

Modicon X80

Seriellles Verbindungsmodul

BMXNOM0200

Benutzerhandbuch

Übersetzung der Originalbetriebsanleitung

09/2020

Die Informationen in der vorliegenden Dokumentation enthalten allgemeine Beschreibungen und/oder technische Leistungsmerkmale der hier erwähnten Produkte. Diese Dokumentation dient keinesfalls als Ersatz für die Ermittlung der Eignung oder Verlässlichkeit dieser Produkte für bestimmte Verwendungsbereiche des Benutzers und darf nicht zu diesem Zweck verwendet werden. Jeder Benutzer oder Integrator ist verpflichtet, angemessene und vollständige Risikoanalysen, Bewertungen und Tests der Produkte im Hinblick auf deren jeweils spezifischen Verwendungszweck vorzunehmen. Weder Schneider Electric noch deren Tochtergesellschaften oder verbundene Unternehmen sind für einen Missbrauch der Informationen in der vorliegenden Dokumentation verantwortlich oder können diesbezüglich haftbar gemacht werden. Verbesserungs- und Änderungsvorschläge sowie Hinweise auf angetroffene Fehler werden jederzeit gern entgegengenommen.

Sie erklären, dass Sie ohne schriftliche Genehmigung von Schneider Electric dieses Dokument weder ganz noch teilweise auf beliebigen Medien reproduzieren werden, ausgenommen zur Verwendung für persönliche nichtkommerzielle Zwecke. Darüber hinaus erklären Sie, dass Sie keine Hypertext-Links zu diesem Dokument oder seinem Inhalt einrichten werden. Schneider Electric gewährt keine Berechtigung oder Lizenz für die persönliche und nichtkommerzielle Verwendung dieses Dokument oder seines Inhalts, ausgenommen die nichtexklusive Lizenz zur Nutzung als Referenz. Das Handbuch wird hierfür „wie besehen“ bereitgestellt, die Nutzung erfolgt auf eigene Gefahr. Alle weiteren Rechte sind vorbehalten.

Bei der Montage und Verwendung dieses Produkts sind alle zutreffenden staatlichen, landesspezifischen, regionalen und lokalen Sicherheitsbestimmungen zu beachten. Aus Sicherheitsgründen und um die Übereinstimmung mit dokumentierten Systemdaten besser zu gewährleisten, sollten Reparaturen an Komponenten nur vom Hersteller vorgenommen werden.

Beim Einsatz von Geräten für Anwendungen mit technischen Sicherheitsanforderungen sind die relevanten Anweisungen zu beachten.

Die Verwendung anderer Software als der Schneider Electric-eigenen bzw. einer von Schneider Electric genehmigten Software in Verbindung mit den Hardwareprodukten von Schneider Electric kann Körperverletzung, Schäden oder einen fehlerhaften Betrieb zur Folge haben.

Die Nichtbeachtung dieser Informationen kann Verletzungen oder Materialschäden zur Folge haben!

© 2020 Schneider Electric. Alle Rechte vorbehalten.



	Sicherheitshinweise	7
	Über dieses Buch	11
Teil I	Hardwaretechnische Installation für die serielle Modbus- und Zeichenmoduskommunikation	15
Kapitel 1	Einführung in die serielle Kommunikation	17
	Kommunikation im seriellen Modbus- und im Zeichenmodus	18
	Beschreibung des BMXNOM0200 Moduls	20
	Abmessungen des X80-SL-Moduls (Serielle Verbindung) BMXNOM0200(H)	26
	Normen und Zertifizierungen	27
	Installation des Moduls BMXNOM0200	28
Kapitel 2	Serielle Kommunikationsarchitekturen	33
	Modbus-Leitungsabschluss und Polarisierung (RS485)	34
	Verbindung von Modbus-Geräten (RS485)	36
	Anschließen von Datenendeinrichtungen (DEE) (RS232)	38
	Anschließen von Datenübertragungseinrichtungen (DÜE) (RS232) ..	40
	Verkabelung	43
Teil II	Softwaretechnische Implementierung der seriellen Modbus- und Zeichenmoduskommunikation	47
Kapitel 3	BMXNOM0200 - Einschränkungen und Implementierungsregeln	49
	Beschränkungsregeln für BMXNOM0200-Module	50
	Implementierungsregeln für BMXNOM0200	52
Kapitel 4	Serielle Modbus-Kommunikation	55
4.1	Allgemeine Informationen	56
	Über Modbus Serial	57
	Leistung	58
	Zugriff auf die seriellen Verbindungsparameter	60
4.2	Konfiguration der seriellen Modbus-Kommunikation	63
	Konfigurationsfenster für die serielle Modbus-Kommunikation	64
	Anwendungsrelevante Modbus-Parameter	66
	Parameter der Signal- und der physischen Leitung im Modbus-Modus ..	68
	Übertragungsbezogene Modbus-Parameter	71
	Einstellen der BMXNOM0200-MODBUS-Slaveadresse ohne Control Expert?	74

4.3	Programmierung der seriellen Modbus-Kommunikation	76
	Von einem Master-Modul in einer Modbus-Verbindung unterstützte Dienste	77
	Von einem Slave-Modul in einer Modbus-Verbindung unterstützte Dienste	78
	Details zum Modbus Experten-Modus	80
4.4	Debuggen einer seriellen Modbus-Kommunikation	88
	Debug-Fenster der seriellen Modbus-Kommunikation	88
Kapitel 5	Kommunikation im Zeichenmodus	91
5.1	Allgemeine Informationen	92
	Informationen zur Kommunikation im Zeichenmodus	92
5.2	Konfiguration der Zeichenmoduskommunikation	93
	BMXNOM0200 Konfigurationsfenster für die Zeichenmodus-Kommunikation	94
	Parameter zur Erkennung des Nachrichtenendes im Zeichenmodus	96
	Übertragungsparameter im Zeichenmodus	98
	Parameter der Signal- und der physischen Leitung im Zeichenmodus	100
5.3	Programmierung der Zeichenmoduskommunikation	103
	Zeichenmodus-Kommunikationsfunktionen	104
	Details zum Experten-Modus für den Zeichenmodus	109
5.4	Debuggen der Zeichenmoduskommunikation	112
	Debug-Fenster der Zeichenmoduskommunikation	112
Kapitel 6	BMXNOM0200 Moduldiagnose	115
	Detaillierte Diagnose nach Kommunikationskanal	116
	Diagnose eines BMXNOM0200-Moduls	118
Kapitel 7	Sprachobjekte der Modbus- und Zeichenmoduskommunikation	121
7.1	Sprachobjekte und IODDTs der Modbus- und Zeichenmoduskommunikation	122
	Einführung in die Sprachobjekte für die Modbus- und Zeichenmodus-Kommunikation	123
	Implizite Austauschsprachobjekte der anwendungsspezifischen Funktion	124
	Explizite Austauschsprachobjekte der anwendungsspezifischen Funktion	125
	Verwaltung von Austauschvorgängen und Berichten mit expliziten Objekten	127

7.2	Allgemeine Sprachobjekte und IODDTs für Kommunikationsprotokolle	130
	Beschreibung der impliziten IODDT-Austauschobjekte vom Typ T_COM_STS_GEN	131
	Beschreibung der expliziten IODDT-Austauschobjekte vom Typ T_COM_STS_GEN	132
7.3	Sprachobjekte und IODDTs der Modbus-Kommunikation	134
	Beschreibung der expliziten Austauschsprachobjekte für eine Modbus- Funktion	135
	Beschreibung der impliziten IODDT-Austauschobjekte vom Typ T_COM_MB_BMX und T_COM_MB_BMX_CONF_EXT	136
	Beschreibung der expliziten IODDT-Austauschobjekte vom Typ T_COM_MB_BMX und T_COM_MB_BMX_CONF_EXT	137
	Beschreibung der Sprachobjekte für die Konfiguration des Modbus- Modus	140
7.4	Sprachobjekte und IODDTs der Zeichenmoduskommunikation	142
	Beschreibung der expliziten Austauschsprachobjekte für die Kommunikation im Zeichenmodus	143
	Beschreibung der impliziten IODDT-Austauschobjekte vom Typ T_COM_CHAR_BMX	144
	Beschreibung der expliziten IODDT-Austauschobjekte vom Typ T_COM_CHAR_BMX	145
	Beschreibung der Sprachobjekte für die Konfiguration des Zeichenmodus	148
7.5	IODDT Type T_GEN_MOD, anwendbar auf alle Module	150
	Beschreibung der Sprachobjekte des IODDT vom Typ T_GEN_MOD	150
7.6	Sprachobjekte und gerätespezifische IODDTs der Modbus- Kommunikation	152
	BMX NOM 0200.x-Geräte-DDT	153
	Beschreibung des Bytes MOD_FLT	156
Kapitel 8	Dynamischer Protokollwechsel	157
	Ändern des Protokolls mit BMXNOM0200-Modulen	157
Teil III	Schnellstart: BMXNOM0200 als Modbus-Slave über eine Quantum SPS	161
Kapitel 9	Übersicht	163
	Produktübersicht	164
	Übersicht über die Architektur	165
	Einschränkungen	167

Kapitel 10	Konfiguration in Control Expert	169
	Einsetzen des Moduls	170
	Fenster "Module Configuration"	171
Glossar		175
Index		183



Wichtige Informationen

HINWEISE

Lesen Sie sich diese Anweisungen sorgfältig durch und machen Sie sich vor Installation, Betrieb, Bedienung und Wartung mit dem Gerät vertraut. Die nachstehend aufgeführten Warnhinweise sind in der gesamten Dokumentation sowie auf dem Gerät selbst zu finden und weisen auf potenzielle Risiken und Gefahren oder bestimmte Informationen hin, die eine Vorgehensweise verdeutlichen oder vereinfachen.



Wird dieses Symbol zusätzlich zu einem Sicherheitshinweis des Typs „Gefahr“ oder „Warnung“ angezeigt, bedeutet das, dass die Gefahr eines elektrischen Schlags besteht und die Nichtbeachtung der Anweisungen unweigerlich Verletzung zur Folge hat.



Dies ist ein allgemeines Warnsymbol. Es macht Sie auf mögliche Verletzungsgefahren aufmerksam. Beachten Sie alle unter diesem Symbol aufgeführten Hinweise, um Verletzungen oder Unfälle mit Todesfälle zu vermeiden.



GEFAHR

GEFAHR macht auf eine gefährliche Situation aufmerksam, die, wenn sie nicht vermieden wird, Tod oder schwere Verletzungen **zur Folge hat**.



WARNUNG

WARNUNG macht auf eine gefährliche Situation aufmerksam, die, wenn sie nicht vermieden wird, Tod oder schwere Verletzungen **zur Folge haben kann**.



VORSICHT

VORSICHT macht auf eine gefährliche Situation aufmerksam, die, wenn sie nicht vermieden wird, leichte Verletzungen **zur Folge haben kann**.

HINWEIS

HINWEIS gibt Auskunft über Vorgehensweisen, bei denen keine Verletzungen drohen.

BITTE BEACHTEN

Elektrische Geräte dürfen nur von Fachpersonal installiert, betrieben, bedient und gewartet werden. Schneider Electric haftet nicht für Schäden, die durch die Verwendung dieses Materials entstehen.

Als qualifiziertes Fachpersonal gelten Mitarbeiter, die über Fähigkeiten und Kenntnisse hinsichtlich der Konstruktion und des Betriebs elektrischer Geräte und deren Installation verfügen und eine Schulung zur Erkennung und Vermeidung möglicher Gefahren absolviert haben.

BEVOR SIE BEGINNEN

Dieses Produkt nicht mit Maschinen ohne effektive Sicherheitseinrichtungen im Arbeitsraum verwenden. Das Fehlen effektiver Sicherheitseinrichtungen im Arbeitsraum einer Maschine kann schwere Verletzungen des Bedienpersonals zur Folge haben.

WARNUNG

UNBEAUF SICHTIGTE GERÄTE

- Diese Software und zugehörige Automatisierungsgeräte nicht an Maschinen verwenden, die nicht über Sicherheitseinrichtungen im Arbeitsraum verfügen.
- Greifen Sie bei laufendem Betrieb nicht in das Gerät.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

Dieses Automatisierungsgerät und die zugehörige Software dienen zur Steuerung verschiedener industrieller Prozesse. Der Typ bzw. das Modell des für die jeweilige Anwendung geeigneten Automatisierungsgeräts ist von mehreren Faktoren abhängig, z. B. von der benötigten Steuerungsfunktion, der erforderlichen Schutzklasse, den Produktionsverfahren, außergewöhnlichen Bedingungen, behördlichen Vorschriften usw. Für einige Anwendungen werden möglicherweise mehrere Prozessoren benötigt, z. B. für ein Backup-/Redundanzsystem.

Nur Sie als Benutzer, Maschinenbauer oder -integrator sind mit allen Bedingungen und Faktoren vertraut, die bei der Installation, der Einrichtung, dem Betrieb und der Wartung der Maschine bzw. des Prozesses zum Tragen kommen. Demzufolge sind allein Sie in der Lage, die Automatisierungskomponenten und zugehörigen Sicherheitsvorkehrungen und Verriegelungen zu identifizieren, die einen ordnungsgemäßen Betrieb gewährleisten. Bei der Auswahl der Automatisierungs- und Steuerungsgeräte sowie der zugehörigen Software für eine bestimmte Anwendung sind die einschlägigen örtlichen und landesspezifischen Richtlinien und Vorschriften zu beachten. Das National Safety Council's Accident Prevention Manual (Handbuch zur Unfallverhütung; in den USA landesweit anerkannt) enthält ebenfalls zahlreiche nützliche Hinweise.

Für einige Anwendungen, z. B. Verpackungsmaschinen, sind zusätzliche Vorrichtungen zum Schutz des Bedienpersonals wie beispielsweise Sicherheitseinrichtungen im Arbeitsraum erforderlich. Diese Vorrichtungen werden benötigt, wenn das Bedienpersonal mit den Händen oder anderen Körperteilen in den Quetschbereich oder andere Gefahrenbereiche gelangen kann und somit einer potenziellen schweren Verletzungsgefahr ausgesetzt ist. Software-Produkte allein können das Bedienpersonal nicht vor Verletzungen schützen. Die Software kann daher nicht als Ersatz für Sicherheitseinrichtungen im Arbeitsraum verwendet werden.

Vor Inbetriebnahme der Anlage sicherstellen, dass alle zum Schutz des Arbeitsraums vorgesehenen mechanischen/elektronischen Sicherheitseinrichtungen und Verriegelungen installiert und funktionsfähig sind. Alle zum Schutz des Arbeitsraums vorgesehenen Sicherheitseinrichtungen und Verriegelungen müssen mit dem zugehörigen Automatisierungsgerät und der Softwareprogrammierung koordiniert werden.

HINWEIS: Die Koordinierung der zum Schutz des Arbeitsraums vorgesehenen mechanischen/elektronischen Sicherheitseinrichtungen und Verriegelungen geht über den Umfang der Funktionsbaustein-Bibliothek, des System-Benutzerhandbuchs oder andere in dieser Dokumentation genannten Implementierungen hinaus.

START UND TEST

Vor der Verwendung elektrischer Steuerungs- und Automatisierungsgeräte ist das System zur Überprüfung der einwandfreien Funktionsbereitschaft einem Anlauftest zu unterziehen. Dieser Test muss von qualifiziertem Personal durchgeführt werden. Um einen vollständigen und erfolgreichen Test zu gewährleisten, müssen die entsprechenden Vorkehrungen getroffen und genügend Zeit eingeplant werden.

WARNUNG

GEFAHR BEIM GERÄTEBETRIEB

- Überprüfen Sie, ob alle Installations- und Einrichtungsverfahren vollständig durchgeführt wurden.
- Vor der Durchführung von Funktionstests sämtliche Blöcke oder andere vorübergehende Transportsicherungen von den Anlagekomponenten entfernen.
- Entfernen Sie Werkzeuge, Messgeräte und Verschmutzungen vom Gerät.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

Führen Sie alle in der Dokumentation des Geräts empfohlenen Anlauftests durch. Die gesamte Dokumentation zur späteren Verwendung aufbewahren.

Softwaretests müssen sowohl in simulierten als auch in realen Umgebungen stattfinden.

Sicherstellen, dass in dem komplett installierten System keine Kurzschlüsse anliegen und nur solche Erdungen installiert sind, die den örtlichen Vorschriften entsprechen (z. B. gemäß dem National Electrical Code in den USA). Wenn Hochspannungsprüfungen erforderlich sind, beachten Sie die Empfehlungen in der Gerätedokumentation, um eine versehentliche Beschädigung zu verhindern.

Vor dem Einschalten der Anlage:

- Entfernen Sie Werkzeuge, Messgeräte und Verschmutzungen vom Gerät.
- Schließen Sie die Gehäusetür des Geräts.
- Alle temporären Erdungen der eingehenden Stromleitungen entfernen.
- Führen Sie alle vom Hersteller empfohlenen Anlauftests durch.

BETRIEB UND EINSTELLUNGEN

Die folgenden Sicherheitshinweise sind der NEMA Standards Publication ICS 7.1-1995 entnommen (die Englische Version ist maßgebend):

- Ungeachtet der bei der Entwicklung und Fabrikation von Anlagen oder bei der Auswahl und Bemessung von Komponenten angewandten Sorgfalt, kann der unsachgemäße Betrieb solcher Anlagen Gefahren mit sich bringen.
- Gelegentlich kann es zu fehlerhaften Einstellungen kommen, die zu einem unbefriedigenden oder unsicheren Betrieb führen. Für Funktionseinstellungen stets die Herstelleranweisungen zu Rate ziehen. Das Personal, das Zugang zu diesen Einstellungen hat, muss mit den Anweisungen des Anlagenherstellers und den mit der elektrischen Anlage verwendeten Maschinen vertraut sein.
- Bediener sollten nur über Zugang zu den Einstellungen verfügen, die tatsächlich für ihre Arbeit erforderlich sind. Der Zugriff auf andere Steuerungsfunktionen sollte eingeschränkt sein, um unbefugte Änderungen der Betriebskenngrößen zu vermeiden.

Über dieses Buch



Auf einen Blick

Ziel dieses Dokuments

In diesem Handbuch werden die Grundlagen der Hardware- und Softwareimplementierung für die Zeichenmodus- und Modbus-Kommunikation mit Kommunikationsmodulen des Typs BMXNOM0200 beschrieben.

Gültigkeitsbereich

Diese Dokumentation ist gültig ab EcoStruxure™ Control Expert 15.0.

Die technischen Merkmale der hier beschriebenen Geräte sind auch online abrufbar. So greifen Sie auf diese Informationen online zu:

Schritt	Aktion
1	Gehen Sie zur Homepage von Schneider Electric www.schneider-electric.com .
2	Geben Sie im Feld Search die Referenz eines Produkts oder den Namen einer Produktreihe ein. • Die Referenz bzw. der Name der Produktreihe darf keine Leerstellen enthalten. • Wenn Sie nach Informationen zu verschiedenen vergleichbaren Modulen suchen, können Sie Sternchen (*) verwenden.
3	Wenn Sie eine Referenz eingegeben haben, gehen Sie zu den Suchergebnissen für technische Produktdatenblätter (Product Datasheets) und klicken Sie auf die Referenz, über die Sie mehr erfahren möchten. Wenn Sie den Namen einer Produktreihe eingegeben haben, gehen Sie zu den Suchergebnissen Product Ranges und klicken Sie auf die Reihe, über die Sie mehr erfahren möchten.
4	Wenn mehrere Referenzen in den Suchergebnissen unter Products angezeigt werden, klicken Sie auf die gewünschte Referenz.
5	Je nach der Größe der Anzeige müssen Sie ggf. durch die technischen Daten scrollen, um sie vollständig einzusehen.
6	Um ein Datenblatt als PDF-Datei zu speichern oder zu drucken, klicken Sie auf Download XXX product datasheet .

Die in diesem Dokument vorgestellten Merkmale sollten denen entsprechen, die online angezeigt werden. Im Rahmen unserer Bemühungen um eine ständige Verbesserung werden Inhalte im Laufe der Zeit möglicherweise überarbeitet, um deren Verständlichkeit und Genauigkeit zu verbessern. Sollten Sie einen Unterschied zwischen den Informationen im Dokument und denen online feststellen, nutzen Sie die Online-Informationen als Referenz.

Verwandte Dokumente

Titel der Dokumentation	Referenznummer
Modicon M580, M340 und X80 I/O-Plattformen, Normen und Zertifizierungen	EIO0000002726 (Englisch), EIO0000002727 (Französisch), EIO0000002728 (Deutsch), EIO0000002730 (Italienisch), EIO0000002729 (Spanisch), EIO0000002731 (Chinesisch)
Modicon M340 – Prozessoren, Konfigurationshandbuch	35012676 (Englisch), 35012677 (Französisch), 35013351 (Deutsch), 35013352 (Italienisch), 35013353 (Spanisch), 35013354 (Chinesisch)
Modicon M580 – Hardware, Referenzhandbuch	EIO0000001578 (Englisch), EIO0000001579 (Französisch), EIO0000001580 (Deutsch), EIO0000001582 (Italienisch), EIO0000001581 (Spanisch), EIO0000001583 (Chinesisch)
EcoStruxure™ Control Expert – Betriebsarten	33003101 (Englisch), 33003102 (Französisch), 33003103 (Deutsch), 33003104 (Spanisch), 33003696 (Italienisch), 33003697 (Chinesisch)
EcoStruxure™ Control Expert – Kommunikation, Bausteinbibliothek	33002527 (Englisch), 33002528 (Französisch), 33002529 (Deutsch), 33003682 (Italienisch), 33002530 (Spanisch), 33003683 (Chinesisch)
EcoStruxure™ Control Expert – E/A-Verwaltung, Bausteinbibliothek	33002531 (Englisch), 33002532 (Französisch), 33002533 (Deutsch), 33003684 (Italienisch), 33002534 (Spanisch), 33003685 (Chinesisch)
EcoStruxure™ Control Expert – Programmiersprachen und Struktur, Referenzhandbuch	35006144 (Englisch), 35006145 (Französisch), 35006146 (Deutsch), 35013361 (Italienisch), 35006147 (Spanisch), 35013362 (Chinesisch)

Sie können diese technischen Veröffentlichungen sowie andere technische Informationen von unserer Website herunterladen: www.schneider-electric.com/en/download.

Produktbezogene Informationen

WARNUNG

UNBEABSICHTIGTER GERÄTEBETRIEB

Die Anwendung dieses Produkts erfordert Fachkenntnisse bezüglich der Entwicklung und Programmierung von Steuerungssystemen. Nur Personen mit solchen Fachkenntnissen sollten dieses Produkt programmieren, installieren, ändern und anwenden.

Befolgen Sie alle landesspezifischen und örtlichen Sicherheitsnormen und -vorschriften.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

Teil I

Hardwaretechnische Installation für die serielle Modbus- und Zeichenmoduskommunikation

Inhalt dieses Teils

Dieser Abschnitt enthält eine Beschreibung der hardwaretechnischen Installation für die serielle Modbus- und Zeichenmoduskommunikation.

Inhalt dieses Teils

Dieser Teil enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
1	Einführung in die serielle Kommunikation	17
2	Serielle Kommunikationsarchitekturen	33

Kapitel 1

Einführung in die serielle Kommunikation

Inhalt dieses Kapitels

Dieses Kapitel gibt eine Einführung in die serielle Kommunikation und enthält eine Beschreibung der BMXNOM0200-Module.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Kommunikation im seriellen Modbus- und im Zeichenmodus	18
Beschreibung des BMXNOM0200 Moduls	20
Abmessungen des X80-SL-Moduls (Serielle Verbindung) BMXNOM0200(H)	26
Normen und Zertifizierungen	27
Installation des Moduls BMXNOM0200	28

Kommunikation im seriellen Modbus- und im Zeichenmodus

Allgemeines

Serielle Verbindungen unterstützen zwei Kommunikationsprotokolle:

- Modbus seriell
- Zeichenmodus

Modbus-Protokoll

Modbus ist ein Standardprotokoll und weist folgende Eigenschaften auf:

- Das Protokoll baut eine Client/Server-Kommunikation zwischen verschiedenen Modulen über eine Bus- oder serielle Verbindung auf. Der Client wird vom Master identifiziert und die Slave-Module fungieren als Server.
- Das Protokoll basiert auf einem Datenaustauschmodus, bestehend aus Requests und Antworten zur Bereitstellung von Diensten anhand unterschiedlicher Funktionscodes.
- Das Protokoll ermöglicht den Austausch der Frames Modbus-fähiger Anwendungen in zwei Arten von Code:
 - RTU-Modus
 - ASCII-Modus

Die Verwaltung des Austauschs verläuft folgendermaßen:

- Nur jeweils ein Gerät kann Daten auf dem Bus senden.
- Der Datenaustausch wird vom Master verwaltet. Nur der Master kann den Austausch initiieren. Slaves können keine Nachrichten senden, wenn sie nicht zuvor eine entsprechende Aufforderung erhalten.
- Bei einem ungültigen Austausch wiederholt der Master seinen Request. Der Slave, an den der Request gerichtet ist, wird vom Master als abwesend eingestuft, wenn dieser nicht innerhalb eines vorgegebenen Zeitraums antwortet.
- Wenn der Slave den Request nicht versteht bzw. nicht verarbeiten kann, sendet er eine Ausnahmeantwort an den Master. In diesem Fall kann der Master den Request erneut senden oder nicht.

Zwischen Master und Slave(s) sind zwei Typen von Dialog möglich:

- Der Master sendet einen Request an eine bestimmte Slave-Nummer und wartet auf die Antwort des betreffenden Slaves.
- Der Master sendet einen Request an alle Slaves, ohne auf eine Antwort zu warten (das entspricht dem allgemeinen Broadcast-Prinzip).

Kommunikation im Zeichenmodus

Bei der Kommunikation im Zeichenmodus handelt es sich um einen Datenaustausch im Punkt-zu-Punkt-Modus zwischen zwei Einheiten. Im Gegensatz zum Modbus-Protokoll wird in diesem Modus keine hierarchisch strukturierte Kommunikation über eine serielle Verbindung aufgebaut bzw. es werden keine Dienste über Funktionscodes bereitgestellt.

Der Zeichenmodus ist asynchron. Jedes Element der Textinformationen wird Zeichen für Zeichen in unregelmäßigen Zeitintervallen gesendet oder empfangen. Die Dauer des jeweiligen Austauschs wird durch Folgendes bestimmt:

- Ein oder zwei Frame-Endzeichen
- Timeout
- Anzahl der Zeichen

Beschreibung des BMXNOM0200 Moduls

Allgemeines

Das serielle Verbindungsmodul BMXNOM0200 ist ein asynchrones bidirektionales serielles Verbindungsmodul, das die serielle Modbus-Kommunikation (Master oder Slave) und die Zeichenmoduskommunikation unterstützt.

Das BMXNOM0200-Modul ist ein dediziertes Modul im einfachen Format, das auf einem Modicon X80-Rack installiert werden kann.

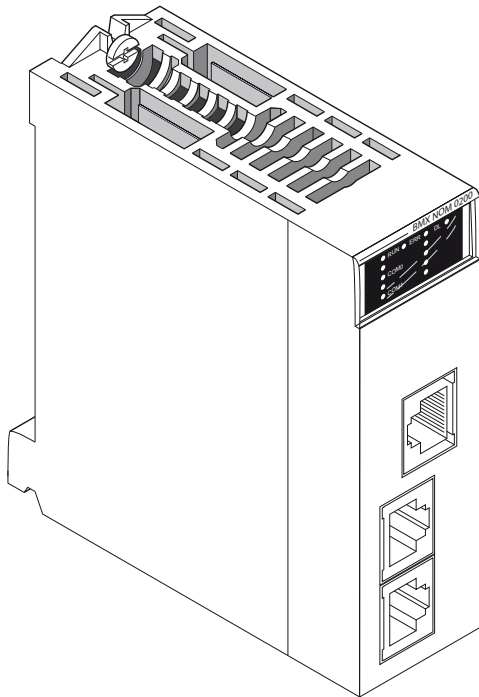
Verstärkte Version

Das BMXNOM0200H-Gerät (Hardened) ist die verstärkte Version des BMXNOM0200-Standardgeräts. Es kann auch bei extremen Temperaturen und unter chemisch aggressiven Umgebungsbedingungen eingesetzt werden.

Weitere Informationen finden Sie im Kapitel *Installation in besonders rauen Umgebungen* (siehe *Modicon M580-, M340- und X80 I/O-Plattformen, Normen und Zertifizierungen*).

Einführung in das Modul

Die folgende Abbildung zeigt die physikalischen Merkmale des BMXNOM0200-Moduls:



Dieses BMX NOM 0200-Modul setzt sich aus den in der folgenden Tabelle aufgeführten Elementen zusammen:

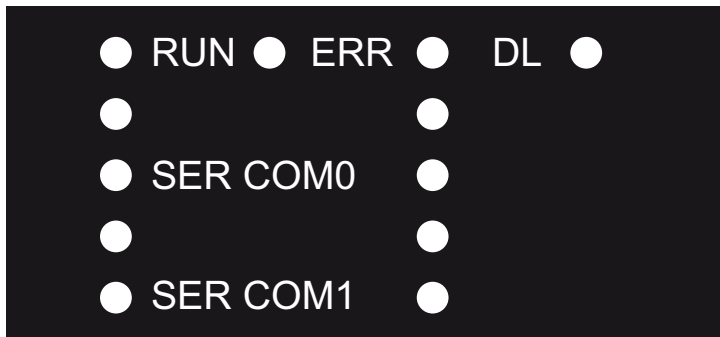
Taste	Beschreibung
1	Fünf LED-Anzeigen an der Vorderseite des Moduls: • RUN und ERR zeigen den Modulstatus an. • SER COM zeigt den Datenverkehrsstatus am seriellen Port port 0 RS232 oder port 0 RS485 (Kanal 0) an. • SER COM1 zeigt den Datenverkehrsstatus am seriellen Port Port 1 RS485 (Kanal 1) an. • DL zeigt den Status des Firmware-Downloads an.
2	Integrierter dedizierter Kanal 0 für die serielle Verbindung mit zwei seriellen Ports: • Port 0 RS232 • Port 0 RS485 Hinweis: Es kann jeweils nur ein serieller Port aktiv sein.
3	Integrierter dedizierter Kanal 1 für die serielle Verbindung mit einem seriellen Port: • Port 1 RS485

HINWEIS: In einigen Betriebsarten können die LEDs weitere Informationen anzeigen (*siehe Seite 21*).

Visuelle Diagnose

An der Frontblende des BMXNOM0200-Moduls befinden sich fünf LEDs. Sie zeigen den Betriebsstatus des Moduls und den Kommunikationsstatus der integrierten seriellen Verbindung an.

LED-Anzeige:



- RUN = Das Modul ist mit Strom versorgt und korrekt konfiguriert.
- ERR = Das Modul hat einen Fehler erkannt und kann nicht korrekt funktionieren.
- DL = Die Firmware wird heruntergeladen.
- SER COM0 = Kommunikation an Kanal 0 (**Port 0 RS232** oder **Port 0 RS485**) erkannt.
- SER COM1 = Kommunikation an Kanal 1 (**Port 1 RS485**) erkannt.

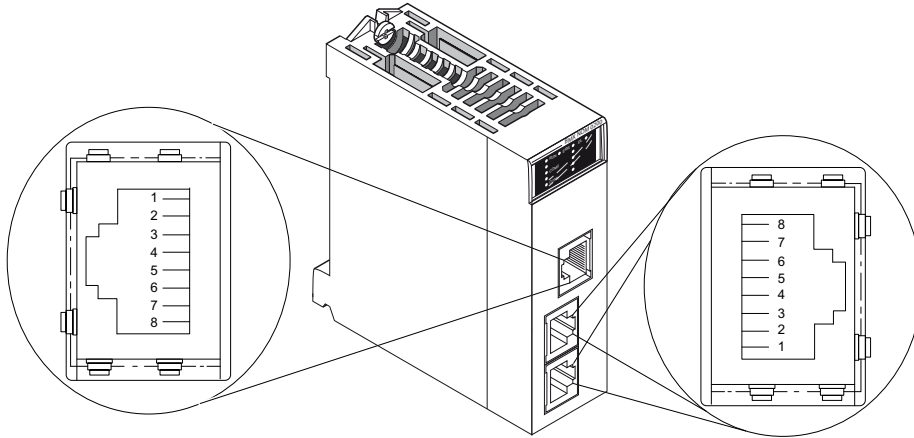
Bedeutung der LED:

- Jede LED kann einen der folgenden Zustände annehmen:
 - 1 = Ein
 - 0 = Aus
 - B = Blinkend
- Beim Starten des Moduls leuchten alle EIN- und AUS-LEDs auf. So wird geprüft, dass alle LEDs einwandfrei funktionieren.

RUN	ERR	SER COM0	SER COM1	DL	Diagnose
0	–	–	–	–	Das Modul ist nicht mit Strom versorgt oder nicht betriebsbereit.
0	B	–	–	–	Das Modul ist nicht konfiguriert.
1	1	–	–	–	Das Modul ist nicht vorschriftsmäßig betriebsbereit.
1	1	1	0	–	Das Modul hat ein Problem auf Kanal 0 erkannt.
1	1	1	B	–	Das Modul hat ein Problem auf Kanal 0 erkannt, auf Kanal 1 werden Daten ausgetauscht.
1	1	0	1	–	Das Modul hat ein Problem auf Kanal 1 erkannt.
1	1	B	1	–	Das Modul hat ein Problem auf Kanal 1 erkannt, auf Kanal 0 werden Daten ausgetauscht.
1	0	B	–	–	Auf Kanal 0 werden Daten ausgetauscht.
1	0	–	B	–	Auf Kanal 1 werden Daten ausgetauscht.
B	B	–	–	0	Die CPU fehlt.
B	B	B	B	–	Das Modul führt Selbsttests durch.
–	–	–	–	B	Modul-Firmware wird heruntergeladen.
–	–	–	–	1	Die Firmware wurde heruntergeladen; das Modul muss zurückgesetzt werden.

Einführung in die seriellen Schnittstellen

Die folgende Abbildung zeigt die RJ45-Anschlüsse der seriellen Schnittstellen am BMXNOM0200:



Die folgende Tabelle zeigt die Anschlussbelegung des seriellen Ports am BMXNOM0200:

Pin-Nr.	Serieller Port RS485	Serieller Port RS232
1	–	RXD (Empfangsdaten)
2	–	TXD (Sendedaten)
3	–	RTS (Sendeanforderung)
4	D1 (B/B4)	DTR (Endgerät betriebsbereit)
5	D0 (A/A4)	DSR (Betriebsbereitschaft)
6	–	CTS (Sendebereit)
7	–	DCD (Datenträgererkennung)
8	Potenzialerdung der seriellen Verbindung (0 V)	Potenzialerdung der seriellen Verbindung (0 V)

HINWEIS:

- Die beiden RS485-Leitungen sind galvanisch getrennt. Die Isolationsspannung zwischen den beiden seriellen 500 V-Verbindungen und zwischen den beiden isolierten seriellen Verbindungen und dem Baugruppenträger beträgt 500 VAC.
- Die 7-adrige RS232-Schnittstelle und die 2-adrige RS485-Schnittstelle verwenden die gleichen RJ45-Buchse. Lediglich die Signalverdrahtung ist unterschiedlich.

Technische Daten der Kanäle

Die Kanäle des BMXNOM0200-Moduls enthalten:

- Zwei potentialgetrennte, physikalische RS485-Schnittstellen,
- Eine nicht potentialgetrennte physikalische RS232-Schnittstelle,
- Kommunikationstypen Modbus seriell (ASCII und RTU) und Zeichenmodus

Technische Kenndaten der Verbindung für die zwei Protokolle:

	Modbus seriell / RS485	Modbus seriell / RS232	Zeichenmodus / RS485	Zeichenmodus / RS232
Typ	Master/Slave	Master/Slave	Half Duplex	Full Duplex
Datenfluss	19200 Baud. Die Parameter können auf einen Wert zwischen 300 und 57600 Baud eingestellt werden.	19200 Baud. Die Parameter können auf einen Wert zwischen 300 und 115200 Baud eingestellt werden.	9600 Baud. Die Parameter können auf einen Wert zwischen 300 und 57600 Baud eingestellt werden.	9600 Baud. Die Parameter können auf einen Wert zwischen 300 und 115200 Baud eingestellt werden.
Anzahl Geräte	32	32	–	–
Zulässige Slave-Adressen	1 bis 247	1 bis 247	–	–
Max. Buslänge ohne Abzweige	Beachten Sie die Tabelle unten (15 m mit Verzweigung).	15 m	Beachten Sie die Tabelle unten (15 m mit Verzweigung).	15 m
Nachrichtengröße	Modbus seriell: • RTU: 256 Byte (252 Byte an Daten) • ASCII: 513 Byte (2x252 Byte an Daten)	Modbus seriell: • RTU: 256 Byte (252 Byte an Daten) • ASCII: 513 Byte (2x252 Byte an Daten)	1024 Byte	1024 Byte
Dienstprogramme	Lesen von Wörtern/Bits Schreiben von Wörtern/Bits Diagnose	Lesen von Wörtern/Bits Schreiben von Wörtern/Bits Diagnose	Senden von Zeichenfolgen Empfangen von Zeichenfolgen	Senden von Zeichenfolgen Empfangen von Zeichenfolgen
Hardware-Flusskontrolle	–	Optional über RTS/CTS-Signale.	–	Optional über RTS/CTS-Signale.

HINWEIS: Bei der gleichzeitigen Verwendung mehrerer Funktionsbausteine zur Kommunikation pro Kanal nimmt das Modul BMXNOM0200 ggf. die gesamte Modbus-Bandbreite in Anspruch.

Die nachstehende Tabelle zeigt die maximal verwendbare Länge des Kabels RS485 je nach ausgewählter Baudrate:

Ausgewählte Baudrate (Bit/s)	Länge (m)	Bestellreferenz
300	1000	(1)
600	1000	(1)
1200	1000	(1)
2400	1000	(1)
9600	1000	(1)
19200	600	(1)
38400	300	(1) oder (2)
57600	200	(1) oder (2)

- (1): Paarig verdrehtes geschirmtes Kabel, Querschnitt AWG24 (TSX CSA 100, TSX CSA 200, TSX CSA 500)
- (2): Kabelkategorie 5 oder höher

Betriebsbedingungen je nach Höhenlage

Die Kenndaten in der nachstehenden Tabelle gelten für die Module BMXNOM0200 und BMXNOM0200H bei einem Einsatz in einer Höhe bis 2000 m (6560 ft). Wenn die Module in einer Höhe über 2000 m (6560 ft) zum Einsatz kommen, muss die Temperatur herabgesetzt werden.

Detaillierte Informationen finden Sie im Kapitel *Betriebs- und Lagerbedingungen (siehe Modicon M580-, M340- und X80 I/O-Plattformen, Normen und Zertifizierungen)*.

Betriebstemperatur

Modul	Temperaturbereich
BMXNOM0200	0 bis 60 °C (32 bis 140 °F)
BMXNOM0200H	-25 bis 70 °C (-13 bis 158 °F)

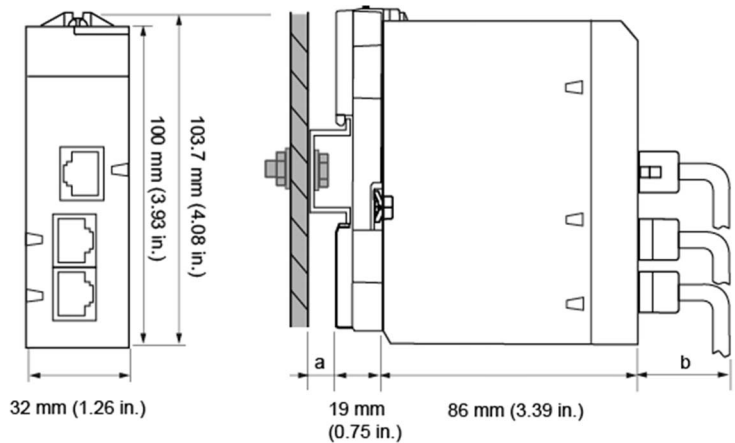
Stromverbrauch des Moduls

Die folgende Tabelle zeigt die Leistungsaufnahme des BMXNOM0200-Moduls:

Spannung	Typischer Strom	Höchststrom	Typische Verlustleistung	Maximale Verlustleistung
24 VDC	80 mA	130 mA	1,92 W	3,12 W

Abmessungen des X80-SL-Moduls (Serielle Verbindung) BMXNOM0200(H)

Allgemeine Beschreibung des X80-SL-Moduls (Serielle Verbindung) BMXNOM0200(H)



- a Tiefe der DIN-Schiene: Der Wert ist von dem in Ihrer Plattform verwendeten DIN-Schientyp abhängig.
- b Verdrahtungstiefe: Der Wert ist vom Steckanschluss und den in Ihrer Plattform verwendeten Drähten abhängig.

Abmessungen des X80-SL-Moduls (Serielle Verbindung) BMXNOM0200(H)

Modulreferenz	Modulabmessungen			Installationstiefe ⁽¹⁾
	Breite	Höhe	Tiefe	
BMXNOM0200(H)	32 mm (1.26 in.)	103,7 mm (4.08 in.)	86 mm (3.39 in.)	105 mm (4.13 in.) ⁽¹⁾
(1) DIN-Schientiefe (a) und Verdrahtungstiefe (b) nicht inbegriffen.				

HINWEIS: Beachten Sie die Abmessungen der Anschlüsse und sehen Sie ausreichende Abstände für die Kabelinstallation und rund um die Racks vor.

Normen und Zertifizierungen

Download

Klicken Sie auf die Verknüpfung für Ihre bevorzugte Sprache, um die Normen und Zertifizierungen für die Module dieser Produktfamilie (im PDF-Format) herunterzuladen:

Titel	Sprachen
Modicon M580, M340 und X80 I/O-Plattformen, Normen und Zertifizierungen	• Englisch: EIO0000002726 • Französisch: EIO0000002727 • Deutsch: EIO0000002728 • Italienisch: EIO0000002730 • Spanisch: EIO0000002729 • Chinesisch: EIO0000002731

Installation des Moduls BMXNOM0200

Allgemeines

Das Modul BMXNOM0200 wird in einem Modicon X80-Rack in einem beliebigen ungenutzten Steckplatz installiert, mit Ausnahme der Steckplätze, die für die Stromversorgung, den Prozessor, das Drop-end-Modul oder das Rack-Erweiterungsmodul erforderlich sind. Bei der Installation müssen die Installationsanweisungen des Racks beachtet werden.

WARNUNG

UNBEABSICHTIGTER GERÄTEBETRIEB

Die Anwendung dieses Produkts erfordert Fachkenntnisse bezüglich der Entwicklung und Programmierung von Steuerungssystemen. Diese Produkte dürfen nur von Personen programmiert, installiert, modifiziert und verwendet werden, die über das entsprechende Fachwissen verfügen.

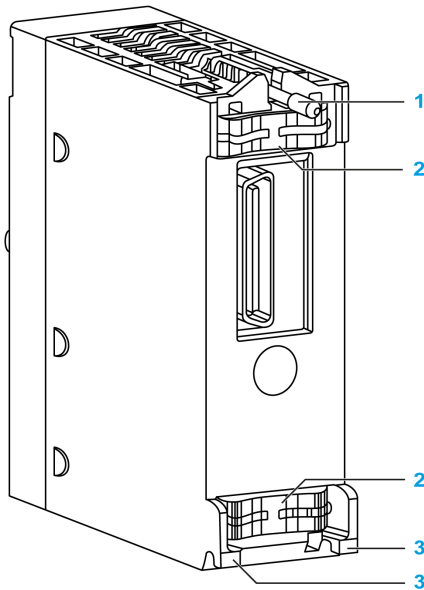
Befolgen Sie alle landesspezifischen und örtlichen Sicherheitsnormen und -vorschriften.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

HINWEIS: Das BMXNOM0200-Modul kann im Rack installiert werden, während die Anwendung auf der SPS läuft.

Erdung des Moduls

Das BMXNOM0200-Modul ist zu Erdungszwecken auf der Rückseite mit Kontaktleisten ausgestattet:



- 1 Montageschraube
- 2 Kontaktleiste
- 3 Führungsstifte

Wenn das Modul ordnungsgemäß im Rack installiert wird, stellt die Kontaktleiste eine Verbindung zwischen dem Erdungsbus des Moduls und dem Erdungsbus des Racks her.

⚡ ⚠ GEFAHR

GEFAHR EINES ELEKTRISCHEN SCHLAGS

Stellen Sie sicher, dass die Erdungskontaktleiste vorhanden und nicht verbogen ist.

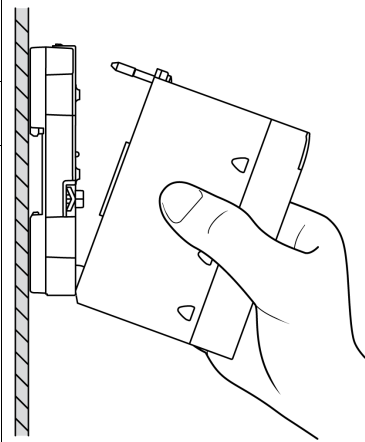
Die Nichtbeachtung dieser Anweisungen führt zu Tod oder schweren Verletzungen.

HINWEIS: Wenn die Erdungskontakte nicht verfügbar oder verbogen sind, verwenden Sie das Modul nicht und wenden Sie sich an Ihren Schneider Electric-Vertreter.

Installation des Moduls

Gehen Sie zur Installation eines BMXNOM0200-Moduls in einem Rack vor wie folgt:

Schritt	Aktion
1	Entfernen Sie die Schutzabdeckung vom Modulsteckplatz auf dem Modicon X80-Rack.
2	Positionieren Sie die Unverwechselbarkeitsstifte auf der Rückseite des Moduls (am unteren Teil) in dem entsprechenden Steckplatz am Rack.
3	Drehen Sie das Modul in Richtung Oberseite des Racks, sodass das Modul bündig mit dem Rack abschließt.
4	Ziehen Sie die Montageschraube an der Moduloberseite fest, um das Modul in seiner Position im Rack zu sichern. Anzugsmoment: 0,4 bis 1,5 N•m (0.30 bis 1.10 lbf-ft).



WARNUNG

UNBEABSICHTIGTER GERÄTEBETRIEB

Stellen Sie sicher, dass die Montageschraube ordnungsgemäß festgezogen ist, um die sichere Befestigung des Moduls am Rack zu gewährleisten.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

An das Modul kann je nach angestrebtem Netzwerk ein entsprechender RJ45-Steckverbinder angeschlossen werden.

Anschließen/Trennen

Das BMXNOM0200-Modul kann bei eingeschalteter Stromversorgung angeschlossen oder getrennt werden.

WARNUNG

UNBEABSICHTIGTER GERÄTEBETRIEB

Obwohl die Module des Typs BMXNOM0200 angeschlossen und von den Anschlüssen getrennt werden können, während die Station eingeschaltet ist, kann die aktuell ausgeführte Anwendung dadurch unterbrochen werden.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

Wenn das Modul vom Rack getrennt wird, gehen die Inhalte des internen Speichers verloren. Bei einem erneuten Anschluss an die Backplane durchläuft das Modul eine Initialisierung.

Die folgenden Situationen können eine temporäre Unterbrechung der Anwendung oder Kommunikation bewirken:

- Der Steckverbinder der RJ45 wird angeschlossen oder getrennt, während die Stromversorgung eingeschaltet ist.
- Die Module werden neu initialisiert, wenn die Stromversorgung wieder eingeschaltet wird.

Anwendungsfall: Zusätzlicher Verbindungspunkt

Ein BMXNOM0200-Modul (mit Firmware-Version ≥ 1.2) kann ohne konfiguriert zu sein an einem beliebigen freien Steckplatz in einem beliebigen Rack eingesetzt werden. Dies ist besonders nützlich, um Control Expert anzuschließen, während die CPU nicht konfiguriert ist, oder um einen zusätzlichen Verbindungspunkt zu schaffen. In diesem Fall liegt das BMXNOM0200-Modul in der Standardkonfiguration vor.

Die Standardkonfiguration des BMXNOM0200-Moduls erfolgt als MODBUS-Slave mit Adresse 248, RTU (Verzögerung zwischen Frames = 2 ms), 8 Bit Daten, 1 Stoppbit, gerader Parität, RS232 bei 115.200 Bit/s auf Kanal 0 und RS485 bei 57.600 Bit/s auf Kanal 1.

Die Adresse 248 ist die Punkt-zu-Punkt-Adresse, auf die jedes BMXNOM0200-Slave-Modul antwortet. Diese Funktionalität ermöglicht die direkte Verbindung zu einem beliebigen Slave-Modul, dessen Adresse unbekannt ist.

Firmware-Aktualisierung

Die Firmware von BMXNOM0200 kann über den SPS-Baugruppenträger mithilfe eines der folgenden Tools aktualisiert werden:

- EcoStruxure™ Automation Device Maintenance
- Unity Loader

Kapitel 2

Serielle Kommunikationsarchitekturen

Inhalt dieses Kapitels

Dieses Kapitel bietet eine Einführung in Architekturen, die auf eine serielle Kommunikation zurückgreifen, und enthält die Anforderungen an die Verdrahtung.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Modbus-Leitungsabschluss und Polarisierung (RS485)	34
Verbindung von Modbus-Geräten (RS485)	36
Anschließen von Datenendeinrichtungen (DEE) (RS232)	38
Anschließen von Datenübertragungseinrichtungen (DÜE) (RS232)	40
Verkabelung	43

Modbus-Leitungsabschluss und Polarisierung (RS485)

Übersicht

Ein Mehrpunkt-Modbus-Netzwerk erfordert einen Leitungsabschluss und eine Polarisierung.

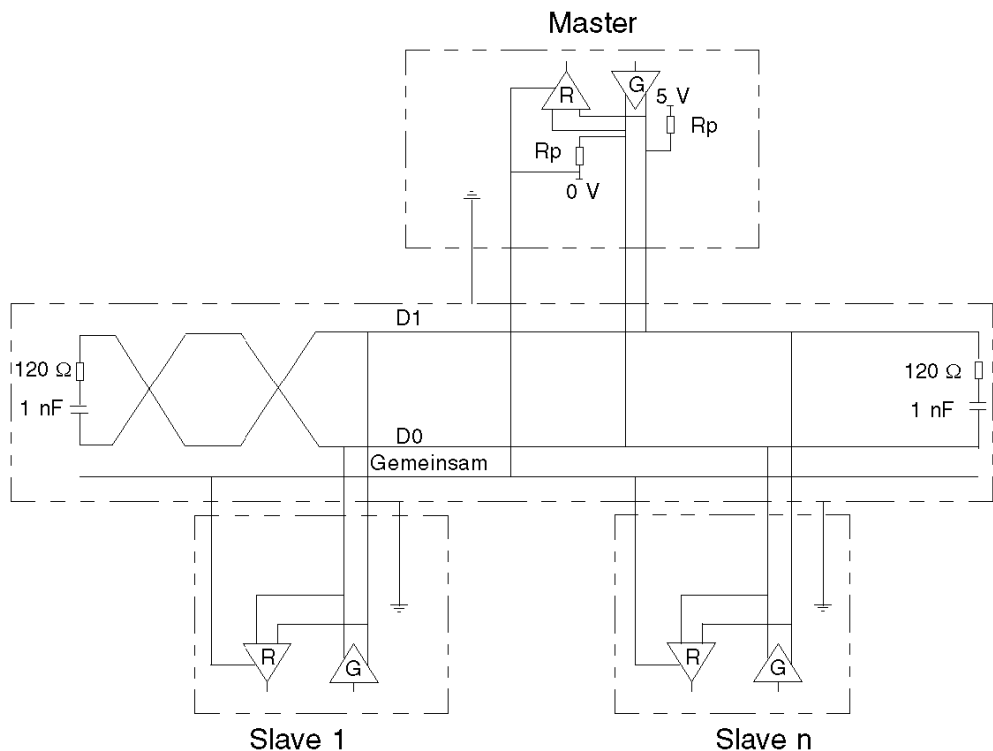
An diesen Bus sind folgende Geräte anschließbar:

- Andere SPS wie M340, Premium, Quantum, Twido oder Nano
- Geräte von Schneider Automation wie Altivar, Sicherheitsmodule XPS, SEPAM, XBT oder Momentum
- Andere Modbus-kompatible Geräte
- Modem, Hub

Ein Beispiel für ein **Mehrpunkt-Modbus-Netzwerk** (*siehe Seite 37*) mit einem BMXNOM0200-Modul wird in diesem Handbuch vorgestellt.

HINWEIS: Modbus-Netzwerke können auch als Punkt-zu-Punkt-Netzwerk realisiert werden.

Elektrische Grundlagen zu einem Leitungsabschluss und einer Polarisierung:



Leitungsabschluss

Der Leitungsabschluss erfolgt extern: Er besteht aus zwei 120- Ω -Widerständen und einen 1-nF-Kondensator an jedem Ende des Netzwerks (VW3 A8 306 RC oder VW3 A8 306 DRC). Platzieren Sie den Leitungsabschluss nicht am Ende von Abzweigkabeln.

Leitungspolarisierung

Bei einem Modbus ist eine Polarisierung des RS485-Netzwerks erforderlich.

- Wenn das BMXNOM0200-Modul als Master eingesetzt wird, erfolgt diese automatisch durch das System, so dass keine externe Polarisierung erforderlich ist.
- Wenn das BMXNOM0200-Modul als Slave eingesetzt wird, muss die Polarisierung in Form von zwei Widerständen (R_p) von 450 bis 650 Ω , je einem pro RS485-Kreis erfolgen:
 - ein Pull-Up-Widerstand gegen 5 V im D1-Stromkreis,
 - ein Pull-Down-Widerstand gegen Masse im D0-Stromkreis.

HINWEIS:

Im Zeichenmodus ist die Leitungspolarisierung unter Control Expert konfigurierbar. Zur Auswahl stehen:

- Polarisierung mit niedriger Impedanz wie in einem Modbus-Netzwerk (Sinn dieser Polarisierungsart ist es, dem Master das Halten des Standardzustands zu überlassen),
- Polarisierung mit hoher Impedanz (wo jedes Gerät zum Halten des Standardstatus beiträgt),
- Keine Polarisierung (wenn eine externe Polarisierung verwendet wird).

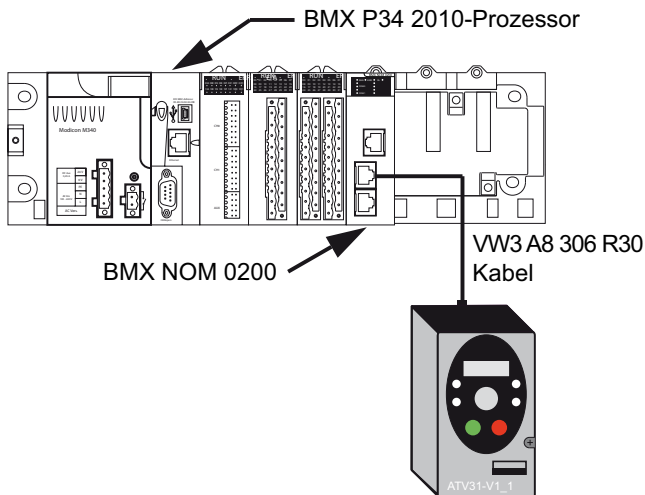
Verbindung von Modbus-Geräten (RS485)

Allgemeines

Die folgenden Seiten zeigen ein Beispiel für den Anschluss von Modbus-Geräten und eine serielle Modbus-Verbindungsarchitektur.

Anschließen von Modbus-Geräten, die nicht über die serielle Verbindung gespeist werden

Die folgende Abbildung zeigt, wie ein BMXNOM0200-Modul mit einem ATV31-Antrieb verbunden wird:



Die Geräte sind folgendermaßen konfiguriert:

- Ein BMXP342010-Prozessor
- Ein als Master konfiguriertes BMXNOM0200-Modul
- Ein als Slave konfiguierter ATV31-Antrieb

Das VW3A8306R30-Kabel weist folgende Eigenschaften auf:

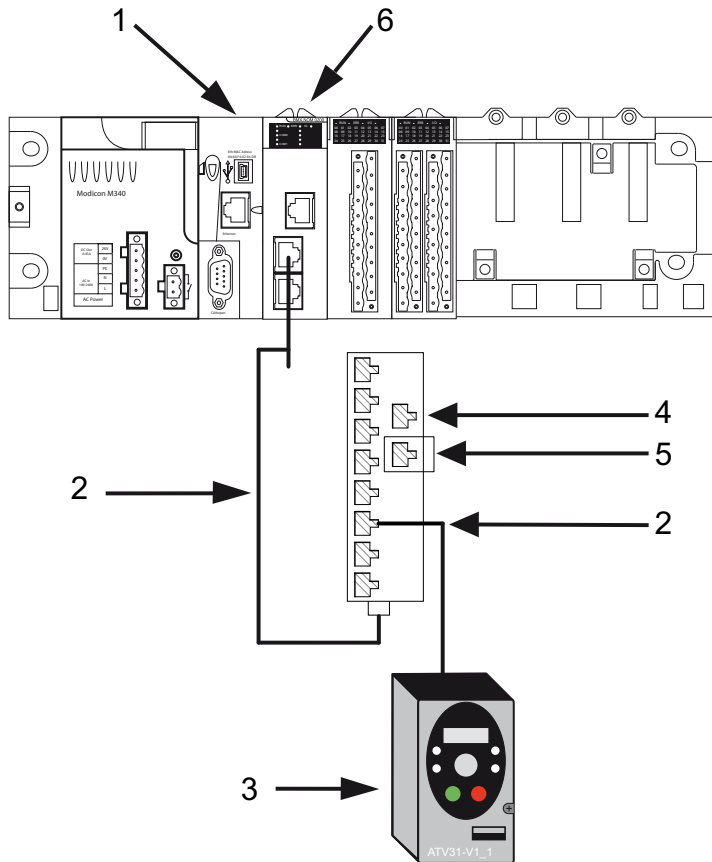
- Anschluss: 2 RTJ45-Steckverbinder
- Verdrahtung: 2 Drähte für die physische RS485-Leitung

Serielle Modbus-Verbindungsarchitektur

Die Architektur einer seriellen Modbus-Verbindung umfasst folgende Elemente:

- Ein BMXP342010-Prozessor
- Ein als Master konfiguriertes BMXNOM0200-Modul
- Einen LU9GC3-Verteilerblock
- Zwei als Slaves konfigurierte ATV31-Antriebe

Die folgende Abbildung zeigt die oben beschriebene Architektur der seriellen Verbindung:



- 1 BMXP342010-Prozessor
- 2 VW3A8306R**-Kabel
- 3 ATV31-Laufwerk
- 4 LU9GC3-Anschlussblock
- 5 Modbus-Leitungsabschluss VW3A8306RC
- 6 BMXNOM0200-Modul

Anschließen von Datenendeinrichtungen (DEE) (RS232)

Allgemein

Der Begriff Datenendeinrichtung beschreibt beispielsweise folgende Geräte:

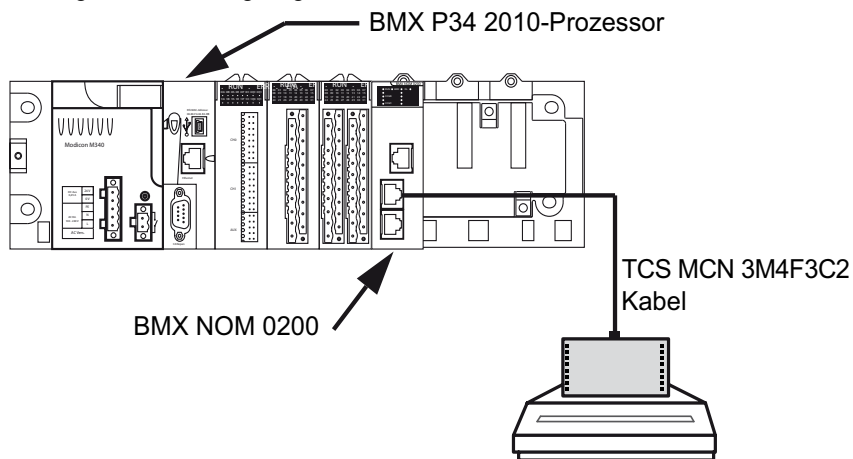
- Herkömmliche Peripheriegeräte (Drucker, Tastaturbildschirm, Endgeräte in Werkstätten usw.)
- Spezielle Peripheriegeräte (Strichcodelesegeräte usw.)
- PCs

Bei einem DEE-Gerät werden die RTS- und CTS-Leitungen gekreuzt.

Jede Datenendeinrichtung wird mit einem seriellen, gekreuzten Kabel über eine physischen RS232-Verbindung mit einem BMXNOM0200-Modul verbunden.

Anschließen von Datenendeinrichtungen

Die folgende Abbildung zeigt einen Drucker, der an ein BMXNOM0200-Modul angeschlossen ist:



Als Kommunikationsprotokoll wird der Zeichenmodus verwendet.

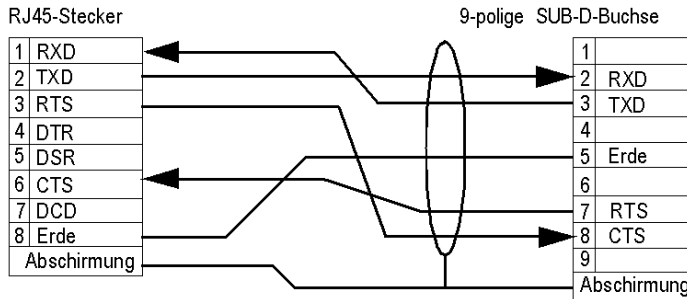
HINWEIS: Mit dem BMXNOM0200-Modul kann jeweils nur eine Datenendeinrichtung verbunden werden.

Seriell, gekreuztes RS232-Kabel

Das serielle, gekreuzte TCSMCN3M4F3C2-Kabel weist zwei Anschlüsse auf:

- RJ45-Stecker,
- 9-polige SUB-D-Buchse.

Die folgende Abbildung zeigt die Anschlussbelegung eines seriellen, gekreuzten TCSMCN3M4F3C2-Kabels:



Verbinden von Kabeln und Zubehör

Die folgende Tabelle führt die Referenznummern der Kabel und Adapter auf, die gemäß dem seriellen Anschluss der Datenendeinrichtung verwendet werden müssen:

Serieller Anschluss für Datenendeinrichtung	Verdrahtung
9-poliger SUB-D-Stecker	TCSMCN3M4F3C2-Kabel
25-poliger SUB-D-Stecker	• TCSMCN3M4F3C2-Kabel • TSXCTC07-Adapter
25-polige SUB-D-Buchse	• TCSMCN3M4F3C2-Kabel • TSXCTC10-Adapter

Anschließen von Datenübertragungseinrichtungen (DÜE) (RS232)

Allgemein

Der Begriff Datenübertragungseinrichtung (DÜE) beschreibt Geräte wie beispielsweise Modems. Bei einem DÜE-Gerät werden die RTS- und CTS-Leitungen direkt angeschlossen (nicht gekreuzt). Jede Datenendeinrichtung wird über ein serielles, ungekreuztes Kabel mittels der physischen RS232-Verbindung mit einem BMXNOM0200-Modul verbunden.

HINWEIS: Die Unterschiede beim Anschluss von DÜE und DEE liegen im Wesentlichen in der Steckerbelegung und den Signalrichtungen (Eingang oder Ausgang). Zum Beispiel ist ein PC in der Regel eine Datenendeinrichtung (DEE), während ein Modem eine Datenübertragungseinrichtung (DÜE) ist.

Technische Daten der Modems

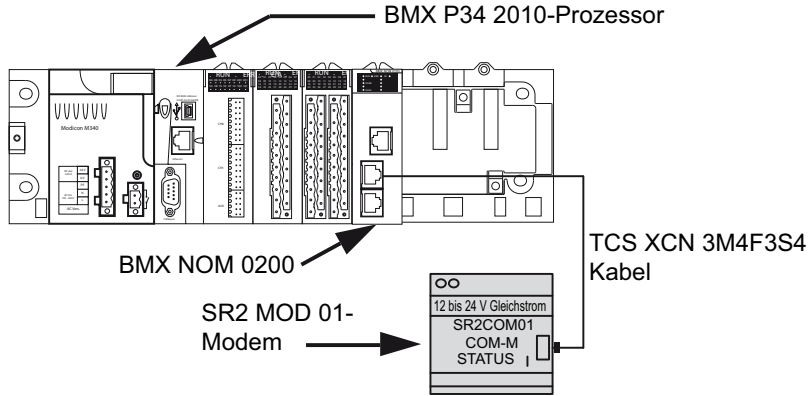
Das BMXNOM0200-Modul unterstützt die meisten handelsüblichen Modems. Um ein Modem an den seriellen Port eines BMXNOM0200-Moduls anzuschließen, muss das Modem folgende Merkmale aufweisen:

- 10 bzw. 11 Bits pro Zeichen unterstützen, falls der Terminal-Port im seriellen Modbus verwendet wird:
 - 7 bzw. 8 Datenbits
 - 1 bzw. 2 Stoppbits
 - Ungerade, gerade oder keine Parität
- Ohne Datenträgerprüfung arbeiten.

Die Signale CTS, DTR, DSR und DCD können von der Anwendung verwaltet werden.

Anschließen von Datenendeinrichtungen

Die folgende Abbildung zeigt ein Modem, das an ein BMXNOM0200-Modul angeschlossen ist:



Für den Anschluss des Modems ist ein spezielles Modemkabel erforderlich.

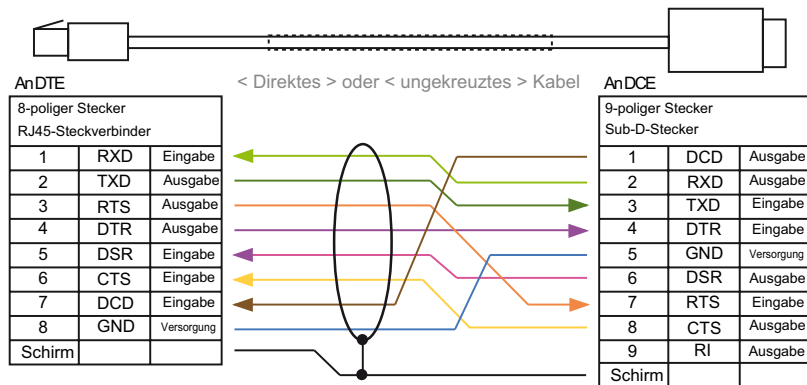
Seriell, ungekreuztes RS232-Kabel

Kabelbeispiel TCSXCN3M4F3S4:

Das serielle ungekreuzte Kabel TCSXCN3M4F3S4 ist 8-adrig und hat zwei Steckverbinder:

- RJ45-Stecker,
- 9-poliger SUB-D-Stecker.

Die folgende Abbildung zeigt die Anschlussbelegung eines seriellen, ungekreuzten Kabels TCSXCN3M4F3S4



Verbinden von Kabeln und Zubehör

Die folgende Tabelle führt die Referenznummern der Kabel und Adapter auf, die gemäß dem seriellen Anschluss der Datenendeinrichtung verwendet werden müssen:

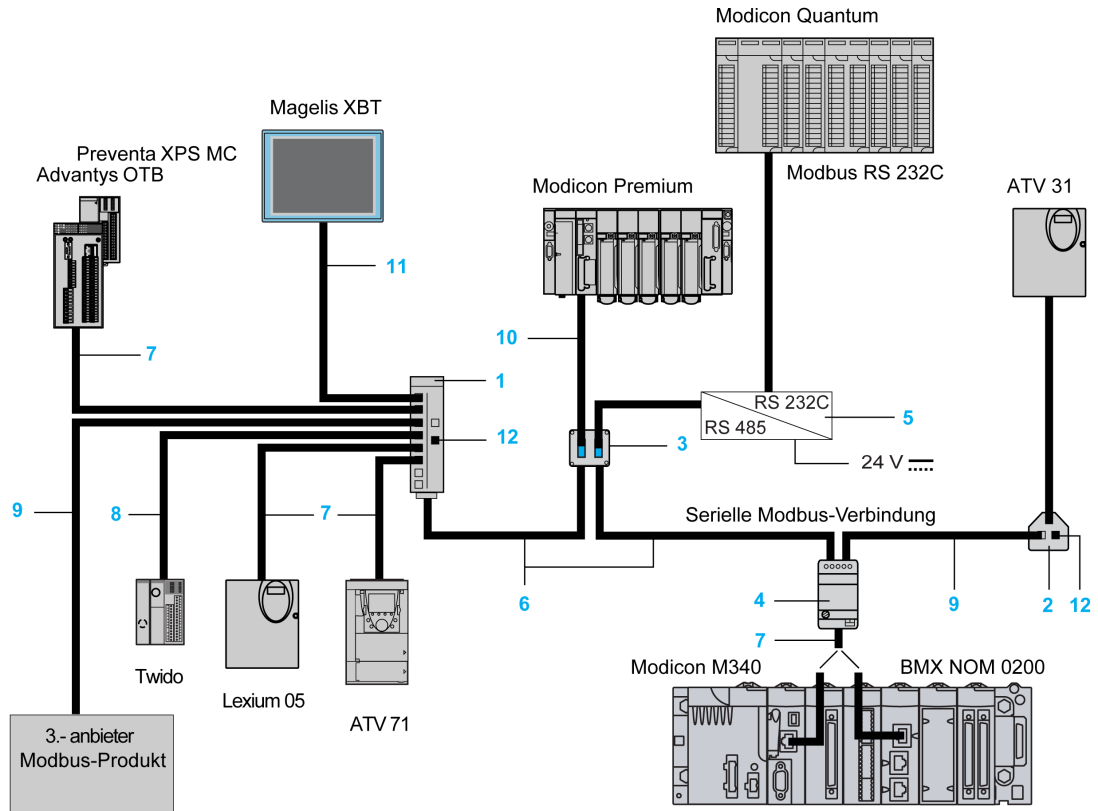
Serieller Anschluss für Datenendeinrichtung	Verkabelung
9-polige SUB-D-Buchse	● TCSMCN3M4M3S2-Kabel ● TCSXCN3M4F3S4-Kabel
25-polige SUB-D-Buchse	● TCSMCN3M4M3S2-Kabel ● TSXCTC09-Adapter

Verkabelung

Verkabelungssystem

Zur Einrichtung einer seriellen Verbindung sind mehrere Kabel und Zubehörteile erforderlich.

Die nachstehende Abbildung zeigt ein Beispiel für das Verkabelungssystem einer seriellen Modbus-Verbindung im Zeichenmodus. Die in der Abbildung ausgewiesenen **Kabel** (siehe Seite 44) und **Anschlusszubehörteile** (siehe Seite 45) werden in den folgenden Tabellen beschrieben:



Kabel

Die folgende Tabelle zeigt die verfügbaren Kabel, die mit einer seriellen Kommunikation mit diesen Prozessoren und Modulen kompatibel sind:

Abbildungsreferenz	Bezeichnung	Eigenschaften	Länge	Produkterferenz
6	Doppelt geschirmtes, verdrehtes RS485-Twisted-Pair-Trunkkabel	Zwei freie Enden	100 m	TSX CSA 100
			200 m	TSX CSA 200
			500 m	TSX CSA 500
7	Modbus-RS485-Kabel	Zwei RJ45-Stecker	0,3 m	VW3 A8 306 R03
			1 m	VW3 A8 306 R10
			3 m	VW3 A8 306 R30
-	Modbus-RS485-Kabel	• Ein RJ45-Stecker • Ein 15-poliger SUB-D-Steckverbinder	3 m	VW3 A8 306
8	Modbus-RS485-Kabel	• Ein RJ45-Stecker • Ein mini-DIN-Steckverbinder	0,3 m	TWD XCA RJ003
			1 m	TWD XCA RJ010
			3 m	TWD XCA RJ030
9	Modbus-RS485-Kabel	• Ein RJ45-Stecker • Ein freies Ende	3 m	VW3 A8 306 D30
10	Modbus-RS485-Kabel	• Ein Miniaturstecker • Ein 15-poliger SUB-D-Steckverbinder	3 m	TSX SCP CM 4630
11	RS485-Kabel für Maglis XBT-Display und -Bedienpult	• Ein RJ45-Stecker • Eine 25-polige SUB-D-Steckbuchse Hinweis: Dieses Kabel ist nicht mit dem Modul BMX NOM 0200 kompatibel.	2,5 m	XBT-Z938
-	RS485-Kabel für über die serielle Verbindung gespeiste Geräte	Zwei RJ45-Stecker Hinweis: Dieses Kabel ist nicht mit dem Modul BMX NOM 0200 kompatibel.	3 m	XBT-Z9980
-	4-drahtiges RS232-Kabel für Datenendgeräte (DTE: Data Terminal Equipment)	• Ein RJ45-Stecker • Eine 9-polige SUB-D-Steckbuchse	3 m	TCS MCN 3M4F3C2

Abbildungsreferenz	Bezeichnung	Eigenschaften	Länge	Produktreferenz
-	4-drahtiges RS232-Kabel für Datenübertragungsgeräte (DCE: Data Circuit-terminating Equipment)	• Ein RJ45-Stecker • Ein 9-poliger SUB-D-Steckverbinder	3 m	TCS MCN 3M4M3S2
-	7-drahtiges RS232-Kabel für Datenübertragungsgeräte (DCE: Data Circuit-terminating Equipment)	• Ein RJ45-Stecker • Ein 9-poliger SUB-D-Steckverbinder	3 m	TCS XCN 3M4F3S4

Anschlusszubehör

Die folgende Tabelle zeigt die verfügbaren Anschlusszubehörteile, die mit einer seriellen Kommunikation mit diesen Prozessoren und Modulen kompatibel sind:

Abbildungsreferenz	Bezeichnung	Eigenschaften	Produktreferenz
1	Modbus-Splitterbox	• Zehn RJ45-Stecker • Eine Schraubklemmenleiste	LU9 GC3
2	T-Anschlusskasten	• Zwei RJ45-Stecker • Integriertes Kabel 0,3 m mit RJ45-Stecker	VW3 A8 306 TF03
		• Zwei RJ45-Stecker • Integriertes Kabel 1 m mit RJ45-Stecker	VW3 A8 306 TF10
-	Passiver T-Anschlusskasten	• Drei Schraubklemmenleisten • RC-Leitungsendadapter	TSX SCA 50
3	Passives 2-Kanal-Teilnehmer-Socket	• Zwei 15-polige SUB-D-Steckbuchsen • Zwei Schraubklemmenleisten • RC-Leitungsendadapter	TSX SCA 62
4	Isolierter RS485-T-Anschlusskasten	• Ein RJ45-Steckverbinder • Eine Schraubklemmenleiste	TWD XCA ISO
-	T-Anschlusskasten	Drei RJ45-Steckverbinder	TWD XCA T3RJ

Abbildungsreferenz	Bezeichnung	Eigenschaften	Produktreferenz
-	Modbus-/Bluetooth-Adapter	• Ein Bluetooth-Adapter mit einem RJ45-Stecker • Ein Kabelsatz für PowerSuite mit zwei RJ45-Steckern • Ein Kabelsatz für TwidoSuite mit einem RJ45- und einem mini-DIN-Stecker • Ein 9-poliger Adapter RJ45/SUB-D für ATV-Regelantriebe	VW3 A8 114
5	RS232C/RS485-Leitungsadapter ohne Modemsigale	19,2 KBit/s	XGS Z24
12	Leitungsabschluss für RJ45-Stecker	• Widerstand 120 Ω • Kapazität 1 nF	VW3 A8 306 RC
-	Leitungsabschluss für Schraubklemmenleiste	• Widerstand 120 Ω • Kapazität 1 nF	VW3 A8 306 DRC
-	Adapter für Nicht-Standardgeräte	• Zwei 25-polige SUB-D-Steckverbinder	XBT ZG999
-	Adapter für Nicht-Standardgeräte	• Ein 25-poliger SUB-D-Steckverbinder • Ein 9-poliger SUB-D-Steckverbinder	XBT ZG909
-	Adapter für Datenendgeräte	• Ein 9-poliger SUB-D-Steckverbinder • Eine 25-polige SUB-D-Steckbuchse	TSX CTC 07
-	Adapter für Datenendgeräte	• Ein 9-poliger SUB-D-Steckverbinder • Ein 25-poliger SUB-D-Steckverbinder	TSX CTC 10
-	Adapter für Datenübertragungsgeräte (DCE)	• Eine 9-polige SUB-D-Steckbuchse • Ein 25-poliger SUB-D-Steckverbinder	TSX CTC 09

HINWEIS: Diese Liste mit Kabeln und Zubehörteilen ist nicht vollständig.

Teil II

Softwaretechnische Implementierung der seriellen Modbus- und Zeichenmoduskommunikation

Inhalt dieses Teils

Dieser Abschnitt enthält eine Beschreibung der softwaretechnischen Implementierung der seriellen Modbus- und Zeichenmoduskommunikation mit Control Expert-Software.

Inhalt dieses Teils

Dieser Teil enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
3	BMXNOM0200 - Einschränkungen und Implementierungsregeln	49
4	Serielle Modbus-Kommunikation	55
5	Kommunikation im Zeichenmodus	91
6	BMXNOM0200 Moduldiagnose	115
7	Sprachobjekte der Modbus- und Zeichenmoduskommunikation	121
8	Dynamischer Protokollwechsel	157

Kapitel 3

BMXNOM0200 - Einschränkungen und Implementierungsregeln

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Beschränkungsregeln für BMXNOM0200-Module	50
Implementierungsregeln für BMXNOM0200	52

Beschränkungsregeln für BMXNOM0200-Module

Übersicht

Die Anzahl der BMXNOM0200-Module in einer Hardwarekonfiguration ist verbunden mit dem Folgenden:

- Plattform (M340, M580 und Quantum)
- Modulinstallation (in lokalen Racks oder in der X80-Station).
- Kanalkonfiguration (Master oder Slave).

HINWEIS: Jeder konfigurierte Kanal des BMXNOM0200-Moduls zählt als Expertenkanal, wenn Sie die maximale Anzahl an Expertenkanälen in einer Konfiguration berechnen.

Wenn die Anwendung generiert wird, prüft Control Expert, dass die entsprechenden Grenzen nicht überschritten werden.

M340-Plattform

HINWEIS: Bei der Installation mit einer M340-SPS, muss für das Modul BMXNOM0200 eine CPU mit einer Betriebssystemversion von mind. 02.10 installiert sein.

Die maximale Anzahl an BMXNOM0200-Modulen, die in einer M340 SPS-Station konfiguriert werden können, ist verbunden mit Folgendem:

- Den Funktionen der Modicon M340-SPS (*siehe Modicon M340, Prozessoren, Konfigurationshandbuch*)
- Der Anzahl der bereits konfigurierten Expertenkanäle
- Der Anzahl der auf jedem BMXNOM0200-Modul konfigurierten Kanäle

M580-Plattform

In einem M580-System ergibt sich die maximale Anzahl an konfigurierbaren BMXNOM0200-Modulen aus den jeweiligen Beschränkungen für lokale Racks und X80-Stationen (dezentrale Racks).

Lokale Racks: Die maximale Anzahl an BMXNOM0200-Modulen, die in lokalen M580-Racks konfiguriert werden können (d. h. lokale und lokale erweiterte Racks), ist verbunden mit Folgendem:

- Der maximal zulässigen Anzahl an Expertenkanälen in einer lokalen Konfiguration (*siehe Modicon M580, Hardware, Referenzhandbuch*)
- Der Anzahl der bereits konfigurierten Expertenkanäle
- Der Anzahl der auf jedem BMXNOM0200-Modul konfigurierten Kanäle

X80-Station: Die maximale Anzahl an BMXNOM0200-Modulen, die in jeder X80-Station konfiguriert werden kann (mit einem X80 Performance-EIO-Adaptermodul BMXCRA31210 oder BMECRA31210) folgt den folgenden Regeln:

- maximal 36 Expertenkanäle
- maximal sechs konfigurierte BMXNOM0200-Module

HINWEIS: Bei der M580-CPU mit Betriebssystemversion $\leq$ V2.40, ist die maximale Anzahl auf vier konfigurierte BMXNOM0200-Module beschränkt.

HINWEIS: In einem M580 Hot Standby-System kann das BMXNOM0200-Modul nur in X80-Stationen konfiguriert werden (dezentrale Haupt- oder Erweiterungsracks).

Quantum-Plattform

In einem Quantum-System können maximal **16** BMXNOM0200-Module konfiguriert werden, einschließlich der folgenden Beschränkung für die X80-Station:

X80-Station: Die maximale Anzahl an BMXNOM0200-Modulen, die in jeder X80-Station konfiguriert werden kann (mit einem X80 Performance-EIO-Adaptermodul BMXCRA31210 oder BMECRA31210) folgt den folgenden Regeln:

- maximal 36 Expertenkanäle
- maximal vier konfigurierte BMXNOM0200-Module
- maximal vier als Master konfigurierte Kanäle

Beispielsweise kann die maximale Konfiguration in einer X80-Station mit zwei BMXNOM0200-Modulen erreicht werden, wenn beide Kanäle des jeweiligen Moduls als Master konfiguriert sind.

Implementierungsregeln für BMXNOM0200

Übersicht

Die Verfügbarkeit der BMXNOM0200-Modulfunktionen ist verbunden mit dem Folgenden:

- Plattform (M340, M580 und Quantum)
- Modulinstallation (in lokalen Racks oder in der X80-Station)
- Firmware-Version des Moduls

Die folgenden Tabellen zeigen die Verfügbarkeit und Einschränkungen der BMXNOM0200-Modulfunktionalitäten nach Plattform. Die Tabellen enthalten ebenfalls die zu konfigurierenden Control Expert **Hardware-Katalog**-Geräte.

M340-Plattform

Modulfunktionen und -anforderungen			Lokales M340-Rack	
BMXNOM0200 Kommunikationsprotokolle	Modbus	Master	Ja	Ja ⁽¹⁾
		Slave	Ja	Ja ⁽¹⁾
	Character Mode		Ja	Ja ⁽¹⁾
BMXNOM0200 Anforderungen	Firmware-Version		V1.0	Mindestens V1.2
	Control Expert Hardware-Katalog -Gerät		BMXNOM0200	BMXNOM0200.2 (SV>=1.2)
(1) Der Expertenmodus zur Konfiguration von Zeitüberschreitungen ist individuell mit der Anwendung verbunden und passt sich an die spezifischen Eigenschaften bestimmter Modems an. Auf der physischen RS-232-Leitung ermöglicht die Konfiguration der Hardware zur Verwaltung der Signale eine Auswahl zwischen den Modi DTE und DCE.				

M580-Plattform

Das Modul BMXNOM0200 kann sowohl in den lokalen M580-Racks als auch in der Modicon X80-Station mit einem X80-Leistungs-EIO-Adaptermodul (BMXCRA31210 oder BMECRA31210) installiert und konfiguriert werden.

HINWEIS: In einem M580 Hot Standby-System kann das BMXNOM0200-Modul nur in einer Modicon X80-Station (dezentrales Haupt- oder Erweiterungsrack) installiert und konfiguriert werden.

Modulfunktionen und -anforderungen			Lokale M580-Racks (Haupt- und Erweiterungsracks)	X80-Station über M580 PAC (dezentrale Haupt- und Erweiterungsracks)		
BMXNOM0200 Kommunikationsprotokolle		Modbus	Master	Ja	Ja	
			Slave	Ja	Nein	Ja ⁽¹⁾
		Character Mode		Ja	Ja	Ja
Anforderungen	BMXNOM0200	Firmware-Version		Mindestens V1.2	Mindestens V1.4	Mindestens V1.5
		Control Expert Hardware-Katalog -Gerät		BMXNOM0200.2 (SV>=1.2)	BMXNOM0200.3 (SV>=1.4)	BMXNOM0200.4 (SV>=1.5)
	BMXCRA31210 Drop-end- Kommunikator	Firmware-Version		–	Mindestens V2.08	Mindestens V2.14
		Control Expert Hardware-Katalog -Gerät		–	Alle ⁽²⁾	BMXCRA31210 (SV>=2.10)
	Oder					
	BMECRA31210 Drop-end- Kommunikator	Firmware-Version		–	Mindestens V2.00	Mindestens V2.14
Control Expert Hardware-Katalog -Gerät		–	Alle ⁽²⁾	BMECRA31210 (SV>=2.10)		
(1) Der Modbus-Slave ist nur auf der physischen RS-232 Leitung und im RTU-Modus verfügbar und konfigurierbar. (2) Wählen Sie das Control Expert Hardware-Katalog -Gerät entsprechend der Firmware-Version des Drop-End-Kommunikators aus. – Ohne Bedeutung						

Quantum-Plattform

Das Modul BMXNOM0200 kann nur in einer EIO-Modicon-X80-Station mit einem X80-Leistungs-EIO-Adaptermodul (BMXCRA31210 oder BMECRA31210) installiert und konfiguriert werden.

HINWEIS: Die Konfiguration eines BMXNOM0200-Moduls als Modbus-Slave ist nur mit einem BMXCRA31210 möglich und erfordert eine Verbindung des Quantum 140NOC78•00 zum Quantum 140CRP31200. Für weitere Informationen, siehe das Kapitel Schenllstart: BMXNOM0200 als Modbus-Slave über eine Quantum SPS (*siehe Seite 161*).

Modulfunktionen und -anforderungen			X80-Station über M580-SPS (dezentrale Haupt- oder Erweiterungsracks)		
BMXNOM0200 Kommunikationsprotokolle		Modbus	Master	Ja	Ja
			Slave	Nein	Ja ⁽¹⁾
		Character Mode		Ja	Ja
Anforderungen	BMXNOM0200	Firmware-Version		Mindestens V1.4	Mindestens V1.5
		Control Expert Hardware-Katalog -Gerät		BMXNOM0200.3 (SV>=1.4)	BMXNOM0200.4 (SV>=1.5)
	BMXCRA31210 Drop-end-Kommunikator	Firmware-Version		Alle	Mindestens V2.14
		Control Expert Hardware-Katalog -Gerät		Alle ⁽²⁾	BMXCRA31210 (SV>=2.13)
	Oder				
	BMECRA31210 Drop-end-Kommunikator	Firmware-Version		Alle	–
		Control Expert Hardware-Katalog -Gerät		Alle ⁽²⁾	–
(1) Der Modbus-Slave ist nur auf der physischen RS-232 Leitung und im RTU-Modus verfügbar und konfigurierbar. (2) Wählen Sie das Control Expert Hardware-Katalog -Gerät entsprechend der Firmware-Version des Drop-End-Kommunikators aus. – Ohne Bedeutung					

Kapitel 4

Serielle Modbus-Kommunikation

Inhalt dieses Kapitels

Dieses Kapitel beschreibt die Software-Implementierung der seriellen Modbus-Kommunikation für BMXNOM0200-Module.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
4.1	Allgemeine Informationen	56
4.2	Konfiguration der seriellen Modbus-Kommunikation	63
4.3	Programmierung der seriellen Modbus-Kommunikation	76
4.4	Debuggen einer seriellen Modbus-Kommunikation	88

Abschnitt 4.1

Allgemeine Informationen

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die allgemeine Beschreibung der seriellen Modbus-Kommunikation und ihrer Dienste.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Über Modbus Serial	57
Leistung	58
Zugriff auf die seriellen Verbindungsparameter	60

Über Modbus Serial

Einführung

Die Kommunikation über Modbus ermöglicht den Datenaustausch zwischen allen mit dem Bus verbundenen Geräten. Modbus seriell ist ein Protokoll, dass eine hierarchische Struktur (ein Master und mehrere Slaves) erstellt.

Der Master verwaltet den Austausch auf zwei Arten:

- Der Master tauscht Daten mit dem Slave aus und wartet auf die Antwort.
- Der Master tauscht Daten mit allen Slaves aus, ohne auf eine Antwort zu warten (Broadcast).

HINWEIS: Achten Sie darauf, dass zwei Master (auf demselben Bus) nicht gleichzeitig einen Request senden, denn dann gehen die Requests verloren und jede Rückmeldung enthält ein negatives Ergebnis, z. B. 16#0100 (Request konnte nicht verarbeitet werden) oder 16#ODFF (Slave nicht vorhanden).

WARNUNG

KRITISCHER DATENVERLUST

Verwenden Sie die Kommunikationsports ausschließlich für die nicht-kritische Datenübertragung.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

Leistung

Einführung

Die folgenden Tabellen können zur Beurteilung der typischen Austauschzeiten bei der Modbus-Kommunikation nach verschiedenen Kriterien herangezogen werden.

Die angezeigten Ergebnisse entsprechen der durchschnittlichen Ausführungszeit der `READ_VAR`-Funktion in Millisekunden.

Definition der Austauschzeit

Die Austauschzeit ist die Zeit zwischen Einrichtung und Ende eines Austauschs. Sie umfasst die Zeit der Kommunikation über die serielle Verbindung.

Der Austausch wird eingerichtet, sobald die Kommunikationsfunktion aufgerufen wird.

Er endet, wenn eines der folgenden Ereignisse auftritt:

- Daten werden empfangen.
- Eine Anomalität tritt auf.
- Der Timeout-Zeitraum läuft ab.

Austauschzeit für ein Wort

Die nachstehende Tabelle zeigt die Austauschzeiten für ein Wort bei der Modbus-Kommunikation auf einem BMX NOM 0200-Modul (der Modbus-Slave ist ein zyklischer BMX P34 1000):

Austauschzeit ⁽¹⁾ in ms		Zykluszeit in ms		
		Zyklisch	10	50
Baudrate der Kommunikation in Bits pro Sekunde	4800	65	68	100
	9600	38	47	50
	19200	29	38	50
	38400	24	30	50
	57600	17	20	50
	115200	17	20	50
(1) Alle oben aufgeführten Austauschzeiten basieren auf Messwerten mit einer Genauigkeitsmarge von +/- 10 ms				

Austauschzeit für 100 Wörter

Die nachstehende Tabelle zeigt die Austauschzeiten für 100 Wörter bei der Modbus-Kommunikation auf einem BMX NOM 0200-Modul (der Modbus-Slave ist ein zyklischer BMX P34 1000):

Austauschzeit ⁽¹⁾ in ms		Zykluszeit in ms		
		Zyklisch	10	50
Baudrate der Kommunikation in Bits pro Sekunde	4800	560	560	600
	9600	286	295	300
	19200	152	160	200
	38400	86	90	100
	57600	56	60	100
	115200	36	40	50
(1) Alle oben aufgeführten Austauschzeiten basieren auf Messwerten mit einer Genauigkeitsmarge von +/- 10 ms				

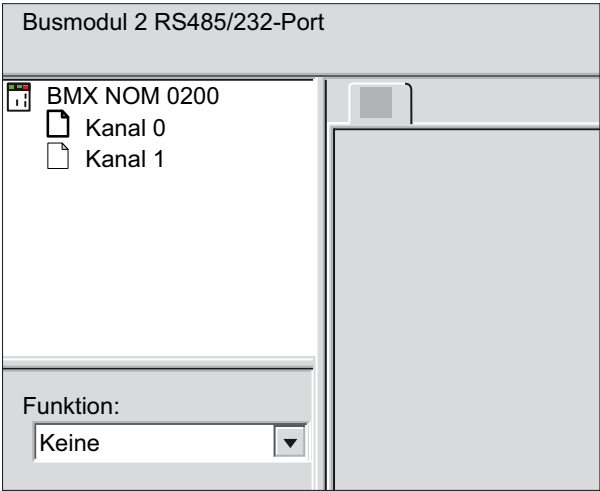
Zugriff auf die seriellen Verbindungsparameter

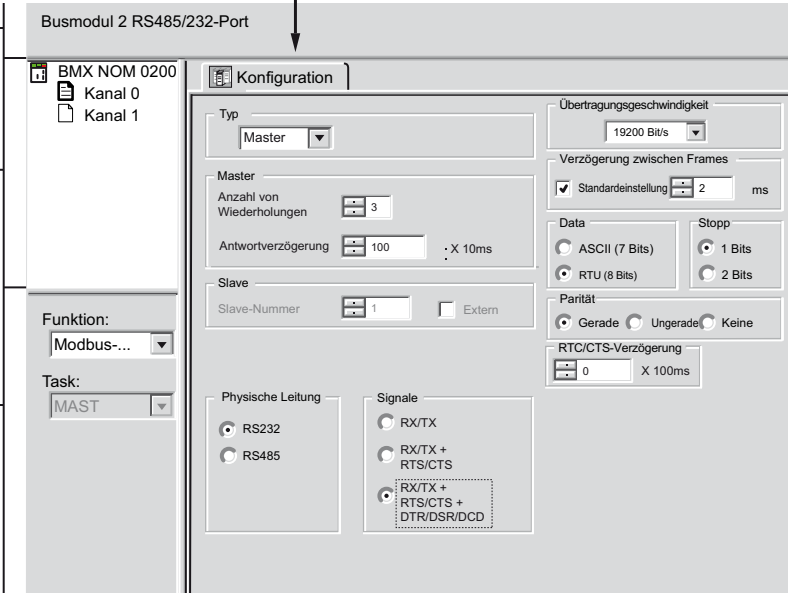
Einführung

Auf den folgenden Seiten wird der Zugriff auf das Fenster zur Konfiguration des seriellen Anschlusses für das BMXNOM0200-Modul beschrieben. Ferner werden die allgemeinen Elemente der Fenster zur Konfiguration und Fehlerbehebung der Modbus- und Zeichenmodusverbindung vorgestellt:

Zugriff auf die serielle Verbindung

Die folgende Tabelle beschreibt den Zugriff auf die serielle Verbindung eines BMXNOM0200-Moduls:

Schritt	Aktion
1	Öffnen Sie den Hardware-Konfigurations-Editor.
2	Doppelklicken Sie auf das BMXNOM0200-Modul.
3	Wählen Sie den zu konfigurierenden Kanal (Kanal 0 oder Kanal 1). Ergebnis bei Auswahl von Kanal 0: 

Schritt	Aktion
4	Wählen Sie die Funktion Modbus-Verbindung. Ergebnis bei Auswahl von Kanal 0: 
	5

Beschreibung des Fensters „Konfiguration“:

In der folgenden Tabelle sind die verschiedenen Elemente des Konfigurationsfensters aufgeführt:

Taste	Element	Funktion
1	Registerkarten	Die Registerkarte im Vordergrund zeigt die aktuell verwendete Betriebsart (in diesem Beispiel Konfiguration). Jeder Modus kann über die entsprechende Registerkarte ausgewählt werden. Folgende Modi sind verfügbar: ● Konfiguration ● Debuggen (Zugriff nur im Online-Modus möglich). ● Diagnose (Zugriff nur im Online-Modus möglich).
2	Modul Bereich	Zeigt eine Modulreferenz und im Online-Modus den LED-Status des Moduls an.
3	Kanalbereich	Ermöglicht Folgendes: ● Durch Klicken auf ein BMX NOM 0200 werden folgende Registerkarten angezeigt: ○ Übersicht enthält die Merkmale des Geräts. ○ E/A-Objekte zur Vorsymbolisierung der Eingangs-/Ausgangsobjekte. ○ Fehler zum Zugriff auf die Gerätefehler bietet (Zugriff nur im Online-Modus). ● Durch Klicken auf den Kanal 0 oder Kanal 1 werden folgende Registerkarten angezeigt: ○ Konfiguration ○ Debuggen ○ Fehler ● Die Anzeige des vom Benutzer (im Variableneditor) festgelegten Namens und Symbols des Kanals.
4	Bereich der allgemeinen Parameter	Dieser Bereich ermöglicht die Auswahl der allgemeinen Parameter für den Kanal: ● Funktion: Die verfügbaren Funktionen sind Keine, Modbus-Verbindung und Zeichenmodus-Verbindung. Standardmäßig ist diese Funktion konfiguriert als Keine. ● Task: Legt die Master-Task fest, in der die impliziten Austauschobjekte des Kanals ausgetauscht werden. Dieser Bereich erscheint grau und kann daher nicht konfiguriert werden.
5	Konfigurations-, Debugging- oder Fehler-Bereich	Dieser Bereich dient im Konfigurationsmodus der Konfiguration der Kanalparameter. Im Debug-Modus wird er zum Debuggen des Kommunikationskanals verwendet. Im Diagnosemodus wird sie zur Anzeige der aktuellen Fehler entweder auf Modul- oder auf Kanalebene verwendet.

Abschnitt 4.2

Konfiguration der seriellen Modbus-Kommunikation

Inhalt dieses Abschnitts

In diesem Abschnitt wird die Softwarekonfiguration für die serielle Modbus-Kommunikation beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Konfigurationsfenster für die serielle Modbus-Kommunikation	64
Anwendungsrelevante Modbus-Parameter	66
Parameter der Signal- und der physischen Leitung im Modbus-Modus	68
Übertragungsbezogene Modbus-Parameter	71
Einstellen der BMXNOM0200-MODBUS-Slaveadresse ohne Control Expert?	74

Konfigurationsfenster für die serielle Modbus-Kommunikation

Allgemein

Die folgenden Seiten geben eine Einführung in das Konfigurationsfenster für die serielle Modbus-Kommunikation.

Zugriff auf das Konfigurationsfenster

Die folgende Tabelle beschreibt die Vorgehensweise zum Zugreifen auf das Konfigurationsfenster der seriellen Modbus-Kommunikation:

Schritt	Aktion
1	Öffnen Sie im Projekt-Browser das Unterverzeichnis des BMX NOM 0200 (<i>siehe Seite 60</i>).
2	Wählen Sie in der eingblendeten Anzeige den zu konfigurierenden Kanal und die Funktion "Modbus-Verbindung".

Abbildung

Die folgende Abbildung zeigt das Standardkonfigurationsfenster der seriellen Modbus-Kommunikation an Kanal 0:

Konfiguration

Typ

Slave

Master

Anz. Wiederholungen0

Antwortverzögerung1 X 10 ms

Slave

Slave-Nummer1

Übertragungsgeschwindigkeit

19200 Bits/s

Verzögerung zwischen Frames

☒ Standard2 ms

Daten

☐ ASCII(7 Bits)

☒ RTU(8 Bits)

Stopp

☒ 1 Bit

☐ 2 Bits

Parität

☒ Gerade

☐ Ungerade

☐ Keine

RTS/CTS-Verzögerung

0 X 100 ms

Physikalische Leitung

☐ RS232

☒ RS485

Signale

☒ RX/TX

☐ RX/TX + RTS/CTS

☐ RX/TX + RTS/CTS + DTR/DSR

Beschreibung

Diese Bereiche dienen der Konfiguration der Kanalparameter. Der Zugriff auf diese Zonen ist im Online-Modus möglich. Im Offline-Modus kann zwar auf diese Zonen, jedoch möglicherweise nicht auf einige Parameter zugegriffen werden. Diese werden deshalb ausgegraut angezeigt.

Die folgende Tabelle zeigt die verschiedenen Bereiche des Konfigurationsfensters der Modbus-Verbindung:

Element	Kommentar
Parameter der Anwendung (siehe Seite 66)	Diese Parameter sind in drei verschiedenen Bereichen verfügbar: • Typ • Master • Slave
Parameter der Signal- und physischen Leitung (siehe Seite 68)	Diese Parameter sind in drei verschiedenen Bereichen verfügbar: • Physische Leitung • Signale • RTS/CTS-Verzögerung
Parameter der Übertragung (siehe Seite 71)	Diese Parameter sind in fünf verschiedenen Bereichen verfügbar: • Übertragungsgeschwindigkeit • Verzögerung zwischen Frames • Datenbits • Stoppbits • Parität

HINWEIS: Bei der Konfiguration der seriellen Modbus-Kommunikation im Master-Modus wird der Bereich **Slave** grau abgeblendet dargestellt und kann nicht geändert werden - und umgekehrt.

Standardwerte

Die folgende Tabelle zeigt die Standardwerte der Parameter für die serielle Modbus-Kommunikation:

Konfigurationsparameter		Wert
Parameter der Anwendung	Typ	Slave
	Slave-Nummer	1
Parameter der Signal- und physischen Leitung	Physische Leitung	RS485
	Signale	RX/TX
Parameter der Übertragung	Übertragungsgeschwindigkeit	19200 Bit/s
	Verzögerung zwischen Frames	2 ms
	Daten	RTU (8 Bits)
	Stopp	1 Bit
	Parität	Gerade

Anwendungsrelevante Modbus-Parameter

Einführung

Nach der Konfiguration des Kommunikationskanals müssen Sie die anwendungsspezifischen Parameter eingeben.

Auf diese Parameter kann von drei Konfigurationsbereiche aus zugegriffen werden:

- Bereich **Typ**
- Bereich **Master**
- Bereich **Slave**

Bereich Typ

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:

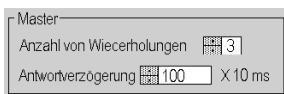


In dieser Zone können Sie die Rolle des Moduls für die serielle Modbus-Kommunikation konfigurieren:

- **Master:** Wenn das Modul der Master ist.
- **Slave:** Wenn das Modul der Slave ist.

Bereich Master

Auf die unten abgebildete Konfigurationszone kann nur zugegriffen werden, wenn im Bereich **Typ** der Wert **Master** ausgewählt ist:



Diese Zone ermöglicht die Eingabe folgender Parameter:

- **Anzahl von Wiederholungen:** Anzahl der Verbindungsaufbauversuche, die der Master unternimmt, bevor er den Slave als nicht vorhanden betrachtet.
Der Standardwert ist 3.
Die möglichen Werte liegen zwischen 0 und 15.
Der Wert 0 gibt an, dass der Master über keine Wiederholschleife verfügt.
- **Antwortverzögerung:** Die Zeit, die zwischen dem Versand der Anforderung durch den Master und dem Beginn der Wiederholschleife vergeht, wenn der Slave nicht antwortet. Sie entspricht der maximalen Zeit zwischen dem Versand des letzten Zeichens der vom Master versandten Anforderung und dem Empfang des ersten Zeichens der vom Slave zurückgesandten Anforderung.
Der Standardwert ist 1 Sekunde (100 * 10 ms).
Die möglichen Werte liegen zwischen 10 ms und 10 Sekunden.

HINWEIS: Die Antwortverzögerung des Masters muss mindestens der längsten Antwortverzögerung der am Bus vorhandenen Slaves entsprechen.

HINWEIS: Im Broadcast-Modus wird der als **Antwortverzögerung** konfigurierte Wert als Broadcast-Verzögerung verwendet: Minstdauer zwischen zwei Austauschvorgängen im Broadcast-Modus.

Der Bereich Slave

Auf die unten abgebildete Konfigurationszone kann nur zugegriffen werden, wenn im Bereich **Typ** der Wert **Slave** ausgewählt ist:



Diese Zone ermöglicht die Eingabe der Slave-Nummer des Prozessors:

Der Standardwert ist 1.

Die möglichen Werte liegen zwischen 1 und 247.

Bei der Auswahl von **Extern** ist das Feld **Slave-Nummer** grau abgeblendet und das Modul verwendet den in seinem internen (*siehe Seite 74*) FLASH-Speicher abgelegten Wert für die Slave-Adresse.

HINWEIS: Wenn die im FLASH-Speicher abgelegte Adresse keine gültige Modbus-Adresse ist, wird die Standardeinstellung der Slave-Adresse von 248 verwendet.

Bei einer Aktualisierung der Firmware des Moduls wird die im FLASH-Speicher abgelegte Slave-Adresse auf 248 gesetzt. Die FLASH-Adresse muss dann mit einem neuen Befehl initialisiert werden.

Parameter der Signal- und der physischen Leitung im Modbus-Modus

Einführung

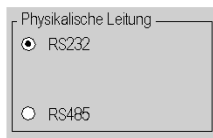
Nach der Konfiguration des Kommunikationskanals müssen Sie die Signal- und physischen Leitungsparameter eingeben.

Diese Parameter sind in drei verschiedenen Bereichen verfügbar:

- Bereich **Physische Leitung**
- Bereich **Signale**
- Bereich **RTS/CTS-Verzögerung**

Bereich Physische Leitung

Dieser unten gezeigte Konfigurations-Bereich ist nur für Kanal 0 verfügbar (und für Kanal 1 ausgegraut und auf RS485 konfiguriert):



In diesem Bereich können Sie zwischen zwei Arten von physischer Leitung für den seriellen Port am BMXNOM0200-Modul wählen:

- RS232-Leitung
- RS485-Leitung

Bereich Signale

HINWEIS: Wenn die Konfiguration **Physische Leitung** als **RS485** erfolgt, wird der gesamte Bereich grau unterlegt (ausgeblendet) und der Standardwert ist **RX/TX**.

Die verfügbaren RS232-Signale sind abhängig vom Control Expert **Hardware-Katalog**-Gerät, wie nachstehend gezeigt:

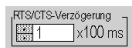
BMXNOM0200	BMXNOM0200.2 BMXNOM0200.3 BMXNOM0200.4
Signale ☒ RX/TX ☐ RX/TX + RTS/CTS ☐ RX/TX + RTS/CTS + DTR/DSR	Signale ☒ RX/TX ☐ RX/TX + RTS/CTS DTE mode ☐ RX/TX + RTS/CTS DCE mode ☐ RX/TX + RTS/CTS + DTR/DSR/DCD

In diesem Bereich können Sie die von der physischen RS232-Leitung unterstützten Signale auswählen:

- RX/TX
- Signale zur Hardware-Datenflusskontrolle:
 - RX/TX + RTS/CTS
 - RX/TX + RTS/CTS DTE mode
 - RX/TX + RTS/CTS DCE mode
- Modemsignale:
 - RX/TX + RTS/CTS + DTR/DSR
 - RX/TX + RTS/CTS + DTR/DSR/DCD

Bereich RTS/CTS-Verzögerung

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:

The image shows a small rectangular configuration window titled "RTS/CTS-Verzögerung". Inside the window, there is a text input field containing the number "1". To the right of the input field is a label "x100 ms".

Der Bereich für die **RTS/CTS-Verzögerung** ist verfügbar, wenn ein Signal mit einer RTS/CTS-Hardware-Datenflusskontrolle konfiguriert wird.

Der Algorithmus zur RTS/CTS-Hardware-Datenflusskontrolle soll den Überlauf des Empfangspuffers (Voll duplex) verhindern.

Die RTS/CTS-Verzögerung entspricht der Timeout-Verzögerung zwischen dem Setzen von RTS und dem Setzen von CTS. Eine RTS/CTS-Verzögerung ungleich 0 gibt auch die maximale Wartezeit zwischen zwei Zeichen an, nachdem RTS und CTS gesetzt wurden. Wenn der Wert auf 0 gesetzt ist, können UARTs für eine unbegrenzte Zeit in einem Wartezustand auf das Setzen von CTS verharren, so dass der Wert 0 nur in bestimmten Fällen verwendet wird, wenn zum Beispiel das RTS-Signal auf das CTS-Signal geschleift wird, um den korrekten Betrieb aller Verbindungen zu prüfen.

HINWEIS: Der Standardwert beträgt 0 ms.

Übertragungsbezogene Modbus-Parameter

Einführung

Nach der Konfiguration des Kommunikationskanals müssen Sie die Parameter für die Übertragung eingeben.

Auf diese Parameter kann von fünf Bereichen aus zugegriffen werden:

- Bereich **Übertragungsgeschwindigkeit**
- Bereich **Verzögerung zwischen Frames**
- Bereich **Daten**
- Bereich **Stopp**
- Bereich **Parität**

Bereich Übertragungsgeschwindigkeit

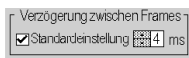
Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:



Sie können sie zur Auswahl der Übertragungsgeschwindigkeit der seriellen Modbusverbindung verwenden. Die ausgewählte Geschwindigkeit muss mit den anderen Geräten übereinstimmen. Die konfigurierbaren Werte sind 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 und (nur für Kanal 0 im RS232-Modus) 115200 Bits pro Sekunde.

Bereich Verzögerung zwischen Frames

Diese unten gezeigte Konfigurationszone ist nur im RTU-Modus zugänglich (und im ASCII-Modus grau abgeblendet):



Der Bereich **Verzögerung zwischen Frames** ist der Mindestzeitraum zwischen zwei Frames beim Empfang. Diese Verzögerung wird verwaltet, wenn das BMXNOM0200-Modul (Master oder Slave) Nachrichten empfängt.

HINWEIS: Der Standardwert ist von der ausgewählten Übertragungsgeschwindigkeit abhängig.

HINWEIS: Die Verzögerung zwischen Frames sollte der Standardwert sein, damit die Kompatibilität mit Modbus gewährleistet werden kann. Bei einem nicht konformen Slave kann der Wert geändert werden und sollte für Master und sämtlichen Slaves auf dem Bus identisch sein.

Bereich Daten

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:



Diese Zone ermöglicht die Eingabe des Codetyps, der bei der Verwendung der seriellen Modbus-Verbindung verwendet werden soll. Das Feld wird in Übereinstimmung mit den anderen mit dem Bus verbundenen Geräten eingestellt. Zwei konfigurierbare Modi stehen zur Auswahl:

- RTU-Modus:
 - Die Zeichen sind 8-Bit-codiert.
 - Das Ende des Frames wird erkannt, wenn für mindestens 3,5 Zeichen Stille herrscht.
 - Die Integrität des Frames wird mithilfe eines Worts überprüft, das als CRC-Prüfsumme bekannt ist und im Frame enthalten ist.
- ASCII-Modus:
 - Die Zeichen sind 7-Bit-codiert.
 - Der Anfang des Frames wird erkannt, wenn das Zeichen „:“ empfangen wird.
 - Das Ende des Frames wird mittels Wagenrücklauf und Zeilenvorschub erkannt.
 - Die Integrität des Frames wird mithilfe eines Byte überprüft, das als LRC-Prüfsumme bekannt ist und im Frame enthalten ist.

Bereich Stopp

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:

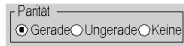


Der Bereich **Stopp** ermöglicht die Eingabe der für die Kommunikation verwendeten Anzahl an Stoppbits. Das Feld wird in Übereinstimmung mit den anderen Geräten eingestellt. Konfigurierbare Werte:

- **1 Bit**
- **2 Bit**

Bereich Parität

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:

The image shows a small rectangular window titled "Parität". Inside the window, there are three radio buttons labeled "Gerade", "Ungerade", and "Keine". The "Gerade" radio button is selected, indicated by a filled circle.

Verwenden Sie diesen Bereich, um anzugeben, ob ein Paritätsbit hinzugefügt wird oder nicht. Hier können Sie auch den Typ des Paritätsbits festlegen. Das Feld wird in Übereinstimmung mit den anderen Geräten eingestellt. Konfigurierbare Werte:

- **Gerade**
- **Ungerade**
- **Ohne**

Einstellen der BMXNOM0200-MODBUS-Slaveadresse ohne Control Expert?

Bedingungen und Voraussetzungen

Die FLASH-Adresse kann in jedem Modus aktualisiert werden, die Aktualisierung wird jedoch nur in einer Betriebsart berücksichtigt.

Die nachstehende Liste verweist auf die Bedingungen und Voraussetzungen für eine Einstellung der BMXNOM0200-MODBUS-Adresse ohne Control Expert:

- Um die FLASH-Adresse verwenden zu können, muss das Modul konfiguriert werden:
 - Im MODBUS-Slaveprotokoll über das Kontrollkästchen **EXTERN**.
 - Im MODBUS-Masterprotokoll oder im **ZEICHEN**-Modus und dann Wechsel zum MODBUS-Slaveprotokoll.

Aktualisieren der MODBUS-Slaveadresse im FLASH über Anwendungsbefehle

Die nachstehende Tabelle zeigt die Vorgehensweise zur Aktualisierung der MODBUS-Slaveadresse im FLASH mithilfe von Anwendungsbefehlen:

Schritt	Aktion
1	Speichern Sie die Slaveadresse in %MW _{r.m.c.25} .
2	Setzen Sie das Bit %MW _{r.m.c.24.7} .
3	Senden Sie den Befehl WRITE_CMD an den Modulkanal.
4	Prüfen Sie den Befehlsabschluss (%MW _{r.m.c.0.1} fallende Flanke) und die Annahme des Befehls (%MW _{r.m.c.1.1} steht auf Null, d. h. kein Fehler) => Der FLASH wird aktualisiert.
5	Führen Sie eine der folgenden Betriebsarten im Kanal aus, damit die neue Adresse berücksichtigt wird: ● Laden einer Anwendung ● Kaltstart ● Warmstart ● Hot-Swapping ● Protokolltausch (TO SLAVE)
6	Führen Sie die Funktion READ_STS im Kanal aus, um die Slaveadresse im höherwertigen Byte von %MW _{r.m.c.3} zu prüfen.

HINWEIS: Verschiedene Anweisungen können im selben Befehl eingebettet werden. Wenn eine der Anweisungen nicht ausgeführt werden kann, wird der ganze Befehl zurückgewiesen und keine Anweisung wird ausgeführt.

Aktualisieren der MODBUS-Slaveadresse im FLASH über die serielle Verbindung

Die nachstehende Tabelle zeigt die Vorgehensweise zur Aktualisierung der MODBUS-Slaveadresse im FLASH über die serielle Verbindung:

Schritt	Aktion
1	Konfigurieren Sie das MASTER-Gerät mit denselben seriellen Verbindungsparametern wie ein Kanal des Moduls.
2	Verbinden Sie den MASTER über eine Punkt-zu-Punkt-Verbindung mit dem Modul.
3	Senden Sie den Request 0x11 an die Punkt-zu-Punkt-Adresse: 0xF8 0x11 0x01 channelnumber(0 oder 1) slaveID(0..0xF8)
4	Vergewissern Sie sich, dass die Rückmeldung OK ist => Der FLASH wird aktualisiert.
5	Führen Sie eine Betriebsart im Kanal aus, damit die in Schritt 4 vorgenommene Änderung berücksichtigt wird.
6	Senden Sie den Request 0x11, um die neue Slaveadresse zu überprüfen: slaveID 0x11 0x01

HINWEIS: Nehmen Sie regelmäßig Änderungen am FLASH vor, um eine Beschädigung dieser Komponente zu vermeiden (max. 100.000 Schreibzyklen).

Abschnitt 4.3

Programmierung der seriellen Modbus-Kommunikation

Inhalt dieses Abschnitts

In diesem Abschnitt wird das Programmierverfahren für die Implementierung der seriellen Modbus-Kommunikation beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Von einem Master-Modul in einer Modbus-Verbindung unterstützte Dienste	77
Von einem Slave-Modul in einer Modbus-Verbindung unterstützte Dienste	78
Details zum Modbus Experten-Modus	80

Von einem Master-Modul in einer Modbus-Verbindung unterstützte Dienste

Einführung

Wenn ein BMXNOM0200-Modul in einer Modbus-Verbindung als Master verwendet wird, unterstützt es verschiedene Dienste über Kommunikationsfunktionen. Diese Funktionen sind plattformabhängig.

Kommunikationsfunktionen

Spezifische Kommunikationsfunktionen sind für das Senden und den Empfang von Daten über einen Modbus-Kommunikationskanal definiert:

Plattform	Variablen/Register lesen	Variablen/Register schreiben	Modbus-Request senden
M580	READ_VAR	WRITE_VAR	DATA_EXCH
M340	READ_VAR	WRITE_VAR	DATA_EXCH
Quantum	READ_REG_QX	WRITE_REG_QX	EXCH_QX

Nähere Informationen zu diesen Kommunikationsfunktionen finden Sie unter *EcoStruxure™ Control Expert – Kommunikation, Bausteinbibliothek*.

Datenaustausch

Das Lesen oder Schreiben von Variablen erfolgt über die Adressierung der folgenden Requests an das ausgewählte Slave-Gerät:

Diese Requests greifen auf die Kommunikationsfunktionen zurück:

Modbus-Request	Funktionscode	Kommunikationsfunktion
Lesen von Bits	16#01 oder 16#02	READ_VAR, READ_REG_QX
Lesen von Wörtern	16#03 oder 16#04	READ_VAR, READ_REG_QX
Schreiben von Bits	16#0F	WRITE_VAR, WRITE_REG_QX
Schreiben von Wörtern	16#10	WRITE_VAR, WRITE_REG_QX

Allgemeiner formuliert ist es möglich, über die Kommunikationsfunktionen DATA_EXCH oder EXCH_QX (plattformabhängig) beliebige Modbus-Requests an ein Slave-Gerät zu senden.

Von einem Slave-Modul in einer Modbus-Verbindung unterstützte Dienste

Einführung

Wenn ein BMXNOM0200-Modul in einer Modbus-Verbindung als Slave-Prozessor verwendet wird, unterstützt es verschiedene Dienste.

Datenaustausch

Ein Slave-Modul verwaltet die folgenden Requests:

Modbus-Request	Funktionscode	Steuerungsobjekt
Lesen von "n" Ausgangsbits	16#01	%M
Lesen von "n" Ausgangswörtern	16#03	%MW
Schreiben von "n" Ausgangsbits	16#0F	%M
Schreiben von "n" Ausgangswörtern	16#10	%MW
Lesen/Schreiben von „n“ Ausgangswörtern	16#17	%MW

HINWEIS: Lesen/Schreiben mehrerer %MW

Der Request `WRITE` wird vor dem Request `READ` ausgeführt, damit bei der E/A-Abfrage (Scanning) gleichzeitig dieselben Register geschrieben und gelesen werden können. Wenn die Größe des Datenaustauschs für `WRITE` oder `READ` außerhalb des zulässigen Wertebereichs liegt, wird der Status „Ungültige Datenadresse“ zurückgegeben, wenn jedoch nur der Request `READ` scheitert, dann wird der Request `WRITE` mit demselben Status ausgeführt.

Diagnose und Wartung

Die vom einem BMXNOM0200-Modul als Modbus-Slave verwalteten Diagnose- und Wartungsrequests sind nachfolgend aufgeführt:

Bezeichnung	Funktionscode/Unterfunktionscode
Ausnahmestatus lesen	16#07
Kommunikationsoption neu starten	16#08 / 16#01
Diagnoseregister zurückgeben	16#08 / 16#02
ASCII-Eingangsbegrenzer ändern	16#08 / 16#03
Modus "Nur überwachen" forcieren.	16#08 / 16#04
Zähler und Diagnoseregister löschen	16#08 / 16#0A
Rückgabe der Anzahl von Busnachrichten	16#08 / 16#0B
Rückgabe der Anzahl von Buskommunikationsfehlern	16#08 / 16#0C
Rückgabe der Anzahl von Busausnahmefehlern	16#08 / 16#0D
Rückgabe der Anzahl von Slave-Nachrichten	16#08 / 16#0E

Bezeichnung	Funktionscode/Unterfunktionscode
Rückgabe der Anzahl von fehlenden Slave-Antworten	16#08 / 16#0F
Rückgabe der Anzahl von negativen Bestätigungen vom Slave	16#08 / 16#10
Rückgabe der Anzahl von ausgelasteten Slaves	16#08 / 16#11
Rückgabe der Anzahl von Buszeichenüberläufen	16#08 / 16#12
Kommunikationsereignis-Zähler laden	16#0B
Kommunikationsereignis-Protokoll laden	16#0C
Slave-Identifikation melden	16#11
Slave-Identifikation schreiben	16#11 / 16#01

Details zum Modbus Experten-Modus

Kommunikation im Experten-Modus

Der Experten-Modus besteht aus einer Reihe von Befehlen, die an das Modul gesendet werden können, um zusätzliche Funktionen zu erhalten.

Adresse	Standardsymbol	Austauschtyp	Typ	Bedeutung
%MWr.m.c.24	CONTROL	Explizit	INT	Befehlssignal, Protokoll ändern
%MWr.m.c.24.0		Explizit	BOOL	Lokale Zähler löschen
%MWr.m.c.24.1		Explizit	BOOL	Den Wiederholungszähler im MODBUS-Master-Modus dynamisch ändern (%MW26)
%MWr.m.c.24.2		Explizit	BOOL	Die Slave-Antwortverzögerung (%MW28) für einen bestimmten Slave (%MW27) im Master-Modus dynamisch ändern
%MWr.m.c.24.3		Explizit	BOOL	Standard-Slave-Blindzeit ändern; das Zeichen zum Ignorieren des Empfangs nach einem empfangenen Frame, das an die CPU weitergeleitet wird (%MW29)
%MWr.m.c.24.4		Explizit	BOOL	Interne MODBUS-RTU-Zeitgebungen t1,5ch (%MW31), t3,5ch (%MW30) und Austauschverzögerung (%MW32) ändern. Diese Wertaktualisierung kann das Modul u. U. stören, falls es in Betrieb ist.

Adresse	Standardsymbol	Austauschtyp	Typ	Bedeutung
%MWr.m.c.24.6		Explizit	BOOL	HALF/FULL DUPLEX-Modem-Verwaltungsmodus ändern • Wenn dieses Bit gleichzeitig mit RTS_ON gesetzt wird (%MWr.m.c.24.10 funktioniert auch mit RTS_OFF %MWr.m.c.24.11, und DTR verwenden, wenn %MWr.m.r.24.8 oder %MWr.m.r.24.9 verwendet werden), wird der Halbduplex-Modus aktiviert. • Wenn dieses Bit gesetzt ist, aber keine der RTS/DTR-Bits (weder %MWr.m.c.24.8, %MWr.m.c.24.9, %MWr.m.c.24.10 noch %MWr.m.c.24.11), wird der Vollduplex-Modus aktiviert. Mit %MW26 wird die Startverzögerung (StartDelay) und mit %MW27 die Endverzögerung (EndDelay) eingestellt. Also können die Bits %MW24.5 und %MW24.1 und %MW24.2 nicht gleichzeitig verwendet werden. HINWEIS: Unter Umständen muss der Benutzer den ordnungsgemäßen Zustand der RTS/DTR-Signale wiederherstellen, nachdem der Befehl akzeptiert wurde.
%MWr.m.c.24.7	SAVE_SLAVE_ADDR	Explizit	BOOL	Modbus-Slave-Adresse im FLASH speichern (%MW25).
%MWr.m.c.24.8	DTR_ON	Explizit	BOOL	DTR-Signal einstellen (positive Spannung)

Adresse	Standardsymbol	Austauschtyp	Typ	Bedeutung
%MWr.m.c.24.9	DTR_OFF	Explizit	BOOL	DTR-Signal zurücksetzen (negative Spannung)
%MWr.m.c.24.10		Explizit	BOOL	RTS-Signal einstellen (positive Spannung)
%MWr.m.c.24.11		Explizit	BOOL	RTS-Signal zurücksetzen (negative Spannung)
%MWr.m.c.24.12	TO_MODBUS_MASTER	Explizit	BOOL	In den Master-Modus wechseln
%MWr.m.c.24.13	TO_MODBUS_SLAVE	Explizit	BOOL	In den Slave-Modus wechseln
%MWr.m.c.24.14	TO_CHAR_MODE	Explizit	BOOL	In den Zeichenmodus wechseln
%MWr.m.c.25	SLAVE_ADDR	Explizit	INT	Im FLASH zu speichernde Modbus-Slave-Adresse
%MWr.m.c.26		Explizit	INT	NIEDERWERTIGES BYTE: MasterRetries-Zähler: Anzahl Wiederholungen im Master-Modus [0..15], siehe %MW24.1 StartDelay wenn %MW26.6 gesetzt ist. Die Zeit, die gewartet wird, nachdem das CTS OK ist, bevor mit dem Senden des Frames begonnen wird. Dies ist nützlich für Modems, die nach CTS zusätzliche Zeit erfordern oder die das CTS-Signal nicht verwalten (in diesem Fall muss das RTS mit dem CTS verbunden sein). Diese Zeit wird in Millisekunden angegeben, die Genauigkeit beträgt etwa 3 ms. Kann nur im RS232-Modus durchgeführt werden.

Adresse	Standardsymbol	Austauschtyp	Typ	Bedeutung
%MW _{r.m.c.} 27		Explizit	INT	NIEDERWERTIGES BYTE: Slave, für den der Master die Antwortverzögerung anpassen wird [0..248, 255=ALLE], siehe %MW24.2 und %MW28 EndDelay wenn %MW24.6 gesetzt ist. Die Zeit, die nach dem Senden eines Frames gewartet wird, bevor das RTS-Signal ausgelöst wird, um so dem Modem vor der Weitergabe genügend Zeit zu geben, das Senden des Frames abzuschließen. Diese Zeit wird in Millisekunden angegeben, die Genauigkeit beträgt etwa 3 ms. Kann nur im RS232-Modus durchgeführt werden.
%MW _{r.m.c.} 28		Explizit	INT	Spezifische Antwortverzögerung für einen Slave in 10 ms [1..1000], siehe %MW24.2 und %MW27
%MW _{r.m.c.} 29		Explizit	INT	Blindzeit in 10 ms [1..10], siehe %MW24.3
%MW _{r.m.c.} 30		Explizit	INT	T3,5char: Verzögerung zwischen Frames in Millisekunden [0..10000]. Der verwendete Wert hängt von der Geschwindigkeit ab. Falls der Wert kleiner oder größer ist als die möglichen Werte, wird der untere bzw. obere Grenzwert angewendet und der Befehl wird akzeptiert. Ein Wert von 0 bedeutet keine Änderung in RTU. Die Antwortverzögerung wird erneut berechnet.

Adresse	Standardsymbol	Austauschtyp	Typ	Bedeutung
%MWr.m.c.31		Explizit	INT	T1,5char: Verzögerung zwischen Zeichen in Millisekunden [0..9999]. Der verwendete Wert hängt von der Geschwindigkeit ab. Falls der Wert kleiner oder größer ist als die möglichen Werte, wird der untere bzw. obere Grenzwert angewendet und der Befehl wird akzeptiert. Ein Wert von 0 bedeutet eine Berechnung von T1,5 als T3,5ch – 2ch (Standardberechnung).
%MWr.m.c.32		Explizit	INT	Master-Austauschverzögerung im RTU-Modus [0..256] in Millisekunden. Der Wert 0 bedeutet "keine Verzögerung"; wenn der Wert geringer als eine 10-Bit-Dauer ist, wird der Minimalwert von 10 Bit verwendet.

Beispielcode

```
(* Master-seitiges NOM ist in Rack 0 Steckplatz 9 *)
if HalfModemMaster then
  HalfModemMaster:=false;
  %MW0.9.0.24:=16#0450;(* zu Halbduplex-Modus mit RTS wechseln und MODBUS-
Zeitgebungen ändern*)
  %MW0.9.0.26:=12;(* 12 ms vor dem Senden warten, wenn CTS steigt *)
  %MW0.9.0.27:=9; (* RTS 9 ms nach Sendeende auslösen *)
  %MW0.9.0.30:=0;
  %MW0.9.0.31:=0;(*Wert des Konfigurationsbildschirms gleich 6 ms verwenden *)
  %MW0.9.0.32:=50;(* 50 ms Verzögerung, bevor ein neuer Frame gesendet wird *)
  write_cmd(%ch0.9.0);(* Befehl und Daten an den NOM-Kanal senden*)
end_if;
(* Slave-seitiges NOM ist in Rack 0 Steckplatz 3 *)
if HalfModemSlave then
  HalfModemSlave:=false;
```

```
%MW0.3.0.24:=16#0448;(* zu Halbduplex-Modus mit RTS wechseln und Slave-Blindzeit
ändern*)
%MW0.3.0.26:=12;(* 12 ms vor dem Senden warten, wenn CTS steigt *)
%MW0.3.0.27:=9; (* RTS 9 ms nach Sendeende auslösen *)
%MW0.3.0.29:=4; (* 4*10 ms Blindzeit *)
write_cmd(%ch0.3.0);(* Befehl und Daten an den NOM-Kanal senden*)
end_if;
(* optional: den Befehl automatisch senden *)
if %S0 or %S1 or %S13 then
    memoSendCmd:=true;
end_if;
(* den Modulfehler jeden Zyklus kopieren, um das Verschwinden des Moduls zu erkennen *)
memoSendCmd:=%I0.3.0.ERR;
(* wenn Modul OK ist, den Befehl ein Mal senden *)
if FE(memoSendCmd) then
    HalfModemSlave:=true;
end_if;
```

Lesbare interne NOM-Register

Der Zugriff auf interne NOM-Register kann nur im MODBUS-Modus durch Verwendung von READ_VAR EF erfolgen. Beispielcode (das NOM befindet sich in Rack 0 Steckplatz 3 *)

```
if dataCh030GetChannelGlobalInfo then
    read_var(addm('0.3.0'), '%MW', 200, 3, dataCh030Mgt, dataCh030Buff);
    (* Internal_Reg@200 werden in den Puffer dataCh030Buff kopiert *)
    dataCh030GetChannelGlobalInfo := false;
end_if;
```

- Internal_Reg@0: StartDelay in ms (Genauigkeit etwa 3 ms) (Lese- oder Schreibzugriff)
- Internal_Reg@1: EndDelay in ms (Genauigkeit etwa 3 ms) (Lese- oder Schreibzugriff)
- Internal_Reg@200: Schnittstellen-Versionsnummer = 1
- Internal_Reg@201: Im FLASH-Speicher gespeicherte Slave-Adresse
- Internal_Reg@202: 1=FLASH kann geändert werden, 0=darf nicht geändert werden
- Internal_Reg@1000: Interner Modbus-Master-RTU-Code ch0=1110, ch1=2110
- Internal_Reg@1002: 0 = Vollduplex - Hardwareflusssteuerung oder RS485 ; 1 = Halbduplex - Richtung wird automatisch von dem Modul mit RTS verwaltet
- Internal_Reg@1010: Internes Senden der Verzögerung zwischen Zeichen in Bit (nbbits*1000/Geschwindigkeit => Dauer in ms) [T1,5S].

- Internal_Reg@1012: Internes Empfangen der Verzögerung zwischen Zeichen in Bit [T1,5R].
- Internal_Reg@1014: Internes Senden der Verzögerung zwischen Frames in Bit [T3,5R].
- Internal_Reg@1016: Internes Empfangen der Verzögerung zwischen Frames in Bit [T3,5R].
- Internal_Reg@1018: Zeit, die vor dem Senden der nächsten Frames gewartet wird, in Bits.
- Internal_Reg@1090: MasterRetries-Zähler.
- Internal_Reg@1100: Slave-Antwortverzögerung für Broadcast in 10 ms.
- Internal_Reg@1101: Slave-Antwortverzögerung für Slave 1 in 10 ms.
- ...
- Internal_Reg@1348: Slave-Antwortverzögerung für Punkt-zu-Punkt-Adresse (248).
- Internal_Reg@1500: Interner Modbus-RTU-Slave-Code ch0=1120, ch1=2120
- Internal_Reg@1502: 0 = Vollduplex - Hardwareflusssteuerung oder RS485 ; 1 = Halbduplex - Richtung wird automatisch von dem Modul mit RTS verwaltet
- Internal_Reg@1510: Internes Senden der Verzögerung zwischen Zeichen in Bit $(nbbits \cdot 1000 / \text{Geschwindigkeit} \Rightarrow \text{Dauer in ms})$ [T1,5S].
- Internal_Reg@1512: Internes Empfangen der Verzögerung zwischen Zeichen in Bit [T1,5R].
- Internal_Reg@1514: Internes Senden der Verzögerung zwischen Frames in Bit [T3,5R].
- Internal_Reg@1516: Internes Empfangen der Verzögerung zwischen Frames in Bit [T3,5R].
- Internal_Reg@1518: Zeit, die vor dem Senden der nächsten Frames gewartet wird, in Bits.
- Internal_Reg@1602: Blindzeit nach Empfang in ms.
- Internal_Reg@1606: Nur-Lauschen-Modus aktiv = 1, (nicht aktiv = 0).
- Internal_Reg@2000: Interner Modbus-Master-ASCII-Code ch0=1210, ch1=2210
- Internal_Reg@2002: 0 = Vollduplex - Hardwareflusssteuerung oder RS485 ; 1 = Halbduplex - Richtung wird automatisch von dem Modul mit RTS verwaltet
- Internal_Reg@2010: Internes Senden der Verzögerung zwischen Zeichen in Bit $(nbbits \cdot 1000 / \text{Geschwindigkeit} \Rightarrow \text{Dauer in ms})$ [T1,5S].
- Internal_Reg@2012: Internes Empfangen der Verzögerung zwischen Zeichen in Bit [T1,5R].
- Internal_Reg@2014: Internes Senden der Verzögerung zwischen Frames in Bit [T3,5R].
- Internal_Reg@2014: Internes Senden der Verzögerung zwischen Frames in Bit [T3,5R].
- Internal_Reg@2014: Internes Senden der Verzögerung zwischen Frames in Bit [T3,5R].
- Internal_Reg@2016: Internes Empfangen der Verzögerung zwischen Frames in Bit [T3,5R].
- Internal_Reg@2018: Zeit, die vor dem Senden der nächsten Frames gewartet wird, in Bits.
- Internal_Reg@2090: MasterRetries-Zähler.
- Internal_Reg@2100: Slave-Antwortverzögerung für Broadcast in 10 ms.
- Internal_Reg@2101: Slave-Antwortverzögerung für Slave 1 in 10 ms.
- ...
- Internal_Reg@2348: Slave-Antwortverzögerung für Punkt-zu-Punkt-Adresse (248).
- Internal_Reg@2500: Interner Modbus-ASCII-Slave-Code ch0=1220, ch1=2220
- Internal_Reg@2502: 0 = Vollduplex - Hardwareflusssteuerung oder RS485 ; 1 = Halbduplex - Richtung wird automatisch von dem Modul mit RTS verwaltet
- Internal_Reg@2510: Internes Senden der Verzögerung zwischen Zeichen in Bit $(nbbits \cdot 1000 / \text{Geschwindigkeit} \Rightarrow \text{Dauer in ms})$ [T1,5S].
- Internal_Reg@2512: Internes Empfangen der Verzögerung zwischen Zeichen in Bit [T1,5R].
- Internal_Reg@2514: Internes Senden der Verzögerung zwischen Frames in Bit [T3,5R].

- Internal_Reg@2516: Internes Empfangen der Verzögerung zwischen Frames in Bit [T3,5R].
- Internal_Reg@2518: Zeit, die vor dem Senden der nächsten Frames gewartet wird, in Bits.
- Internal_Reg@2600: Verwendete Slave-Adresse.
- Internal_Reg@2602: Blindzeit nach Empfang in ms.
- Internal_Reg@2606: Nur-Lauschen-Modus aktiv = 1, (nicht aktiv = 0).
- Internal_Reg@3000: Interner Zeichenmodus-Code ch0=1000, ch1=2000
- Internal_Reg@3002: 0 = Vollduplex - Hardwareflussteuerung oder RS485 ; 1 = Halbduplex - Richtung wird automatisch von dem Modul mit RTS verwaltet
- Internal_Reg@3100: 0=keine Stopp-Kriterien aktiv, 1= Stopp bei Stille oder Stopp bei Zeichenende
- Internal_Reg@3102: Interne Stille in Bit (Min. ist 2 Bit, Max. ist 65.535 Bit)
- Internal_Reg@3104: Erstes zu verwendendes End-of-Frame-Byte, 16#0100 bedeutet kein Byte
- Internal_Reg@3106: First EOF: 1=End-of-Frame-Byte im Frame zulassen, 0=End-of-Frame-Byte entfernen
- Internal_Reg@3108: Zweites End-of-Frame-Byte
- Internal_Reg@3110: Second EOF: 1=End-of-Frame-Byte im Frame zulassen, 0=End-of-Frame-Byte entfernen

Abschnitt 4.4

Debuggen einer seriellen Modbus-Kommunikation

Debug-Fenster der seriellen Modbus-Kommunikation

Allgemeines

Der Zugriff auf das Debug-Fenster der seriellen Modbus-Kommunikation ist ausschließlich im Online-Modus möglich.

Zugriff auf das Debug-Fenster

Die folgende Tabelle beschreibt die Vorgehensweise zum Zugreifen auf das Debug-Fenster der seriellen Modbus-Kommunikation:

Schritt	Aktion
1	Rufen Sie das Konfigurationsfenster der seriellen Modbus-Kommunikation auf. <i>(siehe Seite 64)</i>
2	Wählen Sie auf dem angezeigten Fenster die Registerkarte "Debuggen" aus.

Beschreibung des Debug-Fensters

Das Debug-Fenster ist in zwei oder drei Zonen aufgeteilt:

- Die Zone für Typ und Slave-Nummer,
- Die Zone "Zähler,
- Die Signal-Zone (bei RS232).

Die Zone für Typ und Slave-Nummer

Wenn ein Modul als Master an der Modbus-Verbindung betrieben wird, sieht diese Zone wie folgt aus:



Wenn ein Modul als Slave an der Modbus-Verbindung betrieben wird, sieht diese Zone wie folgt aus:



Zone "Zähler"

Diese Zone sieht wie folgt aus:

Diese Zone enthält die verschiedenen Debugging-Zähler.

Die Schaltfläche "Zähler zurücksetzen" setzt alle Zähler des Debug-Modus auf Null zurück.

Arbeitsweise der Zähler

Die Debugging-Zähler für die serielle Modbus-Kommunikation sind:

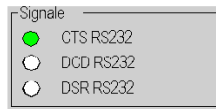
- **Busnachrichtenzähler:** Dieser Zähler gibt die Anzahl der Nachrichten an, die das Modul auf der seriellen Verbindung erkannt hat. Nachrichten mit einem negativen CRC-Prüfergebnis werden nicht gezählt.
- **Zähler für Buskommunikationsfehler:** Dieser Zähler gibt die Anzahl der negativen CRC-Prüfergebnisse an, die vom Modul gezählt werden. Falls ein Zeichenfehler (Überlauf, Paritätsfehler) auftritt oder die Nachricht weniger als 3 Byte umfasst, kann das System, das die Daten empfängt, die CRC-Prüfung nicht durchführen. In solchen Fällen wird der Zähler entsprechend erhöht.
- **Zähler für Slave-Ausnahmefehler:** Dieser Zähler gibt die Anzahl der Modbus-Ausnahmefehler an, die vom Modul erkannt werden.
- **Slave-Nachrichtenzähler:** Dieser Zähler gibt die Anzahl der von der Modbus-Verbindung empfangenen und verarbeiteten Nachrichten an.
- **Zähler für fehlende Slave-Antworten:** Dieser Zähler gibt die Anzahl der vom dezentralen System gesendeten Nachrichten an, für die keine Antwort empfangen wurde (weder normale noch Ausnahmeantwort). Es wird auch die Anzahl der im Broadcast-Modus empfangenen Nachrichten gezählt.
- **Zähler für negative Slave-Bestätigungen:** Dieser Zähler gibt die Anzahl der an das dezentrale System gesendeten Nachrichten an, für die es eine negative Bestätigung ausgegeben hat.
- **Slave-Ausgelastet-Zähler:** Dieser Zähler gibt die Anzahl der an das dezentrale System gesendeten Nachrichten an, für die es eine Ausnahmenachricht ausgegeben hat, dass der Slave ausgelastet ist.
- **Buszeichenüberlauf-Zähler:** Dieser Zähler gibt die Anzahl der an das Modul gesendeten Nachrichten an, dass es wegen eines Zeichenüberlaufs auf dem Bus keine Erfassung vornehmen kann. Ein Überlauf kann folgende Ursachen haben:
 - Daten vom Typ "Zeichen" werden auf dem seriellen Port schneller übertragen, als sie gespeichert werden können.
 - Datenverlust wegen eines Hardwareereignisses.

HINWEIS: Alle Zähler beginnen die Zählung beim letzten Neustart, nach einer Zählerlöschung bzw. beim Modulstart.

Die Signal-Zone

Diese Zone wird nur angezeigt, wenn RS232 im Konfigurationsfenster ausgewählt wurde. Beim Auswahl von RS485 im Konfigurationsfenster wird dieses Fenster nicht angezeigt.

Die Zone "Signale" sieht wie folgt aus:



Dieser Bereich gibt die Aktivität der Signale an:

- **CTS RS232:** Zeigt die Aktivität des CTS-Signals an.
- **DCD RS232:** Zeigt die Aktivität des DCD-Signals an.
- **DSR RS232:** Zeigt die Aktivität des DSR-Signals an.

Kapitel 5

Kommunikation im Zeichenmodus

Inhalt dieses Kapitels

In diesem Kapitel wird die Softwareimplementierung für die Kommunikation im Zeichenmodus für das BMXNOM0200-Modul beschrieben.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
5.1	Allgemeine Informationen	92
5.2	Konfiguration der Zeichenmoduskommunikation	93
5.3	Programmierung der Zeichenmoduskommunikation	103
5.4	Debuggen der Zeichenmoduskommunikation	112

Abschnitt 5.1

Allgemeine Informationen

Informationen zur Kommunikation im Zeichenmodus

Einführung

Bei der Kommunikation im Zeichenmodus können Dialog- und Kommunikationsfunktionen mit folgenden Geräten ausgeführt werden:

- Herkömmliche Peripheriegeräte (Drucker, Tastaturbildschirm, Endgeräte in Werkstätten usw.)
- Spezielle Peripheriegeräte (Strichcode-Lesegeräte usw.)
- Rechner (Prüfung, Produktionsmanagement usw.)
- Heterogene Geräte (numerische Befehle, Drehzahlregler usw.)
- Externe Modems

WARNUNG

KRITISCHER DATENVERLUST

Verwenden Sie die Kommunikationsports ausschließlich für die nicht-kritische Datenübertragung.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

Abschnitt 5.2

Konfiguration der Zeichenmoduskommunikation

Inhalt dieses Abschnitts

In diesem Abschnitt wird das Konfigurationsverfahren für die Implementierung der Zeichenmoduskommunikation beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
BMXNOM0200 Konfigurationsfenster für die Zeichenmodus-Kommunikation	94
Parameter zur Erkennung des Nachrichtenendes im Zeichenmodus	96
Übertragungsparameter im Zeichenmodus	98
Parameter der Signal- und der physischen Leitung im Zeichenmodus	100

BMXNOM0200 Konfigurationsfenster für die Zeichenmodus-Kommunikation

Allgemein

Die folgenden Seiten geben eine Einführung in das Konfigurationsfenster für die Zeichenmodus-Kommunikation.

Zugriff auf das Konfigurationsfenster

Die folgende Tabelle beschreibt die Vorgehensweise zum Zugriff auf das Konfigurationsfenster für die Zeichenmodus-Kommunikation:

Schritt	Aktion
1	Öffnen Sie im Projekt-Browser das Unterverzeichnis des BMX NOM 0200 (<i>siehe Seite 60</i>).
2	Wählen Sie in der eingblendeten Anzeige den zu konfigurierenden Kanal und die Funktion "Zeichenmodusverbindung".

Konfigurationsfenster der Zeichenmoduskommunikation

Die folgende Abbildung zeigt das Standardkonfigurationsfenster der Zeichenmoduskommunikation an Kanal 0:

The screenshot shows the 'Konfiguration' (Configuration) window for serial communication. It is divided into several sections:

- Stopp bei Empfang** (Stop at reception):
 - Zeichen 1** (Character 1):
 - ☐ Stopp
 - ☐ CR ☐ LF
 - ☐ Inkl. Zeichen
 - Zeichen 2** (Character 2):
 - ☐ Stopp
 - ☐ CR ☐ LF
 - ☐ Inkl. Zeichen
- Physikalische Leitung** (Physical connection):
 - ☒ RS232
 - ☐ RS485
- Signale** (Signals):
 - ☒ RX/TX
 - ☐ RX/TX + RTS/CTS DTE mode
 - ☐ RX/TX + RTS/CTS DCE mode
 - ☐ RX/TX + RTS/CTS + DTR/DSR/DCD
- Übertragungsgeschwindigkeit** (Baud rate):
 - 9600 Bits/s
- Stopp bei Ruhe** (Stop at idle):
 - ☐ Stopp ms
- Daten** (Data):
 - ☐ 7 Bits
 - ☒ 8 Bits
- Stopp** (Stop):
 - ☒ 1 Bit
 - ☐ 2 Bits
- Parität** (Parity):
 - ☐ Gerade
 - ☒ Ungerade
 - ☐ Keine
- RTS/CTS-Verzögerung** (RTS/CTS delay):
 - X 100 ms
- Polarisierung** (Polarity):
 - ☐ Keine
 - ☐ Eindeutige Polarisation
 - ☐ Verteilte Polarisation

HINWEIS: In diesem Beispiel sind die Bereiche **Polarisierung** und **RTS/CTS-Verzögerung** ausgegraut dargestellt, weil eine physische RS232-Leitung und RX/TX-Signale ausgewählt wurde.

Beschreibung

Diese Bereiche dienen der Konfiguration der Kanalparameter. Der Zugriff auf diese Bereiche ist im Online-Modus möglich. Im Offline-Modus kann zwar auf diese Zonen, jedoch möglicherweise nicht auf einige Parameter zugegriffen werden. Diese werden deshalb ausgegraut angezeigt.

Die folgende Tabelle zeigt der verschiedenen Bereiche des Konfigurationsfensters der Zeichenmodus-Kommunikation:

Element	Kommentar
Parameter zur Erkennung des Nachrichtenendes (<i>siehe Seite 96</i>)	Diese Parameter sind in zwei verschiedenen Bereichen verfügbar: • Stopp bei Empfang • Stopp bei Stille
Parameter der Signal- und physischen Leitung (<i>siehe Seite 100</i>)	Auf diese Parameter kann über vier Zonen zugegriffen werden: • Physische Leitung • Signale • RTS/CTS-Verzögerung, • Polarisierung.
Parameter der Übertragung (<i>siehe Seite 98</i>)	Diese Parameter sind in vier verschiedenen Bereichen verfügbar: • Übertragungsgeschwindigkeit • Daten • Stoppbits • Parität

Standardwerte

Die folgende Tabelle zeigt die Standardwerte der Parameter für die Zeichenmodus-Kommunikation:

Konfigurationsparameter		Kanal 0	Kanal 1
Parameter zur Erkennung des Nachrichtenendes	Stopp bei Empfang	Ohne	Ohne
	Stopp bei Stille	Ohne	Ohne
Parameter der Signal- und physischen Leitung	Physische Leitung	RS232	RS485
	Signale	RX/TX	RX/TX
	RTS/CTS-Verzögerung	Nicht zutreffend	Nicht zutreffend
	Polarisierung	Nicht zutreffend	Ohne
Parameter der Übertragung	Übertragungsgeschwindigkeit	9600 Bit/s	9600 Bit/s
	Daten	8 Bit	8 Bit
	Stopp	1 Bit	1 Bit
	Parität	Ungerade	Ungerade

Parameter zur Erkennung des Nachrichtendes im Zeichenmodus

Einführung

Nach der Konfiguration des Kommunikationskanals müssen Sie die Nachrichtendeparameter eingeben.

Diese Parameter sind in zwei verschiedenen Bereichen verfügbar:

- Bereich **Stopp bei Empfang**: Stopp bei Empfang eines Sonderzeichens.
- Bereich **Stopp bei Stille**: Stopp bei Stille.

Verwendungsbedingungen

Die Auswahl von **Stopp bei Stille** bedeutet, dass die Auswahl von **Stopp bei Empfang** aufgehoben wird und umgekehrt.

Bereich Stopp bei Empfang

Dieser Konfigurationsbereich wird wie folgt im Fenster angezeigt:

Stopp bei Empfang

Zeichen 1

☒ Stopp

☐ CR ☒ LF

☐ Inklusive Zeichen

Zeichen 2

☒ Stopp

☒ CR ☐ LF

☐ Inklusive Zeichen

Ein Empfangsrequest kann bei Empfang eines Sonderzeichens beendet werden.

Durch die Auswahl der Option **Stop** kann ein **Stopp bei Empfang** mit einem bestimmten Nachrichtenendzeichen aktiviert werden:

- **CR**: Das Nachrichtenende wird anhand eines Wagenrücklaufs erkannt.
- **LF**: Das Nachrichtenende wird anhand eines Zeilenvorschubs erkannt.
- **Dateneingabefeld**: Hier können Sie ein anderes Nachrichtenendzeichen als einen Wagenrücklauf oder Zeilenvorschub in Form eines Dezimalwerts eingeben:
 - Zwischen 0 und 255, wenn die Daten über 8 Bits codiert sind.
 - Zwischen 0 und 127, wenn die Daten über 7 Bits codiert sind.
- **Zeichen enthalten**: Das Nachrichtenendzeichen wird in die Empfangstabelle der SPS-Anwendung aufgenommen.

Es besteht die Möglichkeit, zwei Nachrichtenendzeichen zu konfigurieren. In obigem Fenster wird das Ende des Nachrichtenempfangs durch einen Zeilenvorschub oder Wagenrücklauf erkannt.

Bereich Stopp bei Stille

Dieser Konfigurationsbereich wird wie folgt im Fenster angezeigt:



Er ermöglicht die Erkennung des Nachrichtenendes beim Empfang anhand des Fehlens eines Nachrichtenendzeichens während eines bestimmten Zeitraums.

Stopp bei Stille wird durch Aktivierung des **Stopp**-Kontrollkästchens ausgewählt. Die Dauer der Stille (ausgedrückt in Millisekunden) wird im Dateneingabefeld festgelegt.

Der minimale Wert dieser Dauer ist die Zeit, die der Übertragung von 1,5 Zeichen entspricht. In einer Anzahl von Bits und in Abhängigkeit von der Konfiguration der Start- und Stopp-Bits gilt für die minimale Dauer der Stille:

Zeichenlänge insg. (Bit)	Min. Dauer der Stille (Bit)
8	12
9	12
10	15
11	15

Konvertieren Sie die Zahl in der rechten Spalte gemäß der konfigurierten Übertragungsgeschwindigkeit in eine Dauer.

HINWEIS: Der zulässige Wertebereich reicht von 1 ms bis 10000 ms und ist von der jeweils ausgewählten Übertragungsgeschwindigkeit abhängig.

Übertragungsparameter im Zeichenmodus

Einführung

Nach der Konfiguration des Kommunikationskanals müssen Sie die Parameter für die Übertragung eingeben.

Diese Parameter sind in vier verschiedenen Bereichen verfügbar:

- Bereich **Übertragungsgeschwindigkeit**
- Bereich **Daten**
- Bereich **Stopp**
- Bereich **Parität**

Bereich Übertragungsgeschwindigkeit

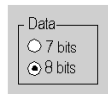
Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:



Sie können in diesem Bereich die Übertragungsgeschwindigkeit des Zeichenmodus-Protokolls auswählen. Die ausgewählte Geschwindigkeit muss mit den anderen Geräten übereinstimmen. Die konfigurierbaren Werte sind 300, 600, 1200, 2400, 4800, 9600, 19200, 57600 und (nur für Kanal 0 im RS232-Modus) 115200 Bits pro Sekunde.

Bereich Daten

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:



In diesem Bereich können Sie die Größe der über die Verbindung ausgetauschten Daten angeben.

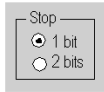
Konfigurierbare Werte:

- **7 Bit**
- **8 Bit**

Es wird empfohlen, die Anzahl der Datenbits in Übereinstimmung mit dem verwendeten dezentralen Gerät einzustellen.

Bereich Daten

Dieser Bereich sieht wie folgt aus:



Der Bereich **Stopp** ermöglicht die Eingabe der für die Kommunikation verwendeten Anzahl an Stoppbits. Es wird empfohlen, die Anzahl der Stoppbits in Übereinstimmung mit dem verwendeten dezentralen Gerät einzustellen.

Konfigurierbare Werte:

- **1 Bit**
- **2 Bit**

Bereich Parität

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:



Verwenden Sie diesen Bereich, um anzugeben, ob ein Paritätsbit hinzugefügt wird oder nicht. Hier können Sie auch den Typ des Paritätsbits festlegen. Es wird empfohlen, die Parität in Übereinstimmung mit dem verwendeten dezentralen Gerät einzustellen.

Konfigurierbare Werte:

- **Gerade**
- **Ungerade**
- **Ohne**

Parameter der Signal- und der physischen Leitung im Zeichenmodus

Einführung

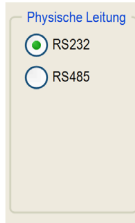
Nach der Konfiguration des Kommunikationskanals müssen Sie die Signal- und physischen Leitungsparameter eingeben.

Diese Parameter sind in drei verschiedenen Bereichen verfügbar:

- Bereich **Physische Leitung**
- Bereich **Signale**
- Bereich **RTS/CTS-Verzögerung**

Bereich Physische Leitung

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:



In diesem Bereich können Sie zwischen zwei Arten von physischer Leitung für den seriellen Port am BMXNOM0200-Modul wählen:

- Die RS 232-Leitung
- Die RS 485-Leitung

Bereich Signale

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:

Signale

☒ RX/TX

☐ RX/TX + RTS/CTS Vollduplex (DTE-Modus)

☐ RX/TX + RTS/CTS Halbduplex (DCE-Modus)

☐ RX/TX + RTS/CTS + DTR/DSR/DCD

In dieser Zone können Sie die Signale wählen, die von der physischen RS 232-Leitung unterstützt werden:

- RX/TX
- RX/TX + RTS/CTS Vollduplex (DTE-Modus)
- RX/TX + RTS/CTS Halbduplex (DCE-Modus)
- RX/TX + RTS/CTS + DTR/DSR/DCD

Wenn eine RS 485-Leitung konfiguriert ist, wird die gesamte Zone ausgegraut dargestellt. Die Standardeinstellung ist in diesem Fall RX/TX.

Bereich RTS/CTS-Verzögerung

Dieser Konfigurationsbereich wird wie folgt auf dem Bildschirm angezeigt:

RTS/CTS-Verzögerung

X 100 ms

Der Bereich **RTS/CTS-Verzögerung** ist nur verfügbar, wenn beide Kontrollkästchen RS232 und RX/TX+RTS/CTS oder RX/TX+RTS/CTS+DTR/DSR/DCD aktiviert sind. Eine RTS/CTS-Hardware-Datenflusskontrolle wird durchgeführt.

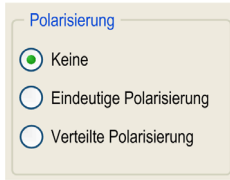
Der Algorithmus zur RTS/CTS-Hardware-Datenflusskontrolle soll den Überlauf des Empfangspuffers (Vollduplex) verhindern.

Die RTS/CTS-Verzögerung entspricht der Timeout-Verzögerung zwischen dem Setzen von RTS und dem Setzen von CTS. Eine RTS/CTS-Verzögerung ungleich 0 gibt auch die maximale Wartezeit zwischen zwei Zeichen an, nachdem RTS und CTS gesetzt wurden. Wenn der Wert auf 0 gesetzt ist, können UARTs für eine unbegrenzte Zeit in einem Wartezustand auf das Setzen von CTS verharren, so dass der Wert 0 nur in bestimmten Fällen verwendet wird, wenn zum Beispiel das RTS-Signal auf das CTS-Signal geschleift wird, um den korrekten Betrieb aller Verbindungskabel zu prüfen.

HINWEIS: Der Standardwert beträgt 0 ms.

Der Bereich Polarisierung

Der unten gezeigte Konfigurationsbereich ist bei Auswahl von **RS485** im Bereich **Physikalische Leitung** verfügbar:



The image shows a configuration window titled 'Polarisierung' with a light beige background. It contains three radio button options: 'Keine' (which is selected, indicated by a green dot), 'Eindeutige Polarisierung', and 'Verteilte Polarisierung'.

Diese Zone bietet drei Typen zur Konfiguration der Polarisierung für den Kanal:

- **Keine** wird verwendet, wenn Sie einen eigenen Abschluss verwenden.
- **Eindeutige Polarisierung** ist eine Polarisierung mit niedriger Impedanz wie in einem Modbus-Netzwerk (Sinn dieser Polarisierungsart ist es, dem Master das Halten des Standardzustands zu überlassen).
- **Verteilte Polarisierung** ist eine Polarisierung mit hoher Impedanz (wo jedes Gerät zum Halten des Standardstatus beiträgt).

Abschnitt 5.3

Programmierung der Zeichenmoduskommunikation

Inhalt dieses Abschnitts

In diesem Abschnitt wird das Programmierverfahren für die Implementierung der Zeichenmodus-kommunikation beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Zeichenmodus-Kommunikationsfunktionen	104
Details zum Experten-Modus für den Zeichenmodus	109

Zeichenmodus-Kommunikationsfunktionen

Einführung

Bestimmte Kommunikationsfunktionen sind für das Senden und den Empfang von Daten über einen Kommunikationskanal im Zeichenmodus definiert. Diese Funktionen sind plattformabhängig.

Kommunikationsfunktionen

Bestimmte Kommunikationsfunktionen sind für das Senden und den Empfang von Daten über einen Kommunikationskanal im Zeichenmodus definiert:

Plattform	Senden einer Zeichenfolge	Lesen einer Zeichenfolge	Lesen eines Byte-Arrays
M580	PRINT_CHAR	INPUT_CHAR	INPUT_BYTE
M340	PRINT_CHAR	INPUT_CHAR	INPUT_BYTE
Quantum	PRINT_CHAR_QX	INPUT_CHAR_QX	–

Nähere Informationen zu diesen Kommunikationsfunktionen finden Sie unter *EcoStruxure™ Control Expert – Kommunikation, Bausteinbibliothek*.

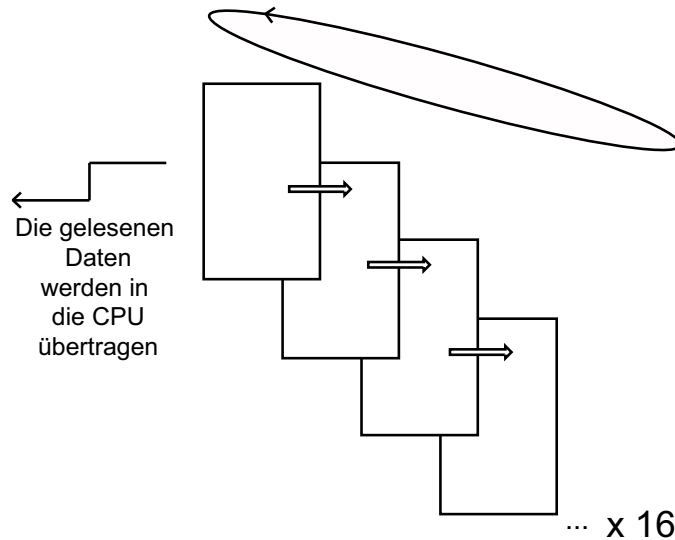
HINWEIS: Wenn der Kanal ohne **Stopp bei Stille** konfiguriert ist, ist für die Funktion INPUT_CHAR ein konfigurierter Timeout erforderlich, um das Aktivitätsbit der Funktion zu quittieren.

Interner Mechanismus des BMXNOM0200-Moduls

Das Modul BMXNOM0200 kann beim Senden und Empfangen insgesamt 16 Frames speichern. Die Frames in den Puffern werden in der FIFO-Reihenfolge verwaltet. Sie werden über RS-232-Leitungen im Vollduplex-Modus verwaltet.

Die empfangenen Daten werden der Reihe nach in 16 umlaufenden Puffern mit jeweils 1.024 Bytes gespeichert.

Die nachstehende Abbildung illustriert diesen Mechanismus:



Anzahl der empfangenen Frames in den Pufferspeichern

Wenn der serielle Port im Zeichenmodus konfiguriert ist, verweist %MW.r.m.c.7 auf die Anzahl der im Empfangspuffer des BMXNOM0200-Moduls enthaltenen Frames.

Dieses Wort (WORD) wird bei jedem Empfang eines weiteren Frames durch das BMXNOM0200-Modul über eine RS-232-Leitung inkrementiert.

Empfang von Daten

Die Frames werden von dem Anwendungsprogramm mit den Funktionen zum Empfangen von Zeichen `INPUT_CHAR` und `INPUT_CHAR_QX` EF abgerufen, um einen String zu empfangen, oder mit der Funktion `INPUT_BYTE` EF, um binäre Daten zu empfangen.

Die EF zum Empfangen von Daten kann ausgeführt werden, bevor die Daten vom Modul empfangen werden. In diesem Fall wartet das Modul auf Daten aus der Leitung und sendet sie dann an die CPU.

Die EF kann auch ausgeführt werden, wenn der Frame bereits empfangen wurde (z. B. nach der Prüfung von %MWr.m.c.7 mit `READ_STS`). In diesem Fall sendet das Modul den gepufferten Frame direkt an die CPU.

Es besteht ebenfalls die Möglichkeit, das Modul zum Warten auf Daten aus der Leitung zu forcieren, indem der Reset-Parameter der EF auf 1 gesetzt wird. In diesem Fall werden die zuvor gepufferten Daten beseitigt und das BMXNOM0200-Modul wartet auf das Senden neuer Daten an die CPU.

Das Modulverhalten ist abhängig von:

- der Kanalkonfiguration (**mit** oder **ohne** Stoppparameter
- den Eingangsparametern der Kommunikationsfunktionen
- den Status des Puffers vor Aktivierung der Kommunikationsfunktion

HINWEIS: Die max. Datengröße eines vom BMXNOM0200-Modul an die CPU gesendeten Frames beträgt 1.024 Bytes. Intern gilt für den Empfang des Frames jedoch eine max. Größe von 1.025 Byte, wenn ein End-of-Frame-Byte konfiguriert wurde und dieses Byte nicht in die Daten integriert werden soll, die an die CPU gesendet werden.

Die folgende Tabelle enthält das Modulverhalten unter folgenden Bedingungen:

- Kanal **ohne** Stoppparameter konfiguriert
- Eingangsparameter der EF (`NB` oder `INPUT_LEN`) auf 0 gesetzt.

Problem	Vorgehen
Der Puffer ist vor der Aktivierung der EF nicht leer	Das Modul sendet den Inhalt des Puffers mit max. 1.024 Zeichen an die CPU.
Der Puffer ist vor der Aktivierung der EF leer	Das Modul wartet mit dem Senden der Zeichen an die CPU bis zum Empfang der ersten Zeichen.
Der Reset-Puffer wird vor der Aktivierung der EF ausgewählt	Das Modul leert zunächst den Puffer und wartet bis zum Empfang der nächsten Zeichen.

Die folgende Tabelle enthält das Modulverhalten unter folgenden Bedingungen:

- Kanal **ohne** Stoppparameter konfiguriert und
- Eingangsparameter der EF (NB oder INPUT_LEN) auf einen Wert größer als 0 gesetzt.

Problem	Vorgehen
Der Puffer ist vor der Aktivierung der EF nicht leer	Das Modul wartet mit dem Senden an die CPU bis der Puffer NB oder INPUT_LEN Bytes enthält.
Der Puffer ist vor der Aktivierung der EF leer	Das Modul wartet mit dem Senden der Zeichen an die CPU bis zum Empfang der NB- oder INPUT_LEN-Zeichen.
Der Reset-Puffer wird vor der Aktivierung der EF ausgewählt	Das Modul leert zunächst den Puffer und wartet bis zum Empfang der nächsten NB- oder INPUT_LEN-Zeichen.

Die folgende Tabelle enthält das Modulverhalten unter folgenden Bedingungen:

- Kanal konfiguriert **mit** Stoppparametern (Stopp bei Empfang eines Sonderzeichens oder Stopp bei Stille) und
- Eingangsparameter der EF (NB oder INPUT_LEN) auf 0 gesetzt.

Problem	Vorgehen
Der Puffer enthält vor der Aktivierung der EF eine Nachricht	Das Modul sendet die Nachricht mit max. 1.024 Zeichen an die CPU.
Der Puffer ist vor der Aktivierung der EF leer	Das Modul wartet mit dem Senden von max. 1.024 Zeichen an die CPU bis zum Empfang der ersten Nachricht.
Der Reset-Puffer wird vor der Aktivierung der EF ausgewählt	Das Modul leert den Puffer und wartet bis zum Empfang der nächsten Nachrichten.

Null-Größen-Frames

Null-Größen-Frames bleiben unberücksichtigt. Wenn ein End-of-Frame-Byte konfiguriert ist und nicht als Teil der Daten angefordert wurde, wird der vom BMXNOM0200-Modul empfangene Null-Größen-Frame nicht an die CPU gesendet. In diesem Fall bleibt der Frame bei Empfang eines End-of-Frame-Bytes ohne vorangestellte Daten unberücksichtigt und es werden keine Informationen an die CPU gesendet.

Empfang mehrerer Frames während einer MAST-Task

Während einer MAST-Task können mehrere Frames vom BMXNOM0200-Modul an die CPU weitergeleitet werden. Parallel dazu können mehrere Instanzen der EF INPUT_CHAR an das gleiche BMXNOM0200-Modul adressiert werden. Das kann sich als notwendig erweisen, wenn eine große Menge Daten über eine serielle Leitung übertragen wird.

Die Funktionen „Cancel“ und „Timeout“

Die Funktionen „Cancel“ und „Timeout“ werden an das BMXNOM0200-Modul weitergeleitet. Die auf eine Instanz von `INPUT_CHAR` angewendeten Timeout-Bedingungen und Cancel-Befehle werden an das BMXNOM0200-Modul weitergeleitet. Die entsprechende zurückgestellte Task wird aus der Task-Warteschlange des BMXNOM0200-Moduls entfernt.

Interner Mechanismus des BMXNOM0200-Moduls: Senden

Verwenden Sie die EF `PRINT_CHAR` oder `PRINT_CHAR_QX` zum Senden von Daten über die serielle Leitung des BMXNOM0200-Moduls.

HINWEIS: Wenn mehrere Frames gesendet (mehrere EF-Instanzen aufgerufen) wurden und eine Stille konfiguriert wurde, fügt das BMXNOM0200-Modul zwischen jedem Frame eine Stille ein.

Es können bis zu 16 EF-Requests ausgegeben werden: Diese werden nacheinander mit jeweils einer Stille zwischen jedem Request übertragen.

Details zum Experten-Modus für den Zeichenmodus

Kommunikation im Experten-Modus

Der Experten-Modus besteht aus einer Reihe von Befehlen, die an das Modul gesendet werden können, um zusätzliche Funktionen zu erhalten.

Adresse	Standardsymbol	Austauschtyp	Typ	Bedeutung
%MWr.m.c.24	CONTROL	Explizit	INT	Befehlssignal, Protokoll ändern
%MWr.m.c.24.0		Explizit	BOOL	Lokale Zähler löschen
%MWr.m.c.24.4		Explizit	BOOL	Interne Zeitgebungen für Stille ändern (%MW30). Diese Wertaktualisierung kann das Modul u. U. stören, falls es in Betrieb ist.
%MWr.m.c.24.5		Explizit	BOOL	End-of-Frame-Byte 0 (%MW26) und Byte 1 (%MW27) für den Zeichenmodus ändern
%MWr.m.c.24.6		Explizit	BOOL	HALF/FULL DUPLEX-Modem-Verwaltungsmodus ändern • Wenn dieses Bit gleichzeitig mit RTS_ON gesetzt wird (%MWr.m.c.24.10 funktioniert auch mit RTS_OFF %MWr.m.c.24.11, und DTR verwenden, wenn .8 oder .9 verwendet werden), wird der Halbduplex-Modus aktiviert. • Wenn dieses Bit gesetzt ist, aber keine der RTS/DTR-Bits (weder %MWr.m.c.24.8, %MWr.m.c.24.9, %MWr.m.c.24.10 noch %MWr.m.c.24.11), wird der Vollduplex-Modus aktiviert. Mit %MW26 wird die Startverzögerung (StartDelay) und mit %MW27 die Endverzögerung (EndDelay) eingestellt. Also können die Bits %MW24.5 und %MW24.1 und %MW24.2 nicht gleichzeitig verwendet werden. HINWEIS: Unter Umständen muss der Benutzer den ordnungsgemäßen Zustand der RTS/DTR-Signale wiederherstellen, nachdem der Befehl akzeptiert wurde.
%MWr.m.c.24.7		Explizit	BOOL	Modbus-Slave-Adresse im FLASH speichern (%MW25).
%MWr.m.c.24.8	DTR_ON	Explizit	BOOL	DTR-Signal einstellen (positive Spannung)

Adresse	Standardsymbol	Austauschtyp	Typ	Bedeutung
%MW _r .m.c.24.9	DTR_OFF	Explizit	BOOL	DTR-Signal zurücksetzen (negative Spannung)
%MW _r .m.c.24.10		Explizit	BOOL	RTS-Signal einstellen (positive Spannung)
%MW _r .m.c.24.11		Explizit	BOOL	RTS-Signal zurücksetzen (negative Spannung)
%MW _r .m.c.24.12	TO_MODBUS_MASTER	Explizit	BOOL	In den Master-Modus wechseln
%MW _r .m.c.24.13	TO_MODBUS_SLAVE	Explizit	BOOL	In den Slave-Modus wechseln
%MW _r .m.c.24.14	TO_CHAR_MODE	Explizit	BOOL	In den Zeichenmodus wechseln
%MW _r .m.c.25		Explizit	INT	Im FLASH zu speichernde Modbus-Slave-Adresse
%MW _r .m.c.26		Explizit	INT	Neues EOF im Zeichenmodus (entspricht %KW6) wenn %MW24.5 gesetzt ist: ● Bit 0: 1 byte 1 is set below, 0 no more byte 1 ● Bit 1: 1 - Byte 1 hinzufügen, 0 - Byte 1 nicht hinzufügen ● Bit2..7: Muss Null sein. HOCHWERTIGES BYTE: das End-of-Frame-Byte 1 StartDelay wenn %MW26.6 gesetzt ist. Die Zeit, die gewartet wird, nachdem das CTS OK ist, bevor mit dem Senden des Frames begonnen wird. Dies ist nützlich für Modems, die nach CTS zusätzliche Zeit erfordern oder die das CTS-Signal nicht verwalten (in diesem Fall muss das RTS mit dem CTS verbunden sein). Diese Zeit wird in Millisekunden angegeben, die Genauigkeit beträgt etwa 3 ms. Kann nur im RS232-Modus durchgeführt werden.

Adresse	Standardsymbol	Austauschtyp	Typ	Bedeutung
%MWr.m.c.27		Explizit	INT	Neues EOF im Zeichenmodus (entspricht %KW7) wenn %MW24.5 gesetzt ist: • Bit 0: 1 byte 2 is set below, 0 no more byte 2 • Bit 1: 1 - Byte 2 hinzufügen, 0 - Byte 2 nicht hinzufügen • Bit2..7: Muss Null sein. HOCHWERTIGES BYTE: das End-of-Frame-Byte 2 StartDelay wenn %MW24.6 gesetzt ist. Die Zeit, die nach dem Senden eines Frames gewartet wird, bevor das RTS-Signal ausgelöst wird, um so dem Modem vor der Weitergabe genügend Zeit zu geben, das Senden des Frames abzuschließen. Diese Zeit wird in Millisekunden angegeben, die Genauigkeit beträgt etwa 3 ms. Kann nur im RS232-Modus durchgeführt werden.
%MWr.m.c.28		Explizit	INT	Reserviert
%MWr.m.c.29		Explizit	INT	Reserviert
%MWr.m.c.30		Explizit	INT	Stille: Verzögerung zwischen Frames in Millisekunden [0..10000]. Der verwendete Wert hängt von der Geschwindigkeit ab. Falls der Wert kleiner oder größer ist als die möglichen Werte, wird der untere bzw. obere Grenzwert angewendet und der Befehl wird akzeptiert. Ein Wert von 0 bedeutet keine Stille.
%MWr.m.c.31		Explizit	INT	Reserviert
%MWr.m.c.32		Explizit	INT	Reserviert

Beispielcode

```

if HalfModemSlave then
  HalfModemMaster:=false;

  %MW0.9.0.24:=16#0440; (* zu Halbduplex-Modus mit RTS wechseln*)
  %MW0.9.0.26:=12; (* 12 ms vor dem Senden warten, wenn CTS steigt *)
  %MW0.9.0.27:=9; (* RTS 9 ms nach Sendeende auslösen *)

  write_cmd(%ch0.9.0); (* Befehl und Daten an den NOM-Kanal senden*)
end_if;

```

Abschnitt 5.4

Debuggen der Zeichenmoduskommunikation

Debug-Fenster der Zeichenmoduskommunikation

Allgemein

Auf das Debug-Fenster für den Zeichenmodus kann im Online-Modus zugegriffen werden.

Zugriff auf das Debug-Fenster

Die folgende Tabelle beschreibt die Vorgehensweise zum Zugreifen auf das Debug-Fenster der Zeichenmoduskommunikation:

Schritt	Aktion
1	Greifen Sie auf das Konfigurationsfenster der Zeichenmoduskommunikation zu. <i>(siehe Seite 94)</i>
2	Wählen Sie auf dem angezeigten Fenster die Registerkarte Debuggen aus.

Beschreibung des Debug-Fensters

Das Debug-Fenster umfasst die Bereiche **Fehler** und **Signale** (bei RS232).

Der Bereich Fehler

Der Bereich **Fehler** sieht wie folgt aus:

Dieser Bereich gibt die Anzahl der Kommunikationsunterbrechungen an, die vom Modul gezählt werden:

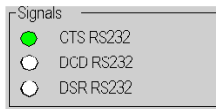
- **Beim Senden:** Entspricht der Anzahl der Unterbrechungen beim Senden (Abbild des %MW4-Worts).
- **Beim Empfang:** Entspricht der Anzahl der Unterbrechungen beim Empfang (Abbild des %MW5-Worts).

Die Schaltfläche **Zähler zurücksetzen** setzt beide Zähler auf Null zurück.

Bereich Signale

Dieser Bereich wird nur angezeigt, wenn RS232 im Konfigurationsfenster ausgewählt wurde. Bei der Auswahl von RS485 im Konfigurationsfenster wird dieses Fenster nicht angezeigt.

Der Bereich **Signale** sieht wie folgt aus:



Dieser Bereich gibt die Aktivität der Signale an:

- **CTS RS232:** Zeigt die Aktivität des CTS-Signals an.
- **DCD RS232:** Zeigt die Aktivität des DCD-Signals an.
- **DSR RS232:** Zeigt die Aktivität des DSR-Signals an.

Kapitel 6

BMXNOM0200 Moduldiagnose

Inhalt dieses Kapitels

Dieses Kapitel beschreibt die Diagnoseaspekte der Implementierung von BMXNOM0200-Kommunikationsmodulen.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Detaillierte Diagnose nach Kommunikationskanal	116
Diagnose eines BMXNOM0200-Moduls	118

Detaillierte Diagnose nach Kommunikationskanal

Auf einen Blick

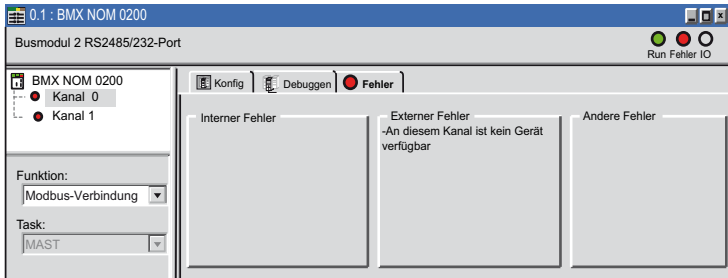
Die Kanaldiagnosefunktion zeigt aktuelle Fehler, sofern vorhanden, gemäß ihrer jeweiligen Kategorie an:

- **Interner Modulfehler**
 - Selbsttest wird ausgeführt
- **Externe Ereignisse**
 - Gerät nicht vorhanden
 - Gerät nicht betriebsbereit
 - Timeout der seriellen Kommunikation
- **Andere erkannte Fehler**
 - Leitungstool-Fehler
 - Konfigurationsfehler
 - Kommunikationsverlust
 - Anwendungsfehler

Ein Kanalfehler wird auf der Registerkarte **Debuggen** durch eine rote Anzeige  in der Spalte **Fehler** angezeigt.

Zugriff auf das Kanal-Diagnosefenster

In der folgenden Tabelle ist die Vorgehensweise zum Aufrufen des Kanal-Diagnosefensters beschrieben.

Schritt	Aktion
1	Rufen Sie das Debug-Fenster des Moduls auf.
2	Klicken Sie für den Kanal außer Funktion auf die Schaltfläche  in der Spalte Fehler. Ergebnis: Die Liste der Kanalfehler wird angezeigt. 
	Hinweis: Auf Kanaldiagnoseinformationen kann auch über ein Programm zugegriffen werden (READ_STS-Anweisung).

Kanalfehlerliste

Die folgende Übersichtstabelle zeigt die Fehler, die für eine konfigurierte serielle Verbindung erkannt werden:

Fehlerklassifizierung	Sprachobjekte
Interner Fehler: ● Selbsttest wird ausgeführt	● %MWr.m.c.2.4
Externer Fehler: ● An diesem Kanal ist kein Gerät verfügbar ● Gerätefehler ● Timeout-Fehler (CTS)	● %MWr.m.c.2.0 ● %MWr.m.c.2.1 ● %MWr.m.c.2.3
Andere Fehler: ● Leitungstool-Fehler ● Hardware-Konfigurationsfehler ● Problem bei der Kommunikation mit der Steuerung ● Anwendungsfehler	● %MWr.m.c.2.2 ● %MWr.m.c.2.5 ● %MWr.m.c.2.6 ● %MWr.m.c.2.7

Diagnose eines BMXNOM0200-Moduls

Einführung

Die Moduldiagnosefunktion zeigt aktuelle Fehler – sofern vorhanden – gemäß ihrer jeweiligen Kategorie an:

- **Interner Modulfehler:**
 - Modulereignis
- **Externe Ereignisse:**
 - Verdrahtungsüberwachung (Drahtbruch, Überlast oder Kurzschluss)
- **Andere Fehler:**
 - Nicht betriebsbereiter Kanal
 - Konfigurationsfehler
 - Modul nicht vorhanden oder ausgeschaltet

Ein fehlerhaftes Modul wird dadurch angezeigt, dass bestimmte Anzeige-LEDs auf Rot wechseln, z. B.:

- Im Konfigurationseditor auf Rack-Ebene:
 - Anzeige-LED für die Racknummer
 - Anzeige-LED für die Steckplatznummer des Moduls im Rack
- im Konfigurationseditor der Modul-Ebene:
 - Die Anzeige-LEDs **Err** und **I/O**, je nach Fehlertyp
 - Die Anzeige-LED **Kanal** im Feld **Kanal**

Zugriff auf das Modul-Diagnosefenster

In der folgenden Tabelle ist die Vorgehensweise zum Aufrufen des Modul-Diagnosefensters beschrieben.

Schritt	Aktion
1	Rufen Sie das Debug-Fenster des Moduls auf.
2	Klicken Sie auf die Referenz des Moduls im Kanalbereich, und wählen Sie die Registerkarte Fehler. Ergebnis: Es erscheint eine Liste mit den Modulfehlern.
	Hinweis: Ein Zugriff auf den Moduldiagnose-Bildschirm ist nicht möglich, wenn ein Konfigurationsfehler, ein schwerwiegender Ausfallfehler oder ein Fehler wegen eines fehlenden Moduls auftritt. Auf dem Bildschirm erscheint die folgende Nachricht: „Modul nicht vorhanden oder unterscheidet sich von dem in dieser Position konfigurierten Modul.“

Modulfehlerliste

Die folgende Übersichtstabelle zeigt die Fehler, die für ein Kommunikationsmodul erkannt werden:

Fehlerklassifizierung	Sprachobjekte
Interner Fehler: ● Modulausfall	● %MWr.m.MOD.2.0
Externer Fehler: ● Klemmenleiste	● %MWr.m.MOD.2.2
Andere Fehler: ● Fehlerhafte Kanäle ● Hardware-Konfigurationsfehler ● Modul nicht vorhanden oder ausgeschaltet	● %MWr.m.MOD.2.1 ● %MWr.m.MOD.2.5 ● %MWr.m.MOD.2.6

Kapitel 7

Sprachobjekte der Modbus- und Zeichenmoduskommunikation

Inhalt des Kapitels

In diesem Kapitel werden die Sprachobjekte der Modbus- und der Zeichenmoduskommunikation und deren verschiedene Verwendungsmöglichkeiten beschrieben.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
7.1	Sprachobjekte und IODDTs der Modbus- und Zeichenmoduskommunikation	122
7.2	Allgemeine Sprachobjekte und IODDTs für Kommunikationsprotokolle	130
7.3	Sprachobjekte und IODDTs der Modbus-Kommunikation	134
7.4	Sprachobjekte und IODDTs der Zeichenmoduskommunikation	142
7.5	IODDT Type T_GEN_MOD, anwendbar auf alle Module	150
7.6	Sprachobjekte und gerätespezifische IODDTs der Modbus-Kommunikation	152

Abschnitt 7.1

Sprachobjekte und IODDTs der Modbus- und Zeichenmoduskommunikation

Inhalt dieses Abschnitts

Dieser Abschnitt bietet einen Überblick über die allgemeinen Gesichtspunkte von IODDTs und Sprachobjekten für die Modbus- und Zeichenmoduskommunikation.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Einführung in die Sprachobjekte für die Modbus- und Zeichenmodus-Kommunikation	123
Implizite Austauschsprachobjekte der anwendungsspezifischen Funktion	124
Explizite Austauschsprachobjekte der anwendungsspezifischen Funktion	125
Verwaltung von Austauschvorgängen und Berichten mit expliziten Objekten	127

Einführung in die Sprachobjekte für die Modbus- und Zeichenmodus-Kommunikation

Allgemeines

IODDTs werden durch den Hersteller vordefiniert. Sie enthalten Eingangs-/Ausgangs-Sprachobjekte, die zum Kanal eines anwendungsspezifischen Moduls gehören.

Für die Modbus- und Zeichenmodus-Kommunikation sind drei zugehörige IODDTs vorhanden:

- **T_COM_STS_GEN:** Anwendbar auf alle Kommunikationsprotokolle ausgenommen Fipio und Ethernet.
- **T_COM_MB_BMX:** Spezifisch für die Modbus-Kommunikation.
- **T_COM_CHAR_BMX:** Spezifische für die Zeichenmodus-Kommunikation.

HINWEIS: IODDT-Variablen können auf zwei Arten erstellt werden:

- Verwendung der Registerkarte der E/A-Objekte (*siehe EcoStruxure™ Control Expert, Betriebsarten*)
- Verwendung des Dateneditors (*siehe EcoStruxure™ Control Expert, Betriebsarten*)

Sprachobjekttypen

Jeder IODDT beinhaltet eine Reihe von Sprachobjekten, die der Steuerung und Überprüfung der ordnungsgemäßen Funktionsweise des IODDT dienen.

Es gibt zwei Arten von Sprachobjekten:

- **Implizite Austauschobjekte:** Diese Objekte werden automatisch bei jedem Zyklusdurchlauf der dem Modul zugeordneten Task ausgetauscht.
- **Explizite Austauschobjekte:** Diese Objekte werden unter Verwendung der Anweisungen zum expliziten Austausch auf Anforderung der Anwendung ausgetauscht.

Der implizite Austausch betrifft den Status der Prozessoren, Kommunikationssignale, Slaves usw.

Der explizite Austausch ermöglicht die Definition der Prozesseureinstellungen und die Durchführung von Diagnosen.

Implizite Austauschsprachobjekte der anwendungsspezifischen Funktion

Einführung

Die Verwendung einer integrierten anwendungsspezifischen Schnittstelle oder das Hinzufügen eines Moduls erweitert automatisch das Projekt von Sprachobjekten, welche das Programmieren dieser Schnittstelle oder dieses Moduls ermöglichen.

Diese Objekte entsprechen den Abbildern der Ein-/Ausgänge und Softwareinformationen des Moduls oder der integrierten anwendungsspezifischen Schnittstelle.

Zur Erinnerung

Die Moduleingänge (%I und %IW) werden zu Beginn des Tasks im SPS-Speicher aktualisiert bzw. wenn sich die SPS im Modus RUN oder STOP befindet.

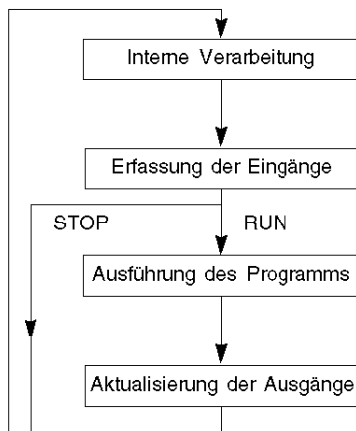
Die Ausgänge (%Q und %QW) werden am Ende des Tasks aktualisiert, jedoch nur, wenn sich die Steuerung im Modus "RUN" befindet.

HINWEIS: Befindet sich der Task im Modus STOP, so erfolgt abhängig von der gewählten Konfiguration Folgendes:

- Die Ausgänge werden auf die Position Fehlerwert gesetzt (Fehlermodus).
- Die Ausgänge werden auf ihrem letzten Wert gehalten (Modus "Wert halten").

Beschreibung

Das folgende Diagramm zeigt den Betriebszyklus der SPS-Aufgabe (zyklische Ausführung):



Explizite Austauschsprachobjekte der anwendungsspezifischen Funktion

Auf einen Blick

Ein expliziter Austausch ist ein Austausch, der auf Anforderung des Anwenderprogramms mithilfe der folgenden Anweisungen durchgeführt wird:

- `READ_STS` (siehe *EcoStruxure™ Control Expert, E/A-Verwaltung, Bausteinbibliothek*): Lesen von Statuswörtern
- `WRITE_CMD` (siehe *EcoStruxure™ Control Expert, E/A-Verwaltung, Bausteinbibliothek*): Schreiben von Befehlswörtern

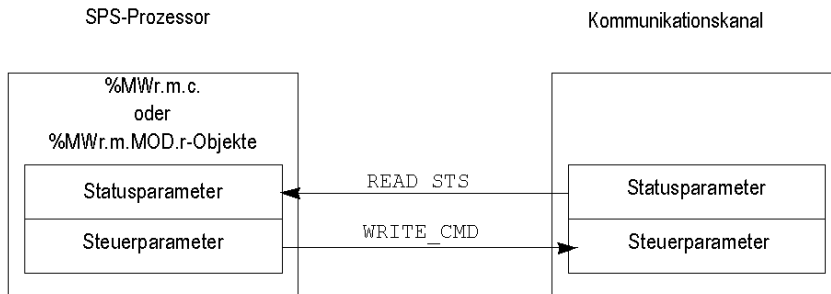
Diese Austauschvorgänge gelten für einen Satz von `%MW` Objekten desselben Typs (Status, Befehle oder Parameter), die zu einem Kanal gehören.

HINWEIS: Diese Objekte bieten Informationen über den Prozessor oder das Modul und können für Befehle an diese verwendet werden (z. B.: Steuerung der Flipflops) sowie, um deren Betriebsarten festzulegen (Speichern und Wiederherstellen der ausgeführten Regelungsparameter).

HINWEIS: Die Anweisungen `READ_STS` und `WRITE_CMD` werden gleichzeitig und immer fehlerfrei als der Task ausgeführt, von dem sie aufgerufen werden. Das Ergebnis dieser Anweisungen steht unmittelbar nach dem Ausführen zur Verfügung.

Allgemeines Nutzungsprinzip der expliziten Anweisungen

Die folgende Abbildung zeigt die unterschiedlichen Arten expliziter Austauschvorgänge, die zwischen dem Prozessor und dem Kommunikationskanal ausgeführt werden können:



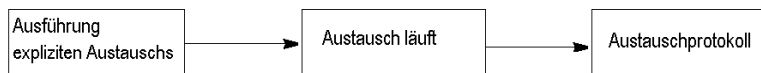
Verwaltung des Austauschs

Während eines expliziten Austauschs muss der Ablauf dieses Austauschs überwacht werden, damit die Daten nur dann berücksichtigt werden, wenn der Austausch ordnungsgemäß durchgeführt wurde.

Hierzu sind zwei Informationstypen verfügbar:

- Informationen über den laufenden Austausch (*siehe EcoStruxure™ Control Expert, E/A-Verwaltung, Bausteinbibliothek*)
- Protokoll des Austauschs (*siehe EcoStruxure™ Control Expert, E/A-Verwaltung, Bausteinbibliothek*)

Die folgende Abbildung veranschaulicht das Verwaltungsprinzip eines Austauschs:



HINWEIS: Um mehrere simultane explizite Austauschvorgänge für ein und denselben Kanal zu vermeiden, muss der Wert des Worts EXCH_STS (`%MWx.m.c.0`) des zum Kanal gehörenden IODDT getestet werden, bevor eine Elementarfunktion, die diesen Kanal nutzt, aufgerufen wird.

Verwaltung von Austauschvorgängen und Berichten mit expliziten Objekten

Auf einen Blick

Werden Daten zwischen SPS-Speicher und Modul ausgetauscht, kann die Bestätigung dieser Informationen durch das Modul mehrere Taskzyklen erfordern.

Um den Austausch zu verwalten, besitzen alle IODDT zwei Wörter:

- EXCH_STS (%MW_{r.m.c.0}): Austausch läuft.
- EXCH_RPT (%MW_{r.m.c.1}): Rückmeldung.

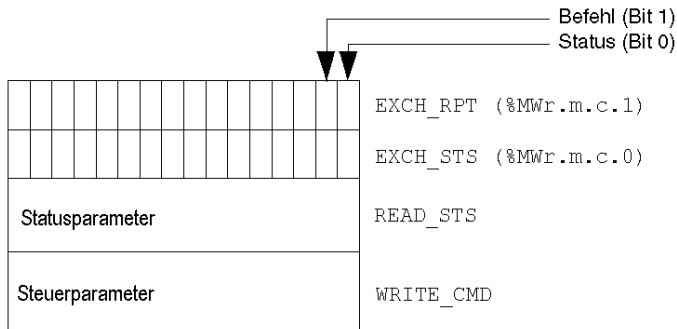
HINWEIS:

Je nach Lokalisierung des Moduls wird die Verwaltung der expliziten Austauschvorgänge (z.B. %MW0.0.MOD.0.0 von der Anwendung nicht erkannt:

- Bei Modulen im Rack erfolgt der Austausch sofort auf dem lokalen SPS-Bus und wird vor der Fertigstellung der Ausführungstask abgeschlossen, so dass beispielsweise READ_STS immer fertig gestellt ist, wenn das Bit %MW0.0.mod.0.0 von der Applikation überprüft wird.
- Bei dezentralen Busmodulen (z. B. Fipio) sind explizite Austauschvorgänge nicht mit der Task-Ausführung synchronisiert, somit ist eine Erkennung für die Anwendung möglich.

Abbildung

Die folgende Abbildung zeigt die unterschiedlichen signifikanten Bits für die Verwaltung der Austauschvorgänge:



Beschreibung der signifikanten Bits

Jedes Bit der Wörter `EXCH_STS` (`%MWr.m.c.0`) und `EXCH_RPT` (`%MWr.m.c.1`) ist einem Parametertyp zugewiesen:

- Die Bits mit dem Stellenwert 0 sind den Statusparametern zugeordnet:
 - Das Bit `STS_IN_PROGR` (`%MWr.m.c.0.0`) zeigt an, ob eine aktuelle Aufforderung zum Lesen der Statuswörter vorhanden ist.
 - Das Bit `STS_ERR` (`%MWr.m.c.1.0`) zeigt an, ob eine Aufforderung zum Lesen der Statuswörter vom Kanal des Moduls akzeptiert wird.
- Die Bits mit dem Stellenwert 1 sind den Befehlsparametern zugeordnet:
 - Das Bit `CMD_IN_PROGR` (`%MWr.m.c.0.1`) gibt an, ob die Steuerparameter an den Modulkanal gesendet werden oder nicht.
 - Das Bit `CMD_ERR` (`%MWr.m.c.1.1`) zeigt an, ob die Steuerparameter vom Kanal des Moduls akzeptiert werden.

HINWEIS: „r“ entspricht der Racknummer und „m“ der Position des Moduls im Rack, während „c“ der Kanalnummer im Modul entspricht.

HINWEIS: Austausch- und Berichtswörter existieren auch auf Modulebene `EXCH_STS` (`%MWr.m.MOD.0`) und `EXCH_RPT` (`%MWr.m.MOD.1`) des IODDT-Typs `T_GEN_MOD`.

Ausführungsanzeiger eines expliziten Austauschs: `EXCH_STS`

Die folgende Tabelle zeigt die Steuerbits der expliziten Austauschvorgänge des Worts `EXCH_STS` (`%MWr.m.c.0`):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
<code>STS_IN_PROGR</code>	BOOL	R	Lesen der Statuswörter des aktuellen Kanals	<code>%MWr.m.c.0.0</code>
<code>CMD_IN_PROGR</code>	BOOL	R	Befehlsparameter werden ausgetauscht	<code>%MWr.m.c.0.1</code>
<code>ADJ_IN_PROGR</code>	BOOL	R	Einstellparameter werden ausgetauscht	<code>%MWr.m.c.0.2</code>
<code>RECONF_IN_PROGR</code>	BOOL	R	Aktuelle Neueinstellung des Moduls	<code>%MWr.m.c.0.15</code>

HINWEIS: Wenn das Modul nicht vorhanden oder getrennt ist, werden die expliziten Austauschobjekte (z. B. `READ_STS`) nicht an das Modul gesendet (`STS_IN_PROG` (`%MWr.m.c.0.0`) = 0), aber die Worte werden aktualisiert.

Protokoll des expliziten Austauschs: EXCH_RPT

Die folgende Tabelle zeigt die Berichtsbits des Worts EXCH_RPT (%MWr.m.c.1):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
STS_ERR	BOOL	R	Fehler beim Lesen der Statuswörter des Kanals erkannt (1 = Fehler)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Fehler beim Austausch von Befehlsparametern erkannt (1 = Fehler)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Unterbrechungen beim Austausch von Einstellparametern (1 = Fehler)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Unterbrechungen bei der Neukonfiguration des Kanals (1 = Fehler)	%MWr.m.c.1.15

Abschnitt 7.2

Allgemeine Sprachobjekte und IODDTs für Kommunikationsprotokolle

Inhalt dieses Abschnitts

Dieser Abschnitt beschreibt die generischen Sprachobjekte und IODDTs, die sich auf alle Kommunikationsprotokolle beziehen.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Beschreibung der impliziten IODDT-Austauschobjekte vom Typ T_COM_STS_GEN	131
Beschreibung der expliziten IODDT-Austauschobjekte vom Typ T_COM_STS_GEN	132

Beschreibung der impliziten IODDT-Austauschobjekte vom Typ T_COM_STS_GEN

Einführung

In der folgenden Tabelle sind die Sprachobjekte für den impliziten Austausch des IODDT des Typs T_COM_STS_GEN aufgeführt, der für alle Kommunikationsprotokolle außer Fipio gültig ist.

Fehlerbit

Die folgende Tabelle enthält die Bedeutung des Fehlerbits CH_ERROR (%Ir.m.c.ERR):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
CH_ERROR	EBOOL	R	Fehlerbit des Kommunikationskanals	%Ir.m.c.ERR

Beschreibung der expliziten IODDT-Austauschobjekte vom Typ T_COM_STS_GEN

Auf einen Blick

Dieser Abschnitt beschreibt die Objekte mit explizitem Austausch des IODDT-Typs `T_COM_STS_GEN`, der auf alle Kommunikationsprotokolle außer Fipio und Ethernet anwendbar ist. Der Teil umfasst die Objekte des Typs Wort, deren Bits eine besondere Bedeutung haben. Diese Objekte werden im Folgenden ausführlich erläutert.

In diesem Abschnitt weist die Variable `IODDT_VAR1` den Typ `T_COM_STS_GEN` auf.

Bemerkungen

Die Bedeutung eines Bit wird im Allgemeinen für den Zustand 1 des Bit gegeben. In spezifischen Fällen wird eine Erläuterung zu jedem Zustand des Bit gegeben.

Es werden nicht alle Bits verwendet.

Ausführungsanzeiger eines expliziten Austauschs: EXCH_STS

In der folgenden Tabelle sind die Bedeutungen der Kanal-Austauschsteuerungsbits des Kanals `EXCH_STS (%MWr.m.c.0)` aufgeführt:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
STS_IN_PROGR	BOOL	R	Lesen der Statuswörter des aktuellen Kanals	%MW _r .m.c.0.0
CMD_IN_PROGR	BOOL	R	Austausch der Befehlsparameter läuft	%MW _r .m.c.0.1

Rückmeldung von expliziten Austauschvorgängen: EXCH_RPT

In der folgenden Tabelle ist die Bedeutung der Austauschberichtsbits `EXCH_RPT (%MWr.m.c.1)` aufgeführt:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
STS_ERR	BOOL	R	Fehler beim Lesen der Statuswörter des Kanals erkannt	%MW _r .m.c.1.0
CMD_ERR	BOOL	R	Fehler beim Austausch von Befehlsparametern erkannt.	%MW _r .m.c.1.1

Kanalspezifische Standardfehler: CH_FLT

In der folgenden Tabelle wird die Bedeutung der Bits des Statusworts CH_FLT (%MW_{r.m.c.2}) aufgeführt:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
NO_DEVICE	BOOL	R	Kein Gerät auf dem Kanal funktioniert	%MW _{r.m.c.2.0}
ONE_DEVICE_FLT	BOOL	R	Ein Gerät auf dem Kanal ist nicht funktionsfähig	%MW _{r.m.c.2.1}
BLK	BOOL	R	Klemmenleiste ist nicht angeschlossen	%MW _{r.m.c.2.2}
TO_ERR	BOOL	R	Timeout übernommen (Analyse erforderlich)	%MW _{r.m.c.2.3}
INTERNAL_FLT	BOOL	R	Interner Modulfehler oder Selbsttest des Kanals	%MW _{r.m.c.2.4}
CONF_FLT	BOOL	R	Unterschiedliche Hard- und Softwarekonfiguration	%MW _{r.m.c.2.5}
COM_FLT	BOOL	R	Kommunikationsanalyse mit Kanal erforderlich.	%MW _{r.m.c.2.6}
APPLI_FLT	BOOL	R	Anwendungsfehler erkannt (Anpassungs- oder Konfigurationsfehler)	%MW _{r.m.c.2.7}

Das Lesen erfolgt durch die Anweisung READ_STS (IODDT_VAR1).

Abschnitt 7.3

Sprachobjekte und IODDTs der Modbus-Kommunikation

Inhalt dieses Abschnitts

In diesem Abschnitt werden die Sprachobjekte und IODDTs der Modbus-Kommunikation erläutert.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Beschreibung der expliziten Austauschsprachobjekte für eine Modbus-Funktion	135
Beschreibung der impliziten IODDT-Austauschobjekte vom Typ T_COM_MB_BMX und T_COM_MB_BMX_CONF_EXT	136
Beschreibung der expliziten IODDT-Austauschobjekte vom Typ T_COM_MB_BMX und T_COM_MB_BMX_CONF_EXT	137
Beschreibung der Sprachobjekte für die Konfiguration des Modbus-Modus	140

Beschreibung der expliziten Austauschsprachobjekte für eine Modbus-Funktion

Auf einen Blick

Die folgende Tabelle führt die Sprachobjekte für die Modbus-Kommunikation im Master- bzw. Slavemodus auf. Diese Objekte sind nicht in die IODDTs integriert.

Liste der expliziten Austauschobjekte im Master- bzw. Slavemodus

Die folgende Tabelle führt die expliziten Austauschobjekte auf:

Adresse	Typ	Zugriff	Bedeutung
%MW _r .m.c.4	INT	R	Anzahl der fehlerfrei empfangenen Antworten.
%MW _r .m.c.5	INT	R	Anzahl der Antworten, die mit CRC-Fehler empfangen wurden
%MW _r .m.c.6	INT	R	Anzahl der Antworten, die im Slavemodus mit einem Ausnahmecode empfangen wurden
%MW _r .m.c.7	INT	R	Anzahl der im Slavemodus gesendeten Nachrichten
%MW _r .m.c.8	INT	R	Anzahl der im Slavemodus gesendeten Nachrichten ohne Antwort
%MW _r .m.c.9	INT	R	Anzahl der Antworten, die mit einer negativen Bestätigung empfangen wurden.
%MW _r .m.c.10	INT	R	Anzahl der im Slavemodus wiederholten Nachrichten
%MW _r .m.c.11	INT	R	Anzahl der erkannten Zeichenfehler
%MW _r .m.c.24.0	BOOL	RW	Zurücksetzen des Zählers für erkannte Fehler

Beschreibung der impliziten IODDT-Austauschobjekte vom Typ T_COM_MB_BMX und T_COM_MB_BMX_CONF_EXT

Einführung

Die nachstehenden Tabellen zeigen die impliziten Austauschobjekte der IODDTs vom Typ T_COM_MB_BMX und T_COM_MB_BMX_CONF_EXT, die bei der seriellen Modbus-Kommunikation zum Einsatz kommen. Sie unterscheiden sich in Bezug auf die **Verfügbarkeit der Konfigurationsobjekte** (*siehe Seite 139*).

Bit CH_ERROR

In der folgenden Tabelle wird die Bedeutung des Fehlerbits CH_ERROR (%Ir.m.c.ERR) erläutert:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
CH_ERROR	EBOOL	R	Fehlerbit des Kommunikationskanals	%Ir.m.c.ERR

Wortobjekt im Modbus-Master-Modus

In der folgenden Tabelle wird die Bedeutung des Bits des Worts INPUT_SIGNALS (%IW.r.m.c.0) erläutert:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
DCD	BOOL	R	RS232-Signal DCD (Datenträgererkennung) (nur für das Modul BMX NOM 0200)	%IW.r.m.c.0.0
CTS	BOOL	R	Bereit zum Senden des RS232-Signals	%IW.r.m.c.0.2
DSR	BOOL	R	RS232-Signal DSR (Datensatz bereit) (nur für das Modul BMX NOM 0200)	%IW.r.m.c.0.3

HINWEIS: %IW.r.m.c.0.2 wird auf 1 gesetzt, wenn das CTS-Signal eine positive Spannung aufweist. Es ist ebenfalls auf DCD und DSR anwendbar.

Wortobjekt im Modbus-Slave-Modus

Die Sprachobjekte entsprechen denjenigen der Modbus-Master-Funktion. Nur die Objekte in der nachstehenden Tabelle sind unterschiedlich.

In der folgenden Tabelle wird die Bedeutung des Bits des Worts INPUT_SIGNALS (%IW.r.m.c.0) erläutert:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
LISTEN_ONLY	BOOL	R	Nur-Hören-Modus	%IW.r.m.c.0.8

Beschreibung der expliziten IODDT-Austauschobjekte vom Typ T_COM_MB_BMX und T_COM_MB_BMX_CONF_EXT

Einführung

Dieser Abschnitt beschreibt die expliziten Austauschobjekte der IODDT-Typen T_COM_MB_BMX und T_COM_MB_BMX_CONF_EXT, die für die serielle Modbus-Kommunikation verwendet werden können und sich hinsichtlich der **Verfügbarkeit der Konfigurationsobjekte** (*siehe Seite 139*) unterscheiden. Hierzu gehören Objekte des Typs Wort, deren Bits eine besondere Bedeutung haben. Diese Objekte werden im Folgenden ausführlich erläutert.

In diesem Abschnitt weist die Variable IODDT_VAR1 den Typ T_COM_STS_GEN auf.

Anmerkungen

Prinzipiell wird die Bedeutung der Bitstatus 1 angegeben. In bestimmten Fällen wird jeder Bitstatus erläutert.

Es werden nicht alle Bits verwendet.

Flags für die Ausführung des expliziten Austauschs: EXCH_STS

Die folgende Tabelle verdeutlicht die verschiedenen Bedeutungen der Austauschsteuerungsbits des Kanals EXCH_STS (%MW_{r.m.c.0}):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
STS_IN_PROGR	BOOL	R	Lesen der Statuswörter des aktuellen Kanals.	%MW _{r.m.c.0.0}
CMD_IN_PROGR	BOOL	R	Austausch der Befehlsparameter läuft	%MW _{r.m.c.0.1}
ADJ_IN_PROGR	BOOL	R	Austausch der Einstellparameter läuft (nicht für Module vom Typ BMX NOM 0200)	%MW _{r.m.c.0.2}

Rückmeldung zum expliziten Austausch: EXCH_RPT

Die folgende Tabelle verdeutlicht die verschiedenen Bedeutungen der Austauschberichtsbits EXCH_RPT (%MW_{r.m.c.1}):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
STS_ERR	BOOL	R	Fehler beim Lesen der Statuswörter des Kanals erkannt	%MW _{r.m.c.1.0}
CMD_ERR	BOOL	R	Unregelmäßigkeit während eines Austauschs von Befehlsparametern.	%MW _{r.m.c.1.1}
ADJ_ERR	BOOL	R	Unregelmäßigkeit während eines Austauschs von Einstellparametern (nicht für Module des Typs BMX NOM 0200).	%MW _{r.m.c.1.2}

Kanalspezifische erkannte Standardfehler: CH_FLT

Die folgende Tabelle verdeutlicht die verschiedenen Bedeutungen der Bits des Statusworts CH_FLT (%MWr.m.c.2):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
NO_DEVICE	BOOL	R	Kein Gerät auf dem Kanal funktioniert	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	R	Ein Gerät auf dem Kanal ist nicht funktionsfähig	%MWr.m.c.2.1
BLK	BOOL	R	Klemmenleiste ist nicht angeschlossen	%MWr.m.c.2.2
TO_ERR	BOOL	R	Timeout übernommen (Analyse erforderlich)	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Interner Modulfehler oder Selbsttest des Kanals	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Unterschiedliche Hard- und Softwarekonfiguration	%MWr.m.c.2.5
COM_FLT	BOOL	R	Kommunikationsanalyse mit Kanal erforderlich.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Anwendungsfehler erkannt (Anpassungs- oder Konfigurationsfehler)	%MWr.m.c.2.7

Das Lesen erfolgt durch die Anweisung READ_STS (IODDT_VAR1).

Spezifischer Kanalstatus: %MWr.m.c.3

Die folgende Tabelle verdeutlicht die verschiedenen Bedeutungen der Bits des Kanalstatusworts PROTOCOL (%MWr.m.c.3):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
PROTOCOL	INT	R	Byte 0 = 16#06 für Modbus-Mastermodus. Byte 0 = 16#07 für Modbus-Slavemodus. Byte 0 = 16#03 für Zeichenmodus.	%MWr.m.c.3

Das Lesen erfolgt durch die Anweisung READ_ST (SIODDT_VAR1).

Kanalbefehl: %MWr.m.c.24

In der folgenden Tabelle sind die verschiedenen Bedeutungen der Bits des Worts `CONTROL` (%MWr.m.c.24) aufgeführt:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
DTR_ON	BOOL	R/W	Signal "Datenübertragungseinrichtung bereit" einstellen.	%MWr.m.c.24.8
DTR_OFF	BOOL	R/W	Signal "Datenübertragungseinrichtung bereit" zurücksetzen.	%MWr.m.c.24.9
TO_MODBUS_MASTER	BOOL	R/W	Wechseln Sie vom Zeichenmodus beziehungsweise Modbus-Slavemodus in den Modbus-Mastermodus.	%MWr.m.c.24.12
TO_MODBUS_SLAVE	BOOL	R/W	Wechseln Sie vom Zeichenmodus beziehungsweise Modbus-Mastermodus in den Modbus-Slavemodus.	%MWr.m.c.24.13
TO_CHAR_MODE	BOOL	R/W	Wechseln Sie vom Modbus- in den Zeichenmodus.	%MWr.m.c.24.14

Der Befehl wird durch die Anweisung `WRITE_CMD (IODDT_VAR1)` ausgeführt.

Weitere Informationen zur Änderung von Protokollen finden Sie unter **Protokolländerungen** (siehe Seite 157).

Externe Konfigurationsobjekte des Typs T_COM_MB_BMX_CONF_EXT: %MWr.m.c.24.7 und %MWr.m.c.25

Die folgende Tabelle beschreibt die Bedeutung des Bits `CONTROL` (%MWr.m.c.24.7 und des Worts `CONTROL_DATA` (%MWr.m.c.25), die speziell für die Programmierung des Moduls BMX NOM 0200 vorgesehen sind:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
SAVE_SLAVE_ADDR	BOOL	R/W	Speichert die Steuerungsdaten in den Flash-Speicher	%MWr.m.c.24.7
SLAVE_ADDR	INT	R/W	In Flash zu speichernde Modbus-Slave-Adresse, von 0 bis 248 (0 für Master). HINWEIS: Beachten Sie, dass diese Funktionalität optional ist und es keinen Grund gibt, sie häufig anzuwenden. Da als Technologie Flash genutzt wird, kann der Chip beschädigt werden.	%MWr.m.c.25

Beschreibung der Sprachobjekte für die Konfiguration des Modbus-Modus

Auf einen Blick

Die folgenden Tabellen führen alle Konfigurationssprachobjekte für die Kommunikation im Modbus-Modus auf. Diese Objekte sind nicht in die IODDTs integriert und können vom Anwendungsprogramm angezeigt werden.

Liste der Objekte mit explizitem Austausch für den Mastermodus

In der folgenden Tabelle sind die Objekte mit explizitem Austausch aufgeführt.

Adresse	Typ	Zugriff	Bedeutung
%KWr.m.c.0	INT	R	Byte 0 dieses Worts entspricht dem folgenden Typ: • Wert 6 entspricht dem Master • Wert 7 entspricht dem Slave
%KWr.m.c.1	INT	R	Byte 0 dieses Worts entspricht der Übertragungsgeschwindigkeit. Dieses Byte kann mehrere Werte haben: • Wert -2 (0xFE) entspricht 300 Bit/s • Wert -1 (0xFF) entspricht 600 Bit/s • Wert 0 (0x00) entspricht 1200 Bit/s • Wert 1 (0x01) entspricht 2400 Bit/s • Wert 2 (0x02) entspricht 4800 Bit/s • Wert 3 (0x03) entspricht 9600 Bit/s • Wert 4 (0x04) entspricht 19200 Bit/s (Standardwert) • Wert 5 (0x05) entspricht 38400 Bit/s • Wert 6 (0x06) entspricht 57600 Bit/s (nur für BMX NOM 0200-Module) • Wert 7 (0x07) entspricht 115200 Bit/s (nur für BMX NOM 0200-Module) Byte 1 dieses Worts entspricht dem Format: • Bit 8: Anzahl der Bits (1 = 8 Bits (RTU), 0 = 7 Bits (ASCII)) • Bit 9 = 1: Paritätsverwaltung (1 = mit, 0 = ohne) • Bit 10: Paritätstyp (1 = ungerade, 0 = gerade) • Bit 11: Anzahl der Stoppbits (1 = 1 Bit, 0 = 2 Bits) • Bit 13: Physische Leitung (1 = RS232, 0 = RS485) • Bit 14: DTR/DSR/DCD-Modemsignale (nur für BMX NOM 0200-Module und physische RS232-Leitung) Wenn dieses Bit 1 auf gesetzt ist, werden die Modemsignale verwaltet. • Bit 15: Hardware-Datenflusskontrolle über RTS/CTS-Signale. Bei Auswahl von RS232 kann dieses Bit 2 Werte haben: 0 für RX/TX und 1 für RX/TX + RTS/CTS. Bei Auswahl von RS485 ist der Standardwert 0. Dieser Wert entspricht RX/TX.

Adresse	Typ	Zugriff	Bedeutung
%KWr.m.c.2	INT	R	Verzögerung zwischen Frames (nur in RTU-Modus): Wert in ms von 2 bis 10000 ms (abhängig von der ausgewählten Übertragungsgeschwindigkeit und vom ausgewählten Format). Der Standardwert ist 2 ms, wenn das Kontrollkästchen "Standard" aktiviert ist. 10 s entspricht unendlichem Warten.
%KWr.m.c.3	INT	R	Im Modbus-Mastermodus entspricht dieses Objekt der Antwortverzögerung in ms von 10 ms bis 1000 ms. 100 ms ist der Standardwert. 10 s entspricht unendlichem Warten.
%KWr.m.c.4	INT	R	Nur verfügbar im Modbus-Mastermodus. Byte 0 dieses Worts ist die Anzahl der Wiederholungen von 0 bis 15. Der Standardwert ist 3.
%KWr.m.c.5	INT	R	Bei Auswahl von RS232 entspricht dieses Wort von 0 bis 100 der RTS/CTS-Verzögerungszeit in Schritten von Hundert Millisekunden. Bei Auswahl von RS485 ist der Standardwert 0.

Liste der Objekte mit explizitem Austausch für den Slavemodus

Die Sprachobjekte für die Modbus-Slavefunktion sind identisch mit denen der Modbus-Masterfunktion. Lediglich die nachfolgend aufgeführten Objekte unterscheiden sich:

Adresse	Typ	Zugriff	Bedeutung
%KWr.m.c.3	INT	R	Im Modbus-Slavemodus entspricht Byte 0 dieses Objekts der Slave-Nummer [0/1, 247]. Für BMX NOM 0200-Module bedeutet ein Wert von 0, dass die Slave-Nummer im FLASH-Speicher abgelegt ist.
%KWr.m.c.4	INT	R	Nur im Modbus-Mastermodus.

Abschnitt 7.4

Sprachobjekte und IODDTs der Zeichenmoduskommunikation

Inhalt dieses Abschnitts

In diesem Abschnitt werden die Sprachobjekte und IODDTs der Zeichenmoduskommunikation erläutert.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Beschreibung der expliziten Austauschsprachobjekte für die Kommunikation im Zeichenmodus	143
Beschreibung der impliziten IODDT-Austauschobjekte vom Typ T_COM_CHAR_BMX	144
Beschreibung der expliziten IODDT-Austauschobjekte vom Typ T_COM_CHAR_BMX	145
Beschreibung der Sprachobjekte für die Konfiguration des Zeichenmodus	148

Beschreibung der expliziten Austauschsprachobjekte für die Kommunikation im Zeichenmodus

Auf einen Blick

Die folgenden Tabellen führen alle Konfigurationssprachobjekte für die Kommunikation im Zeichenmodus auf. Diese Objekte sind nicht in die IODDTs integriert.

Liste der expliziten Austauschobjekte

Die folgende Tabelle führt die expliziten Austauschobjekte auf:

Adresse	Typ	Zugriff	Bedeutung
%MW _r .m.c.4	INT	R	Unregelmäßigkeit bei gesendeten Zeichen.
%MW _r .m.c.5	INT	R	Unregelmäßigkeit bei empfangenen Zeichen.
%MW _r .m.c.24.0	BOOL	RW	Setzt Fehlerzähler zurück, wenn es auf 1 gesetzt ist
%QW _r .m.c.0 = 16#DEAD	INT	RW	BMX NOM 0200 neu starten.

Beschreibung der impliziten IODDT-Austauschobjekte vom Typ T_COM_CHAR_BMX

Einführung

Die nachstehenden Tabellen zeigen die impliziten Austauschobjekte des IODDT vom Typ T_COM_CHAR_BMX, die bei der Zeichenmodus-Kommunikation zum Einsatz kommen.

Fehlerbit

In der folgenden Tabelle wird die Bedeutung des Fehlerbits CH_ERROR (%Ir.m.c.ERR) erläutert:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
CH_ERROR	EBOOL	R	Fehlerbit des Kommunikationskanals	%Ir.m.c.ERR

Signalobjekt am Eingang

In der folgenden Tabelle wird die Bedeutung des Bits des Worts INPUT_SIGNALS (%IW.r.m.c.0) erläutert:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
DCD	BOOL	R	RS232-Signal DCD (Datenträgererkennung) (nur für das Modul BMX NOM 0200)	%IW.r.m.c.0.0
CTS	BOOL	R	Bereit zum Senden des RS232-Signals	%IW.r.m.c.0.2
DSR	BOOL	R	RS232-Signal DSR (Datensatz bereit) (nur für das Modul BMX NOM 0200)	%IW.r.m.c.0.3

HINWEIS: %IW.r.m.c.0.2 wird auf 1 gesetzt, wenn das CTS-Signal eine positive Spannung aufweist. Es ist ebenfalls auf DCD und DSR anwendbar.

Beschreibung der expliziten IODDT-Austauschobjekte vom Typ T_COM_CHAR_BMX

Einführung

Die nachstehenden Tabellen zeigen die expliziten Austauschobjekte des IODDT vom Typ T_COM_CHAR_BMX, die bei der Zeichenmodus-Kommunikation zum Einsatz kommen. Hierzu gehören Objekte des Typs Wort, deren Bits eine besondere Bedeutung haben. Diese Objekte werden im Folgenden ausführlich erläutert.

In diesem Abschnitt weist die Variable IODDT_VAR1 den Typ T_COM_STS_GEN auf.

Anmerkungen

Prinzipiell wird die Bedeutung für den Bitstatus 1 angegeben. In bestimmten Fällen wird jeder Bitstatus erläutert.

Es werden nicht alle Bits verwendet.

Flag für die Ausführung des expliziten Austauschs: EXCH_STS

Die folgende Tabelle verdeutlicht die verschiedenen Bedeutungen der Austauschsteuerungsbits des Kanals EXCH_STS (%MW_{r.m.c.0}):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
STS_IN_PROGR	BOOL	R	Lesen der Statuswörter des Kanals läuft	%MW _{r.m.c.0.0}
CMD_IN_PROGR	BOOL	R	Austausch der Befehlsparameter läuft	%MW _{r.m.c.0.1}
ADJ_IN_PROGR	BOOL	R	Austausch der Einstellparameter läuft (nicht für das Modul BMX NOM 0200)	%MW _{r.m.c.0.2}

Rückmeldung zum expliziten Austausch: EXCH_RPT

Die Tabelle unten zeigt die Bedeutung der Austauschrückmeldungsbits EXCH_RPT (%MW_{r.m.c.1}):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
STS_ERR	BOOL	R	Fehler beim Lesen der Statuswörter des Kanals erkannt	%MW _{r.m.c.1.0}
CMD_ERR	BOOL	R	Unregelmäßigkeit während eines Austauschs von Befehlsparametern	%MW _{r.m.c.1.1}
ADJ_ERR	BOOL	R	Unregelmäßigkeit während eines Austauschs von Einstellparametern (nicht für das Modul BMX NOM 0200)	%MW _{r.m.c.1.2}

Kanalspezifische Standardfehler, CH_FLT

Die folgende Tabelle verdeutlicht die verschiedenen Bedeutungen der Bits des Statusworts CH_FLT (%MWr.m.c.2):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
NO_DEVICE	BOOL	R	Kein Gerät auf dem Kanal funktioniert.	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	R	Ein Gerät auf dem Kanal ist nicht funktionsfähig	%MWr.m.c.2.1
BLK	BOOL	R	Klemmenleiste ist nicht angeschlossen.	%MWr.m.c.2.2
TO_ERR	BOOL	R	Timeout übernommen (Analyse erforderlich)	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Interner Modulfehler oder Selbsttest des Kanals	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Unterschiedliche Hard- und Softwarekonfiguration	%MWr.m.c.2.5
COM_FLT	BOOL	R	Kommunikationsanalyse erforderlich mit der SPS	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Anwendungsfehler erkannt (Anpassungs- oder Konfigurationsfehler)	%MWr.m.c.2.7

Das Lesen erfolgt durch die Anweisung READ_STS (IODDT_VAR1).

Spezifischer Kanalstatus, %MWr.m.c.3

Die folgende Tabelle verdeutlicht die verschiedenen Bedeutungen der Bits des Kanalstatusworts PROTOCOL (%MWr.m.c.3):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
PROTOCOL	INT	R	Byte 0 = 16#03 für die Zeichenmodus-Funktion	%MWr.m.c.3

Das Lesen erfolgt durch die Anweisung READ_STS (IODDT_VAR1).

Kanalbefehl %MWr.m.c.24

Die folgende Tabelle verdeutlicht die verschiedenen Bedeutungen der Bits des Kanalstatusworts CONTROL (%MWr.m.c.24):

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
DTR_ON	BOOL	R/W	Signal „Datenübertragungseinrichtung bereit“ einstellen	%MWr.m.c.24.8
DTR_OFF	BOOL	R/W	Signal „Datenübertragungseinrichtung bereit“ zurücksetzen	%MWr.m.c.24.9

Der Befehl wird durch die Anweisung WRITE_CMD (IODDT_VAR1) ausgeführt.

Weitere Informationen zur Änderung der Protokolle finden Sie unter Ändern der Protokolle (*siehe Seite 157*).

Wortobjekt %QWr.m.c.0

In der folgenden Tabelle wird die Bedeutung des Bits 0 des Worts %QWr.m.c.0 beschrieben:

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
STOP_EXCH	BOOL	R/W	Anhalten des gesamten Austauschs bei steigender Flanke (nur für das Modul BMX NOM 0200)	%QWr.m.c.0.0

Beschreibung der Sprachobjekte für die Konfiguration des Zeichenmodus

Auf einen Blick

Die folgenden Tabellen führen alle Konfigurationssprachobjekte für die Kommunikation im Zeichenmodus auf. Diese Objekte sind nicht in die IODDTs integriert und können vom Anwendungsprogramm angezeigt werden.

Liste der Objekte mit explizitem Austausch für den Zeichenmodus

In der folgenden Tabelle sind die Objekte mit explizitem Austausch aufgeführt.

Adresse	Typ	Zugriff	Bedeutung
%KWr.m.c.0	INT	R	Byte 0 dieses Worts entspricht dem folgenden Typ. Wert 3 entspricht dem Zeichenmodus.
%KWr.m.c.1	INT	R	Byte 0 dieses Worts entspricht der Übertragungsgeschwindigkeit. Dieses Byte kann mehrere Werte haben: • Wert -2 (0xFE) entspricht 300 Bit/s • Wert -1 (0xFF) entspricht 600 Bit/s • Wert 0 (0x00) entspricht 1200 Bit/s • Wert 1 (0x01) entspricht 2400 Bit/s • Wert 2 (0x02) entspricht 4800 Bit/s • Wert 3 (0x03) entspricht 9600 Bit/s (Standardwert) • Wert 4 (0x04) entspricht 19200 Bit/s • Wert 5 (0x05) entspricht 38400 Bit/s • Wert 6 (0x06) entspricht 57600 Bit/s (nur für BMX NOM 0200-Module wählbar) • Wert 7 (0x07) entspricht 115200 Bit/s (nur für BMX NOM 0200-Module wählbar) Byte 1 dieses Worts entspricht dem Format: • Bit 8: Anzahl der Bits (1 = 8 Bits (RTU), 0 = 7 Bits (ASCII)) • Bit 9 = 1: Paritätsverwaltung (1 = mit, 0 = ohne) • Bit 10: Paritätstyp (1 = ungerade, 0 = gerade) • Bit 11: Anzahl der Stoppbits (1 = 1 Bit, 0 = 2 Bits) • Bit 13: Physische Leitung (1 = RS232, 0 = RS485) • Bit 14: DTR/DSR/DCD-Modemsignale Für BMX NOM 0200-Module und bei Auswahl von RS232 kann dieses 2 Werte haben: 1 bedeutet, dass die Modemsignale verwaltet werden, 0 bedeutet, dass sich nicht verwaltet werden (Standardwert für BMX P34 oder bei Auswahl von RS485) • Bit 15: Hardware-Datenflusskontrolle über RTS/CTS-Signale. Bei Auswahl von RS232 kann dieses Bit 2 Werte haben: 0 für RX/TX und 1 für RX/TX + RTS/CTS. Bei Auswahl von RS485 ist der Standardwert 0. Dieser Wert entspricht RX/TX.

Adresse	Typ	Zugriff	Bedeutung
%KWr.m.c.2	INT	R	Eingegebener Wert in ms für "Stopp bei Stille" (abhängig von der ausgewählten Übertragungsgeschwindigkeit und vom ausgewählten Format). Der Wert 0 bedeutet keine Erkennung von Stille.
%KWr.m.c.3	INT	R	Dieses Wort entspricht dem Polarisierungstyp: • Ein Wert von 0 für Bit 14 und Bit 15 entspricht keiner Polarisierung (diese ist die Standardeinstellung für BMX P34 oder bei Auswahl von RS232) • Bit 14: Ein Wert von 1 entspricht einer Polarisierung mit niedriger Impedanz (wie Modbus) und kann nur für BMX NOM 0200-Module und bei Auswahl von RS485 verwendet werden. • Bit 15: Ein Wert von 1 entspricht einer Polarisierung mit hoher Impedanz und kann nur für BMX NOM 0200-Module und bei Auswahl von RS485 verwendet werden.
%KWr.m.c.5	INT	R	Dieses Wort entspricht der RTS/CTS-Verzögerungszeit in Hundertstel ms von 0 bis 100, wenn RS232 ausgewählt ist. Bei Auswahl von RS485 ist der Standardwert 0.
%KWr.m.c.6	INT	R	Bit 0 von Byte 0 kann zwei Werte haben: • Wert 1 entspricht dem aktivierten Kontrollkästchen "Stopp" im Bereich "Stopp bei Empfang" für Zeichen 1. • Wert 0 entspricht dem deaktivierten Kontrollkästchen "Stopp" im Bereich "Stopp bei Empfang" für Zeichen 1. Bit 1 von Byte 0 kann zwei Werte haben: • Wert 1 entspricht dem aktivierten Kontrollkästchen "Zeichen enthalten" im Bereich "Stopp bei Empfang" für Zeichen 1. • Wert 0 entspricht dem deaktivierten Kontrollkästchen "Zeichen enthalten" im Bereich "Stopp bei Empfang" für Zeichen 1. Byte 1 dieses Worts entspricht dem für die Option "Stopp bei Empfang" von Zeichen 1 eingegebenen Wert von 0 bis 255.
%KWr.m.c.7	INT	R	Bit 0 von Byte 0 kann zwei Werte haben: • Wert 1 entspricht dem aktivierten Kontrollkästchen "Stopp" im Bereich "Stopp bei Empfang" für Zeichen 2. • Wert 0 entspricht dem deaktivierten Kontrollkästchen "Stopp" im Bereich "Stopp bei Empfang" für Zeichen 2. Bit 1 von Byte 0 kann zwei Werte haben: • Wert 1 entspricht dem aktivierten Kontrollkästchen "Zeichen enthalten" im Bereich "Stopp bei Empfang" für Zeichen 2. • Wert 0 entspricht dem deaktivierten Kontrollkästchen "Zeichen enthalten" im Bereich "Stopp bei Empfang" für Zeichen 2. Byte 1 dieses Worts entspricht dem für die Option "Stopp bei Empfang" von Zeichen 2 eingegebenen Wert von 0 bis 255.

Abschnitt 7.5

IODDT Type T_GEN_MOD, anwendbar auf alle Module

Beschreibung der Sprachobjekte des IODDT vom Typ T_GEN_MOD

Einführung

Die Modicon X80-Module verfügen über einen zugeordneten IODDT vom Typ T_GEN_MOD.

Bemerkungen

Prinzipiell wird die Bedeutung der Bits für den Bitstatus 1 angegeben. In speziellen Fällen wird jeder Status des Bits erläutert.

Einige Bits werden nicht verwendet.

Liste der Objekte

In der folgenden Tabelle werden die Objekte des IODDT aufgeführt.

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
MOD_ERROR	BOOL	R	Modulfehlerbit	%Ir.m.MOD.ERR
EXCH_STS	INT	R	Steuerwort für den Modulaustausch	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Lesen von Statuswörtern des Moduls	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Wort für Austauschrückmeldung	%MWr.m.MOD.1
STS_ERR	BOOL	R	Ereignis beim Lesen von Modulstatuswörtern	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Internes Fehlerwort des Moduls	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	Modul funktionsunfähig	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Funktionsunfähige Kanäle	%MWr.m.MOD.2.1
BLK	BOOL	R	Klemmenleiste falsch verdrahtet	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Hardware- oder Software-Konfigurationsunregelmäßigkeit	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Modul fehlt oder nicht betriebsbereit	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Internes Fehlerwort des Moduls (nur Fipio-Erweiterung)	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Interner Modulfehler, Modul nicht betriebsbereit (nur Fipio-Erweiterung)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Funktionsunfähige Kanäle (nur Fipio-Erweiterung)	%MWr.m.MOD.2.9

Standardsymbol	Typ	Zugriff	Bedeutung	Adresse
BLK_EXT	BOOL	R	Klemmenleiste falsch verdrahtet (nur Fipio-Erweiterung)	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Hardware- oder Software-Konfigurationsunregelmäßigkeit (nur Fipio-Erweiterung)	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	Modul fehlt oder nicht betriebsbereit (nur Fipio-Erweiterung)	%MWr.m.MOD.2.14

Abschnitt 7.6

Sprachobjekte und gerätespezifische IODDTs der Modbus-Kommunikation

Gegenstand dieses Abschnitts

In diesem Abschnitt werden die der Modbus-Kommunikation zugeordneten Sprachobjekte und gerätespezifischen IODDTs erläutert.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
BMX NOM 0200.x-Geräte-DDT	153
Beschreibung des Bytes MOD_FLT	156

BMX NOM 0200.x-Geräte-DDT

Einführung

In diesem Abschnitt wird die in der Namensgebungsregel für Geräte-DDTs (*siehe EcoStruxure™ Control Expert, Programmiersprachen und Struktur, Referenzhandbuch*) vorgegebene Standardnamensgebung für die **NOM-Geräte-DDTs** von Control Expert beschrieben.

Der Name eines Geräte-DDT enthält folgende Informationen:

- Plattform mit:
 - M für Modicon X80- Modul
- Gerätetyp (COM für Kommunikation)
- Funktion (NOM für BMX NOM 0200.x)
- Richtung:
 - IN
 - OUT

Liste der impliziten Geräte-DDTs

In der folgenden Tabelle werden die Geräte-DDTs mit den zugehörigen **X80**-Modulen aufgeführt:

Gerätespezifischer DDT	Modicon X80-Module
T_M_COM_NOM	BMX NOM 0200.x

Beschreibung der impliziten Geräte-DDTs

Die nachstehende Tabelle zeigt die Bits des Statusworts T_M_COM_NOM:

Standardsymbol	Typ	Bedeutung	Zugriff
MOD_HEALTH	BOOL	0 = Modul mit erkanntem Fehler	Lesen
		1 = Modulbetrieb OK	
MOD_FLT	BYTE	Internes Fehlerbyte (siehe Seite 156) des Moduls	Lesen
COM_CH	ARRAY [0...1] von T_M_COM_NOM_CH		Array-Struktur

Die nachstehende Tabelle zeigt die Bits des Statusworts `T_M_COM_NOM_CH[0...1]`:

Standardsymbol		Typ	Bit	Bedeutung	Zugriff
FCT_TYPE		WORD		0 = Kanal wird nicht verwendet	Lesen
				3 = Zeichenmodus	
				6 = MODBUS-Master	
				7 = MODBUS-Slave	
CH_HEALTH		BOOL		0 = Kanal mit erkanntem Fehler	Lesen
				1 = Kanalbetrieb OK	
INPUT_SIGNALS [INT]	DCD	BOOL	0	RS-232-Signal DCD (Datenträgererkennung) (nur für BMX NOM 0200-Modul)	Lesen
	CTS	BOOL	2	Bereit zum Senden des RS-232-Signals	Lesen
	DSR	BOOL	3	RS-232-Signal DSR (Datensatz bereit) (nur für BMX NOM 0200-Modul)	Lesen
COMMAND [INT]	STOP_EXCH	BOOL	0	Steigende Flanke an 1: Der gesamte laufende Austausch wird gestoppt.	Lesen/Schreiben

Beschreibung der expliziten Geräte-DDT-Instanzen

Der explizite Austausch (Lesestatus) - nur gültig für Modicon X80-E/A-Kanäle - wird über die EFB-Instanzen `READ_STS_MX` (Modicon M580) bzw. `READ_STS_QX` (Modicon Quantum) verwaltet.

- Die Adresse des Zielkanals (`ADDR`) kann über den EF `ADDMX` (*siehe EcoStruxure™ Control Expert, Kommunikation, Bausteinbibliothek*) verwaltet werden (dazu wird `ADDMX_OUT` mit `ADDR` verbunden).
- Der Ausgangsparameter `READ_STS_MX` (*siehe EcoStruxure™ Control Expert, E/A-Verwaltung, Bausteinbibliothek*) oder `READ_STS_QX` (*siehe EcoStruxure™ Control Expert, E/A-Verwaltung, Bausteinbibliothek*) (`STS`) kann mit einer DDT-Instanz „`T_M_XXX_YYY_CH_STS`“ (manuell zu erstellende Variable) verbunden werden, wobei Folgendes gilt:
 - `XXX` verweist auf den Gerätetyp.
 - `YYY` verweist auf die Funktion.

Beispiel: `T_M_COM_NOM_CH_STS`

Die nachstehende Tabelle zeigt die Bits des Statusworts `T_M_COM_NOM_CH_STS`:

Typ	Typ	Zugriff
STRUCT	<code>T_M_COM_NOM_CH_STS</code>	

Die nachstehende Tabelle zeigt die Bits des Statusworts `T_M_COM_NOM_CH_STS`:

Standardsymbol		Typ	Bit	Bedeutung	Zugriff
CH_FLT [INT]	NO_DEVICE	BOOL	0	Kein Gerät auf dem Kanal funktioniert.	Lesen
	ONE_DEVICE_FLT	BOOL	1	Nicht betriebsfähiges Gerät auf dem Kanal	Lesen
	BLK	BOOL	2	Klemmenleistenfehler (nicht angeschlossen)	Lesen
	TO_ERR	BOOL	3	Timeout-Fehler (Verdrahtung defekt)	Lesen
	INTERNAL_FLT	BOOL	4	Interner Fehler oder Selbsttest des Kanals	Lesen
	CONF_FLT	BOOL	5	Konfigurationsfehler: Unterschiedliche Hardware- und Softwarekonfigurationen	Lesen
	COM_FLT	BOOL	6	Problem bei der Kommunikation mit der Steuerung	Lesen
	APPLI_FLT	BOOL	7	Anwendungsfehler (Einstellungs- oder Konfigurationsfehler)	Lesen
PROTOCOL		BYTE		6 für Modbus-Master, 3 für Zeichenmodus	Lesen
ADDRESS		BYTE		Adresse des Slaves	Lesen

Beschreibung des Bytes MOD_FLT

Byte MOD_FLT in Geräte-DDT

Struktur des Bytes MOD_FLT:

Bit	Symbol	Beschreibung
0	MOD_FAIL	• 1: Interner erkannter Fehler oder erkannter Modulusfall. • 0: Kein Fehler erkannt.
1	CH_FLT	• 1: Nicht betriebsfähige Kanäle. • 0: Kanäle sind betriebsfähig.
2	BLK	• 1: Fehler in Klemmenleiste erkannt. • 0: Kein Fehler erkannt. HINWEIS: Dieses Bit kann möglicherweise nicht verwaltet werden.
3	–	• 1: Modul führt Selbsttest aus. • 0: Modul führt keinen Selbsttest aus. HINWEIS: Dieses Bit kann möglicherweise nicht verwaltet werden.
4	–	Nicht verwendet
5	CONF_FLT	• 1: Hardware- oder Software-Konfigurationsfehler erkannt. • 0: Kein Fehler erkannt.
6	NO_MOD	• 1: Modul fehlt oder nicht betriebsbereit. • 0: Modul ist in Betrieb. HINWEIS: Dieses Bit wird nur von Modulen verwaltet, die sich in einem dezentralen Rack befinden und ein BME CRA 312 10-Adaptermodul haben. Module in einem dezentralen Rack verwalten dieses Bit nicht, das auf 0 bleibt.
7	–	Nicht verwendet

Kapitel 8

Dynamischer Protokollwechsel

Ändern des Protokolls mit BMXNOM0200-Modulen

Allgemein

Dieser Abschnitt beschreibt, wie mit dem Befehl `WRITE_CMD(ODDT_VAR1)` das Protokoll geändert werden kann, das von einer seriellen BMXNOM0200-Kommunikation verwendet wird.

Dieser Befehl kann zum Umschalten zwischen folgenden drei Protokollen eingesetzt werden:

- Modbus-Slave
- Modbus-Master
- Zeichenmodus

HINWEIS: Die Variable `ODDT_VAR1` muss entweder vom Typ `T_COM_MB_BMX` oder `T_COM_MB_BMX_CONF_EXT` sein.

Ändern des Protokolls: Grundprinzip

Erstellen Sie zunächst eine `ODDT`-Variable für den seriellen Kanal. Setzen Sie dann das Bit im Wort `ODDT_VAR1.CONTROL` (`%MWr.m.c.24`), das der gewünschten Protokolländerung entspricht, auf 1:

- `TO_MODEBUS_MASTER` (Bit 12): Das aktuelle Protokoll wird zu Modbus Master geändert.
- `TO_MODEBUS_SLAVE` (Bit 13): Das aktuelle Protokoll wird zu Modbus Slave geändert.
- `TO_CHAR_MODE` (Bit 14): Das aktuelle Protokoll wird zu Zeichenmodus geändert.

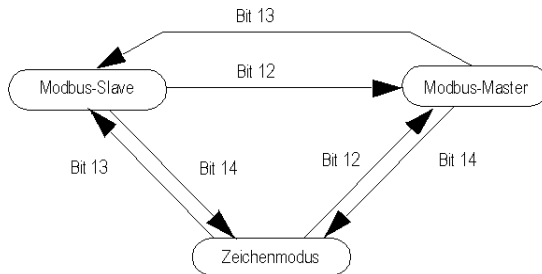
HINWEIS: Es kann jeweils ein einzelnes Bit auf 1 gesetzt werden: das Setzen mehrerer Bits auf 1 gleichzeitig führt zu einem Fehler.

HINWEIS: `ODDT_VAR1.CONTROL` (`%MWr.m.c.24`) ist Teil der `ODDT`-Variable `ODDT_VAR1`.

Wenden Sie danach die Anweisung `WRITE_CMD` auf die `ODDT`-Variable des seriellen Kanals an.

HINWEIS: Achten Sie darauf, dass zwei Master (auf demselben Bus) nicht gleichzeitig einen Request senden, denn dann gehen die Requests verloren und jede Rückmeldung enthält ein negatives Ergebnis, z. B. `16#0100` (Request konnte nicht verarbeitet werden) oder `16#ODFF` (Slave nicht vorhanden).

Das folgende Diagramm zeigt die Protokolländerungen, die gemäß der auf 1 gesetzten Bits des Worts IODDT_VAR1.CONTROL (%MWx.m.c.24) vorgenommen werden müssen:



Verwendung

Drei Protokolländerungen werden verwendet:

- Wechsel vom Modbus-Slave zum Modbus-Master:
Ziel der Modbus-Master-Konfiguration ist das Senden von Informationen über ein Ereignis an eine andere SPS. Bei einer Umschaltung von der Modbus-Slave- zur Modbus-Master-Konfiguration bleiben die Parameter für Übertragung, Signal- und physische Leitung unverändert. Nur die Werte der folgenden spezifischen Parameter der Modbus-Master-Konfiguration werden geändert:
 - Die Verzögerung zwischen Frames wird auf den Standardwert gesetzt, der von der Übertragungsgeschwindigkeit abhängig ist.
 - Die Antwortverzögerung wird auf 3 s eingestellt.
 - Die Anzahl der Wiederholungen wird auf 0 gesetzt.
- Wechsel von der Modbus-Slave/Master-Konfiguration zur Zeichenmoduskonfiguration
Der Wechsel zum Zeichenmodus ermöglicht das Senden von AT-Befehlen an ein Modem. Beim Wechsel von der Modbus-Konfiguration zur Zeichenmoduskonfiguration bleiben die Send-, Signal- und physischen Leitungsparameter gleich. Lediglich der Zeichenmodus-spezifische Parameter zur Erkennung des Nachrichtenendes wird auf den Empfang eines `x0d` als Endezeichen gesetzt.
- Wechsel vom Zeichenmodus zu Modbus-Master und Modbus-Slave:
Ziel der Zeichenmodus-Konfiguration ist die Kommunikation mit einem privaten Protokoll (einem Modem beispielsweise). Nachdem der Austausch abgeschlossen ist, wechselt der Benutzer zur Modbus-Master-Konfiguration (mit einer Antwortverzögerung von 3 s und einer Anzahl von Wiederholungen von 0), um Informationen über Ereignisse an andere SPS zu senden. Sobald die Nachricht gesendet wurde, stellt der Benutzer wieder die Modbus-Slave-Konfiguration her: die Slave-Nummer wird auf den im FLASH-Speicher abgelegten Wert gesetzt oder auf 248, wenn kein Wert abgelegt ist.

Kalt- und Warmstart

Protokolländerungen werden von den Bits %S0 und %S1 nicht beeinflusst (die Bits werden während eines Kalt- und Warmstarts jeweils auf 1 gesetzt). Der Kalt- oder Warmstart der SPS bewirkt jedoch die Konfiguration des seriellen Ports mit dessen Standardwerten bzw. den in der Anwendung programmierten Werten.

HINWEIS: Die Standardkonfiguration des Moduls ist wie folgt: Um eine einfache Konfiguration über einen Computer wie z. B. einen PC zu ermöglichen, ist der Kanal 0 auf den RS232-Slave-Modus und der Kanal 1 auf den RS485-Modus konfiguriert. Andere Parameter sind: 19200 Bauds, RTU, gerade Parität, 1 Stoppbit, keine Datenflusskontrolle, 1,75 ms als Standardverzögerung zwischen Frames, Slave-Nummer 248.

Teil III

Schnellstart: BMXNOM0200 als Modbus-Slave über eine Quantum SPS

Übersicht

In diesem Teil wird beschrieben, wie das Modul BMXNOM0200 als Modbus RS-485 RTU-Slave in einer Modicon-X80-Station über eine Quantum-SPS konfiguriert wird.

Das im Control Expert **Hardware-Katalog** zu konfigurierende Gerät ist BMXNOM0200.4.

Inhalt dieses Teils

Dieser Teil enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
9	Übersicht	163
10	Konfiguration in Control Expert	169

Kapitel 9

Übersicht

Voraussetzungen

Zum Konfigurieren des BMXNOM0200.4 ist Folgendes erforderlich:

- Folgende Firmware-Versionen verwenden:
 - BMXCRA31210: Minimal V2.14
 - BMXNOM0200: Minimal V1.5
- Verbinden Sie ein Quantum 140NOC78•00 zum Quantum 140CRP31200

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Produktübersicht	164
Übersicht über die Architektur	165
Einschränkungen	167

Produktübersicht

Beschreibung

Das BMXNOM0200.4 ist ein neues generisches Control Expert-Gerät, das Sie im Hardware-katalog Control Expert in der Kommunikationsfamilie finden können.

Bevor Sie das BMXNOM0200.4-Gerät in die Modicon X80-Station hinzufügen können, müssen Sie erste ein Drop-End-Kommunikatorgerät hinzufügen. Wählen Sie im Control Expert **Hardware-Katalog** das Gerät BMXCRA31210 (SV>=2.13) aus.

Unterstützte Protokolle

Für die Module des Typs BMXNOM0200 gilt:

- Kanal 0 ist RS232 oder RS485
- Kanal 1 ist nur RS485

Das Deklarieren des Moduls BMXNOM0200 als BMXNOM0200.4 in Control Expertermöglicht die Konfiguration des Moduls als:

- Modbus-RTU-Slave auf RS-485
- Modbus-Serial-RTU- und ASCII-Master auf RS-232 und RS-485
- Zeichenmodus

Kompatibilität

Dieses Angebot ist kompatibel mit dem Standardangebot: BMXNOM0200, und Quantum-CPU.140CRP31200BMXCRA31210

Übersicht über die Architektur

Beschreibung

Modbus-Slave-Nachrichten, die das BMXNOM0200.4 empfängt, werden an das Kopfmodul (BMXCRA31210) der Station übertragen. Anschließend leitet das Kopfmodul die Nachricht über Ethernet-E/A an die Quantum-CPU weiter.

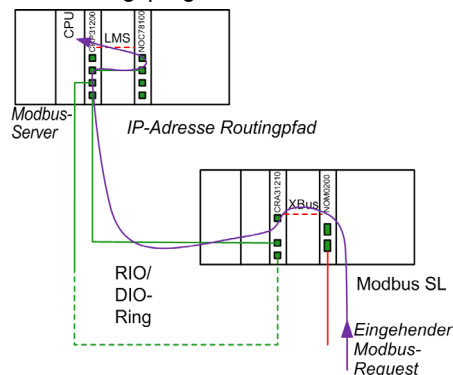
Das Quantum 140CRP31200 bearbeitet keine eingehenden Modbus-Nachrichten. Sie müssen ein zusätzliches Ethernet-Modul des Typs 140NOC78•00 in das Quantum-Haupttrack einsetzen, um es mit dem CRP-Modul zu verbinden.

Nach der Verbindung kann das Kopfmodul der Station die Modbus-Nachrichten an die 140NOC78•00 senden. Das 140NOC78•00 leitet die Nachrichten an die CPU weiter.

Dafür müssen Sie die IP-Adresse des 140NOC78•00 (Modbus-Server-Routingpfad (*siehe Seite 171*)) in die Konfiguration des Drop-End-Kommunikatormoduls (BMXCRA31210) eingeben.

Abbildung

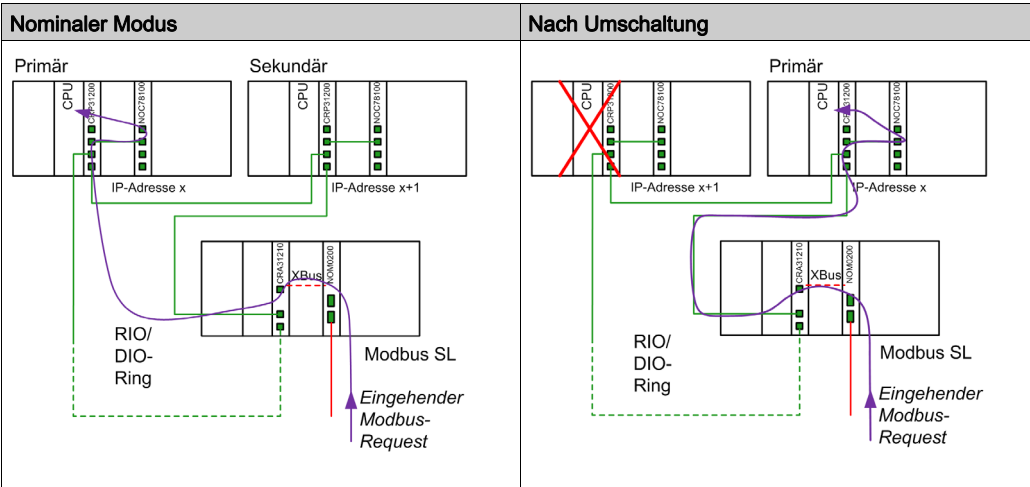
Das Quantum-CPU-System behandelt die Modbus-Requests ohne jegliches Anwendungsprogramm:



HINWEIS: Für das Routing der Modbus-Antwort wird der gleiche Pfad verwendet.

HSBY-Spezialfall

Die IP-Adresse des 140NOC78• 00 wechselt bei einer SPS-Umschaltung. Die Modbus-Requests werden dennoch weiterhin an die betriebsfähige CPU weitergeleitet:



HINWEIS: Die Modbus-Master-Anwendung verwaltet die Wiederholung von Requests im Falle eines Nachrichtenverlust, z. B. bei einer SPS-Umschaltung.

Einschränkungen

Maximale Konfiguration

Aus dieser Tabelle geht die maximale Konfiguration des BMXNOM0200.4 hervor:

Element	Maximale Konfiguration
Master-Kanal	4 pro konfigurierter Station mit maximal 36 Expertenkanälen pro Station. HINWEIS: Jeder konfigurierte Kanal des BMXNOM0200.4 zählt als Expertenkanal.
Station	4 BMXNOM0200.4 pro Station.
Quantum-System	16 BMXNOM0200
Modbus-Frame-Länge	256 Bytes

IP-Adresse

Sie müssen die IP-Adresse des Modbus-Routingpfads für jedes BMXCRA31210 konfigurieren, das ein BMXNOM0200.4 als Modbus-Slavemodul unterstützt.

Control Expert stellt keine Kontrolle der Konsistenz dieser IP-Adressen zur Verfügung.

WARNUNG

UNBEABSICHTIGTER GERÄTEBETRIEB

Stellen Sie sicher, dass die IP-Adresse tatsächlich zu dem Quantum gehört, das den Modbus-Server unterstützt.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

Unterstützte Protokolle

Nur Modbus RTU wird als Slave-Protokoll unterstützt.

Wenn Modbus-Slave ausgewählt ist, wird nur RS-485 unterstützt.

Unterstützte Modbus-Funktionscodes

In dieser Tabelle werden die Modbus-Funktionscodes (FC) aufgeführt, die der Quantum-Server unterstützt:

Verknüpfung mit -> Modbus-FC:	Variablentyp	Code	Funktion
01	%M	0X	Merkerbitstatus lesen (Ausgangsbit)
02	%M	1X	Eingabestatus lesen (Ausgangsbit)
03	%MW	4X	Halteregister lesen
05	%M	0X	Einzelmerkerbit forcieren
04	%MW	3X	Eingangsregister lesen
06	%MW	4X	Einzelnes Register schreiben
15	%M	0X	Mehrere Merkerbits schreiben
16	%MW	4X	Mehrere Register schreiben
23	%MW	4X	Lese-/Schreibzugriff auf mehrere Register

Kapitel 10

Konfiguration in Control Expert

Einführung

Die meisten Betriebsarten sind mit den zuvor unterstützten Versionen von BMXNOM0200 identisch.

In diesem Kapitel wird nur aufgeführt, was speziell für die Konfiguration des BMXNOM0200.4 in Control Expert gilt.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Einsetzen des Moduls	170
Fenster "Module Configuration"	171

Einsetzen des Moduls

Beschreibung

In einer Quantum-Ethernet-E/A-Architektur können Sie Module des Typs BMXNOM0200.4 nur in eine EIO-Modicon-X80-Station einsetzen, mit BMXCRA31210 (SV >= 2.13) als EIO-Adaptermodul.

Verfahren

Gehen Sie wie folgt vor, um das Modul BMXNOM0200.4 in eine dezentrale Modicon-X80-Station einzusetzen:

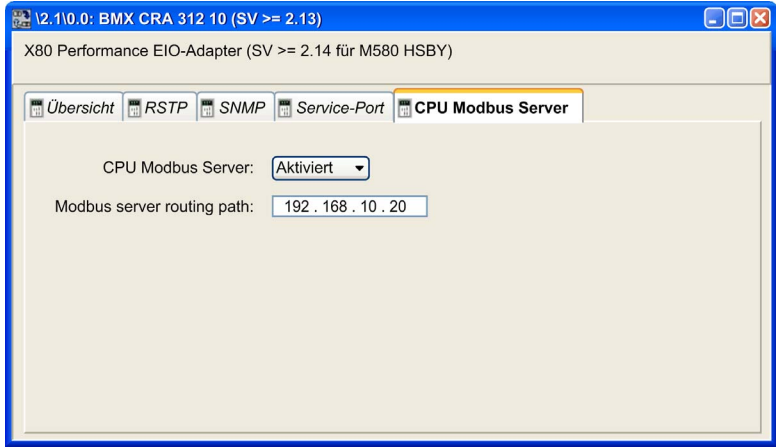
Schritt	Aktion
1	Setzen Sie das Modul 140CRP31200 in das lokale Quantum-Rack ein.
2	Erstellen Sie auf dem EIO-Bus eine EIO-Modicon-X80-Station mit einem BMXCRA31210 (SV >= 2.13).
3	Setzen Sie das neue Modul BMXNOM0200.4 in die Station ein.
4	Setzen Sie ein 140NOC78•00 in den Lokalen Bus des Quantum.

Fenster "Module Configuration"

Konfiguration des Modbus-Server-Routingpfads:

Diese Konfiguration ist nur im Offlinemodus möglich (SPS nicht angeschlossen).

Gehen Sie wie folgt vor, um den Routingpfad für den Modbus-Server zu konfigurieren:

Schritt	Aktion
1	Doppelklicken Sie auf das Modul BMXCRA31210 im Konfigurationseditor.
2	Wählen Sie die Registerkarte CPU Modbus-Server . 
3	Wählen Sie im Feld CPU Modbus-Server die Option Aktiviert aus.
4	Geben Sie die IP-Adresse des 140NOC78•00 in das Feld Routingpfad Modbus-Server ein. Das 140NOC78•00 verwaltet das Routing der Modbus-Frames zwischen der Ethernet-E/A und der CPU.

Control Expert stellt keine Kontrolle der Konsistenz dieser IP-Adressen zur Verfügung.

WARNUNG

UNBEABSICHTIGTER GERÄTEBETRIEB

Stellen Sie sicher, dass die IP-Adresse tatsächlich zu dem Quantum gehört, das den Modbus-Server unterstützt.

Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.

HINWEIS: Sie müssen die IP-Adresse des Modbus-Routingpfads für jedes BMXCRA31210 konfigurieren, das ein BMXNOM0200.4 als Modbus-Slavemodul unterstützt.

Zugriff auf das Konfigurationsfenster für Kanäle

Gehen Sie wie folgt vor, um die Kanalkonfigurationsfenster des Moduls BMXNOM0200.4 aufzurufen:

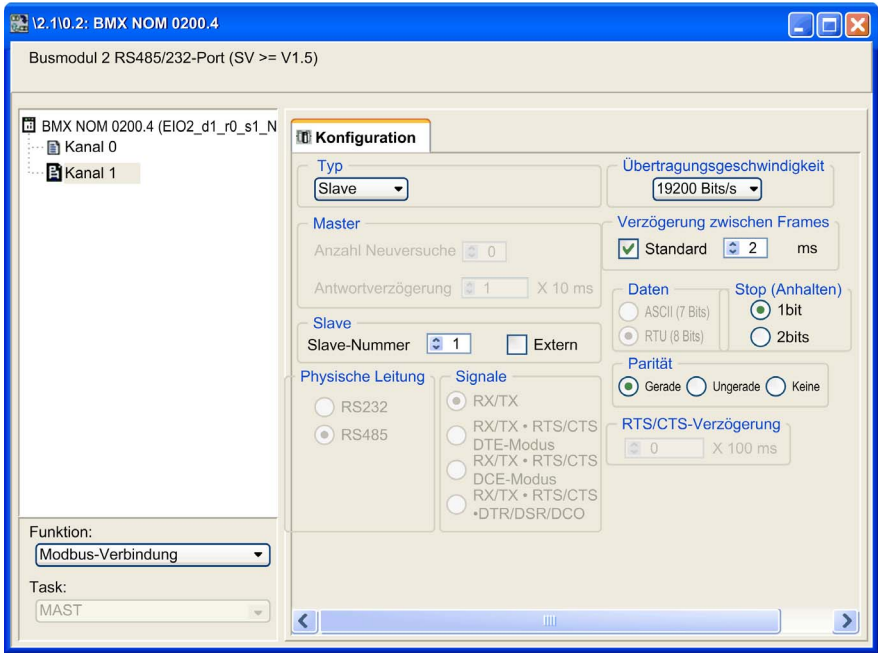
Schritt	Aktion
1	Öffnen Sie das Unterverzeichnis BMXNOM0200.4 im Projektbrowser.
2	Wählen Sie den zu konfigurierenden Kanal aus. Standardeinstellungen: ● Kanal 0 ist mit der Funktion Zeichenmodus-Verbindung konfiguriert. ● Kanal 1 ist mit der Funktion Modbus-Verbindung konfiguriert. HINWEIS: Einige Parameter sind nicht verfügbar und deshalb ausgegraut.

Informationen zum Konfigurieren der seriellen Modbus-Kommunikation im Master-Modus finden Sie im Kapitel *Serielle Modbus-Kommunikation (siehe Seite 55)*.

Informationen zum Konfigurieren der Zeichenmodus-Kommunikation finden Sie im Kapitel *Zeichenmodus-Kommunikation (siehe Seite 91)*.

Konfigurationsfenster für die Slave-Modbus-Verbindung

In der folgenden Abbildung ist das Slave-Konfigurationsfenster des Moduls BMXNOM0200.4 dargestellt:



Diese Tabelle enthält die Standardwerte der Parameter für das Modbus-Slave-Konfigurationsfenster:

Konfigurationsparameter	Standardwert
Typ	Slave
Slave-Nummer	1
Physische Leitung	Nur RS-485
Signale	Nur RX/TX
Übertragungsgeschwindigkeit	19200 Bit/s
Verzögerung zwischen Frames	2 ms
Daten	Nur 8 Bit
Anhalten	1 Bit
Parität	Gerade

HINWEIS: Modbus ist ein Standardprotokoll. Dieses Modul basiert auf einem Einzelmodus für den Datenaustausch.

Bei der Konfiguration der seriellen Modbus-Kommunikation im Master-Modus werden die Slave-Parameter ausgegraut und können nicht bearbeitet werden.



!

%I

In Übereinstimmung mit der IEC-Norm bezeichnet %I ein Sprachobjekt vom Typ "diskreter Eingang".

%IW

In Übereinstimmung mit der IEC-Norm bezeichnet %IW ein Sprachobjekt vom Typ "Analogeingang".

%KW

In Übereinstimmung mit der IEC-Norm bezeichnet %KW ein Sprachobjekt vom Typ "Konstantenwort".

%M

In Übereinstimmung mit der IEC-Norm bezeichnet %M ein Sprachobjekt vom Typ "Speicherbit".

%MW

In Übereinstimmung mit der IEC-Norm bezeichnet %MW ein Sprachobjekt vom Typ "Speicherwort".

%Q

In Übereinstimmung mit der IEC-Norm bezeichnet %Q ein Sprachobjekt vom Typ "diskreter Ausgang".

%QW

In Übereinstimmung mit der IEC-Norm bezeichnet %QW ein Sprachobjekt vom Typ "Analogausgang".

A

Adresse

In einen Netzwerk ist dies die Kennung einer Station. In einem Frame ist es eine Gruppe von Bits, die Quelle oder Ziel dieses Frames angeben.

Altivar

AC-Frequenzumrichter.

ARRAY

Ein ARRAY ist eine Tabelle, die Elemente eines einzelnen Typs enthält. Die Syntax lautet wie folgt: ARRAY [<Grenzen>] OF <Typ> Beispiel: ARRAY [1..2] OF BOOL ist eine eindimensionale Tabelle mit zwei Elementen des Typs BOOL. ARRAY [1..10, 1..20] OF INT ist eine zweidimensionale Tabelle mit 10x20 Elementen des Typs INT.

ASCII

ASCII ist die Abkürzung für American Standard Code for Information Interchange (amerikanischer Standardcode für den Informationsaustausch). Dies ist ein amerikanischer Code (der inzwischen zu einem internationalen Standard geworden ist), der 7 Bit zur Definition aller im Englischen verwendeten alphanumerischen Zeichen und Satzzeichen, bestimmter Graphikzeichen sowie verschiedener Befehle verwendet.

B

BOOL

BOOL bezeichnet den booleschen Datentyp. Dabei handelt es sich um ein grundlegendes Datenelement der Informatik. Eine Variable vom Typ BOOL besitzt einen der folgenden zwei Werte: 0 (FALSE) oder 1 (TRUE). Ein aus einem Wort extrahiertes Bit ist vom Typ BOOL. Beispiel: %MW10.4.

Broadcast

Bei Broadcast-Kommunikationen werden Pakete von einer Station an jedes Netzwerkziel gesendet. Broadcastnachrichten beziehen sich auf jedes Netzwerkgerät oder nur auf ein Gerät, dessen Adresse unbekannt ist.

BYTE

Eine Gruppe von 8 Bit bilden ein BYTE. Ein BYTE wird entweder im binären oder im oktalen Modus ausgedrückt. Der Datentyp BYTE wird in einem 8-Bit-Format kodiert. Im hexadezimalen Format erstreckt er sich von 16#00 bis 16#FF.

C

Control Expert

SPS-Programmiersoftware von Schneider Automation.

CPU

CPU ist die Abkürzung für Central Processing Unit (zentrale Verarbeitungseinheit): Allgemeiner Name für Prozessoren von Schneider Electric.

CRC

CRC ist die Abkürzung für den englischen Begriff Cyclic Redundancy Checksum (zyklische Redundanzprüfsumme): Sie gibt an, ob Zeichen während der Übertragung verändert, also "beschädigt" wurden.

D

DFB

DFB steht für Derived Function Block (abgeleiteter Funktionsbaustein). DFB-Typen sind Funktionsbausteine, die vom Benutzer in den Sprachen ST (Strukturierter Text), IL (Anweisungsliste), LD (Kontaktplan) oder FBD (Funktionsbausteinsprache) definiert werden können. Der Einsatz dieser DFB-Typen in Anwendungen ermöglicht Folgendes:

- Vereinfachen des Entwurfs und der Eingabe des Programms;
- Verbessern der Lesbarkeit des Programms;
- Leichtere Ausführung der Debugging-Funktion;
- Reduzierung der Menge des generierten Codes.

DINT

DINT steht für das 32-Bit-kodierte Format Double INTeger. Der gültige Wertebereich ist folgender: $-(2 \text{ hoch } 31)$ bis $(2 \text{ hoch } 31) - 1$. Beispiel: -2147483648, 2147483647, 16#FFFFFFFF.

Diskretes Modul

Modul mit diskreten (digitalen) Eingängen/Ausgängen.

E

EBOOL

EBOOL steht für Extended BOOLean (erweiterter boolescher Datentyp). Eine Variable vom Typ EBOOL besitzt entweder den Wert 0 (FALSE) oder den Wert 1 (TRUE), sowie steigende oder fallende Flanken und Forcierungsfunktionen. Eine Variable vom Typ EBOOL belegt ein Byte im Speicher. Das Byte enthält die folgenden Informationen:

- ein Bit für den Wert;
- ein Protokollbit (jedes Mal, wenn sich der Status des Objekts ändert, wird der Wert in das Protokollbit kopiert);
- ein Forcierungsbit (0 = keine Forcierung, 1 = Forcierung).

Der Standardwert der einzelnen Bits ist 0 (FALSE).

EF

EF bedeutet Elementary Function (elementare Funktion). Es handelt sich um einen Baustein, der in einem Programm verwendet wird und dort eine vordefinierte Funktion ausführt. Eine Funktion besitzt keine Informationen über den internen Status. Jedes Mal, wenn dieselbe Funktion mit denselben Eingangsparametern aufgerufen wird, liefert sie auch dieselben Ausgangswerte. Informationen zum grafischen Aufbau des Funktionsaufrufs finden Sie im "[Funktionsbaustein (Instanz)]". Im Gegensatz zum Aufruf der Funktionsbausteine enthalten die Funktionsaufrufe lediglich einen unbenannten Ausgang, dessen Name mit dem Namen der Funktion identisch ist. In FBD wird jeder Aufruf mittels des Grafikbausteins durch eine eindeutige [Nummer] bezeichnet. Diese Nummer wird automatisch verwaltet und kann nicht geändert werden. Der Entwickler positioniert und konfiguriert diese Funktionen in seinem Programm so, wie sie in der Anwendung ausgeführt werden sollen. Mithilfe des Software Development Kits SDKC können auch andere Funktionen entwickelt werden.

F

FBD

FBD bedeutet Function Block Diagram (Funktionsbausteindiagramm). FBD ist eine grafische Programmiersprache, die wie ein Ablaufdiagramm funktioniert. Durch Hinzufügen von einfachen Logikbausteinen wie AND und OR werden die einzelnen Funktionen bzw. Funktionsbausteine des Programms in diesem grafischen Format dargestellt. Bei jedem Baustein befinden sich die Eingänge links und die Ausgänge rechts. Die Ausgänge der Bausteine können mit den Eingängen weiterer Bausteine verbunden werden und auf diese Weise komplexe Ausdrücke bilden.

Fipio

Ein Feldbus zur Verbindung von Feldgeräten wie Sensoren oder Aktoren.

FLASH-Speicher

Der FLASH-Speicher ist ein nicht flüchtiger, überschreibbarer Speicher. Er wird in einem speziellen EEPROM gespeichert, der gelöscht und neu programmiert werden kann.

Frame

Ein Frame ist eine Gruppe von Bits, die einen digitalen Informationsblock bilden. Frames enthalten Netzwerk-Steuerungsinformationen oder Daten. Die Größe und Zusammensetzung eines Frames wird durch die verwendete Netzwerktechnologie bestimmt.

H

Halbduplex

Eine Methode, bei der Daten in beiden Richtungen übertragen werden können, jedoch zu einem Zeitpunkt immer nur in eine Richtung.

Hub

Ein Gerät, das eine Reihe flexibler und zentralisierter Module verbindet, um ein Netzwerk einzurichten.

I

INT

INT steht für das Format Single INTeget (Ganzzahl vom Typ Single, 16-Bit-kodiert). Der gültige Wertebereich ist folgender: $-(2 \text{ hoch } 15)$ bis $(2 \text{ hoch } 15) - 1$. Beispiel: 32768, 32767, 2#1111110001001001, 16#9FA4.

IODDT

IODDT bedeutet Input/Output Derived Data Type (abgeleiteter E/A-Datentyp). Mit dem Begriff IODDT wird ein strukturierter Datentyp bezeichnet, der ein Modul oder einen Kanal eines SPS-Moduls darstellt. Jedes Expertenmodul besitzt seine eigenen IODDTs.

K

Konfiguration

In der Konfiguration werden Daten zusammengefasst, die die Maschine (invariant) charakterisiert und die für den Betrieb des Moduls erforderlich sind. Alle diese Informationen werden im Speicherbereich %KW für die Steuerungskonstanten abgelegt. Sie können von der SPS-Anwendung nicht geändert werden.

L

LED

LED ist die Abkürzung für den Begriff "Licht emittierende Diode". Ein zur Anzeige verwendetes Bauelement, das im stromdurchflossenen Zustand leuchtet. Sie zeigt den Betriebsstatus des Kommunikationsmoduls an.

LRC

LRC ist die Abkürzung für den englischen Begriff Longitudinal Redundancy Check (Längssummenprüfung): Sie wurde entwickelt, weil die Fehlererkennung durch die Paritätsprüfung nicht hoch genug ist.

M

Master-Task

Hauptprogramm-Task. Sie ist obligatorisch und wird zur Ausführung der sequenziellen Verarbeitung der SPS verwendet.

Momentum

E/A-Module, die mehrere offene Standardkommunikationsnetzwerke verwenden.

N

Netzwerk

Das Wort "Netzwerk" hat zweierlei Bedeutung.

- Im Kontaktplan (LD): Ein Netzwerk ist eine Gruppe von untereinander verknüpften Grafikelementen. Die Reichweite eines Netzes beziehungsweise Netzwerks gilt in Bezug auf die organisatorische Einheit (Section) des Programms, in dem sich das Netz befindet, als lokal.
- Für Expertenkommunikationsmodule: Hier bedeuten die Begriffe Netz bzw. Netzwerk eine Gruppe von Stationen, die miteinander kommunizieren. Außerdem werden die Begriffe "Netz" und "Netzwerk" auch hier verwendet, um eine Gruppe von miteinander verbundenen grafischen Elementen zu bezeichnen. Eine solche Gruppe bildet dann einen Teil eines Programms, das aus einer Netzgruppe bestehen kann.

P

Protokoll

Beschreibt Meldungsformate und eine Reihe von Regeln, die von zwei oder mehr Geräten verwendet werden, um mittels dieser Formate zu kommunizieren.

R

RS232

Serieller Kommunikationsstandard, der die Spannung für den folgenden Dienst definiert:

- ein Signal von +12 V steht für logisch 0,
- ein Signal von -12 V steht für logisch 1.

Es gibt jedoch für den Fall eines gedämpften Signals eine mögliche Erkennung bis zu den Grenzwerten -3 V und +3 V. Zwischen diesen beiden Grenzwerten wird das Signal als ungültig betrachtet. RS-232-Verbindungen reagieren recht empfindlich auf Störungen. Die Norm gibt an, dass ein Abstand von 15 m oder maximal 9600 Baud (Bit/s) nicht überschritten werden darf.

RS485

Serieller Kommunikationsstandard, der bei 10 V/+5 V arbeitet. Er verwendet zwei Drähte für Senden/Empfangen. Deren Ausgänge mit drei möglichen Zuständen ermöglichen ihnen, am Ende der Übertragung in den Lausch-Modus umzuschalten.

RTU

RTU ist die Abkürzung für den englischen Begriff "Remote Terminal Unit" (entfernte Datenerfassungsstation). Im RTU-Modus werden Daten als zwei Hexadezimalzeichen mit vier Bit gesendet, wodurch bei gleicher Baudrate ein höherer Durchsatz als im ASCII-Modus erreicht wird. Modbus RTU ist ein binäres Protokoll und empfindlicher gegenüber Zeitverzögerungen als das ASCII-Protokoll.

S

Section

Programmmodul, das zu einer Task gehört und in einer vom Programmierer gewählten Sprache geschrieben sein kann (FBD, LD, ST, IL oder SFC). Eine Task kann aus mehreren Sections bestehen; die Reihenfolge der Ausführung der Sections entspricht der Reihenfolge, in der sie erzeugt werden. Die Reihenfolge kann bearbeitet werden.

SEPAM

Digitales Schutzrelais für Schutz, Steuerung und Überwachung in elektrischen Energieverteilungsnetzen.

Socket

Eine Kombination aus Port und IP-Adresse, die zur Identifikation von Sender oder Empfänger dient.

SPS

SPS ist die Abkürzung für speicherprogrammierbare Steuerung. Die SPS ist das Gehirn eines industriellen Fertigungsverfahrens. Im Gegensatz zu Relaisregelungssystemen automatisiert sie einen Prozess. Eine SPS ist ein Computer, der für die rauen Bedingungen industrieller Umgebungen geeignet ist.

ST

ST bedeutet Structured Text (Strukturierter Text). Unter "Strukturierter Text" versteht man eine Programmiersprache, die an die Hochsprachen der Programmierung angelehnt ist. Sie ermöglicht die Strukturierung einer Folge von Anweisungen.

STRING

Eine Variable vom Typ STRING ist eine aus ASCII-Zeichen zusammengesetzte Zeichenfolge. Eine solche Zeichenfolge kann maximal 65.534 Zeichen enthalten.

T**TAP**

TAP die Abkürzung für den englischen Begriff "Transmission Access Point" (Übertragungszugangspunkt): die Busanschlusseinheit.

Task

Eine Gruppe von Sections und Unterprogrammen, die zyklisch oder periodisch für die Task MAST oder periodisch für die Task FAST ausgeführt werden. Eine Task besitzt eine bestimmte Prioritätsstufe und ist mit Ein- und Ausgängen der SPS verknüpft. Diese E/A werden nacheinander aktualisiert.

V**Variable**

Speichereinheit vom Typ BOOL, WORD, DWORD usw., deren Inhalt durch das aktuell ausgeführte Programm geändert werden kann.

Vollduplex

Eine Methode zur Datenübertragung, bei der über den gleichen Kanal gleichzeitig gesendet und empfangen werden kann.

W

WORD

Der 16-Bit-kodierte Datentyp WORD wird verwendet, um Bitketten zu verarbeiten.
Die folgende Tabelle zeigt die Wertebereiche der einzelnen verwendbaren Basissysteme an:

Basis	Unterer Grenzwert	Oberer Grenzwert
Hexadezimal	16#0	16#FFFF
Oktal	8#0	8#177777
Binär	2#0	2#1111111111111111

Beispiele für die Darstellung:

Daten	Darstellung im entsprechenden Zahlensystem
0000000011010011	16#D3
1010101010101010	8#125252
0000000011010011	2#11010011

X

XBT

Grafikbasiertes Bediengerät.

XPS

Sicherheitsmodul zur Verarbeitung von Sicherheitssignalen zur Überwachung von Komponenten und Verdrahtung eines Sicherheitssystem, bestehend aus Geräten zur allgemeinen Überwachung sowie anwendungsspezifischen Modellen.



A

Ändern von Protokollen, *157*

B

Beschränkungen

BMXNOM0200, *50*

BMXNOM0200, *17*

Beschränkungen, *50*

Beschreibung, *20*

BMXNOM0200.4

Quantum SPS, *161*

BMXNOM0200H

Beschreibung, *20*

D

Debuggen der Modbus-Kommunikation, *88*

I

INPUT_BYTE, *104*

INPUT_CHAR, *104*

INPUT_CHAR_QX, *104*

K

Kanaldatenstruktur für alle Module

T_GEN_MOD, *150, 150*

Kanaldatenstruktur für die Modbus-Kommunikation

T_COM_MB_BMX, *136*

Kanaldatenstruktur für die Zeichenmodus-Kommunikation

T_COM_CHAR_BMX, *144, 145*

Kanaldatenstruktur für Kommunikationsprotokolle

T_COM_STS_GEN, *131, 132*

Kanaldatenstruktur für Modbus-Kommunika-

tion

T_COM_MB_BMX, *137*

M

MOD_FLT, *156*

N

Normen, *27*

P

Parametereinstellungen, *121*

PRINT_CHAR, *104*

PRINT_CHAR_QX, *104*

T

T_COM_CHAR_BMX, *144, 145*

T_COM_MB_BMX, *136, 137*

T_COM_STS_GEN, *131, 132*

T_GEN_MOD, *150, 150*

T_M_COM_NOM, *153*

V

Verbindungsgeräte, *33*

Verdrahtungszubehör, *43*

Verkabelung, *43*

Z

Zertifizierungen, *27*

