación, iniciar de nuevo la función de comunicación.

NOTA: Este ejemplo de programación afecta a la función `READ_VAR`, pero es igualmente aplicable a las funciones `WRITE_VAR` y `DATA_EXCH`.

NOTA: La función `CANCEL` utiliza una palabra de informe en la función `CANCEL` (`%MW185` en este ejemplo).

Descripción de los parámetros de la función ADDM

En la tabla siguiente se describen los distintos parámetros de la función ADDM:

Parámetro	Tipo	Descripción
IN	STRING	Dirección de dispositivo en una conexión bus o serie. La sintaxis de la dirección es del tipo 'r.m.c.node'. La dirección se compone de los siguientes parámetros: ● r: Número de bastidor del módulo ● m: Número de ranura del módulo en el bastidor ● c: Número de canal del módulo ● nodo: Número de esclavo al que se envía la solicitud.
OUT	ARRAY [0..7] OF INT	Matriz que representa la dirección de un dispositivo. Este parámetro se puede utilizar como parámetro de entrada para varias funciones de comunicación.

Descripción de los parámetros de la función `READ_VAR`

En la tabla siguiente se describen los distintos parámetros de la función `READ_VAR`:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección de la entidad de destino proporcionada por el parámetro OUT de la función <code>ADDMM</code> .
OBJ	STRING	Tipo de objeto que se va a leer. Los tipos disponibles son: ● %M: bit interno ● %MW: palabra interna ● %I: bit de entrada externa ● %IW: palabra de entrada externa
NUM	DINT	Dirección del primer objeto que se va a leer.
NB	INT	Número de objetos consecutivos que se van a leer.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Número de intercambio ● Byte de menor valor: Bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Informe de operación ● Byte de menor valor: Informe de comunicación ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.
RECP	ARRAY [n..m] OF INT	Tabla de palabras que contiene el valor de los objetos leídos.

Descripción de los parámetros de la función `WRITE_VAR`

En la tabla siguiente se describen los distintos parámetros de la función `WRITE_VAR`:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección de la entidad de destino proporcionada por el parámetro OUT de la función <code>ADDM</code> .
OBJ	STRING	Tipo de objeto que se va a escribir. Los tipos disponibles son: ● <code>%M</code>: bit interno ● <code>%MW</code>: palabra interna Nota: <code>WRITE_VAR</code> no puede utilizarse para las variables <code>%I</code> y <code>%IW</code> .
NUM	DINT	Dirección del primer objeto que se va a escribir.
NB	INT	Número de objetos consecutivos que se van a escribir.
EMIS	ARRAY [n..m] OF INT	Tabla de palabras que contienen el valor de los objetos que van a escribirse.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Número de intercambio ● Byte de menor valor: Bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Informe de operación ● Byte de menor valor: Informe de comunicación ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.

Descripción de los parámetros de la función DATA_EXCH

En la tabla siguiente se describen los distintos parámetros de la función DATA_EXCH:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección de la entidad de destino proporcionada por el parámetro OUT de la función ADDM.
TYPE	INT	Para PLC Modicon M340, el único valor posible es el siguiente: 1 : transmisión de una matriz EMIS ; a continuación el PLC espera la recepción de una matriz RECP .
EMIS	ARRAY [n..m] OF INT	Tabla de enteros que se enviará al dispositivo destino de la solicitud. Nota: Para que la función se ejecute correctamente, es obligatorio que la longitud de los datos que se van a enviar (en bytes) se asignen a la cuarta palabra de la tabla de gestión antes de iniciar la función.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Número de intercambio. ● Byte de menor valor: Bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: Informe de operación ● Byte de menor valor: Informe de comunicación ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.
RECP	ARRAY [n..m] OF INT	Tabla de enteros que contiene los datos recibidos. Nota: El sistema escribe automáticamente tamaño de los datos recibidos (en bytes) en la cuarta palabra de la tabla de gestión.

Servicios admitidos por un módulo esclavo de conexión Modbus

Presentación

Cuando se utiliza como esclavo en una conexión Modbus, un módulo BMX NOM 0200 admite varios servicios.

Intercambios de datos

Un módulo esclavo gestiona las siguientes peticiones:

Petición Modbus	Código de función	Objeto del PLC
Lectura de n bits de salida	16#01	%M
Lectura de n palabras de salida	16#03	%MW
Escritura de n bits de salida	16#0F	%M
Escritura de n palabras de salida	16#10	%MW
Lectura/Escritura de n palabras de salida	16#17	%MW

NOTA: Leer/Escribir varios %MW

La función `WRITE` se realiza antes de la función `READ` para poder escribir y leer los mismos registros al mismo tiempo como `IOscanning`. Si el tamaño de intercambio de la función `WRITE` o `READ` está fuera del límite, el estado de retorno será "DIRECCIÓN DE DATOS NO VÁLIDA". Sin embargo, si solo falla la función `READ`, la función `WRITE` se realizará con el mismo estado.

Diagnóstico y mantenimiento

A continuación se enumeran las solicitudes de mantenimiento y diagnóstico gestionadas por un módulo BMX NOM 0200 esclavo Modbus:

Designación	Código de función/código de subfunción
Lee el estado de excepción	16#07
Opción para reiniciar comunicaciones	16#08 / 16#01
Devolver registro de diagnóstico	16#08 / 16#02
Cambiar delimitador de entrada ASCII	16#08 / 16#03
Forzar modalidad de sólo escucha	16#08 / 16#04
Limpiar contadores y registro de diagnóstico	16#08/16#0A
Devolver contador de mensajes del bus	16#08/16#0B
Devolver contador de errores de comunicaciones del bus	16#08/16#0C
Devolver contador de errores de excepciones del bus	16#08/16#0D
Devolver contador de mensajes del esclavo	16#08/16#0E
Devolver recuento de esclavos sin respuesta	16#08/16#0F
Devolver contador de confirmaciones de esclavo negativas	16#08 / 16#10
Devolver recuento de esclavos ocupados	16#08 / 16#11
Devolver recuento de desbordamiento de caracteres del bus	16#08 / 16#12
Obtener contador de eventos de comunicación	16#0B
Obtener registro de eventos de comunicación	16#0C
Comunicar identificación de esclavos	16#11
Escribir identificación de esclavos	16#11 / 16#01

7.4 Depuración de una comunicación Modbus serie

Pantalla de depuración de la comunicación Modbus serie

General

A la pantalla de depuración de la comunicación Modbus serie sólo puede accederse en la modalidad online.

Acceso a la pantalla de depuración

En la siguiente tabla se describe el procedimiento para acceder a la pantalla de depuración para la comunicación Modbus serie:

Paso	Acción
1	Acceder a la pantalla de configuración de la comunicación Modbus serie. <i>(véase página 135)</i>
2	En la pantalla que aparece, seleccionar la ficha "Depuración".

Descripción de la pantalla Depuración

La pantalla de depuración se divide en dos o tres áreas:

- Área Tipo y número de esclavo
- Área Contadores
- Área Señales (si RS232).

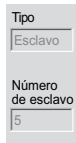
Área Tipo y número de esclavo

Si el módulo tiene la función de maestro en la conexión Modbus, está área tiene el siguiente aspecto:



Tipo
Maestro

Si el módulo tiene la función de esclavo en la conexión Modbus, está área tiene el siguiente aspecto:

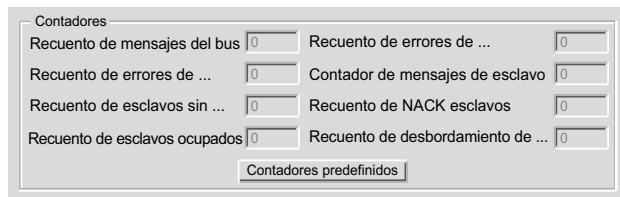


Tipo
Esclavo

Número de esclavo
5

Área Contadores

Esta área tiene un aspecto similar al siguiente:



Esta área muestra los distintos contadores de depuración.

El botón **Resetea** contadores restablece a cero todos los contadores de la modalidad de depuración.

Funcionamiento del contador

Los contadores de depuración de la comunicación Modbus serie son los siguientes:

- **Contador de mensajes del bus:** este contador indica el número de mensajes que el módulo ha detectado en la conexión serie. No incluye mensajes con un resultado negativo de suma de control CRC.
- **Contador de errores de comunicación del bus:** este contador indica el número de resultados negativos de suma de control CRC que ha contado el módulo. Si se detecta un error de caracteres (error de paridad, desborde), o el mensaje es inferior a tres bytes de longitud, el sistema que recibe los datos no puede realizar la suma de control CRC. En estos casos, el contador se incrementa en consecuencia.
- **Contador de errores de excepción de esclavo:** este contador indica el número de errores de excepción de Modbus detectados por el módulo.
- **Contador de mensajes de esclavo:** este contador indica el número de mensajes recibidos y procesados por la conexión Modbus.
- **Contador de esclavos sin respuesta:** este contador indica el número de mensajes enviados por el sistema remoto de los que no ha recibido respuesta (ni una respuesta normal ni una respuesta de excepción). Incluye además el número de mensajes recibidos en modalidad Broadcast.
- **Contador de confirmaciones de esclavo negativas:** este contador indica el número de mensajes enviados al sistema remoto para los que ha devuelto una confirmación negativa.

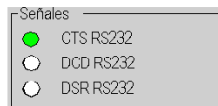
- **Contador de esclavos ocupados:** este contador indica el número de mensajes enviados al sistema remoto para los que ha devuelto un mensaje de excepción de esclavo ocupado.
- **Contador de desbordamiento de caracteres del bus:** este contador indica el número de mensajes enviados al módulo que no pueden adquirirse debido al desbordamiento de caracteres del bus. El desbordamiento ha sido provocado por uno de estos motivos:
 - Datos de tipo carácter que se transmiten por el puerto serie más rápidamente de lo que pueden almacenarse.
 - Pérdida de datos debida a un evento de hardware.

NOTA: Para todos los contadores, la cuenta comienza en el reinicio, la operación de borrado de contadores o el arranque del módulo más reciente.

Área Señales

Esta área sólo se muestra si se selecciona RS232 en la pantalla de configuración. Si se selecciona RS485 en la pantalla de configuración, esta ventana no se mostrará.

El área Señales tiene un aspecto similar al siguiente:



Esta área indica la actividad de las señales:

- **CTS RS232:** muestra la actividad de la señal CTS.
- **DCD RS232:** muestra la actividad de la señal DCD.
- **DSR RS232:** muestra la actividad de la señal DSR.

Comunicación en modalidad de caracteres para BMX NOM 0200



Objeto de esta sección

En este capítulo se presenta el proceso de implementación del software de comunicación en modalidad de caracteres para BMX NOM 0200.

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
8.1	Generalidades	164
8.2	Configuración de la comunicación en modalidad de caracteres	165
8.3	Programación de la comunicación en modalidad de caracteres	176
8.4	Depuración de la comunicación en modalidad de caracteres	183

8.1 Generalidades

Acerca de la comunicación en modalidad de caracteres

Introducción

La comunicación en modalidad de caracteres permite llevar a cabo funciones de diálogo y comunicación entre los PLC y los siguientes dispositivos:

- periféricos normales (impresora, teclado–pantalla, terminal de taller, etc.),
- periféricos especializados (lectores de código de barras, etc.) y
- calculadoras (comprobación, gestión de producción, etc.),
- dispositivos heterogéneos (comandos numéricos, controladores de velocidad variable, etc.) y
- módem externo.

ADVERTENCIA
--

PÉRDIDA DE DATOS CRÍTICOS

Los puertos de comunicación deben utilizarse exclusivamente para transferencias de datos que no sean fundamentales.

Si no se siguen estas instrucciones pueden producirse lesiones personales graves o mortales o daños en el equipo.
--

8.2 Configuración de la comunicación en modalidad de caracteres

Objeto de esta sección

En esta sección se describe el proceso de configuración que se utiliza al implementar la comunicación en modalidad de caracteres.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Pantalla de configuración de la comunicación en modalidad de caracteres	166
Funciones accesibles en modalidad de caracteres	168
Valores predeterminados para los parámetros de comunicación en modalidad de caracteres	169
Parámetros de detección de final de mensaje en modalidad de caracteres	170
Parámetros de transmisión en modalidad de caracteres	172
Parámetros de línea física y señales en modalidad de caracteres	174

Pantalla de configuración de la comunicación en modalidad de caracteres

General

En las páginas siguientes se ofrece una introducción a la pantalla de configuración de la comunicación en modalidad de caracteres.

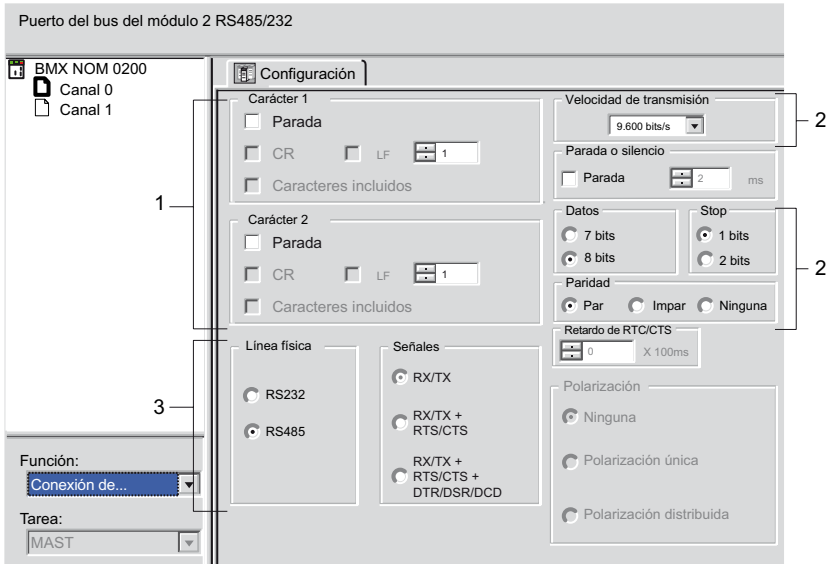
Acceso a la pantalla Configuración

En la siguiente tabla se describe el procedimiento para acceder a la pantalla de configuración para la comunicación en modalidad de caracteres:

Paso	Acción
1	Abra el subdirectorio BMX NOM 0200 del explorador de proyectos (véase página 130).
2	En la pantalla que aparece, seleccione el canal que se configurará y la función de conexión en modalidad de caracteres.

Ilustración

En la siguiente figura se muestra la pantalla de configuración predeterminada de la comunicación en modalidad de caracteres en el canal 0:



Descripción

Esta área se utiliza para configurar parámetros de canal. En la modalidad online, no se puede acceder a esta área, que estará atenuada. En la modalidad offline, puede accederse a esta área, pero es posible que no pueda accederse a algunos parámetros que, por tanto, estarán atenuados.

En la siguiente tabla se muestran las distintas áreas de las pantallas de configuración de la comunicación en modalidad de caracteres:

Tecla	Elemento	Comentario
1	Parámetros de detección de final de mensaje (véase página 170)	Puede accederse a estos parámetros mediante dos áreas: ● Parada al recibir. ● Parada en silencio.
2	Parámetros de transmisión (véase página 172)	Puede accederse a estos parámetros mediante cuatro áreas: ● Velocidad de transmisión. ● Datos. ● Bits de parada. ● Paridad.
3	Parámetros de línea física y señales (véase página 174)	Puede accederse a estos parámetros mediante cuatro áreas: ● Línea física. ● Señales. ● Retardo de RTS/CTS ● Polarización

NOTA: En este ejemplo, las áreas "Polarización" y "Retraso RTS/CTS" están atenuadas porque se ha elegido una línea física RS485 y se han elegido las señales RX/TX.

Funciones accesibles en modalidad de caracteres

Presentación

La accesibilidad a funciones para la configuración de la conexión serie de un BMX NOM 0200 mediante el protocolo modalidad de caracteres varía en función de la conexión física que se utilice.

Funciones accesibles

En la siguiente tabla se muestran las distintas funciones configurables en función del tipo de conexión serie utilizado:

Función	Conexión RS 485 (canal 0 o canal 1)	Conexión RS 232 (canal 0)
Velocidad de transmisión	X	X
Datos	● 7 bits ● 8 bits	● 7 bits ● 8 bits
Parada	● 1 bit ● 2 bits	● 1 bit ● 2 bits
Paridad	● Impar ● Par ● Ninguna	● Impar ● Par ● Ninguna
Parada al recibir	X	X
Parada en silencio	X	X
Señales RX/TX	X	X
Señales RTS/CTS	-	X
Retardo de RTS/CTS	-	X
Señales DTR/DSR/DCD	-	X
Polarización	X	-

X Función accesible

- Función inaccesible

Valores predeterminados para los parámetros de comunicación en modalidad de caracteres

Presentación

Todos los parámetros de comunicación en modalidad de caracteres tienen valores predeterminados.

Valores predeterminados

En la siguiente tabla se muestran los valores predeterminados de los parámetros de comunicación de la modalidad de caracteres en el canal 0 y en el canal 1 del módulo BMX NOM 0200:

Parámetro de configuración	Valor en el canal 0	Valor en el canal 1
Línea física	RS232	RS485
Señales	RX/TX	RX/TX (valor único)
Velocidad de transmisión	9600 bits/s	9600 bits/s
Paridad	Impar	Impar
Bits de datos	8 bits	8 bits
Bits de parada	1 bit	1 bit
Polarización	Ninguna (valor único)	Ninguna

Parámetros de detección de final de mensaje en modalidad de caracteres

Presentación

Una vez configurado el canal de comunicación, deben introducirse los parámetros de detección de final de mensaje.

Puede accederse a estos parámetros mediante dos áreas:

- Área Parada al recibir: condición de parada al recibir un carácter especial.
- Área Parada en silencio: condición de parada por silencio.

Condiciones de uso

Al seleccionar Parada en silencio se anula la selección de Parada al recibir y viceversa.

Área Parada al recibir

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:

Parada al recibir

Carácter 1

☒ Parada

☐ CR ☒ LF

☐ Caracteres incluidos

Carácter 2

☒ Parada

☒ CR ☐ LF

☐ Caracteres incluidos

Una petición de recepción puede terminarse a partir de la recepción de un carácter determinado.

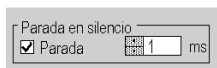
Al marcar la opción Parada, es posible configurar Parada al recibir para que se active mediante un carácter de final de mensaje determinado:

- CR: permite detectar el final del mensaje por un retorno de carro.
- LF: permite detectar el final del mensaje por un retorno de línea.
- Campo de entrada Datos: permite identificar un carácter de final de mensaje distinto de los caracteres CR o LF utilizando un valor decimal:
 - Entre 0 y 255 si los datos se codifican en 8 bits
 - Entre 0 y 127 si los datos se codifican en 7 bits
- Carácter incluido: permite incluir un carácter de final de mensaje en la tabla de recepción de la aplicación del PLC.

Pueden configurarse dos caracteres de fin de recepción. En la ventana siguiente, el fin de recepción de un mensaje se detecta mediante un carácter LF o CR.

Área Parada en silencio:

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



The image shows a small rectangular window titled "Parada en silencio". Inside the window, there is a checked checkbox labeled "Parada". To the right of the checkbox is a numeric input field containing the number "1", followed by the unit "ms".

Esta área permite detectar el final de un mensaje en la recepción mediante la ausencia de caracteres de final de mensaje durante un tiempo determinado.

Parada en silencio se valida al marcar la casilla Detener. La duración del silencio (expresada en milisegundos) se establece mediante el campo de entrada de datos.

NOTA: Los valores disponibles están comprendidos entre 1 ms y 10000 ms y varían en función de la velocidad de transmisión seleccionada.

Parámetros de transmisión en modalidad de caracteres

Presentación

Una vez configurado el canal de comunicación, deben introducirse los parámetros reservados de transmisión.

Puede accederse a estos parámetros mediante cuatro áreas:

- Área Velocidad de transmisión
- Área Datos
- Área Parada
- Área Paridad

Área Velocidad de transmisión

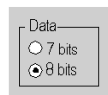
Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



Puede utilizar esta área para seleccionar la velocidad de transmisión del protocolo de modalidad de caracteres. La velocidad seleccionada tiene que ser coherente con los demás dispositivos. Los valores configurables son 300, 600, 1200, 2400, 4800, 9600, 19200, 57600 y 115200 (sólo en el canal 0 en el modo RS232) bits por segundo.

Área Datos

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



En esta área, puede especificarse el tamaño de los datos que se intercambian en la conexión.

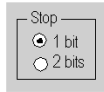
Los valores disponibles son:

- 7 bits
- 8 bits

Se recomienda ajustar el número de bits de datos en función de los dispositivos remotos.

Área Parada

Esta área tiene un aspecto similar al siguiente:



El área Parada permite introducir el número de bits de parada utilizados en la comunicación. Se recomienda ajustar el número de bits de parada en función de los dispositivos remotos.

Los valores configurables son:

- 1 bit
- 2 bits

Área Paridad

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



Esta área permite determinar si se agrega un bit de paridad, así como su tipo. Se recomienda ajustar la paridad en función de los dispositivos remotos.

Los valores configurables son:

- Par
- Impar
- Ninguna

Parámetros de línea física y señales en modalidad de caracteres

Introducción

Una vez configurado el canal de comunicación, deben introducirse los parámetros de línea física y señales. Estos parámetros son idénticos a los parámetros de línea física y señales de la comunicación Modbus (*véase página 144*), excepto una zona de polarización adicional a la que sólo se puede acceder si la línea física que se ha seleccionado es RS485.

Área Retardo RTS/CTS

Esta área de configuración aparece en la pantalla tal y como se muestra a continuación:



La zona de retardo RTS/CTS sólo está disponible cuando las casillas RS232 y RX/TX+RTS/CTS o RX/TX+RTS/CTS+DTR/DSR/DCD están seleccionadas. Se realiza un control del flujo de hardware de RTS/CTS.

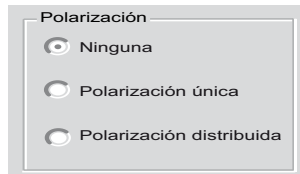
El algoritmo de control del flujo de hardware RTS/CTS está diseñado para prevenir el búfer de recepción de desbordamiento (full duplex).

El retardo de RTS/CTS corresponde al retardo de timeout entre el aumento de RTS y el aumento de CTS. Un valor de retardo de RTS/CTS distinto de 0 también corresponde al tiempo de espera máximo entre cada transmisión de caracteres después del aumento de las señales de RTS y CTS. Si el valor se establece en 0, el UART puede quedarse atascado en un estado de espera durante un tiempo infinito hasta que CTS aumente, por lo que sólo se utiliza el valor 0 en casos concretos, como los bucles de la señal RTS en la señal CTS para comprobar que todos los cables de conexión funcionen correctamente.

NOTA: El valor predeterminado es 0 ms.

Área Polarización

Se puede acceder el área de configuración que se muestra a continuación cuando se selecciona "RS485" en el área "Línea física":



Esta área proporciona la capacidad de elegir entre tres tipos de configuración para la polarización del canal:

- **Ninguna** para no utilizar la polarización en caso de que tenga su propia terminación.
- **Polarización única** para usar una impedancia baja como en las redes Modbus (el objetivo de este tipo de polarización es permitir que el maestro mantenga el estado predeterminado).
- **Polarización distribuida** para usar una polarización con alta impedancia (el objetivo de este tipo de polarización es permitir que cada dispositivo contribuya a mantener el estado predeterminado).

8.3 Programación de la comunicación en modalidad de caracteres

Funciones de comunicación en modalidad de caracteres

Funciones disponibles

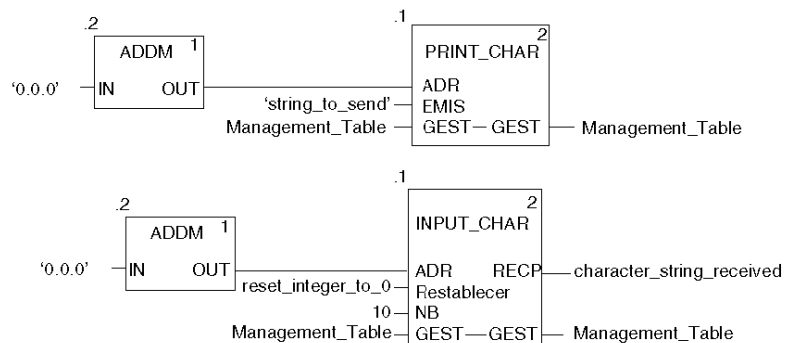
Se definen dos funciones específicas de comunicación para enviar y recibir datos mediante un canal de comunicación en modalidad de caracteres:

- `PRINT_CHAR`: envía una cadena de caracteres de un máximo de 16 x 1.024 bytes.
- `INPUT_CHAR`: lee una cadena de caracteres de un máximo de 16 x 1.024 bytes.

NOTA: Para la función `INPUT_CHAR`, se necesita un timeout configurado si el canal está configurado sin parada en silencio, para confirmar el bit de actividad de la función. Para la función `PRINT_CHAR` es recomendable, pero no necesario, configurar un timeout.

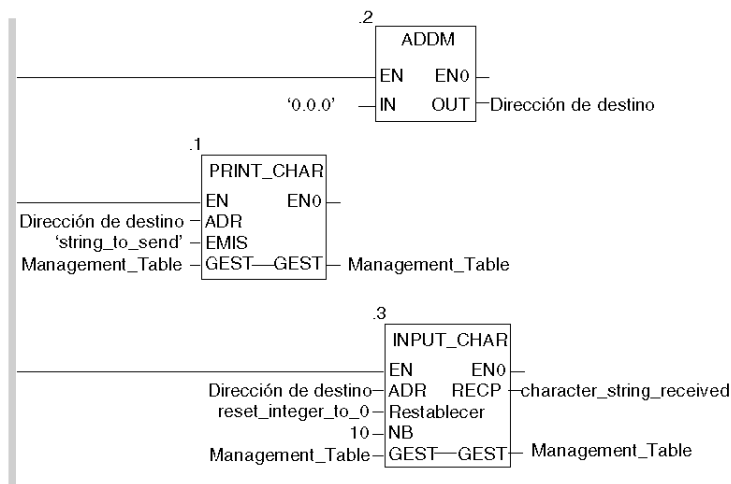
Ejemplo de programación en FBD

En el siguiente diagrama se representa un ejemplo de programación de las funciones de comunicación `PRINT_CHAR` e `INPUT_CHAR` en lenguaje FBD:



Ejemplo de programación en Ladder

En el siguiente diagrama se representa un ejemplo de programación de las funciones de comunicación PRINT_CHAR e INPUT_CHAR en lenguaje Ladder:



Ejemplo de programación en ST

Las siguientes líneas de código representan un ejemplo de programación de las funciones de comunicación PRINT_CHAR e INPUT_CHAR en lenguaje ST:

```
PRINT_CHAR(ADDM('0.1.0'), 'string_to_send',  
Management_Table);  
  
INPUT_CHAR(ADDM('0.1.0'), reset_integer_to_0, 10,  
Management_Table, character_string_received);
```

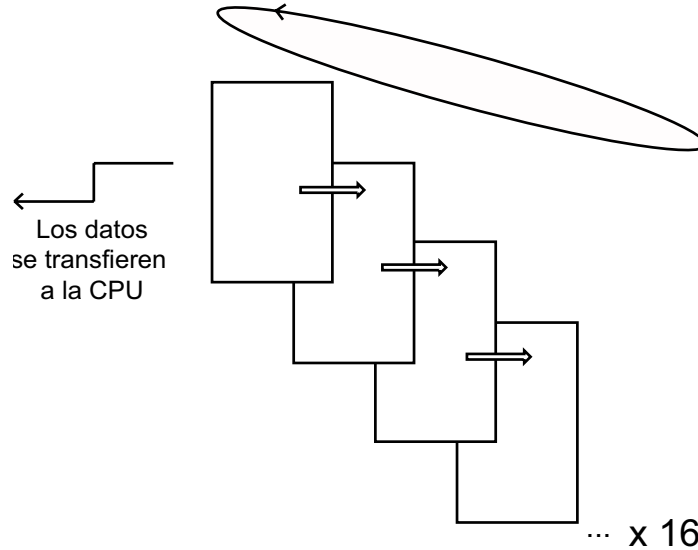
Característica de la función INPUT_CHAR

Si el parámetro de entrada Restablecer está establecido en 1, primero se restablecen todos los búferes y, a continuación, el módulo espera la recepción de datos. Se recomienda utilizar esta función para iniciar correctamente una recepción eliminando los datos antiguos que puedan permanecer en los búferes.

Mecanismo interno del módulo BMX NOM 0200

Los datos recibidos se almacenan en un conjunto cíclico de 16 búferes en serie; cada búfer contiene 1024 bits.

En la siguiente figura se representa este mecanismo:



Dos punteros independientes permiten el acceso para leer y escribir los datos.

Cada acceso de lectura de datos mediante la función INPUT CHAR borra el búfer actual y mueve el puntero de lectura al siguiente búfer.

Cuando se reciben datos:

- Si no se ha configurado ningún parámetro de detección de fin de mensaje, los datos se escriben en el búfer actual hasta que está lleno y, a continuación, el puntero de escritura pasa al siguiente búfer.
- Si se ha configurado un parámetro de detección de fin de mensaje (ya sea un silencio o un determinado carácter), cada vez que se alcanza el último, el puntero de escritura pasa al siguiente búfer, independientemente de su posición en el búfer actual.
- Cuando cualquier bit de un búfer se sobrescribe debido al comportamiento cíclico del mecanismo, se borra el búfer completo antes de la sobrescritura.

Es posible iniciar hasta 16 solicitudes de PRINT CHAR: La última se enviará en serie con un silencio entre cada PRINT CHAR.

Cancelación de un intercambio

Existen dos maneras de programar que habilitan la cancelación de un intercambio ejecutado mediante las funciones `PRINT_CHAR` y `INPUT_CHAR`. Estas dos funciones se presentan a continuación en lenguaje ST:

- Utilización de la función `CANCEL`:

```
IF (%MW40.0) THEN
    %MW200:=SHR(%MW40,8);
    CANCEL(%MW200,%MW185);
END_IF;
```

`%MW40` corresponde al parámetro `GEST` (tabla de gestión). `%MW40.0` corresponde al bit de actividad de la función `PRINT_CHAR` y se establece en 1 cuando la función de comunicación está activa. Si el bit se establece en 1, el programa lleva a cabo las instrucciones siguientes:

- Desplaza los bits `%MW40` un byte (8 bits) hacia la derecha y carga el byte correspondiente al número de intercambio de comunicación en la palabra `%MW200`.
 - Cancela el intercambio cuyo número de intercambio está contenido en la palabra `%MW200` que utiliza la función `CANCEL`.
- Utilización del bit de cancelación de la función de comunicación:

```
IF (%MW40.0) THEN
    SET(%MW40.1);
    PRINT_CHAR(ADDM('0.1.0'), 'string_to_send', %MW40:4);
END_IF;
```

`%MW40` corresponde al parámetro `GEST` (tabla de gestión). `%MW40.0` corresponde al bit de actividad de la función `PRINT_CHAR` y se establece en 1 cuando la función de comunicación está activa. Si este bit se ha establecido en 1, el programa establece el bit `%MW40.1`, el bit de cancelación de la función, en 1. Esto detiene la comunicación de la función `PRINT_CHAR`.

NOTA: Al utilizar el bit de cancelación de la función de comunicación, debe llamarse a la función para habilitar el bit de cancelación contenido en la palabra de gestión de intercambios de la función (`%MW40` en este ejemplo).

NOTA: Al utilizar el bit de cancelación de la función de comunicación, es posible cancelar una comunicación desde una tabla de animación. Para ello, basta con establecer el bit de cancelación de la función en 1 (`%MW40.1` en este ejemplo).

NOTA: Este ejemplo de programación afecta a la función `PRINT_CHAR`, pero es igualmente aplicable a la función `INPUT_CHAR`.

NOTA: La función `CANCEL` utiliza una palabra de informe en la función `CANCEL` (`%MW185` en este ejemplo).

Descripción de los parámetros de la función ADDM

En la tabla siguiente se describen los distintos parámetros de la función `ADDM`:

Parámetro	Tipo	Descripción
IN	STRING	Dirección de dispositivo en una conexión bus o serie. La sintaxis de la dirección es del tipo 'r.m.c.node'. La dirección se compone de los siguientes parámetros: ● r: número de bastidor del sistema de destino, siempre = 0. ● m: número de slot del sistema de destino del bastidor, siempre = 0. ● c: número de canal, siempre = 0, ya que la conexión serie de un sistema remoto es siempre el canal 0. ● nodo: campo opcional que puede ser <code>SYS</code> o estar vacío.
OUT	ARRAY [0..7] OF INT	Tabla que muestra la dirección de un dispositivo. Este parámetro se puede utilizar como parámetro de entrada para varias funciones de comunicación.

Descripción de los parámetros de la función PRINT_CHAR

En la tabla siguiente se describen los distintos parámetros de la función PRINT_CHAR:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección del canal del modo de caracteres que reciben mensajes proporcionada por el parámetro OUT de la función ADDM.
EMIS	STRING	Cadena de caracteres para enviar.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: número de intercambio ● Byte de menor valor: bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: informe de funcionamiento. ● Byte de menor valor: informe de comunicación. ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el usuario, que define la duración del intercambio. ● Si la longitud de este parámetro se establece en 0, el sistema envía toda la cadena. ● Si la longitud de este parámetro es superior a la longitud de la cadena, se devuelve el error 16#0A (Tamaño del búfer de emisión insuficiente) en la segunda palabra de gestión y no se envía ningún carácter.

Descripción de los parámetros de la función INPUT_CHAR

En la tabla siguiente se describen los distintos parámetros de la función INPUT_CHAR:

Parámetro	Tipo	Descripción
ADR	ARRAY [0..7] OF INT	Dirección del canal del modo de caracteres que reciben mensajes proporcionada por el parámetro OUT de la función ADDM.
Restablecer	INT	Este parámetro puede adoptar dos valores: ● Valor 1: restablece en 0 la memoria de recepción del módulo. ● Valor 0: no restablece en 0 la memoria de recepción del módulo.
NB	INT	Longitud de la cadena de caracteres que se va a recibir.
GEST	ARRAY [0..3] OF INT	Tabla de gestión de intercambios que se compone de las siguientes palabras: ● Palabra de rango 1: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: número de intercambio ● Byte de menor valor: bit de actividad (rango 0) y bit de cancelación (rango 1). ● Palabra de rango 2: Palabra administrada por el sistema que se compone de dos bytes: ● Byte de mayor valor: informe de funcionamiento. ● Byte de menor valor: informe de comunicación. ● Palabra de rango 3: Palabra gestionada por el usuario, que define el tiempo de respuesta máximo con una base de tiempo de 100 ms. ● Palabra de rango 4: Palabra gestionada por el sistema, que define la duración del intercambio.
RECP	STRING	Cadena de caracteres recibida. Esta cadena se guarda en una cadena de caracteres.

8.4 Depuración de la comunicación en modalidad de caracteres

Pantalla de depuración de la comunicación en modalidad de caracteres

General

A la pantalla de depuración en modalidad de caracteres sólo puede accederse en la modalidad online.

Acceso a la pantalla de depuración

En la tabla siguiente se describe el procedimiento para acceder a la pantalla de depuración para la comunicación en modalidad de caracteres:

Paso	Acción
1	Acceder a la pantalla de configuración de la comunicación en modalidad de caracteres. (véase <i>página 166</i>)
2	En la pantalla que aparece, seleccionar la ficha "Depuración".

Descripción de la pantalla Depuración

La pantalla de depuración se compone de un área Error y un área Señales (si RS232).

Área Error

El área Error tiene un aspecto similar al siguiente:

Esta área indica el número de interrupciones de comunicación computados por el módulo:

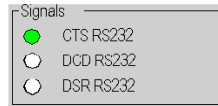
- **Al transmitir:** corresponde al número de interrupciones durante la transmisión (imagen de la palabra `%MW4`).
- **Al recibir:** corresponde al número de interrupciones durante la recepción (imagen de la palabra `%MW5`).

El botón **Resetear contadores** restablece a cero ambos contadores.

Área Señales

Esta zona sólo se muestra si se selecciona RS232 en la pantalla de configuración. Si se selecciona RS232 en la pantalla de configuración, esta ventana no se mostrará.

El área Señales tiene un aspecto similar al siguiente:



Esta área indica la actividad de las señales:

- **CTS RS232:** muestra la actividad de la señal CTS.
- **DCD RS232:** muestra la actividad de la señal DCD.
- **DSR RS232:** muestra la actividad de la señal DSR.

Diagnóstico del módulo BMX NOM 0200



9.1 Diagnóstico del módulo BMX NOM 0200

Objeto de esta sección

En esta sección se describe el diagnóstico de la implementación del módulo de comunicación BMX NOM 0200.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Diagnóstico del módulo BMX NOM 0200	186
Diagnóstico detallado por canal de comunicaciones	188

Diagnóstico del módulo BMX NOM 0200

Introducción

La función de diagnóstico del módulo muestra, si existen, las anomalías que se producen, clasificadas según su categoría:

- **Error interno detectado:**
 - suceso del módulo
- **Evento externo:**
 - control de cableado (conductor interrumpido, sobrecarga o cortocircuito)
- **Otras anomalías:**
 - canal inoperativo
 - anomalía de configuración
 - módulo ausente o desconectado

Algunos indicadores LED cambian a rojo para indicar un error de un módulo detectado, como por ejemplo:

- En el editor de configuración de nivel de bastidor:
 - el indicador LED de número del bastidor
 - el indicador LED de número de ranura del módulo en el bastidor
- En el editor de configuración de nivel de módulo:
 - los indicadores LED **Err** y **E/S**, según el tipo de fallo detectado
 - El indicador LED **Canal** del campo **Canal**.

Acceder a la pantalla de diagnóstico del módulo

La tabla siguiente presenta el procedimiento para acceder a la pantalla de diagnóstico del módulo.

Paso	Acción
1	Acceder a la pantalla de depuración del módulo.
2	Hacer clic en la referencia del módulo que se encuentra en el área de canal y seleccionar la ficha Fallo. Resultado: aparece la lista de errores detectados del módulo.
	Nota: No es posible acceder a la pantalla de diagnóstico del módulo si se detecta un error de configuración, un error grave en la alimentación o un error por ausencia de un módulo. En la pantalla, aparecerá el mensaje siguiente: "Falta el módulo o bien es distinto del que está configurado en esta posición."

Lista de errores del módulo detectados

En la siguiente tabla de resumen se muestran los distintos errores detectados para un módulo de comunicación:

Clasificación de errores detectados	Objetos de lenguaje
Fallo interno: Error del módulo detectado	%MW_r.m.MOD.2.0
Fallo externo: Bloque de terminales	%MW_r.m.MOD.2.2
Otros fallos: - Canal(es) defectuoso(s) - Fallo de configuración del hardware - Módulo ausente o desconectado	%MW_r.m.MOD.2.1 %MW_r.m.MOD.2.5 %MW_r.m.MOD.2.6

Diagnóstico detallado por canal de comunicaciones

Introducción

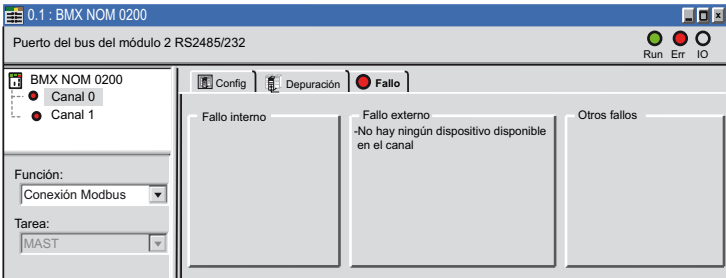
La función de diagnóstico de canal muestra los fallos detectados cuando se producen, clasificados según su categoría:

- **Error interno detectado**
 - Autoverificaciones en curso
- **Eventos externos**
 - Dispositivo ausente
 - Dispositivo no operativo
 - Timeout de de acciones mediante conexión serie
- **Otros errores detectados**
 - Error de la herramienta de línea
 - error de configuración
 - pérdida de comunicación
 - error de aplicación

Un error de canal detectado se indica en la ficha **Depuración** cuando el indicador LED  , situado en la columna **Error**, se ilumina en rojo.

Acceso a la pantalla de diagnóstico del canal

La tabla siguiente presenta el procedimiento para acceder a la pantalla de diagnóstico del canal.

Paso	Acción
1	Acceder a la pantalla de depuración del módulo.
2	Haga clic en el botón  situado en la columna Error para el canal inoperativo. Resultado: aparece la lista de errores del canal detectados.  Nota: A la información de diagnóstico del canal también se puede acceder mediante el programa (instrucción READ_STS).

Lista de errores del canal detectados

En la siguiente tabla de resumen se muestran los distintos errores detectados para una conexión serie configurada:

Clasificación de errores detectados	Objetos de lenguaje
Fallo interno: ● Autoverificaciones en curso	● %MW _r .m.c.2.4
Fallo externo: ● No hay ningún dispositivo disponible en el canal ● Error de dispositivo ● Error de timeout (CTS)	● %MW_r.m.c.2.0 ● %MW_r.m.c.2.1 ● %MW_r.m.c.2.3
Otros fallos: ● Error de la herramienta de línea ● Fallo de configuración del hardware ● Problema en la comunicación con el PLC ● Error de aplicación	● %MW_r.m.c.2.2 ● %MW_r.m.c.2.5 ● %MW_r.m.c.2.6 ● %MW_r.m.c.2.7

Objetos de lenguaje de la comunicación en modalidad de caracteres y Modbus

10

Objeto

En este capítulo se describen los objetos de lenguaje asociados a la comunicación en modalidad de caracteres y Modbus, así como las distintas maneras de utilizarlos.

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
10.1	Objetos de lenguaje e IODDT de las comunicaciones en modalidad de caracteres y Modbus	192
10.2	Objetos de lenguaje general e IODDT para todos los protocolos de comunicación	200
10.3	Objetos de lenguaje e IODDT asociados a la comunicación Modbus	204
10.4	Objetos de lenguaje e IODDT asociados a la comunicación en modalidad de caracteres	214
10.5	El IODDT tipo T_GEN_MOD es aplicable a todos los módulos	222

10.1 Objetos de lenguaje e IODDT de las comunicaciones en modalidad de caracteres y Modbus

Objeto

En este capítulo se ofrece una descripción general de las generalidades relativas a los IODDT y los objetos de lenguaje para las comunicaciones en modalidad de caracteres y Modbus.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Introducción a los objetos de lenguaje para comunicaciones en modalidad de caracteres y Modbus	193
Intercambio implícito de objetos de lenguaje asociados a la función específica de la aplicación	194
Objetos de lenguaje de intercambio explícito asociados a la función específica de aplicaciones	195
Gestión de intercambios y del informe con objetos explícitos	197

Introducción a los objetos de lenguaje para comunicaciones en modalidad de caracteres y Modbus

General

Los IODDT están predefinidos por el fabricante. Los IODDT contienen objetos de lenguaje de entradas/salidas que pertenecen al canal de un módulo de específico de la aplicación.

Las comunicaciones en modalidad de caracteres y Modbus tienen tres IODDT asociados:

- `T_COM_STS_GEN`, que se aplica a todos los protocolos de comunicaciones.
- `T_COM_MB_BMX`, que es específico de la comunicación Modbus.
- `T_COM_CHAR_BMX`, que es específico de la comunicación en modalidad de caracteres.

NOTA: Las variables de IODDT se pueden crear de dos formas diferentes:

- Mediante la ficha Objetos de E/S (véase *Unity Pro, Modalidades de funcionamiento*,).
- Mediante el Editor de datos (véase *Unity Pro, Modalidades de funcionamiento*,).

Tipos de objetos de lenguaje

En cada IODDT encontramos un conjunto de objetos de lenguaje que nos permiten controlarlos y comprobar que funcionan correctamente.

Existen dos tipos de objetos de lenguaje:

- **Objetos de intercambios implícitos:** Estos objetos se intercambian automáticamente en cada revolución cíclica de la tarea asociada al módulo.
- **Objetos de intercambios explícitos:** Estos objetos se intercambian a petición de la aplicación, mediante las instrucciones de intercambio explícito.

Los intercambios implícitos son aquellos relativos al estado de los procesadores, las señales de comunicación, los esclavos, etc.

Los intercambios explícitos se utilizan para definir los ajustes de procesador y realizar diagnósticos.

Intercambio implícito de objetos de lenguaje asociados a la función específica de la aplicación

Presentación

El uso de una interfaz específica de la aplicación integrada o la agregación de un módulo enriquecen automáticamente la aplicación de objetos de lenguaje utilizada para programar la interfaz o el módulo.

Estos objetos corresponden a las imágenes de las entradas/salidas y a los datos de software del módulo o de la interfaz específica de la aplicación integrada.

Notas

Las entradas del módulo (%I y %IW) se actualizan en la memoria del PLC al comienzo de la tarea o cuando el PLC se encuentra en modalidad RUN o STOP.

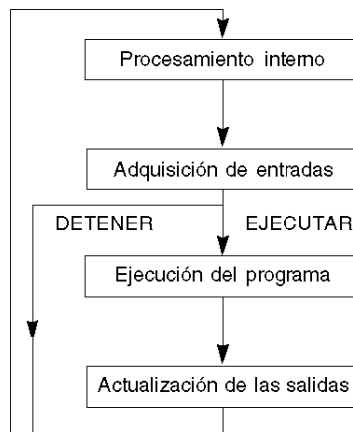
Las salidas (%Q y %QW) se actualizan al final de la tarea, sólo cuando el PLC se encuentra en modalidad RUN.

NOTA: Cuando la tarea está en modalidad STOP, según la configuración elegida, es posible que se produzca una de estas situaciones:

- Las salidas se actualizan en posición de retorno (modo de retorno).
- Las salidas se mantienen en su último valor (modo de mantenimiento).

Ilustración

En el gráfico siguiente se muestra el ciclo de funcionamiento de una tarea del PLC (ejecución cíclica):



Objetos de lenguaje de intercambio explícito asociados a la función específica de aplicaciones

Presentación

Intercambios explícitos son los que se realizan a petición del programa de usuario, mediante las siguientes instrucciones:

- **READ_STS** (véase *Unity Pro, Gestión de E/S, Biblioteca de bloques*): lectura de palabras de estado
- **WRITE_CMD** (véase *Unity Pro, Gestión de E/S, Biblioteca de bloques*): escritura de palabras de comando

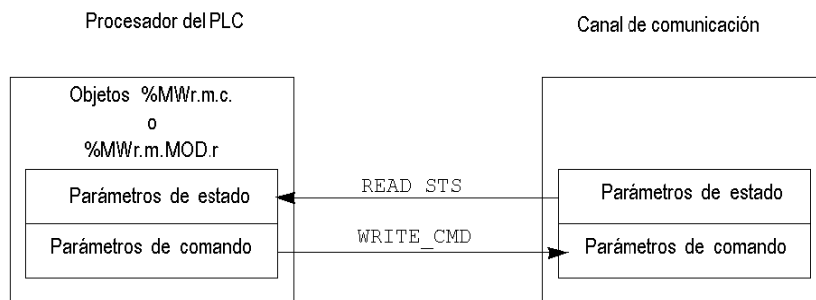
Estos intercambios se aplican a un conjunto de objetos `%MW` del mismo tipo (estado, comandos o parámetros) que pertenece a un canal.

NOTA: Estos objetos proporcionan información acerca del procesador o el módulo, que puede usarse para sus comandos (por ejemplo: comando de conmutación) y definir sus modalidades de funcionamiento (almacenamiento y restauración de los parámetros de ajuste de la aplicación).

NOTA: Las instrucciones **READ_STS** y **WRITE_CMD** se ejecutan al mismo tiempo que la tarea que las llama y siempre correctamente. El resultado de estas instrucciones queda disponible automáticamente después de su ejecución.

Principios generales de uso de las instrucciones explícitas

El siguiente diagrama muestra los distintos tipos de intercambios explícitos que pueden efectuarse entre el procesador y el canal de comunicación:



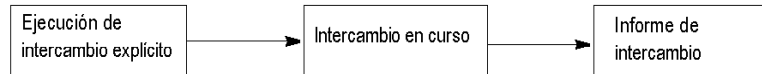
Gestión de intercambios

Durante un intercambio explícito, es necesario comprobar su rendimiento para tener en cuenta los datos únicamente cuando el intercambio se haya ejecutado correctamente.

Para ello, hay dos tipos de información disponibles:

- Información relativa al intercambio en curso (véase *Unity Pro, Gestión de E/S, Biblioteca de bloques*).
- El informe de intercambio (véase *Unity Pro, Gestión de E/S, Biblioteca de bloques*).

El siguiente diagrama muestra el principio de gestión de un intercambio:



NOTA: Para evitar varios intercambios explícitos simultáneos para el mismo canal, es necesario comprobar el valor de la palabra EXCH_STS (%MW_{r.m.c.} 0) del IODDT asociado al canal antes de llamar a cualquier EF utilizando este canal.

Gestión de intercambios y del informe con objetos explícitos

Introducción

Al intercambiar los datos entre la memoria del autómatas y el módulo, la validación puede requerir varios ciclos de la tarea.

Para gestionar los intercambios, todos los IODDT disponen de dos palabras:

- EXCH_STS (%MWr.m.c.0): intercambio en curso.
- EXCH_RPT (%MWr.m.c.1): informe.

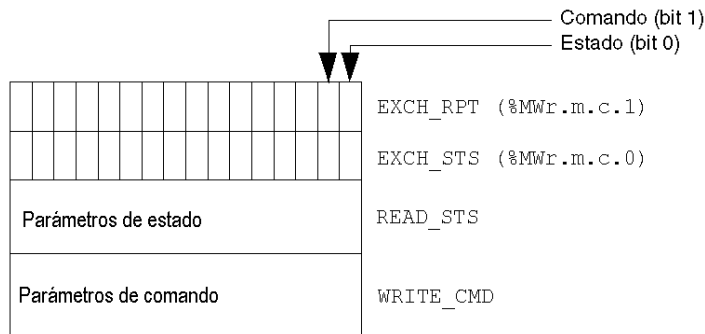
NOTA:

En función de la localización del módulo, la aplicación no detectará la gestión de los intercambios explícitos (%MW0.0.MOD.0.0, por ejemplo):

- Para módulos en bastidor, los intercambios explícitos se realizan inmediatamente en el bus PLC local y se terminan antes del final de la tarea de ejecución, por lo que READ_STS; por ejemplo, siempre termina cuando la aplicación comprueba el bit %MW0.0.mod.0.0
- En el bus remoto (Fipio, por ejemplo), los intercambios explícitos no son sincrónicos con la tarea de ejecución, de forma que la aplicación puede realizar la detección.

Figura

La siguiente ilustración muestra los distintos bits significativos para la gestión de intercambios:



Descripción de los bits significativos

Cada bit de las palabras EXCH_STS (%MWr.m.c.0) y EXCH_RPT (%MWr.m.c.1) está asociado a un tipo de parámetro:

- Los bits de rango 0 están asociados a los parámetros de estado:
 - El bit STS_IN_PROGR (%MWr.m.c.0.0) indica si hay en curso una solicitud de lectura para las palabras de estado.
 - El bit STS_ERR (%MWr.m.c.1.0) especifica si el canal del módulo acepta o no la solicitud de lectura para las palabras de estado.
- Los bits de rango 1 están asociados a los parámetros de comando:
 - El bit CMD_IN_PROGR (%MWr.m.c.0.1) indica si se están enviando los parámetros de comando al canal del módulo.
 - El bit CMD_ERR (%MWr.m.c.1.1) indica si el canal del módulo ha aceptado o no los parámetros de comando.

NOTA: r corresponde al número de bastidor, y m a la posición del módulo en el bastidor, mientras que c representa el número de canal en el módulo.

NOTA: También existen palabras de intercambio e informe en el módulo EXCH_STS (%MWr.m.MOD.0) y EXCH_RPT (%MWr.m.MOD.1) según los según los IODDT de tipo T_GEN_MOD.

Indicadores de ejecución de intercambio explícito: EXCH_STS

En la tabla siguiente se muestran los bits de control de intercambio explícito de la palabra EXCH_STS (%MWr.m.c.0):

Símbolo estándar	tipo	Acceso	Significado	Dirección
STS_IN_PROGR	BOOL	R	Lectura de las palabras de estado del canal actual	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Intercambio de parámetros de comando en curso	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Intercambio de parámetros de ajuste en curso	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Reconfiguración del módulo en curso	%MWr.m.c.0.15

NOTA: Si el módulo no está presente o está desconectado, los intercambios que utilizan objetos explícitos (READ_STS, por ejemplo) no se envían al módulo (STS_IN_PROG (%MWr.m.c.0.0) = 0), pero se actualizan las palabras.

Informe de intercambios explícitos: EXCH_RPT

En la tabla siguiente se muestran los bits de informe de palabra EXCH_RPT (%MWr.m.c.1):

Símbolo estándar	tipo	Acceso	Significado	Dirección
STS_ERR	BOOL	R	Error detectado de lectura de las palabras de estado del canal (1 = fallo detectado)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error detectado durante un intercambio de parámetros de comando. (1 = fallo detectado)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Interrupciones durante un intercambio de parámetros de ajuste (1 = fallo detectado)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Interrupciones durante la reconfiguración del canal (1 = fallo detectado)	%MWr.m.c.1.15

10.2 **Objetos de lenguaje general e IODDT para todos los protocolos de comunicación**

Objeto

En esta sección se presentan los objetos de lenguaje general e IODDT que se aplican a todos los protocolos de comunicación.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Detalles de los objetos de intercambio implícito de IODDT de tipo T_COM_STS_GEN	201
Detalles de los objetos de intercambio explícito de IODDT de tipo T_COM_STS_GEN	202

Detalles de los objetos de intercambio implícito de IODDT de tipo T_COM_STS_GEN

Presentación

En la siguiente tabla se presentan los objetos de intercambio implícito de IODDT de tipo T_COM_STS_GEN que se aplican a todos los protocolos de comunicación excepto Fipio.

Bit de error

En la siguiente tabla se presenta el significado de los bits de error CH_ERROR (%lr.m.c.ERR):

Símbolo estándar	Tipo	Acceso	Significado	Dirección
CH_ERROR	EBOOL	L	Bit de error del canal de comunicación.	%lr.m.c.ERR

Detalles de los objetos de intercambio explícito de IODDT de tipo T_COM_STS_GEN

Presentación

En esta sección se presentan los objetos de intercambio explícito de IODDT de tipo T_COM_STS_GEN que se aplican a todos los protocolos de comunicación excepto Fipio. Agrupa los objetos de tipo palabra cuyos bits tienen un significado determinado. Estos objetos se describen de forma detallada a continuación.

En este apartado, la variable IODDT_VAR1 es de tipo T_COM_STS_GEN.

Observaciones

Por lo general, se ofrece el significado de los bits para el estado 1 del bit. En determinados casos, se explica el estado de cada bit.

No se utilizan todos los bits.

Indicadores de ejecución de intercambio explícito: EXCH_STS

En la siguiente tabla se muestra el significado de los bits de control de intercambio del canal EXCH_STS (%MW_{r.m.c.0}):

Símbolo estándar	Tipo	Acceso	Significado	Dirección
STS_IN_PROGR	BOOL	L	Lectura de palabras de estado del canal en curso.	%MW _{r.m.c.0.0}
CMD_IN_PROGR	BOOL	L	Intercambio de parámetros de comando en curso.	%MW _{r.m.c.0.1}

Informe de los intercambios explícitos: EXCH_RPT

En la siguiente tabla se presenta el significado de los bits de informe de intercambio EXCH_RPT (%MW_{r.m.c.1}):

Símbolo estándar	Tipo	Acceso	Significado	Dirección
STS_ERR	BOOL	L	Error de lectura de palabras de estado del canal detectado.	%MW _{r.m.c.1.0}
CMD_ERR	BOOL	L	Error detectado durante un intercambio de parámetros de comando.	%MW _{r.m.c.1.1}

Fallos estándar del canal: CH_FLT

En la siguiente tabla se muestra el significado de los bits de la palabra de estado CH_FLT (%MWr.m.c.2):

Símbolo estándar	Tipo	Acceso	Significado	Dirección
NO_DEVICE	BOOL	L	No hay ningún dispositivo funcionando en el canal.	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	L	Un dispositivo del canal no está operativo.	%MWr.m.c.2.1
BLK	BOOL	L	El bloque de terminales no está conectado.	%MWr.m.c.2.2
TO_ERR	BOOL	L	Temporización no ajustable (análisis necesario).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	L	Detectado error interno o autoprueba del canal.	%MWr.m.c.2.4
CONF_FLT	BOOL	L	Configuraciones de hardware y software diferentes.	%MWr.m.c.2.5
COM_FLT	BOOL	L	Análisis de comunicación necesario con el canal.	%MWr.m.c.2.6
APPLI_FLT	BOOL	L	Error de aplicación detectado (ajuste o configuración).	%MWr.m.c.2.7

La lectura se lleva a cabo mediante la instrucción READ_STS (IODDT_VAR1).

10.3 Objetos de lenguaje e IODDT asociados a la comunicación Modbus

Objeto

En esta sección se presentan los objetos de lenguaje e IODDT asociados a la comunicación Modbus.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Detalles relativos a los objetos de lenguaje de intercambio explícito para una función Modbus	205
Detalles de los objetos de intercambio implícito de IODDT de tipo T_COM_MB_BMX y T_COM_MB_BMX_CONF_EXT	206
Detalles de los objetos de intercambio explícito de IODDT de tipo T_COM_MB_BMX y T_COM_MB_BMX_CONF_EXT	208
Detalles de los objetos de lenguaje asociados a la modalidad Modbus de configuración	212

Detalles relativos a los objetos de lenguaje de intercambio explícito para una función Modbus

Presentación

En la siguiente tabla se muestran los objetos de lenguaje para la comunicación Modbus en modalidad de maestro o esclavo. Estos objetos no se encuentran integrados en los IODDT.

Lista de objetos de intercambio explícito en modalidad de maestro o esclavo

En la siguiente tabla se muestran los objetos de intercambio explícito:

Dirección	Tipo	Acceso	Significado
%MWr.m.c.4	INT	L	Número de respuestas recibidas correctamente.
%MWr.m.c.5	INT	L	Número de respuestas recibidas con error CRC.
%MWr.m.c.6	INT	L	Número de respuestas recibidas con código de excepción en modalidad de maestro.
%MWr.m.c.7	INT	L	Número de mensajes enviados en modalidad de esclavo.
%MWr.m.c.8	INT	L	Número de mensajes enviados sin respuesta en modalidad de esclavo.
%MWr.m.c.9	INT	L	Número de respuestas recibidas con confirmación negativa.
%MWr.m.c.10	INT	L	Número de mensajes repetidos en modalidad de esclavo.
%MWr.m.c.11	INT	L	Número de errores de caracteres detectados.
%MWr.m.c.24.0	BOOL	L/E	Restablecimiento de los contadores de errores detectados.

Detalles de los objetos de intercambio implícito de IODDT de tipo T_COM_MB_BMX y T_COM_MB_BMX_CONF_EXT

Presentación

En las siguientes tablas se muestran los objetos de intercambio implícito de IODDT de tipo T_COM_MB_BMX y T_COM_MB_BMX_CONF_EXT que se aplican a la comunicación serie Modbus. Difieren en lo respectivo a la **disponibilidad de los objetos de configuración** (véase página 211).

Bit CH_ERROR

En la siguiente tabla se presenta el significado del bit de error CH_ERROR (%Ir.m.c.ERR):

Símbolo estándar	Tipo	Acceso	Significado	Dirección
CH_ERROR	EBOOL	L	El canal de comunicación ha detectado un bit de error.	%Ir.m.c.ERR

Objeto de palabra en modalidad de maestro Modbus

En la siguiente tabla se muestra el significado del bit de la palabra INPUT_SIGNALS (%IWrr.m.c.0):

Símbolo estándar	Tipo	Acceso	Significado	Dirección
DCD	BOOL	L	Señal de detección de portadora de datos RS232 (sólo aplicable al módulo BMX NOM 0200)	%IWrr.m.c.0.0
CTS	BOOL	L	Señal de listo para enviar RS232	%IWrr.m.c.0.2
DSR	BOOL	L	Señal de paquete de datos preparado RS232 (sólo aplicable al módulo BMX NOM 0200)	%IWrr.m.c.0.3

NOTA: Cuando CTS es verde en la unidad P, significa que %IWrr.m.c.0.0 está en 1 y que la tensión de esta señal es positiva. También es aplicable a DCD y DSR.

Objeto de palabra en modalidad de esclavo Modbus

Los objetos de lenguaje son idénticos a los de la función de maestro Modbus. Sólo se diferencian los objetos de la siguiente tabla.

En la siguiente tabla se muestra el significado del bit de la palabra `INPUT_SIGNALS` (`%IWr.m.c.0`):

Símbolo estándar	Tipo	Acceso	Significado	Dirección
LISTEN_ONLY	BOOL	L	Modalidad de sólo escucha	<code>%IW_r.m.c.0.8</code>

Detalles de los objetos de intercambio explícito de IODDT de tipo T_COM_MB_BMX y T_COM_MB_BMX_CONF_EXT

Introducción

En este apartado se presentan los objetos de intercambio explícito de IODDT de tipo T_COM_MB_BMX y T_COM_MB_BMX_CONF_EXT que se aplican a la comunicación serie Modbus y difiere en términos de **disponibilidad de los objetos de configuración** (véase página 211). Agrupa los objetos de tipo palabra, cuyos bits tienen un significado particular. Estos objetos se describen de forma detallada a continuación.

En este apartado, la variable IODDT_VAR1 es de tipo T_COM_STS_GEN.

Observaciones

Por lo general, se ofrece el significado de los bits para el estado 1 del bit. En determinados casos, se explica el estado de cada bit.

No se utilizan todos los bits.

Indicadores de ejecución de intercambio explícito: EXCH_STS

En la siguiente tabla se muestra el significado de los bits de control de intercambio del canal EXCH_STS (%MWr.m.c.0):

Símbolo estándar	tipo	Acceso	Significado	Dirección
STS_IN_PROGR	BOOL	R	Lectura de las palabras de estado del canal en curso.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Intercambio de parámetros de comando en curso.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Intercambio de parámetros de ajuste en curso (no aplicable al módulo BMX NOM 0200).	%MWr.m.c.0.2

Informe de intercambios explícitos: EXCH_RPT

En la siguiente tabla se presenta los distintos significados de los bits de informe de intercambio EXCH_RPT (%MW_{r.m.c.1}):

Símbolo estándar	tipo	Acceso	Significado	Dirección
STS_ERR	BOOL	R	Error de lectura de palabras de estado del canal detectado.	%MW _{r.m.c.1.0}
CMD_ERR	BOOL	R	Anomalía durante un intercambio de parámetros de comando.	%MW _{r.m.c.1.1}
ADJ_ERR	BOOL	R	Anomalía durante el intercambio de parámetros de ajuste (no aplicable al módulo BMX NOM 0200).	%MW _{r.m.c.1.2}

Fallos estándar del canal detectados: CH_FLT

En la siguiente tabla se describen los distintos significados de los bits de palabra de estado CH_FLT (%MW_{r.m.c.2}):

Símbolo estándar	tipo	Acceso	Significado	Dirección
NO_DEVICE	BOOL	R	No hay ningún dispositivo funcionando en el canal.	%MW _{r.m.c.2.0}
ONE_DEVICE_FLT	BOOL	R	Un dispositivo del canal no está operativo.	%MW _{r.m.c.2.1}
BLK	BOOL	R	El bloque de terminales no está conectado.	%MW _{r.m.c.2.2}
TO_ERR	BOOL	R	Temporización no ajustable (análisis necesario).	%MW _{r.m.c.2.3}
INTERNAL_FLT	BOOL	R	Detectado error interno o autoprueba del canal.	%MW _{r.m.c.2.4}
CONF_FLT	BOOL	R	Configuraciones de hardware y software diferentes.	%MW _{r.m.c.2.5}
COM_FLT	BOOL	R	Análisis de comunicación necesario con el canal.	%MW _{r.m.c.2.6}
APPLI_FLT	BOOL	R	Error de aplicación detectado (error de ajuste o configuración).	%MW _{r.m.c.2.7}

La lectura se lleva a cabo mediante la instrucción READ_STS (IODDT_VAR1).

Estado específico del canal: %MWr.m.c.3

En la siguiente tabla se muestran los distintos significados de los bits de la palabra de estado del canal `PROTOCOL` (%MWr.m.c.3):

Símbolo estándar	tipo	Acceso	Significado	Dirección
PROTOCOL	INIT	R	Byte 0 = 16#06 para la función de maestro Modbus.	%MWr.m.c.3
PROTOCOL	INIT	R	Byte 0 = 16#07 para la función de esclavo Modbus.	%MWr.m.c.3

La lectura se lleva a cabo mediante la instrucción `READ_STS` (`IODDT_VAR1`).

Comando de canal: %MWr.m.c.24

En la siguiente tabla se muestran los distintos significados de los bits de la palabra `CONTROL` (%MWr.m.c.24):

Símbolo estándar	tipo	Acceso	Significado	Dirección
DTR_ON	BOOL	L/E	Establece la señal de terminal de datos preparado.	%MWr.m.c.24.8
DTR_OFF	BOOL	L/E	Restablece la señal de terminal de datos preparado.	%MWr.m.c.24.9
TO_MODBUS_MASTER	BOOL	L/E	Cambio de modalidad de caracteres o modalidad de esclavo Modbus a modalidad de maestro Modbus.	%MWr.m.c.24.12
TO_MODBUS_SLAVE	BOOL	L/E	Cambio de modalidad de caracteres o modalidad de maestro Modbus a modalidad de esclavo Modbus.	%MWr.m.c.24.13
TO_CHAR_MODE	BOOL	L/E	Cambio de modalidad Modbus a modalidad de caracteres.	%MWr.m.c.24.14

El comando se realiza mediante la instrucción `WRITE_CMD` (`IODDT_VAR1`).

Para obtener información adicional sobre cómo cambiar protocolos, puede consultar **cambio de protocolos** (véase página 229).

Objetos de configuración externa de tipo T_COM_MB_BMX_CONF_EXT: %MWr.m.c.24.7 y %MWr.m.c.25

En la tabla siguiente se muestra significado del bit `CONTROL` (%MWr.m.c.24.7) y de la palabra `CONTROL_DATA` (%MWr.m.c.25) diseñados específicamente para la programación del módulo BMX NOM 0200:

Símbolo estándar	tipo	Acceso	Significado	Dirección
SAVE_CTRL_DATA	BOOL	L/E	Guarda los datos de control en la memoria FLASH	%MWr.m.c.24.7
CONTROL_DATA	BOOL	L/E	Dirección del esclavo Modbus que se almacena en la memoria FLASH.	%MWr.m.c.25

Detalles de los objetos de lenguaje asociados a la modalidad Modbus de configuración

Presentación

En las siguientes tablas se presentan todos los objetos de lenguaje de configuración para la modalidad de comunicación Modbus. Estos objetos no se encuentran integrados en los IODDT y pueden verse mediante el programa de aplicación.

Lista de objetos de intercambio explícito para la modalidad de maestro

En la tabla siguiente se muestran los objetos de intercambio explícito.

Dirección	Tipo	Acceso	Significado
%KWr.m.c.0	INT	L	El byte 0 de esta palabra corresponde al tipo: - El valor 6 corresponde al maestro. - El valor 7 corresponde al esclavo.
%KWr.m.c.1	INT	L	El byte 0 de esta palabra corresponde a la velocidad de transmisión. Este byte puede tener los valores siguientes: - El valor -2 (0xFE) corresponde a 300 bits/s. - El valor -1 (0xFF) corresponde a 600 bits/s. - El valor 0 (0x00) corresponde a 1.200 bits/s. - El valor 1 (0x01) corresponde a 2.400 bits/s. - El valor 2 (0x02) corresponde a 4.800 bits/s. - El valor 3 (0x03) corresponde a 9.600 bits/s. - El valor 4 (0x04) corresponde a 19.200 bits/s (valor predeterminado). - El valor 5 (0x05) corresponde a 38.400 bits/s. - El valor 6 (0x06) corresponde a 57.600 bits/s (aplicable sólo al módulo BMX NOM 0200). - El valor 7 (0x07) corresponde a 11.5200 bits/s (aplicable sólo al módulo BMX NOM 0200). El byte 1 de esta palabra corresponde al formato: - Bit 8: cantidad de bits (1 = 8 bits (RTU), 0 = 7 bits (ASCII)) - Bit 9 = 1: gestión de paridad (1 = con, 0 = sin) - Bit 10: tipo de paridad (1 = impar, 0 = par) - Bit 11: cantidad de bits de parada (1 = 1 bit, 0 = 2 bits) - Bit 13: línea física (1 = RS232, 0 = RS485) - Bit 14: Señales de módem DTR/DSR/DCD (aplicable sólo al módulo BMX NOM 0200 y sólo para la línea física RS232). Si el bit se establece en 1, las señales del módem se gestionan. - Bit 15: Señales de gestión del flujo de hardware RTS/CTS. Si se selecciona RS232, este bit puede tener 2 valores distintos: 0 para RX/TX y 1 para RX/TX + RTS/CTS. Si se selecciona RS485, el valor predeterminado es 0 y corresponde a RX/TX.

Dirección	Tipo	Acceso	Significado
%KWr.m.c.2	INT	L	Retardo entre tramas (sólo en la modalidad RTU): valor en ms que oscila entre 2 y 10.000 ms (depende del formato y de la velocidad de transmisión seleccionados). Si la casilla está activada de forma predeterminada, el valor predeterminado es 2 ms. 10 s corresponde a una espera infinita.
%KWr.m.c.3	INT	L	En la modalidad de maestro Modbus, este objeto corresponde al retardo de respuesta en ms que oscila entre 10 ms y 1.000 ms, de modo que 100 ms es el valor predeterminado. 10 s corresponde a una espera infinita.
%KWr.m.c.4	INT	L	Disponible únicamente en la modalidad de maestro Modbus. El byte 0 de esta palabra es el número de reintentos que va de 0 a 15. El valor predeterminado es 3.
%KWr.m.c.5	INT	L	Esta palabra corresponde al tiempo de retardo RTS/CTS en centésimas de ms de 0 a 100, en caso de que se haya seleccionado RS232. Si se selecciona RS485, el valor predeterminado es 0.

Lista de objetos de intercambio explícito para la modalidad de esclavo

Los objetos de lenguaje para la función de esclavo Modbus son idénticos a los de la función de maestro Modbus. La única diferencia aparece en los objetos siguientes:

Dirección	Tipo	Acceso	Significado
%KWr.m.c.3	INT	L	En la modalidad de esclavo Modbus, el byte 0 de este objeto corresponde al número de esclavo [0/1, 247]. Para el módulo BMX NOM 0200, el valor 0 significa que el número del esclavo está codificado en la memoria FLASH
%KWr.m.c.4	INT	L	Se utiliza únicamente en la modalidad de maestro Modbus.

10.4 Objetos de lenguaje e IODDT asociados a la comunicación en modalidad de caracteres

Objeto

En esta sección se presentan los objetos de lenguaje e IODDT asociados a la comunicación en modalidad de caracteres.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Detalles relativos a los objetos de lenguaje de intercambio explícito para la comunicación en modalidad de caracteres	215
Detalles de los objetos de intercambio implícito de IODDT de tipo T_COM_CHAR_BMX	216
Detalles de los objetos de intercambio explícito de IODDT de tipo T_COM_CHAR_BMX	217
Detalles de los objetos de lenguaje asociados a la configuración en modalidad de caracteres	220

Detalles relativos a los objetos de lenguaje de intercambio explícito para la comunicación en modalidad de caracteres

Presentación

En las siguientes tablas se muestran todos los objetos de lenguaje de configuración para la comunicación en modalidad de caracteres. Estos objetos no se encuentran integrados en los IODDT.

Lista de objetos de intercambio explícito

En la siguiente tabla se muestran los objetos de intercambio explícito:

Dirección	Tipo	Acceso	Significado
%MW _r .m.c.4	INT	L	Anomalía en caracteres transmitidos.
%MW _r .m.c.5	INT	L	Anomalía en caracteres recibidos.
%MW _r .m.c.24.0	BOOL	L/E	Restablece los contadores de errores cuando se establece en 1.
%QW _r .m.c.0 = 16#DEAD	INT	RW	Reinicie el BMX NOM 0200.

Detalles de los objetos de intercambio implícito de IODDT de tipo T_COM_CHAR_BMX

Presentación

En la siguiente tabla se muestran los objetos de intercambio implícito de IODDT de tipo T_COM_CHAR_BMX que se aplican a la comunicación en modalidad de caracteres.

Bit de error

En la siguiente tabla se presenta el significado del bit de error CH_ERROR (%Ir.m.c.ERR):

Símbolo estándar	Tipo	Acceso	Significado	Dirección
CH_ERROR	EBOOL	L	Bit de error del canal de comunicación.	%Ir.m.c.ERR

Objeto de señal en entrada

En la siguiente tabla se muestra el significado del bit de la palabra INPUT_SIGNALS (%IWIr.m.c.0):

Símbolo estándar	Tipo	Acceso	Significado	Dirección
DCD	BOOL	L	Señal de detección de portadora de datos RS232 (aplicable sólo al módulo BMX NOM 0200).	%IWIr.m.c.0.0
CTS	BOOL	L	Señal de Listo para enviar RS232.	%IWIr.m.c.0.2
DSR	BOOL	L	Señal de detección paquete de datos preparado RS232 (aplicable sólo al módulo BMX NOM 0200).	%IWIr.m.c.0.3

NOTA: Cuando CTS es verde en la unidad P, significa que %IWIr.m.c.0.0 está en 1 y que la tensión de esta señal es positiva. También es aplicable a DCD y DSR.

Detalles de los objetos de intercambio explícito de IODDT de tipo T_COM_CHAR_BMX

Presentación

En este apartado se presentan los objetos de intercambio explícito de IODDT de tipo T_COM_CHAR_BMX que se aplican a la comunicación en modalidad de caracteres. Agrupa los objetos de tipo palabra, cuyos bits tienen un significado particular. Estos objetos se describen de forma detallada a continuación.

En este apartado, la variable IODDT_VAR1 es de tipo T_COM_STS_GEN.

Observaciones

Por lo general, se ofrece el significado de los bits para el estado 1 del bit. En determinados casos, se explica el estado de cada bit.

No se utilizan todos los bits.

Indicadores de ejecución de un intercambio explícito: EXCH_STS

En la siguiente tabla se muestra el significado de los bits de control de intercambio del canal EXCH_STS (%MW_{r.m.c.0}):

Símbolo estándar	tipo	Acceso	Significado	Dirección
STS_IN_PROGR	BOOL	R	Lectura de palabras de estado del canal en curso.	%MW _{r.m.c.0.0}
CMD_IN_PROGR	BOOL	R	Intercambio de parámetros de comando en curso.	%MW _{r.m.c.0.1}
ADJ_IN_PROGR	BOOL	R	Intercambio de parámetros de ajuste en curso (no aplicable al módulo BMX NOM 0200).	%MW _{r.m.c.0.2}

Informe de intercambios explícitos: EXCH_RPT

En la siguiente tabla se presenta el significado de los bits de informe de intercambio EXCH_RPT (%MWr.m.c.1):

Símbolo estándar	tipo	Acceso	Significado	Dirección
STS_ERR	BOOL	R	Error de lectura de palabras de estado del canal detectado.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Anomalía durante un intercambio de parámetros de comando.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Anomalía durante el intercambio de parámetros de ajuste (no aplicable al módulo BMX NOM 0200).	%MWr.m.c.1.2

Fallos de canal estándar detectados, CH_FLT

En la siguiente tabla se describen los distintos significados de los bits de palabra de estado CH_FLT (%MWr.m.c.2):

Símbolo estándar	tipo	Acceso	Significado	Dirección
NO_DEVICE	BOOL	R	No funciona ningún dispositivo en el canal.	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	R	Un dispositivo del canal no está operativo.	%MWr.m.c.2.1
BLK	BOOL	R	El bloque de terminales no está conectado.	%MWr.m.c.2.2
TO_ERR	BOOL	R	Temporización no ajustable (análisis necesario).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Detectado error interno o autoprueba del canal.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Configuraciones de hardware y software diferentes.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Análisis de comunicación necesario con el PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Error de aplicación detectado (error de ajuste o configuración).	%MWr.m.c.2.7

La lectura se lleva a cabo mediante la instrucción READ_STS (IODDT_VAR1).

Estado específico del canal, %MWr.m.c.3

En la siguiente tabla se muestran los distintos significados de los bits de la palabra de estado del canal `PROTOCOL` (%MWr.m.c.3):

Símbolo estándar	tipo	Acceso	Significado	Dirección
PROTOCOL	INT	R	Byte 0 = 16#03 para la función de modalidad de caracteres.	%MWr.m.c.3

La lectura se lleva a cabo mediante la instrucción `READ_STS` (`IODDT_VAR1`).

Comando de canal %MWr.m.c.24

En la siguiente tabla se muestran los distintos significados de los bits de la palabra `CONTROL` (%MWr.m.c.24):

Símbolo estándar	tipo	Acceso	Significado	Dirección
DTR_ON	BOOL	L/E	Establece la señal de terminal de datos preparado.	%MWr.m.c.24.8
DTR_OFF	BOOL	L/E	Restablece la señal de terminal de datos preparado.	%MWr.m.c.24.9

El comando se realiza mediante la instrucción `WRITE_CMD` (`IODDT_VAR1`).

Para obtener información adicional sobre cómo cambiar protocolos, puede consultar cambio de protocolos (*véase página 229*).

Objeto de palabra %QWr.m.c.0

En la tabla siguiente se muestra el significado del bit 0 de la palabra %QWr.m.c.0:

Símbolo estándar	Tipo	Acceso	Significado	Dirección
STOP_EXCH	BOOL	R/W	Detener todos los intercambios en el flanco ascendente (sólo disponible en el módulo BMX NOM 0200).	%QWr.m.c.0.0

Detalles de los objetos de lenguaje asociados a la configuración en modalidad de caracteres

Presentación

En las siguientes tablas se muestran todos los objetos de lenguaje de configuración para la comunicación en modalidad de caracteres. Estos objetos no se encuentran integrados en los IODDT y pueden verse mediante el programa de aplicación.

Lista de objetos de intercambio explícito para la modalidad de caracteres

En la tabla siguiente se muestran los objetos de intercambio explícito.

Dirección	Tipo	Acceso	Significado
%KWr.m.c.0	INT	L	El byte 0 de esta palabra corresponde al tipo. El valor 3 corresponde a la modalidad de caracteres.
%KWr.m.c.1	INT	L	El byte 0 de esta palabra corresponde a la velocidad de transmisión. Este byte puede tener los valores siguientes: ● El valor -2 (0xFE) corresponde a 300 bits/s. ● El valor -1 (0xFF) corresponde a 600 bits/s. ● El valor 0 (0x00) corresponde a 1.200 bits/s. ● El valor 1 (0x01) corresponde a 2.400 bits/s. ● El valor 2 (0x02) corresponde a 4.800 bits/s. ● El valor 3 (0x03) corresponde a 9.600 bits/s (valor predeterminado). ● El valor 4 (0x04) corresponde a 19.200 bits/s. ● El valor 5 (0x05) corresponde a 38.400 bits/s. ● El valor 6 (0x06) corresponde a 57.600 bits/s (sólo se puede usar con el módulo BMX NOM 0200). ● El valor 7 (0x07) corresponde a 115200 bits/s (sólo se puede usar con el módulo BMX NOM 0200). El byte 1 de esta palabra corresponde al formato: ● Bit 8: cantidad de bits (1 = 8 bits (RTU), 0 = 7 bits (ASCII)) ● Bit 9 = 1: gestión de paridad (1 = con, 0 = sin) ● Bit 10: tipo de paridad (1 = impar, 0 = par) ● Bit 11: cantidad de bits de parada (1 = 1 bit, 0 = 2 bits) ● Bit 13: línea física (1 = RS232, 0 = RS485) ● Bit 14: señales de módem DTR/DSR/DCD. Para el módulo BMX NOM 0200 y si se selecciona RS232, este bit puede tener 2 valores distintos: 1 significa que las señales de módem se gestionan, 0 significa que no se gestionan (el valor predeterminado de BMX P34 o si RS485 está seleccionado). ● Bit 15: Señales de gestión del flujo de hardware RTS/CTS. Si se selecciona RS232, este bit puede tener 2 valores distintos: 0 para RX/TX y 1 para RX/TX + RTS/CTS. Si se selecciona RS485, el valor predeterminado es 0 y corresponde a RX/TX.
%KWr.m.c.2	INT	L	Valor introducido en ms de parada en silencio (depende de la velocidad de transmisión y del formato seleccionados). El valor 0 indica que no existe detección de silencio.

Dirección	Tipo	Acceso	Significado
%KWr.m.c.3	INT	L	Esta palabra corresponde al tipo de polarización: ● El valor 0 en los bits 14 y 15 corresponde a no polarización (el valor predeterminado de BMX P34 o si RS232 está seleccionado). ● Bit 14: El valor 1 corresponde a polarización de baja impedancia (de tipo Modbus) y sólo puede usarse para el módulo BMX NOM 0200 y si RS485 está seleccionado. ● Bit 15: El valor 1 corresponde a polarización de alta impedancia y sólo puede usarse para el módulo BMX NOM 0200 y si RS485 está seleccionado.
%KWr.m.c.5	INT	L	Esta palabra corresponde al tiempo de retardo RTS/CTS en centésimas de ms de 0 a 100, en caso de que se haya seleccionado RS232. En caso de seleccionar RS485, el valor predeterminado es 0.
%KWr.m.c.6	INT	L	El bit 0 del byte 0 puede tener dos valores: ● El valor 1 corresponde a la casilla de verificación de detención en el área Parada al recibir para el carácter 1 cuando está activada. ● El valor 0 corresponde a la casilla de verificación de detención en el área Parada al recibir para el carácter 1 cuando está desactivada. El bit 1 del byte 0 puede tener dos valores: ● El valor 1 corresponde a la casilla de verificación Carácter incluido en el área Parada al recibir para el carácter 1 cuando está activada. ● El valor 0 corresponde a la casilla de verificación Carácter incluido en el área Parada al recibir para el carácter 1 cuando está desactivada. El byte 1 de esta palabra corresponde al valor introducido de parada al recibir del carácter 1 de 0 a 255.
%KWr.m.c.7	INT	L	El bit 0 del byte 0 puede tener dos valores: ● El valor 1 corresponde a la casilla de verificación de detención en el área Parada al recibir para el carácter 2 cuando está activada. ● El valor 0 corresponde a la casilla de verificación de detención en el área Parada al recibir para el carácter 2 cuando está desactivada. El bit 1 del byte 0 puede tener dos valores: ● El valor 1 corresponde a la casilla de verificación Carácter incluido en el área Parada al recibir para el carácter 2 cuando está activada. ● El valor 0 corresponde a la casilla de verificación Carácter incluido en el área Parada al recibir para el carácter 2 cuando está desactivada. El byte 1 de esta palabra corresponde al valor introducido de parada al recibir del carácter 2 de 0 a 255.

10.5 El IODDT tipo T_GEN_MOD es aplicable a todos los módulos

Detalles de los objetos de lenguaje del IODDT de tipo T_GEN_MOD

Introducción

Todos los módulos de los PLC Modicon M340 tienen un IODDT asociado de tipo T_GEN_MOD.

Observaciones

En general, el significado de los bits se indica para el estado 1 del bit. En los casos específicos, se explica cada estado del bit.

Algunos bits no se utilizan.

Lista de objetos

La tabla siguiente muestra los objetos del IODDT.

Símbolo estándar	Tipo	Acceso	Significado	Dirección
MOD_ERROR	BOOL	R	Bit de error del módulo detectado	%I.r.m.MOD.ERR
EXCH_STS	INIT	R	Palabra de control de intercambio del módulo	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Lectura de palabras de estado del módulo en curso	%MWr.m.MOD.0.0
EXCH_RPT	INIT	R	Palabra de informe de intercambio	%MWr.m.MOD.1
STS_ERR	BOOL	R	Evento al leer las palabras de estado del módulo	%MWr.m.MOD.1.0
MOD_FLT	INIT	R	Palabra de errores internos detectados del módulo	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	Módulo no operativo	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Canales no operativos	%MWr.m.MOD.2.1
BLK	BOOL	R	Cableado incorrecto del bloque de terminales	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Anomalía de configuración de hardware o software	%MWr.m.MOD.2.5

Símbolo estándar	Tipo	Acceso	Significado	Dirección
NO_MOD	BOOL	R	Falta el módulo o no está operativo	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Palabra de errores internos detectados del módulo (sólo extensión Fipio)	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Error interno detectado, módulo fuera de servicio (sólo extensión Fipio)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Canales no operativos (sólo extensión Fipio)	%MWr.m.MOD.2.9
BLK_EXT	BOOL	R	Bloque de terminales cableado incorrectamente (sólo extensión Fipio)	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Anomalía en la configuración de hardware o software (sólo extensión Fipio)	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	Falta el módulo o no está operativo (sólo extensión Fipio)	%MWr.m.MOD.2.14

Conmutación dinámica de protocolos

11

Objeto

En este capítulo se ofrece una introducción a la conmutación dinámica entre los protocolos Modbus y de modalidad de caracteres.

Contenido de este capítulo

Este capítulo contiene los siguiente apartados:

Apartado	Página
Cambio de protocolo con los procesadores BMX P34 1000/2000/2010/20102/2020	226
Cambio de protocolo con el módulo BMX NOM 0200	229

Cambio de protocolo con los procesadores BMX P34 1000/2000/2010/20102/2020

General

En este apartado se describe cómo cambiar el protocolo que utiliza una comunicación serie de CPU mediante el comando `WRITE_CMD (IODDT_VAR1)`. Este comando puede utilizarse, por ejemplo, para intercambiar los tres protocolos siguientes:

- Esclavo Modbus
- Maestro Modbus
- Modalidad de caracteres

NOTA: La variable `IODDT_VAR1` debe ser de tipo `T_COM_MB_BMX`.

Cambio de protocolo: el principio

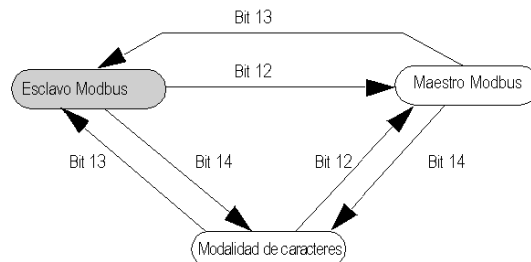
En primer lugar, debe crear una variable `IODDT` vinculada al canal serie del procesador; a continuación, establezca en 1 el bit de la palabra `IODDT_VAR1.CONTROL (%MWr.m.c.24)` que corresponde al cambio de protocolo deseado:

- `TO_MODBUS_MASTER` (Bit 12): el protocolo actual cambia a maestro Modbus.
- `TO_MODBUS_SLAVE` (Bit 13): el protocolo actual cambia a esclavo Modbus.
- `TO_CHAR_MODE` (Bit 14): el protocolo actual cambia a modalidad de caracteres.

NOTA: `IODDT_VAR1.CONTROL (%MWr.m.c.24)` forma parte de la variable `IODDT` `IODDT_VAR1`.

A continuación, aplique la instrucción `WRITE_CMD` a la variable `IODDT` vinculada al canal serie del procesador.

En el diagrama siguiente se muestran los cambios de protocolo que deben realizarse en función de los bits de la palabra `IODDT_VAR1.CONTROL (%MWr.m.c.24)` establecida en 1:



NOTA: Para que se efectúen los cambios de un protocolo a otro, el procesador debe estar configurado inicialmente en modalidad de esclavo Modbus.

Usos

Se utilizan tres cambios de protocolo:

- Transferencia al maestro Modbus: el cambio de protocolo es un proceso de dos fases:
 - Transferencia de la configuración del esclavo Modbus a la configuración del maestro Modbus y
 - vuelta a la configuración del esclavo Modbus inicial.

El objetivo de la configuración del maestro Modbus es enviar información sobre un evento a otro PLC. Cuando se efectúa un cambio de la configuración del esclavo Modbus a la configuración del maestro Modbus, los parámetros de transmisión, señales y línea física permanecen iguales. Sólo se cambian los valores de los siguientes parámetros específicos de la configuración del maestro Modbus:

- El retardo entre tramas se establece en su valor predeterminado, que varía en función de la velocidad de transmisión.
 - El retardo de respuesta se establece en 3.000 ms.
 - El número de reintentos se establece en 3.
- Transferencia a la modalidad de caracteres: este cambio de protocolo es un proceso de dos fases:
 - transferencia de configuración del esclavo Modbus a la configuración de modalidad de caracteres y
 - vuelta a la configuración del esclavo Modbus inicial.

El objetivo de la configuración de modalidad de caracteres es establecer comunicación con un protocolo privado (un módem, por ejemplo). Cuando se efectúa un cambio de configuración del esclavo Modbus a la configuración de modalidad de caracteres, los parámetros de transmisión, señales y línea física permanecen iguales. Sólo se establece el parámetro de final de mensaje específico de la modalidad de caracteres en parada en silencio con un timeout de 1000 ms.

- Transferencia a los protocolos de modalidad de caracteres y maestro Modbus: este cambio de protocolo es un proceso de tres fases:
 - transferencia de la configuración del esclavo Modbus a la configuración de modalidad de caracteres,
 - transferencia de la configuración de modalidad de caracteres a la configuración del maestro Modbus y
 - vuelta a la configuración del esclavo Modbus inicial.

El objetivo de la configuración de modalidad de caracteres es establecer comunicación con un protocolo privado (un módem, por ejemplo). Una vez finalizado el intercambio, el usuario conmuta a la configuración del maestro Modbus para enviar información sobre un evento a otro PLC. Una vez enviado el mensaje, el usuario vuelve a la configuración del esclavo Modbus inicial.

NOTA: En los tres casos, la configuración predeterminada permanece en esclavo Modbus.

Inicios en caliente y en frío

Los cambios de protocolo no se ven afectados por los bits %S0 y %S1 (los bits establecidos en 1 durante un inicio en frío y en caliente, respectivamente). Sin embargo, un inicio en frío o en caliente del PLC configurará el puerto serie a sus valores predeterminados o a valores programados en la aplicación.

Cambio de protocolo con el módulo BMX NOM 0200

General

En este apartado se describe cómo cambiar el protocolo que utiliza una comunicación serie de BMX NOM 0200 mediante el comando

```
WRITE_CMD( IODDT_VAR1 ).
```

Este comando puede utilizarse, por ejemplo, para intercambiar los tres protocolos siguientes:

- Esclavo Modbus
- Maestro Modbus
- Modalidad de caracteres

NOTA: La variable IODDT_VAR1 debe ser de tipo T_COM_MB_BMX o T_COM_MB_BMX CONF EXT.

Cambio de protocolo: el principio

En primer lugar, debe crear una variable IODDT vinculada al canal serie; a continuación, establezca en 1 el bit de la palabra IODDT_VAR1.CONTROL (%MWr.m.c.24) que corresponde al cambio de protocolo deseado:

- TO_MODBUS_MASTER (Bit 12): el protocolo actual cambia a maestro Modbus.
- TO_MODBUS_SLAVE (Bit 13): el protocolo actual cambia a esclavo Modbus.
- TO_CHAR_MODE (Bit 14): el protocolo actual cambia a modalidad de caracteres.

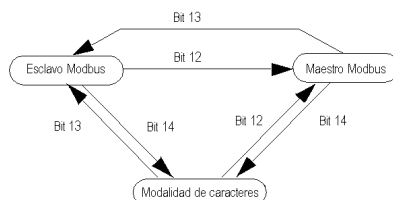
NOTA: Sólo puede establecerse un bit en 1 cada vez; si se establecen varios bits en 1 se puede generar un error.

NOTA: IODDT_VAR1.CONTROL (%MWr.m.c.24) forma parte de la variable IODDT IODDT_VAR1.

A continuación, aplique la instrucción `WRITE_CMD` a la variable IODDT vinculada al canal serie.

NOTA: Procure que dos maestros (en el mismo bus) no envíen dos peticiones simultáneamente. De lo contrario, ambas se perderán y cada informe tendrá un mal resultado, que podría ser 16#0100 (la petición no se ha podido procesar) o 16#ODFF (falta el esclavo).

En el diagrama siguiente se muestran los cambios de protocolo que deben realizarse en función de los bits de la palabra IODDT_VAR1.CONTROL (%MWr.m.c.24) establecida en 1:



Usos

Se utilizan tres cambios de protocolo:

- **Transferencia del esclavo Modbus al maestro Modbus:**
El objetivo de la configuración del maestro Modbus es enviar información sobre un evento a otro PLC. Cuando se efectúa un cambio de la configuración del esclavo Modbus a la configuración del maestro Modbus, los parámetros de transmisión, señales y línea física permanecen iguales. Sólo se cambian los valores de los siguientes parámetros específicos de la configuración del maestro Modbus:
 - El retardo entre tramas se establece en su valor predeterminado, que varía en función de la velocidad de transmisión.
 - El retardo de respuesta se establece en 3s.
 - El número de reintentos se establece en 0.
- **Transferencia del esclavo/maestro Modbus a la modalidad de caracteres**
El cambio a la modalidad de caracteres se utiliza para enviar comandos AT a un módem. Cuando se efectúa un cambio de configuración de Modbus a la configuración de modalidad de caracteres, los parámetros de transmisión, señales y línea física permanecen iguales. Sólo se establece en parada el parámetro de detección de final de mensaje específico de la modalidad de caracteres al recibir el carácter de fin `x0d`.
- **Transferencia de la modalidad de caracteres al maestro Modbus y el esclavo Modbus:**
El objetivo de la configuración de modalidad de caracteres es establecer comunicación con un protocolo privado (un módem, por ejemplo). Una vez finalizado el intercambio, el usuario conmuta a la configuración del maestro Modbus (con el retardo de respuesta establecido en 3s y el número de reintentos establecido en 0) para enviar información sobre un evento a otro PLC. Una vez enviado el mensaje, el usuario vuelve a la configuración del esclavo Modbus: el número de esclavos se establece en el valor almacenado en la memoria FLASH o en 248 si no hay ninguno.

Inicios en caliente y en frío

Los cambios de protocolo no se ven afectados por los bits `%S0` y `%S1` (los bits establecidos en 1 durante un inicio en frío y en caliente, respectivamente). Sin embargo, un inicio en frío o en caliente del PLC configurará el puerto serie a sus valores predeterminados o a valores programados en la aplicación.

NOTA: La configuración predeterminada del módulo es la siguiente: para que un ordenador de tipo PC pueda configurarlo fácilmente, el canal 0 se configura en el modo esclavo RS232, y el canal 1 el modo RS485. Otros parámetros son: 19200 baudios, RTU, par, 1 bit de parada, sin control de paridad, 1,75ms como retardo de tramas predeterminado, número de esclavo 248.

Inicio rápido: Ejemplo de implementación de conexiones serie



Descripción general

Esta sección presenta un ejemplo de implementación de conexiones serie.

Contenido de esta parte

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
12	Descripción de la aplicación	233
13	Instalación de la aplicación con Unity Pro	235
14	Inicio de aplicaciones	265

Descripción de la aplicación

12

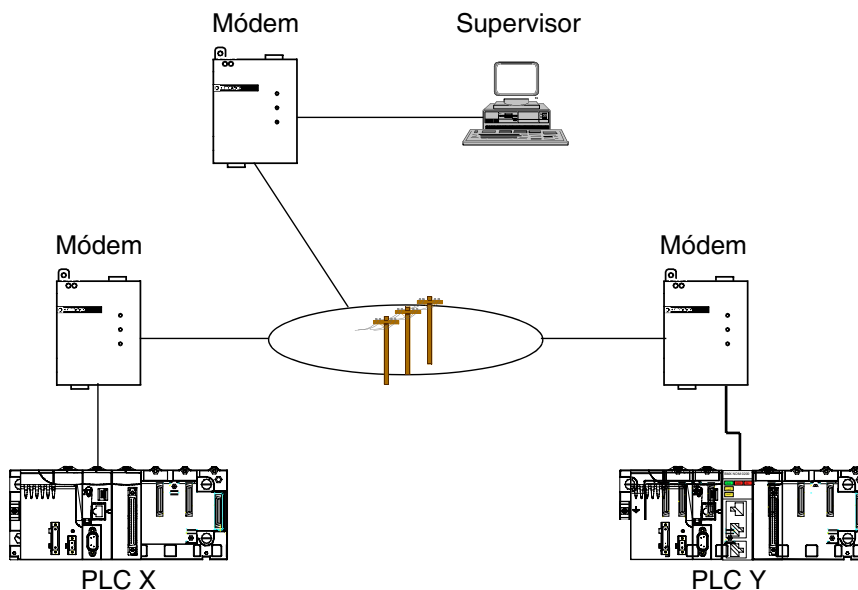
Descripción de la aplicación

Introducción

La aplicación descrita en este documento es una aplicación de comunicación Modbus mediante modems.

Ilustración de ejemplo

La figura que aparece a continuación ilustra el ejemplo:



Los dispositivos se comunican entre sí mediante modems. El supervisor es el maestro Modbus mientras que los PLC X e Y son esclavos.

El objetivo del ejemplo es escribir los valores del área de datos del PLC X en el PLC Y.

Para esto, el PLC X debe convertirse en maestro Modbus.

Cada día, el supervisor se comunica con los PLC para recuperar información.

Si hay una alarma en el PLC X, cambia a la modalidad de maestro Modbus y envía datos al PLC Y.

Para simplificar la programación, los modems se han inicializado con los parámetros correctos mediante un terminal de programación. Estos parámetros están almacenados en una memoria no volátil mediante los comandos AT&W.

Modo de funcionamiento

El funcionamiento de la aplicación es el siguiente:

Paso	Acción
1	El puerto del PLC X cambia a la modalidad de caracteres.
2	El PLC X envía un mensaje de marcación al modem.
3	El puerto del PLC X cambia a la modalidad de maestro Modbus.
4	El PLC maestro (X) envía datos al PLC esclavo (Y).
5	El puerto cambia a la modalidad de caracteres.
6	El PLC X envía un mensaje de desconexión al modem.
7	El puerto del PLC X cambia a la modalidad de esclavo Modbus.

Instalación de la aplicación con Unity Pro

13

Objetivo de este capítulo

En este capítulo se describe el procedimiento para crear la aplicación que se describe. En general y de forma más detallada, se indican los pasos en los que se divide la creación de los distintos componentes de la aplicación.

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
13.1	Presentación de la solución utilizada	236
13.2	Desarrollo de la aplicación	237

13.1 Presentación de la solución utilizada

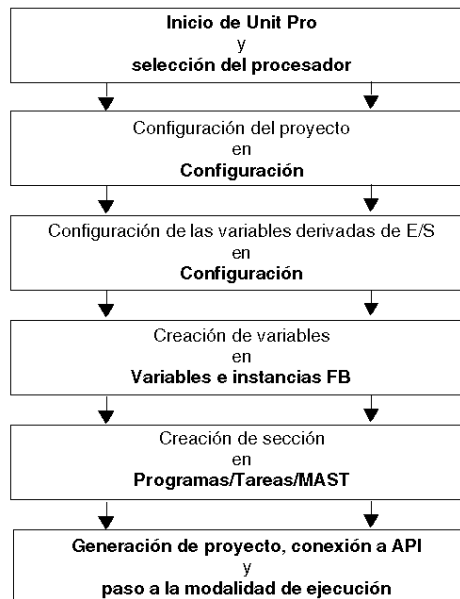
Los distintos pasos del proceso utilizando Unity Pro

Presentación

El esquema lógico siguiente muestra los distintos pasos que deben seguirse para crear la aplicación. Es necesario respetar un orden cronológico con el fin de definir correctamente todos los elementos de la aplicación.

Descripción

Descripción de los distintos tipos:



13.2 Desarrollo de la aplicación

Objeto de este apartado

En esta sección se presenta una descripción detallada del proceso de creación de la aplicación mediante Unity Pro.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Creación del proyecto	238
Declaración de variables	243
Uso de módems	247
Procedimiento de programación	249
Estructura de programación	251
Programación	255

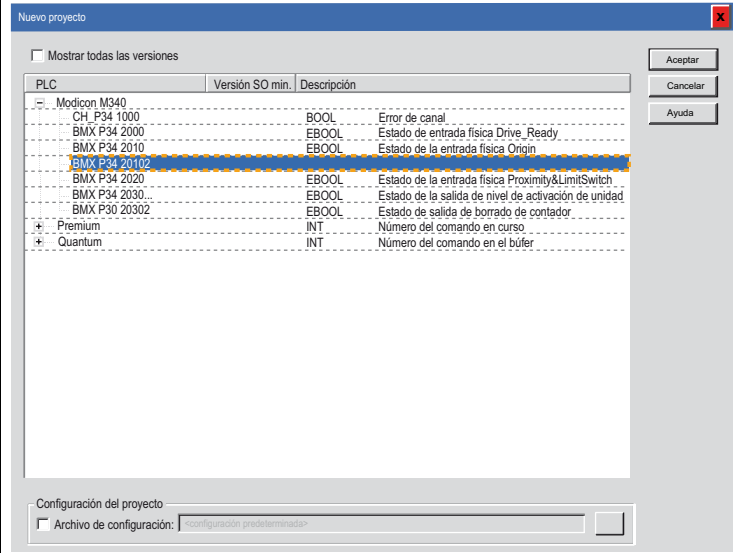
Creación del proyecto

Presentación

Para continuar el desarrollo del ejemplo, es necesario crear un proyecto principal asociado al PLC X para configurar el PLC X, así como declarar todas las variables necesarias y programar la aplicación. Además, es necesario crear un proyecto independiente para la configuración del PLC Y.

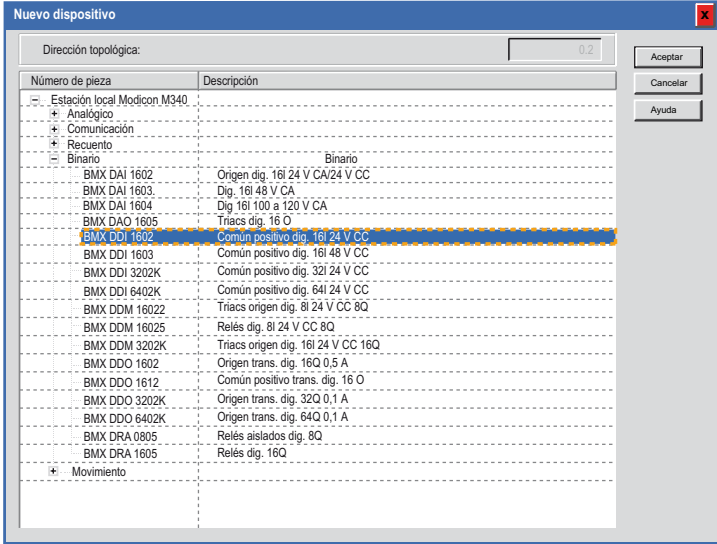
Procedimiento para crear un proyecto

En la tabla siguiente se detalla el procedimiento que debe seguirse para crear un proyecto mediante Unity Pro.

Paso	Acción
1	Inicie el software Unity Pro.
2	Haga clic en Archivo y, a continuación, en Nuevo para seleccionar un procesador BMX P34 20102: 
3	Confirme mediante Aceptar.

Selección del módulo de entrada binario

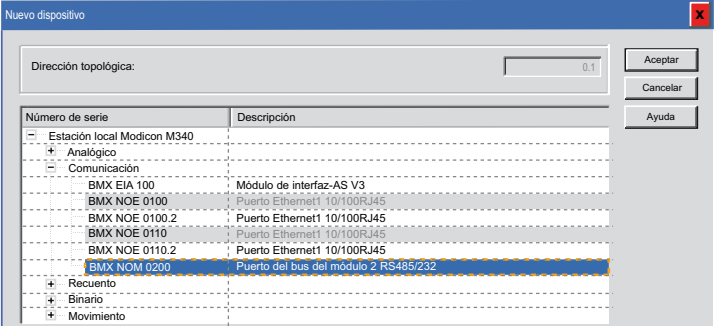
La tabla siguiente presenta el procedimiento para seleccionar el módulo binario que el PLC X necesita.

Paso	Acción
1	En el Explorador de proyectos, haga doble clic en Configuración, 0:Bus PLC y 0:BMX XBP (donde 0 es el número de bastidor).
2	En la ventana Bus PLC, seleccione un slot (por ejemplo, slot 1) y haga doble clic en él.
3	Seleccione el módulo de entrada binario BMX DDI 1602 en la lista de módulos Binarios. 
4	Confirme mediante Aceptar.

Selección del módulo BMX NOM 0200

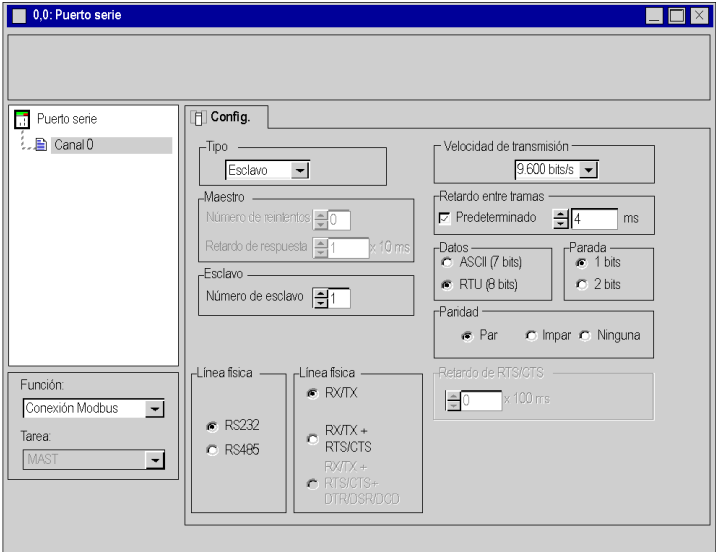
En este ejemplo, se utiliza un módulo BMX NOM 0200 en el PLC Y para la conexión serie con el módem. Por tanto, es necesario añadirlo al proyecto asociado al PLC Y.

En la siguiente tabla se muestra el procedimiento para seleccionar el módulo BMX NOM 0200:

Paso	Acción
1	En el Explorador de proyectos, haga doble clic en Configuración, 0:Bus PLC y 0:BMX XBP (donde 0 es el número de bastidor).
2	En la ventana Bus PLC, seleccione un slot (por ejemplo, slot 1) y haga doble clic en él.
3	Seleccione el módulo de comunicación BMX NOM 0200 en la lista de módulos Comunicación 
4	Confirme mediante Aceptar.

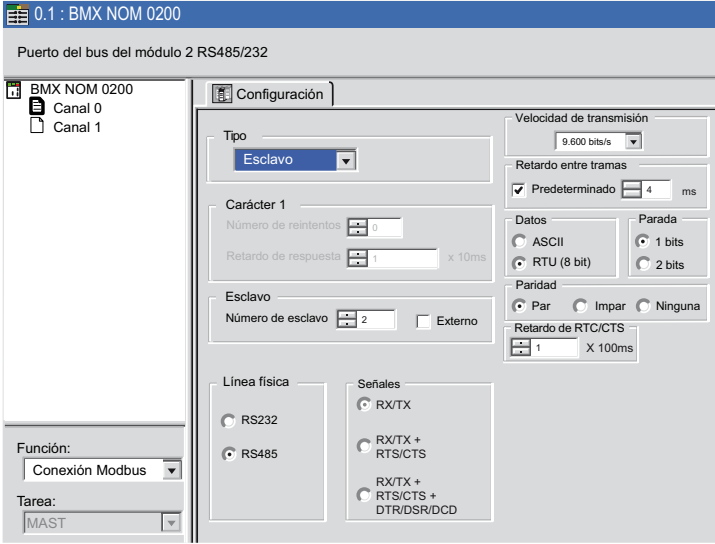
Configuración del puerto serie del procesador

La tabla siguiente muestra el procedimiento para configurar el puerto serie del procesador PLC X como esclavo Modbus:

Paso	Acción
1	En el Explorador de proyectos, haga doble clic en Configuración, 0:BMX XBP 0800 y 0:BMX P34 20102. Después, haga doble clic en Puerto serie para acceder a la ventana 0.0:Puerto serie. 
2	Seleccione el tipo Esclavo.
3	Seleccione 9600 bits/s para la velocidad de transmisión.
4	Seleccione RS232 para la línea física.
5	Seleccione RTU (8 bits) para el tipo de datos.
6	Cierre la ventana y confirme con Aceptar.

Configuración del canal serie BMX NOM 0200

La tabla siguiente muestra el procedimiento para configurar el canal serie del módulo PLC Y BMX NOM 0200 como esclavo Modbus:

Paso	Acción
1	En el Explorador de proyectos, haga doble clic en Configuración, 0:BMX XBP 0800 y 0:BMX NOM 0200 para acceder a la ventana 0.x:BMX NOM 0200 (donde x es el número de slot; por ejemplo x=1). 
2	Seleccione el Canal 0.
3	Seleccione la función conexión Modbus.
4	Seleccione el tipo Esclavo.
5	Seleccione 9600 bits/s para la velocidad de transmisión.
6	Seleccione RS232 para la línea física.
7	Seleccione las señales RX/TX + RTS/CTS + DTR/DSR/DCD.
8	Seleccione el retardo de RTS/CTS 100 ms.
9	Seleccione RTU (8 bits) para el tipo de datos.
10	Cierre la ventana y confirme con Aceptar.

Declaración de variables

Presentación

Es necesario declarar todas las variables utilizadas en las diferentes secciones del programa.

No será posible emplear variables sin declarar en el programa.

NOTA: Para obtener más información, consulte la ayuda en línea de Unity Pro (haga clic en [?, Unity, Software Unity Pro, Modalidades de funcionamiento y Editor de datos](#)).

Procedimiento de declaración de variables

En la tabla siguiente se presenta el procedimiento que debe seguirse para declarar variables de aplicación:

Paso	Acción
1	En el Explorador de proyectos/Variables e instancias FB, haga doble clic en Variables elementales.
2	En la ventana Editor de datos, active el cuadro de la columna Nombre y escriba un nombre para la primera variable.
3	Seleccione un Tipo para esta variable.
4	Después de declarar todas las variables, puede cerrar la ventana.

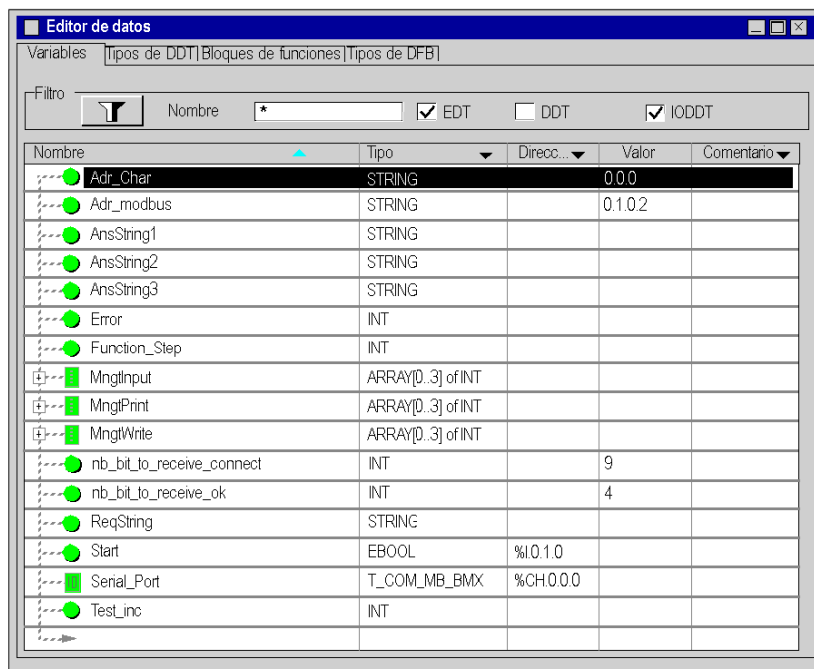
Variables utilizadas para la aplicación

La tabla siguiente contiene los detalles de las variables utilizadas en la aplicación y declaradas en el proyecto asociado al PLC X:

Variable	Tipo	Definición
Adr_Char	STRING	Dirección del puerto serie del PLC maestro.
Adr_modbus	STRING	Dirección del canal serie del PLC esclavo Modbus (canal 0 del módulo BMX NOM 0200).
AnsString1	STRING	Primera cadena de caracteres de respuesta al módem.
AnsString2	STRING	Segunda cadena de caracteres de respuesta al módem.
AnsString3	STRING	Tercera cadena de caracteres de respuesta al módem.
Error	INT	Código de error de la función.
Function_Step	INT	Paso de la función.

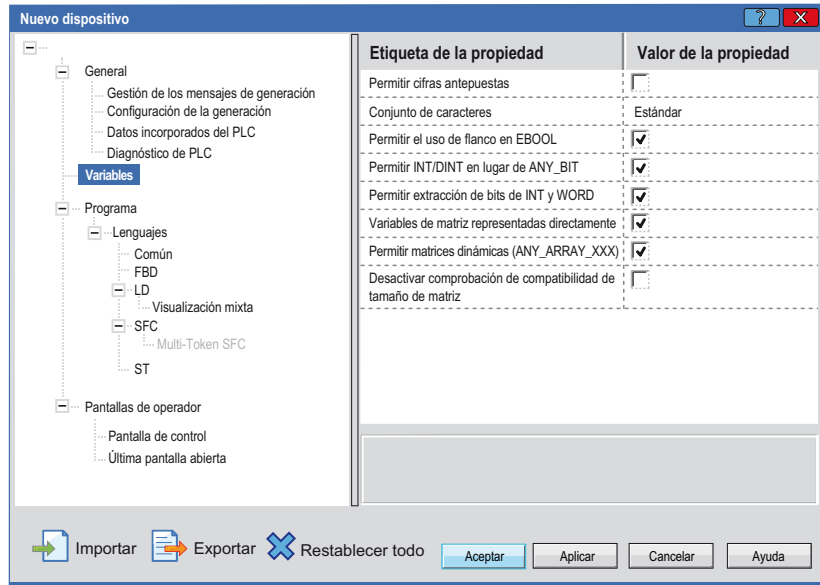
Variable	Tipo	Definición
MngtInput	ARRAY[0..3] of INT	Matriz de los parámetros de comunicación del bloque INPUT_CHAR.
MngtPrint	ARRAY[0..3] of INT	Matriz de los parámetros de comunicación del bloque PRINT_CHAR.
MngtWrite	ARRAY[0..3] of INT	Matriz de los parámetros de comunicación del bloque WRITE_VAR.
nb_charac_to_receive_connect	INT	Número de caracteres que van a recibirse: conexión del módem
nb_charac_to_receive_ok	INT	Número de caracteres que van a recibirse: mensaje de confirmación del módem
ReqString	STRING	Respuesta del módem.
Start	EBOOL	Modo de inicio (señal procedente del canal 0 del módulo BMX DDI 1602).
Serial_Port	T_COM_MB_BMX	Objeto de E/S del puerto serie
Test_inc	INT	Valor de incremento

La pantalla siguiente muestra las variables de aplicación creadas mediante el Editor de datos:

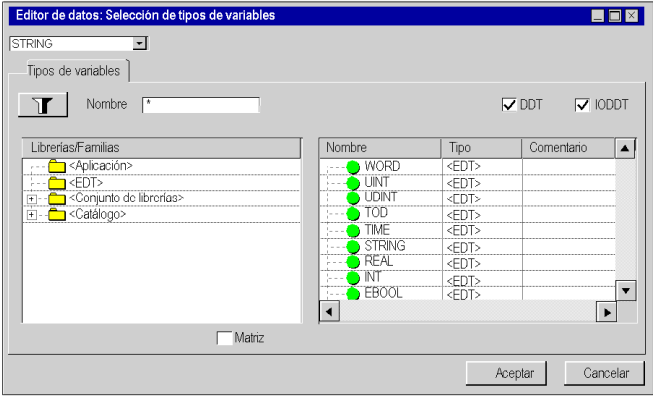
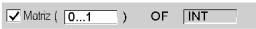


Declaración de un tipo de matriz

Antes de declarar un tipo de matriz, haga clic en **Herramientas/Ajustes del proyecto/Variables**; a continuación, active "Variables de matriz representadas directamente" y "Permitir matrices dinámicas".



La tabla siguiente muestra cómo declarar un tipo de matriz:

Paso	Acción
1	En el Explorador de proyectos, haga clic en Variables e instancias FB.
2	Haga clic en la columna Nombre e introduzca un nombre para la variable.
3	Haga doble clic en la columna Tipo y, a continuación, en el botón . Se abre la ventana Selección de tipos de variables: 
4	Elija el tipo de variable que se desee (por ejemplo, haga clic en <EDT> y seleccione INT) y, a continuación, haga clic en la casilla de verificación Matriz. 
5	Modifique el intervalo y confírmelo con Aceptar.

Declaración de objetos de E/S

Para declarar las variables derivadas de E/S, abra la ventana Selección de tipos de variables como se describe en el anterior procedimiento y haga clic en <Catálogo> para acceder a las variables de tipo <IODDT> (por ejemplo, seleccione T COM MB BMX); a continuación, confirme con Aceptar.

Uso de módems

Descripción

Es necesario conocer tres comandos para actuar como interfaz de módems telefónicos con los PLC. Estos comandos son los siguientes:

- inicializar módem,
- volver a numerar y
- desconectar módem.

Es obligatorio enviar un mensaje de inicialización seguido de un mensaje de marcación al módem antes de enviarle un mensaje ASCII o Modbus.

Cuando la conexión entre los dos módems se realiza correctamente, puede enviar un número ilimitado de mensajes ASCII o Modbus.

Cuando se han enviado todos los mensajes, debe enviar la cadena de desconexión al módem.

Inicialización del módem

Los dos módems deben estar configurados con las mismas características que los puertos serie:

- velocidad de transmisión de datos: 9600 baudios
- trama de caracteres: 8 bits/paridad par/1 bit de parada,
- modulación de línea: V32.

A continuación, defina + como carácter de escape (comando: ATS2=43).

Ejemplo de comando de inicialización:

ATQ0&Q0E0&K0V1

por:

- Q0: habilitar el código de resultado;
- &Q0: DTR siempre se asume (ON);
- E0: deshabilitar el eco de caracteres;
- &K0: sin control de flujo;
- V1: códigos de resultados de palabras.

Marcación del módem

Se utiliza el mensaje de marcación para enviar el número de teléfono al módem.

Sólo los comandos AT relativos a la marcación de un número deben incluirse en este mensaje.

Ejemplo:

- **Marcación por frecuencia:** ATDT6800326<CR><LR>
- **Marcación por pulsos:** ATDP6800326<CR><LF>
- **Marcación por frecuencia con tono de espera:** ATDTW6800326<CR><LF>

Desconexión del módem

En primer lugar, el módem vuelve a la modalidad de comando al recibir tres veces el carácter de escape.

A continuación, puede enviarse el comando de desconexión ATH0.

Secuencia de escape: "+++" (código de resultado del módem: Aceptar);

Comando de desconexión: "ATH0" (código de resultado del módem: Aceptar).

Procedimiento de programación

Procedimiento que debe seguirse

La matriz que figura a continuación muestra el procedimiento de programación de la aplicación.

Paso	Acción	Detalles
1	Preparación del puerto de comunicación.	● Cambie la modalidad de esclavo Modbus a la modalidad de caracteres enviando WRITE_CMD (consulte <i>(véase página 250)</i>) al puerto serie. ● Para una transmisión por módem, envíe el comando HAYES mediante el bloque PRINT_CHAR para configurar el módem <i>(véase página 247)</i>. ● Para una transmisión por módem, envíe el comando HAYES mediante el bloque PRINT_CHAR. Se utiliza el mensaje de marcación para enviar un número de teléfono al módem <i>(véase página 248)</i>.
2	Modalidad de maestro Modbus	● Cambie a la modalidad maestro Modbus mediante la función WRITE_CMD <i>(véase página 250)</i>. ● Envíe los datos que vayan a escribirse en el PLC esclavo.
3	Restablecimiento del puerto de comunicación.	● Cambie a la modalidad de caracteres mediante la función WRITE_CMD <i>(véase página 250)</i>. ● Para una transmisión por módem, envíe el carácter escape y, a continuación, envíe el comando de desconexión para que se envíe un mensaje de desconexión al módem <i>(véase página 248)</i> mediante el bloque PRINT_CHAR. ● Vuelva a la modalidad de inicio del puerto serie (esclavo Modbus) mediante el comando WRITE_CMD <i>(véase página 250)</i>.

Escritura de palabras de comando

Los pasos siguientes deben ejecutarse para enviar WRITE_CMD a un puerto de comunicación:

Paso	Acción	Detalle
1	Prueba para determinar si hay algún comando pendiente.	Antes de ejecutar WRITE_CMD, compruebe si hay algún intercambio en curso actualmente mediante el objeto de lenguaje EXCH_STS (%MW.r.m.c.0). Para actualizar esta palabra, utilice el bloque READ_STS.
2	Asignar la palabra de comando.	A continuación, debe modificar el valor del objeto de lenguaje de comando para realizar el comando necesario. Para una conexión Modbus, el objeto de lenguaje es la palabra interna CONTROL (%MW.r.m.c.24). Por ejemplo, para cambiar de la modalidad Modbus a la modalidad de caracteres, el bit 14 de la palabra %MW.r.m.c.24 se establece en 1. Nota: A continuación, un solo bit de comando debe cambiar de 0 a 1 antes de transmitir WRITE_CMD.
3	Enviar el comando	Finalmente, WRITE_CMD debe ejecutarse para confirmar el comando.

Estructura de programación

Comentarios de pasos

Número de paso	Descripción del paso	Elemento
0	Estado inicial de la función Cuando el bit de inicio cambia a 1, restablezca el error a 0 y vaya al paso 5.	Módem
5	Lea el estado del puerto serie y compruebe que no hay ningún comando activo. Cambie a la modalidad de caracteres y restablezca el contador Test_inc a 0. Vaya al paso 10.	
10	Lea el estado del puerto serie y compruebe que no hay ningún comando activo. Restablezca el bit de comando TO_CHAR_MODE. - Si no se produce ningún error en el puerto serie - y la modalidad de caracteres está activa, vaya al paso 15. - y la modalidad de caracteres está activa, incremente Test_inc y vuelva a intentar el paso 10 hasta un máximo de 1000 veces. Después de 1000 intentos fallidos, establezca Error en 10 y vaya al paso 130. - Si se produce un error en el puerto serie, - Establezca Error en 10. - Vaya al paso 130.	
15	Envíe un comando de marcación al módem a través del bloque PRINT_CHAR. Vaya al paso 20.	
20	Si el resultado de PRINT_CHAR es concluyente, vaya al paso 25; de lo contrario, establezca Error en 20 y vaya al paso 130.	
25	Espere la respuesta del módem a través del bloque INPUT_CHAR. Cuando la cadena de respuesta se haya recibido totalmente, vaya al paso 30.	
30	Si el resultado de INPUT_CHAR es concluyente, vaya al paso 35; de lo contrario, establezca Error en 30 y vaya al paso 130.	
35	Si el módem responde como se espera, vaya al paso 40; de lo contrario, establezca Error en 35 y vaya al paso 130.	

Número de paso	Descripción del paso	Elemento
40	Lea el estado del puerto serie y compruebe que no hay ningún comando activo. Cambie a la modalidad maestro Modbus y restablezca el contador Test_inc a 0. Vaya al paso 45.	Modalidad maestro Modbus
45	Lea el estado del puerto serie y compruebe que no hay ningún comando activo. Restablezca el bit de comando TO_CHAR_MODE. - Si no se produce ningún error en el puerto serie - y la modalidad de caracteres está activa, vaya al paso 50. - y la modalidad de caracteres está activa, incremente Test_inc y vuelva a intentar el paso 45 hasta un máximo de 1000 veces. Después de 1000 intentos fallidos, establezca Error en 45 y vaya al paso 130. - Si se produce un error en el puerto serie, - Establezca Error en 45. - Vaya al paso 130.	
50	Inicialización del parámetro del bloque WRITE_VAR. Envíe datos para escribirlos en el PLC mediante la función WRITE_VAR. Vaya al paso 55.	Función de escritura
55	Si el resultado de WRITE_VAR es concluyente, vaya al paso 60; de lo contrario, establezca Error en 55 y vaya al paso 130.	
60	Lea el estado del puerto serie y compruebe que no hay ningún comando activo. Cambie a la modalidad de caracteres y restablezca el contador Test_inc a 0. Vaya al paso 65.	Modalidad de caracteres
65	Lea el estado del puerto serie y compruebe que no hay ningún comando activo. Restablezca el bit de comando TO_CHAR_MODE. - Si no se produce ningún error en el puerto serie - y la modalidad de caracteres está activa, vaya al paso 70. - y la modalidad de caracteres está activa, incremente Test_inc y vuelva a intentar el paso 65 hasta un máximo de 1000 veces. Después de 1000 intentos fallidos, establezca Error en 65 y vaya al paso 130. - Si se produce un error en el puerto serie, - Establezca Error en 65. - Vaya al paso 130.	

Número de paso	Descripción del paso	Elemento
70	Envíe una secuencia de escape al módem mediante el bloque PRINT_CHAR. Vaya al paso 75.	Módem
75	Si el resultado de PRINT_CHAR es concluyente, vaya al paso 80; de lo contrario, establezca Error en 75 y vaya al paso 130.	
80	Espere la respuesta del módem a través del bloque INPUT_CHAR. Cuando la cadena de respuesta se haya recibido totalmente, vaya al paso 85.	
85	Si el resultado de INPUT_CHAR es concluyente, vaya al paso 90; de lo contrario, establezca Error en 85 y vaya al paso 130.	
90	Si el módem responde como se espera, vaya al paso 95; de lo contrario, establezca Error en 90 y vaya al paso 130.	
95	Envíe un comando de desconexión al módem mediante el bloque PRINT_CHAR. Vaya al paso 100.	
100	Si el resultado de PRINT_CHAR es concluyente, vaya al paso 105; de lo contrario, establezca Error en 100 y vaya al paso 130.	
105	Espere la respuesta del módem a través del bloque INPUT_CHAR. Cuando la cadena de respuesta se haya recibido totalmente, vaya al paso 110.	
110	Si el resultado de INPUT_CHAR es concluyente, vaya al paso 115; de lo contrario, establezca Error en 110 y vaya al paso 130.	
115	Si el módem responde como se espera, vaya al paso 120; de lo contrario, establezca Error en 115 y vaya al paso 130.	

Número de paso	Descripción del paso	Elemento
120	Lea el estado del puerto serie y compruebe que no hay ningún comando activo. Cambie a la modalidad esclavo Modbus y restablezca el contador Test_inc a 0. Vaya al paso 125.	Modo esclavo Modbus
125	Lea el estado del puerto serie y compruebe que no hay ningún comando activo. Restablezca el bit de comando TO_CHAR_MODE. - Si no se produce ningún error en el puerto serie - y la modalidad de caracteres está activa, vaya al paso 130. - y la modalidad de caracteres está activa, incremente Test_inc y vuelva a intentar el paso 125 hasta un máximo de 1000 veces. Después de 1000 intentos fallidos, establezca Error en 125 y vaya al paso 130. - Si se produce un error en el puerto serie, - Establezca Error en 125. - Vaya al paso 130.	
130	Vuelva al paso 0.	

Programación

Programación en lenguaje ST

El ejemplo está programado en lenguaje ST. La sección especializada se encuentra en la misma tarea maestra (MAST).

CASE Function_Step OF

0: (* Initialization *)

IF (Start) THEN (* trigger flag *)

Error := 0;

Function_Step := 5; (* next step *)

END_IF;

5: (* Send command to switch serial port from Slave Modbus mode to Character mode *)

READ_STS(Serial_port); (* read serial port status *)

IF (Serial_port.EXCH_STS = 0) THEN (* no active command *)

Serial_port.CONTROL := 16#00; (* reset control word *)

(* set TO_CHAR_MODE command bit *)

SET(Serial_port.TO_CHAR_MODE);

WRITE_CMD (Serial_port); (* send command *)

Test_inc := 0; (* initialize retry counter *)

Function_Step := 10; (* next step *)

END_IF;

10: (* Test result of switch command to Character mode*)

READ_STS(Serial_port); (* read serial port status *)

IF (Serial_port.EXCH_STS = 0) THEN (* command completed *)

(* reset TO_CHAR_MODE command bit *)

RESET(Serial_port.TO_CHAR_MODE);

IF (Serial_port.EXCH_RPT = 0) THEN (* no error *)

IF (AND(Serial_port.PROTOCOL, 16#0F) = 03)

THEN (* Character mode OK *)

Function_Step := 15; (* next step *)

ELSE

Test_inc := Test_inc + 1;

IF (Test_inc > 1000) THEN

```

        Error := 10; (* error *)
        Function_Step := 130; (* next step = end *)
    END_IF;
END_IF;
ELSE (* error in sending command to port *)
    Error := 10; (* error *)
    Function_Step := 130;
END_IF;
END_IF;

15: (* Send dial command to modem *)
(*Phone number must be inserted between 'ATDT' and '$N'*)
ReqString := 'ATDT4001$N'; (* dial message *)
MngtPrint[2] := 500; (* timeout *)
PRINT_CHAR(ADDM(Adr_Char), ReqString, MngtPrint);
Function_Step := 20;
20: (* Test PRINT_CHAR function result *)
IF (NOT MngtPrint[0].0) THEN
    IF (MngtPrint[1] = 0) THEN
        Function_Step := 25; (* success : next step *)
    ELSE
        Error := 20; (* error *)
        Function_Step := 130; (* next step = end *)
    END_IF;
END_IF;
25: (* Waiting for the response via INPUT_CHAR *)
MngtInput[2] := 500; (* timeout *)
AnsString1:= ' ';
(* wait modem reply *)
INPUT_CHAR(ADDM(Adr_Char), 1, nb_charac_to_receive_connect, MngtInput,
AnsString1);
Function_Step := 30; (* next step *)

```

```

30: (* Test INPUT_CHAR function result *)
IF (NOT MngtInput[0].0) THEN
  IF (MngtInput[1] = 0) THEN
    Function_Step := 35; (* success : next step *)
  ELSE
    Error := 30; (* error *)
    Function_Step := 130; (* next step = end *)
  END_IF;
END_IF;

35: (* Test Modem reply *)
IF (AnsString1 = '$NCONNET') THEN
  Function_Step := 40; (* success : next step *)
ELSE
  Error := 35; (* error *)
  Function_Step := 130; (* next step = end *)
END_IF;

40: (* Send command to switch serial port from character mode to Modbus Master *)
READ_STS(Serial_port); (* read serial port status *)
IF (Serial_port.EXCH_STS = 0) THEN (* no active command *)
  Serial_port.CONTROL := 16#00; (* reset control word *)
  (* set TO_MODBUS_MASTER command bit *)
  SET(Serial_port.TO_MODBUS_MASTER);
  WRITE_CMD (Serial_port); (* send command *)
  Test_inc := 0; (* initialize retry counter *)
  Function_Step := 45; (* next step *)
END_IF;

45: (* Test result of switch command to Modbus Master mode*)
READ_STS(Serial_port); (* read serial port status *)
IF (Serial_port.EXCH_STS = 0) THEN (* command completed *)
  (* TO_MODBUS_MASTER command bit *)

```

```

RESET(Serial_port.TO_MODBUS_MASTER);
IF (Serial_port.EXCH_RPT = 0) THEN (* no error *)
  IF (AND(Serial_port.PROTOCOL, 16#0F) = 06)
    THEN (* Modbus Master mode OK *)
      Function_Step := 50; (* next step *)
    ELSE
      Test_inc := Test_inc + 1;
      IF (Test_inc > 1000) THEN
        Error := 45; (* error *)
        Function_Step := 130; (* next step = end *)
      END_IF;
    END_IF;
  ELSE (* error in sending command to port *)
    Error := 45; (* error *)
    Function_Step := 130;
  END_IF;
END_IF;

50: (*Write information in the second CPU*)
Mngtwrite[2]:=50; (* time outs*)
%MW40:=5; (* value to send *)
WRITE_VAR(ADDM(Adr_modbus),'%MW',100,2,%MW40:2,Mngtwrite);
Function_Step := 55;

55: (* Test WRITE_VAR function result *)
IF (NOT Mngtwrite[0].0) THEN
  IF (Mngtwrite[1] = 0) THEN
    Function_Step := 60; (* success : next step *)
  ELSE
    Error := 55; (* error *)
    Function_Step := 130; (* next step = end *)
  END_IF;
END_IF;

```

```

60: (* Send command to switch serial port from Modbus to character mode *)
READ_STS(Serial_port); (* read serial port status *)
IF (Serial_port.EXCH_STS = 0) THEN (* no active command *)
    Serial_port.CONTROL := 16#00; (* reset control word *)
    (* set TO_CHAR_MODE command bit *)
    SET(Serial_port.TO_CHAR_MODE);
    WRITE_CMD (Serial_port); (* send command *)
    Test_inc := 0; (* initialize retry counter *)
    Function_Step := 65; (* next step *)
END_IF;

65: (* Test result of switch command *)
READ_STS(Serial_port); (* read serial port status *)
IF (Serial_port.EXCH_STS = 0) THEN (* command completed *)
    (* reset TO_CHAR_MODE command bit *)
    RESET(Serial_port.TO_CHAR_MODE);
    IF (Serial_port.EXCH_RPT = 0) THEN (* no error *)
        IF (AND(Serial_port.PROTOCOL, 16#0F) = 03)
            THEN (* character mode OK *)
                Function_Step := 70; (* next step *)
            ELSE
                Test_inc := Test_inc + 1;
                IF (Test_inc > 1000) THEN
                    Error := 65; (* error *)
                    Function_Step := 130; (* next step = end *)
                END_IF;
            END_IF;
        ELSE (* error in sending command to port *)
            Error := 65; (* error *)
            Function_Step := 130; (* next step = end *)
        END_IF;
    END_IF;
END_IF;

```

```

70: (* Hangup modem: step 1 *)
   ReqString := '+++'; (* escape sequence *)
   PRINT_CHAR(ADDM(Adr_Char), ReqString, MngtPrint);
   Function_Step := 75; (* next step *)

75: (* Test PRINT_CHAR function result *)
   IF (NOT MngtPrint[0].0) THEN
     IF (MngtPrint[1] = 0) THEN
       (* Success : next step *)
       Function_Step := 80;
     ELSE
       (* End on error *)
       Error := 75;
       Function_Step := 130;
     END_IF;
   END_IF;

80:
   MngtInput[2] := 50; (* timeout *)
   INPUT_CHAR(ADDM(Adr_Char), 1, nb_charac_to_receive_ok, MngtInput,
AnsString2); (*Wait modem reply*)
   Function_Step := 85; (* next step *)

85: (* Test INPUT_CHAR function result *)
   IF (NOT MngtInput[0].0) THEN
     IF (MngtInput[1] = 0) THEN
       (* Success : next step *)
       Function_Step := 90;
     ELSE
       (* End on error *)
       Error := 85;
       Function_Step := 130;
     END_IF;
   END_IF;

```

```

90: (* Test Modem reply *)
IF (AnsString2 = '$NOK') THEN
    Function_Step := 95; (* success : next step *)
ELSE
    Error := 90; (* error *)
    Function_Step := 130; (* next step = end *)
END_IF;
95: (* Hangup modem: step 2 *)
ReqString := 'ATH0$N'; (* hangup message *)
PRINT_CHAR(ADDM(Adr_Char), ReqString, MngtPrint);
Function_Step := 100; (* next step *)
100: (* Test PRINT_CHAR function result *)
IF (NOT MngtPrint[0].0) THEN
    IF (MngtPrint[1] = 0) THEN
        (* Success : next step *)
        Function_Step := 105;
    ELSE
        (* End on error *)
        Error := 100;
        Function_Step := 130;
    END_IF;
END_IF;
105:
MngtInput[2] := 50; (* timeout *)
INPUT_CHAR(ADDM(Adr_Char), 1, nb_charac_to_receive_ok, MngtInput,
AnsString3); (*Wait modem reply*)
Function_Step := 110; (* next step *)
110: (* Test INPUT_CHAR function result *)
IF (NOT MngtInput[0].0) THEN
    IF (MngtInput[1] = 0) THEN
        (* Success : next step *)
        Function_Step := 115;
    ELSE
        (* End on error *)

```

```

        Error := 110;
        Function_Step := 130;
    END_IF;
END_IF;
115: (* Test Modem reply *)
IF (AnsString3 = '$NOK') THEN
    Function_Step := 120; (* success : next step *)
ELSE
    Error := 115; (* error *)
    Function_Step := 130; (* next step = end *)
END_IF;
120: (* Send command to switch serial port from Character mode to Slave Modbus
mode *)
READ_STS(Serial_port); (* read serial port status *)
IF (Serial_port.EXCH_STS = 0) THEN (* no activecommand *)
    Serial_port.CONTROL := 16#00; (* reset control word *)
    (* set TO_MODBUS_SLAVE command bit *)
    SET(Serial_port.TO_MODBUS_SLAVE);
    WRITE_CMD (Serial_port); (* send command *)
    Test_inc := 0; (* initialize retry counter *)
    Function_Step := 125; (* next step *)
END_IF;

125: (* Test result of switch command *)
READ_STS(Serial_port); (* read serial port status *)
IF (Serial_port.EXCH_STS = 0) THEN (* command completed *)
    (* reset TO_MODBUS_SLAVE command bit *)
    RESET(Serial_port.TO_MODBUS_SLAVE);
    IF (Serial_port.EXCH_RPT = 0) THEN (* no error *)
        IF (AND(Serial_port.PROTOCOL, 16#0F) = 07)
            THEN (* character mode OK *)
                Function_Step := 130; (* next step *)
            ELSE
                Test_inc := Test_inc + 1;

```

```
    IF (Test_inc > 1000) THEN
        Error := 125; (* error *)
        Function_Step := 130; (* next step = end *)
    END_IF;
END_IF;
ELSE (* error in sending command to port *)
    Error := 125; (* error *)
    Function_Step := 130; (* next step = end *)
END_IF;
END_IF;
130: (* End *)
IF (NOT Start) THEN (* trigger flag *)
    Function_Step := 0; (* goto waiting state *)
END_IF;
END_CASE;
```

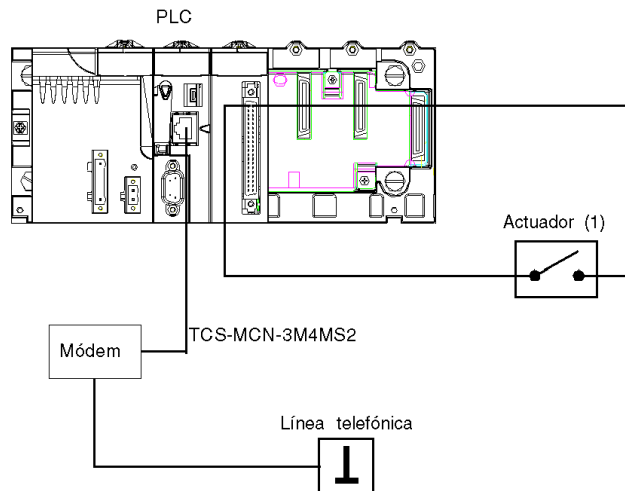

Ejecución de la aplicación en modalidad estándar

Presentación

En Este ejemplo, el funcionamiento en la modalidad estándar requiere el uso de dos PLC, un módulo de entradas binarias, un módulo BMX NOM 0200 y dos módems SR2MOD01.

Cableado del primer PLC esclavo

El primer PLC esclavo está conectado así:



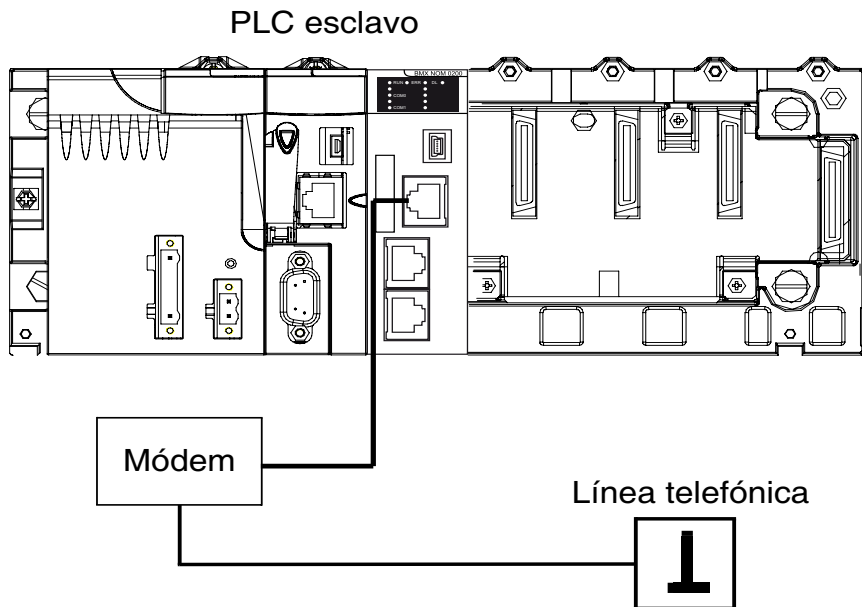
(1): el actuador está conectado al canal 0 del módulo binario.

En este ejemplo, el primer módem se conecta al puerto serie del procesador del primer PLC esclavo.

El estado del actuador controla el estado de la variable `Start` en la aplicación.

Cableado del segundo PLC esclavo

El segundo PLC esclavo está conectado así:



En este ejemplo, el segundo módem está conectado al canal 0 del módulo BMX NOM 0200 del segundo PLC esclavo.

Para obtener una mayor fiabilidad en las comunicaciones, se utiliza el cable TCS XCN 3M4F3S4 para la gestión de señales de módem DTR/DSR/DCD realizada por la aplicación.

Configuración del segundo PLC esclavo

Antes de transferir el proyecto para configurar el segundo PLC esclavo, compruebe que el primer PLC esclavo no está conectado al módem.

En la tabla siguiente se presenta el procedimiento que debe seguirse para transferir el proyecto en modalidad estándar:

Paso	Acción
1	En el menú PLC , haga clic en Modalidad estándar .
2	En el menú Generar , haga clic en Regenerar todos los proyectos . El proyecto se generará y estará listo para transferirse al PLC.
3	En el menú PLC , haga clic en Conectar . Ahora está conectado al PLC.
4	En el menú PLC , haga clic en Transferir proyecto a PLC . Aparece la ventana Transferir proyecto al PLC . Haga clic en Transferir . La aplicación se transfiere al PLC.
5	Conecte el segundo PLC esclavo a un módem SR2MOD01.

Transferencia de aplicaciones al primer PLC esclavo

Antes de transferir la aplicación, compruebe que el primer PLC esclavo no está conectado al módem.

En la tabla siguiente se presenta el procedimiento que debe seguirse para transferir la aplicación en modalidad estándar:

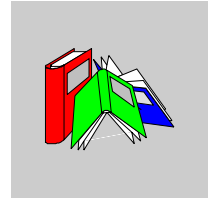
Paso	Acción
1	En el menú PLC , haga clic en Modalidad estándar .
2	En el menú Generar , haga clic en Regenerar todos los proyectos . El proyecto se generará y estará listo para transferirse al PLC. Después de generar el proyecto, aparecerá una ventana de resultados. Si existe un error en el programa, Unity Pro indica su ubicación (al hacer clic en la secuencia resaltada).
3	En el menú PLC , haga clic en Conectar . Ahora está conectado al PLC.
4	En el menú PLC , haga clic en Transferir proyecto a PLC . Aparece la ventana Transferir proyecto al PLC . Haga clic en Transferir . La aplicación se transfiere al PLC.

Ejecución de aplicaciones en el primer PLC esclavo

En la tabla siguiente se presenta el procedimiento que debe seguirse para ejecutar la aplicación en modalidad estándar:

Paso	Acción
1	En el menú PLC, haga clic en Ejecutar. Aparece la ventana Ejecutar. Haga clic en Aceptar. La aplicación se ejecutará en el PLC.
2	Desconecte el PC que está ejecutando el software Unity Pro desde el primer PLC esclavo.
3	Conecte el primer PLC esclavo a un módem SR2MOD01.

Glosario



0-9

%I

Según la norma IEC, %I indica un objeto de lenguaje de entrada binaria.

%IW

Según la norma IEC, %IW indica un objeto de lenguaje de entrada analógica.

%KW

Según la norma IEC, %KW indica un objeto de lenguaje de palabra constante.

%M

Según la norma IEC, %M indica un objeto de lenguaje de bit de memoria.

%MW

Según la norma IEC, %MW indica un objeto de lenguaje de palabra de memoria.

%Q

Según la norma IEC, %Q indica un objeto de lenguaje de salida binaria.

%QW

Según la norma IEC, %QW indica un objeto de lenguaje de salida analógica.

A

Altivar

unidad de velocidad variable CA.

ARRAY

Una ARRAY es una tabla que contiene elementos del mismo tipo. La sintaxis es la siguiente: ARRAY [<límites>] OF <Tipo>Ejemplo:ARRAY [1..2] OF BOOL es una tabla de una dimensión compuesta por dos elementos de tipo BOOL.ARRAY [1..10, 1..20] OF INT es una tabla de dos dimensiones compuesta por 10 x 20 elementos de tipo INT.

ASCII

ASCII son las siglas de American Standard Code for Information Interchange (código normalizado americano para el intercambio de información). Se trata de un código americano (pero que se ha convertido en una norma internacional) que utiliza 7 bits para definir cada carácter alfanumérico que se utiliza en inglés, los símbolos de puntuación, algunos caracteres gráficos y otros comandos de distinta naturaleza.

B

BOOL

BOOL es la forma abreviada del tipo booleano. Se trata del tipo de datos básico en informática. Una variable de tipo BOOL posee uno de estos valores: 0 (FALSO) o 1 (VERDADERO). Un bit extraído de la palabra es de tipo BOOL, por ejemplo: %MW10.4.

BYTE

Cuando se reagrupan ocho bits, se habla de BYTE. La entrada de un BYTE se realiza en modalidad binaria o en base 8. El tipo BYTE está codificado en un formato de 8 bits que, en el formato hexadecimal, va de 16#00 a 16#FF.

C

Concentrador

Dispositivo que conecta una serie de módulos flexibles y centralizados para crear una red.

Conector

Asociación de un puerto a una dirección IP que funciona como una identificación del remitente o el receptor.

Configuración

La configuración recopila los datos que caracterizan a la máquina (no variables) y que son necesarios para que el módulo funcione. Toda esta información se almacena en el área %KW del PLC constante. La aplicación del PLC no puede modificarla.

CPU

Son las siglas de unidad central de proceso (Control Processing Unit): Nombre genérico usado para los procesadores de Schneider Electric.

CRC

CRC son las siglas de Comprobación de redundancia cíclica (del inglés, Cyclic Redundancy Check): Indica si no se ha "deformado" ningún carácter durante la transmisión de las tramas.

D**DFB**

DFB es la forma abreviada de "Derived Function Block" (bloque de funciones derivado). Los tipos DFB son bloques de funciones programables por el usuario en lenguaje ST (texto estructurado), IL (lista de instrucciones), LD (diagrama Ladder) o FBD (diagrama de bloques de funciones). El uso de estos tipos DFB en una aplicación permite:

- Simplificar la concepción y la entrada del programa.
- Aumentar la legibilidad del programa.
- Facilitar su depuración.
- Reducir el volumen de código generado.

Difusión

Las comunicaciones mediante difusión envían paquetes de una estación a todos los destinos de la red. Los mensajes de difusión pertenecen a todos los dispositivos de la red o sólo a un dispositivo del que no se conoce la dirección.

DINT

DINT es la forma abreviada de "Double INTegeR" (entero doble) (codificado en 32 bits). Los límites inferior y superior figuran a continuación: de -(2 elevado a 31) a (2 elevado a 31) - 1. Ejemplo:-2147483648, 2147483647, 16#FFFFFFFF.

Dirección

En una red, identificación de una estación. En una trama, agrupación de bits que identifica el origen o destino de la trama.

E

EBOOL

EBOOL es la versión abreviada de "Extended BOOLean" (booleano extendido). Una variable de tipo EBOOL posee el valor 0 (FALSE) o 1 (TRUE), pero igualmente los flancos ascendentes o descendentes y las funciones de forzado. Una variable de tipo EBOOL ocupa un byte de memoria. El byte se compone de los siguientes elementos:

- Un bit del valor.
- Un bit del historial (cada vez que cambia el objeto de estado, el valor se copia en el bit del historial).
- un bit de forzado (igual a 0 si el objeto no se fuerza, igual a 1 si el bit se fuerza).

El valor predeterminado de cada bit es 0 (FALSE).

EF

EF es la forma abreviada de "Elementary Function" (función elemental). Se trata de un bloque, utilizado en un programa, que realiza una función de software predefinida. Una función no dispone de información sobre el estado interno. Varias llamadas de la misma función con los mismos parámetros de entrada muestran siempre los mismos valores de salida. Encontrará información sobre la forma gráfica de la llamada de la función en el "[bloque funcional (instancia)]". A diferencia de las llamadas de bloques de funciones, las llamadas de función constan únicamente de una salida a la que no se ha asignado un nombre y cuyo nombre coincide con el de la función. En FBD, cada llamada se indica mediante un [número] único a través del bloque gráfico. Este número se genera automáticamente y no puede modificarse. Coloque y configure estas funciones en el programa para ejecutar su aplicación. También puede desarrollar otras funciones mediante el kit de desarrollo de SDKC.

F

FBD

FBD es la forma abreviada de "Function Block Diagram" (lenguaje en bloques funcionales). FBD es un lenguaje de programación gráfica que funciona como un diagrama. Mediante la adición de bloques lógicos simples (Y, O, etc.), cada función o bloque de funciones del programa se representa mediante esta forma gráfica. En cada bloque, las entradas se sitúan a la izquierda y las salidas, a la derecha. Las salidas de los bloques pueden estar vinculadas a las entradas de otros bloques para formar expresiones complejas.

Fipio

Bus de campo usado para conectar dispositivos de tipo sensor o actuador.

Full duplex

Método de transmisión de datos capaz de transmitir y recibir sobre el mismo canal simultáneamente.

H

Half duplex

Método de transmisión de datos capaz de comunicarse en cualquiera de las dos direcciones, pero sólo en una dirección en cada momento.

I

INT

INT es la forma abreviada de "single INTEger" (entero simple) (codificado en 16 bits). Los límites inferior y superior figuran a continuación: de $-(2 \text{ elevado a } 15)$ a $(2 \text{ elevado a } 15) - 1$. Ejemplo: -32768, 32767, 2#1111110001001001, 16#9FA4.

IODDT

IODDT es la forma abreviada de "Input/Output Derived Data Type" (tipo de datos derivados de E/S). El término IODDT designa un tipo de datos estructurado que representa un módulo o un canal de un módulo del PLC. Cada módulo experto posee sus propios IODDT.

L

LED

LED son las siglas de diodo electroluminiscente. Un indicador que se enciende cuando pasa electricidad a través de él. Indica el estado de funcionamiento de un módulo de comunicación.

LRC

LRC son las siglas de Comprobación de redundancia longitudinal (del inglés, Longitudinal Redundancy Check); se ha diseñado para solucionar la baja probabilidad de detección de errores de la comprobación de paridad.

M

Memoria FLASH

La memoria FLASH es una memoria no volátil que se puede sobrescribir. Se almacena en una memoria EEPROM especial que se puede borrar y volver a programar.

Módulo binario

Módulo con entradas/salidas binarias.

Momentum

Módulos de E/S que utilizan varias redes de comunicaciones abiertas estándar.

P

PLC

PLC son las siglas de Controlador lógico programable (del inglés, Programmable Logic Controller). Un PLC es el cerebro de un proceso de fabricación industrial. Automatiza un proceso a diferencia de los sistemas de control por relés. Los PLC son ordenadores adaptados para sobrevivir a las duras condiciones del entorno industrial.

Protocolo

Describe los formatos de los mensajes y establece las reglas que usan dos o más dispositivos para comunicarse mediante esos formatos.

R

Red

Hay dos significados del término "red".

- En LD (diagrama Ladder): una red es un conjunto de elementos gráficos interconectados. El campo de aplicación de una red es local, en relación con la unidad (sección) de organización del programa en el que se encuentra la red.
- Con módulos de comunicación expertos: una red es un grupo de estaciones que se intercomunican. El término "red" se utiliza también para definir un grupo de elementos gráficos interconectados. Dicho grupo constituye una parte de un programa que puede componerse de un grupo de redes.

RS232

Comunicación serie estándar que define el voltaje del servicio siguiente:

- Una señal de +12 V indica un 0 lógico.
- Una señal de -12 V indica un 1 lógico.

Sin embargo, en caso de que se atenúe la señal, existe una detección que se proporciona hasta los límites de -3 V y +3 V. Entre estos dos límites, la señal se considerará no válida. Las conexiones RS232 son bastante sensibles a las interferencias. La normativa especifica que no se debe superar la distancia de 15 m ni los 9.600 baudios (bits/s).

RS485

Normativa de conexión serie que funciona en un diferencial de 10 V/+5 V. Utiliza dos conductores para enviar/recibir. Las salidas de «3 estados» les permiten conectarse a la modalidad de escucha cuando finaliza la transmisión.

RTU

Son las siglas de unidad de terminal remoto (del inglés "Remote Terminal Unit"): en la modalidad RTU, los datos se envían como dos caracteres hexadecimales de cuatro bits que proporcionan un mayor rendimiento que el modo ASCII para la misma velocidad de transmisión. RTU Modbus es un protocolo binario y depende más del retardo que el protocolo ASCII.

S

Sección

Módulo de programa que pertenece a una tarea que se puede escribir en el lenguaje elegido por el programador (FBD, LD, ST, IL o SFC). Una tarea puede estar compuesta por distintas secciones; el orden de ejecución de estas secciones corresponde a su orden de creación. Este orden puede modificarse.

SEPAM

Relé de protección digital para la protección, el control y la supervisión de los sistemas de alimentación.

ST

ST son las siglas del lenguaje de «texto estructurado» (del inglés Structured Text). El lenguaje literal estructurado es un lenguaje elaborado similar a los lenguajes informáticos de programación. Permite estructurar series de instrucciones.

STRING

Una variable de tipo STRING es una cadena de caracteres ASCII. La longitud máxima de una cadena de caracteres es de 65.534 caracteres.

T

TAP

TAP son las siglas de «punto de acceso de transmisión» (del inglés Transmission Access Point): la unidad de conexión del bus.

Tarea

Grupo de secciones y subrutinas ejecutadas cíclica o periódicamente si se trata de la tarea MAST, o periódicamente si se trata de la tarea FAST. Una tarea siempre tiene un nivel de prioridad y tiene asociadas entradas y salidas del PLC. Estas entradas y salidas se actualizarán en consecuencia.

Tarea maestra

Tarea principal del programa. Es obligatoria y se utiliza para ejecutar un procesamiento secuencial del PLC.

Trama

Grupo de bits que forman un bloque de información binario. Las tramas contienen información o datos de control de la red. El tamaño y la composición de una trama están determinados por la tecnología de red utilizada.

U**Unity Pro**

Software de programación del PLC de Schneider Automation.

V**Variable**

Entidad de memoria del tipo BOOL, WORD, DWORD, etc., cuyo contenido puede modificarse mediante el programa que está ejecutándose.

W**WORD**

El tipo WORD se codifica en un formato de 16 bits y se utiliza para realizar tratamientos en las cadenas de bits.

Esta tabla establece los límites inferior y superior de las bases que pueden utilizarse:

Base	Límite inferior	Límite superior
Hexadecimal	16#0	16#FFFF
Octal	8#0	8#177777
Binario	2#0	2#1111111111111111

Ejemplos de representación:

Datos	Representación en una de las bases
0000000011010011	16#D3
1010101010101010	8#125252
0000000011010011	2#11010011

X

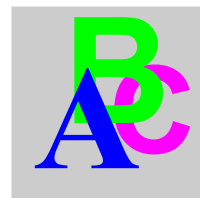
XBT

Terminal del operador gráfico.

XPS

Módulo de seguridad usado para procesar señales de seguridad para supervisar los componentes y el cableado de un sistema de seguridad, incluidos los dispositivos para la supervisión general, así como modelos específicos para aplicaciones.

Índice



B

BMXNOM0200, 19
BMXP341000, 19
BMXP342000, 19
BMXP342010, 19
BMXP3420102, 19
BMXP342020, 19
bus Modbus, 67

C

cableado, 56
cableado, accesorios, 56
cambiar protocolos, 226, 229
canal, estructura de datos para la comunicación en modalidad de caracteres
 T_COM_CHAR_BMX, 216
canal, estructura de datos para la comunicación Modbus
 T_COM_MB_BMX, 206, 208
canal, estructura de datos para los protocolos de comunicación
 T_COM_STS_GEN, 201, 202
canal, estructura de datos para todos los módulos
 T_GEN_MOD, 222
conexión, dispositivos, 35
configurar modalidad de caracteres, 106
configurar, Modbus, 76

D

depurar modalidad de caracteres, 122
depurar, Modbus, 99, 159

E

estructura de datos de canal de todos los módulos
 T_GEN_MOD, 222
estructura de datos de canal para la comunicación en modalidad de caracteres
 T_COM_CHAR_BMX, 217

I

inicio rápido, 231
INPUT_CHAR, 116, 176

M

M340
 Endurecido, 30
 reforzado, 30
modalidad de caracteres, 101

P

parámetros, ajuste, 191
PRINT_CHAR, 116, 176
programar bus Modbus, 88
programar modalidad de caracteres, 116

puesta a tierra, *31*

T

T_COM_CHAR_BMX, *216, 217*

T_COM_MB_BMX, *206, 208*

T_COM_STS_GEN, *201, 202*

T_GEN_MOD, *222, 222*