GERMAN

Modbus-Handbuch

Multifunktions-Leistungsmessgerät für DIN-Tragschiene

MID-Zertifiziert

Bestell-Nr: 236-9293

CE RoHS MID

rspro.com

GERMANY

Merkmale

- MID B+D Zertifiziert
- Zertifikatsnummer 0120/SGS0198
- Klasse B (kWh) EG-Richtlinie 2004/22/EC
- Multifunktionaler 100A Direktanschluss
- Integrierte Impuls- & RS485 Modbus-Ausgänge

1.2 Modbus-Protokoll Übersicht

Dieser Abschnitt bietet grundlegende Informationen für die Anbindung des Smart-Meters an ein Modbus-Protokoll-Netzwerk. Falls Hintergrundinformationen oder mehr Details zur Smart-Implementierung benötigt werden, beziehen Sie sich bitte auf die Abschnitte 2 und 3 dieses Dokuments.

RS485 ist eine bidirektionale, Voll- oder Halbduplex-Kommunikationsbusstruktur, die aus einem einzelnen Master und mindestens einem Slave besteht. Die maximale Anzahl der Slaves kann von System zu System stark variieren; die meisten Hersteller begrenzen die maximale Anzahl der Slaves jedoch auf einen Wert zwischen 16 und 32. Die meisten RS485-Signale arbeiten idealerweise mit einer DC-Grundspannung von 5 Volt.

Die Signale werden abwechselnd angesteuert. Das heißt, jede Leitung arbeitet invers zur anderen und ist elektrisch gesehen auch auf die andere bezogen. Der Empfänger betrachtet die Differenz der beiden Signale, nicht den absoluten Spannungswert. Dies wird als „Line Bias“ bezeichnet und ist in RS-485-Anwendungen kritisch. Eine Bias-Differenz von mehr als 0,3 V wird im Allgemeinen als gültig akzeptiert, kann aber je nach System bis zu 0,7 V betragen. Absolute Werte darunter gelten als „undefiniert“ oder „grau“ und können entweder zu einem hohen oder niedrigen Messwert beim Empfänger führen. In vielen nicht-isolierten Anwendungen wird zusätzlich zu den zwei Datenleitungen ein „Signal Ground“ (Signalmasse) oder eine Schirmverbindung bereitgestellt; dies ist jedoch nicht zwingend erforderlich, da die Signale aufeinander bezogen sind und nicht auf die absolute Erde. Es dient lediglich als Erdungspunkt für die Abschirmung des Kommunikationskabels. Es ist wichtig zu beachten, dass der Schirmgrund bei nicht-isolierten Anwendungen nur an einem einzelnen Stammende angeschlossen werden sollte. Wenn Massen in nicht-isolierten Systemen miteinander verbunden werden, erzeugt bereits ein geringer Spannungsunterschied zwischen den absoluten Erdungspunkten eine „Brummschleife“ (Ground Loop), die schwere Schäden an der Ausrüstung verursachen kann. Bei isolierten Geräten gilt dies nicht, da keine direkte Verbindung zwischen den Erdungspunkten besteht; dennoch sollte in allen Anwendungen eine ordnungsgemäße Schirmung praktiziert werden, um elektrische Störungen zu eliminieren oder zu reduzieren.

Unser Gerät 236-9293 bietet die Option einer RS485-Kommunikationseinrichtung für den direkten Anschluss an SCADA oder andere Kommunikationssysteme unter Verwendung des Modbus-Protokolls RTU (Slave). Das Modbus-Protokoll legt das Format für die Abfrage des Masters fest, indem es die Geräteadresse, einen Funktionscode, der die angeforderte Aktion definiert, alle zu sendenden Daten und ein Fehlerprüffeld enthält. Die Antwortnachricht des Slaves wird ebenfalls nach dem Modbus-Protokoll aufgebaut. Sie enthält Felder, die die durchgeführte Aktion bestätigen, alle zurückzugebenden Daten und ein Fehlerprüffeld. Tritt beim Empfang der Nachricht ein Fehler auf, erfolgt keine Antwort vom 236-9293. Wenn das Gerät die angeforderte Aktion nicht ausführen kann, erstellt es eine Fehlermeldung und sendet diese als Antwort.

Die elektrische Schnittstelle ist 2-Draht-RS485 über 2 Schraubklemmen. Der Anschluss sollte mit einem abgeschirmten Twisted-Pair-Kabel erfolgen (typischerweise 22 Gauge Belden 8761 oder gleichwertig). Alle „A“- und „B“-Anschlüsse werden in Reihe (Daisy Chain) geschaltet. Die Leitungstopologie kann je nach Kabeltyp und -länge Abschlusswiderstände erfordern. Eine Ringtopologie benötigt keinen Abschlusswiderstand. Die Impedanz des Abschlusswiderstands sollte der Kabelimpedanz entsprechen und an beiden Leitungsenden vorhanden sein. Das Kabel sollte an jedem Ende mit einem 120-Ohm-Widerstand (mind. 0,25 W) abgeschlossen werden. Eine Gesamtlänge von maximal 1200 Metern (3900 Fuß) ist für das RS485-Netzwerk zulässig. Maximal 32 elektrische Knoten können angeschlossen werden, einschließlich des Controllers. Die Adresse jedes 236-9293 kann auf einen beliebigen Wert zwischen 1 und 247 eingestellt werden. Der Broadcast-Modus (Adresse 0) wird nicht unterstützt.

Hinweis: Da in einer Modbus-Umgebung üblicherweise mit den durch die Modbus-Organisation verwendeten Fachbegriffen gearbeitet wird, werden einzelne Bezeichnungen in Abbildungen und Tabellen nicht übersetzt.

Format für jedes Byte im RTU-Modus:

Coding System:	8-bit per byte
Data Format:	4 bytes (2 registers) per parameter.
	Floating point format (to IEEE 754)
	Most signifcant register first (Default). The default may be changed if required -See Holding Register “Register Order” parameter.
Error Check Field:	2 byte Cyclical Redundancy Check (CRC)
Framing:	1 start bit
	8 data bits, least signifcant bit sent first
	1 bit for even/odd parity (or no parity)
	1 stop bit if parity is used; 1 or 2 bits if no parity

Datenkodierung

Alle Datenwerte im 236-9293 werden als 32-Bit IEEE754 Gleitkommazahlen übertragen (Eingang und Ausgang). Daher wird jeder Wert über zwei Modbus-Protokoll-Register übertragen. Alle Register- Leseanfragen und Daten-Schreibanfragen müssen eine gerade Anzahl von Registern spezifizieren. Versuche, eine ungerade Anzahl von Registern zu lesen/schreiben, führen dazu, dass das Gerät eine Modbus-Ausnahmenachricht zurückgibt. Zur Kompatibilität mit einigen SCADA-Systemen antwortet das Gerät jedoch auf das Lesen eines einzelnen Eingangs- oder Halteregisters mit einem Instrument spezifischen Wert.

Das 236-9293 kann maximal 40 Werte in einer einzelnen Transaktion übertragen; daher liegt die maximale Anzahl der anforderbaren Register bei 80. Das Überschreiten dieses Limits veranlasst das 236-9293 dazu, eine Ausnahmeantwort (Exception Response) zu generieren.

Die Übertragungsgeschwindigkeit der Daten kann zwischen 2400, 4800, 9600, 19200 und 38400 Baud gewählt werden.

1.2 Eingangsregister (Input Register)

Eingangsregister (Input Registers) werden verwendet, um die aktuellen Werte der gemessenen und berechneten elektrischen Größen anzuzeigen. Jeder Parameter wird in zwei aufeinanderfolgenden 16-Bit-Registern gespeichert. Die folgende Tabelle führt die 3X-Registeradressen sowie die Werte der Adress-Bytes innerhalb der Nachricht detailliert auf. Ein Häkchen ( ) in der Spalte zeigt an, dass der Parameter für das jeweilige Verdrahtungssystem gültig ist. Jeder Parameter mit einem Kreuz (X) gibt den Wert Null zurück. Alle Parameter werden in den 3X-Registern vorgehalten. Zum Zugriff auf sämtliche Parameter wird der Modbus-Protokoll-Funktionscode 04 verwendet.

For example, to request:	Amps 1	Start address	=0006
		No. of registers	=0002
	Amps 2	Start address	=0008
		No. of registers	=0002

Jede Datenabfrage muss auf maximal 40 Parameter beschränkt werden. Das Überschreiten dieses Limits von 40 Parametern führt dazu, dass ein Modbus-Protokoll-Ausnahmecode (Exception Code) zurückgegeben wird.

1.2.1 236-9293 Eingangsregister-Parameter

Address (Register)	236-9293 Input Register Parameter		Modbus Protocol Start Address Hex	
(Register)	Description	Units	Hi Byte	Lo Byte
30001	Voltage	Volts	00	00
30007	Current	Amps	00	06
30013	Active Power	Watts	00	0C
30019	Apparent ower	Volt Amps	00	12
30025	Reactiv Power	VAr	00	18
30031	Factor Power	None	00	1E
30037	Phase Angle	Degrees	00	24
30071	Frequency	Hz	00	46
30073	Import Active Energy	kWh	00	48
30075	Export Active Energy	kWh	00	4A
30077	Import Reactive Energy	kVArh	00	4C
30079	Export Reactive Energy	kVArh	00	4E
30343	Total Active Energy	kWh	01	56
30345	Total Reactive Energy	kVArh	01	58

1.3 Modbus-Protokoll: Holding-Register und Einstellung des Digitalmessgeräts

Holding-Register werden verwendet, um die Konfigurationseinstellungen des Geräts zu speichern und anzuzeigen. Alle Holding-Register, die nicht in der untenstehenden Tabelle aufgeführt sind, sind für den Hersteller reserviert; es sollte nicht versucht werden, deren Werte zu ändern.

Die Parameter der Holding-Register können über das Modbus-Protokoll eingesehen oder geändert werden. Jeder Parameter wird in zwei aufeinanderfolgenden 4X-Registern gespeichert. Zum Lesen der Parameter wird der Modbus-Protokoll-Funktionscode 03 verwendet, zum Schreiben der Funktionscode 16. Pro Nachricht darf nur in einen einzigen Parameter geschrieben werden.

1.3.1 Parameter der Modbus-Protokoll-Holding-Register

Address Register	Parameter	Modbus Protocol Start Address Hex		Valid range	Mode
		High Byte	Low Byte		
40013	Relay Pulse Width	00	OA	Write relay on period in milliseconds: 60, 100 or 200, default 200.	r/w
40019	Network Parity Stop	00	12	Write the network port parity/stop bits for MODBUS Protocol, where: 0 = One stop bit and no parity, default. 1 = One stop bit and even parity. 2 = One stop bit and odd parity.3 = Two stop bits and no parity. Requires a restart to become effective.	r/w
40021	Network Node	00	14	Write the network port node address: 1 to 247 for MODBUS Protocol, default 1. Requires a restart to become effective. Note, both the MODBUS Protocol and Johnson Controls node addresses can be changed via the display setup menus.	r/w
40029	Network Baud Rate	00	1C	Write the network port baud rate for MODBUS Protocol, where: r/w 0 = 2400 baud. 1 = 4800 baud. 2 = 9600 baud, default. 3 = 19200 baud. 4 = 38400 baud. Requires a restart to become effective	

Address Register	Parameter	Modbus Protocol Start Address Hex		Valid range	Mode
		High Byte	Low Byte		
462721	Demand Interval, Slide Time, Automatic Scroll Display Interval (Scroll Time), Backlight Time	F5	00	Data Format: BCD min-min-s-min Scroll Time=0: the display does not scroll automatically Backlight Time=0: Backlight is Always On.	r/w
463761	System Power	F9	10	Default Format: Hex 0000: 0.001kWh (kVArh) /imp (default) 0001: 0.01kWh (kVArh) /imp 0002: 0.1kWh (kVArh) /imp 0003: 1kWh (kVArh) /imp	r/w
463776	Measurement Mode	F9	20	Data Format: Hex 0001: Mode 1 (Total = Import) 0002: Mode 2 (Total = Import + Export) 0003: Mode 3 (Total = Import - Export)	r/w
363792	Pulse Output & LED Indicator Mode	F9	30	Data Format: Hex 0000: Import & Export Energy, LED ashes for Import & Export Energy 0001: Import Energy, LED ashes for Import Energy only 0002: Export Energy, LED ashes for Export Energy only	r/o

2. RS485 – Allgemeine Informationen

Die folgenden Informationen in diesem Abschnitt beziehen sich auf das 236-9293 und dienen zur Unterstützung beim Aufbau eines gemischten Netzwerks (Mixed Network). RS485 (bzw. EIA RS485 – Electronic Industries Association) ist ein symmetrisches Halbduplex-Übertragungssystem, das Übertragungsdistanzen von bis zu 1,2 km ermöglicht. Die folgende Tabelle fasst den RS-485-Standard zusammen

PARAMETER

Mode of Operation	Differential
Number of Drivers and Receivers	32 Drivers, 32 Receivers
Maximum Cable Length	1200 m
Maximum Data Rate	10 M baud
Maximum Common Mode Voltage	12 V to –7 V
Minimum Driver Output Levels (Loaded)	+/– 1.5 V
Minimum Driver Output Levels (Unloaded)	+/– 6 V
Drive Load	Minimum 60 ohms
Driver Output Short Circuit Current Limit	150 mA to Gnd, 250 mA to 12 V 250 mA to –7 V
Minimum Receiver Input Resistance	12 kohms
Receiver Sensitivity	+/– 200 mV

Weitere Informationen zu RS485 sind entweder von der EIA oder den verschiedenen Herstellern von RS485-Komponenten erhältlich, wie zum Beispiel Texas Instruments oder Maxim Semiconductors. Diese Liste erhebt keinen Anspruch auf Vollständigkeit.

2.1 Halbduplex

Halbduplex ist ein System, bei dem ein oder mehrere Sender („Talker“) mit einem oder mehreren Empfängern („Listeners“) kommunizieren können, wobei jedoch zu jedem Zeitpunkt nur ein einziger Sender aktiv sein darf. Man kann sich dies wie eine Unterhaltung vorstellen: Ein Gespräch beginnt mit einer Frage; die Person, die die Frage gestellt hat, hört dann so lange zu, bis sie eine Antwort erhält oder bis sie entscheidet, dass die befragte Person nicht antworten wird.

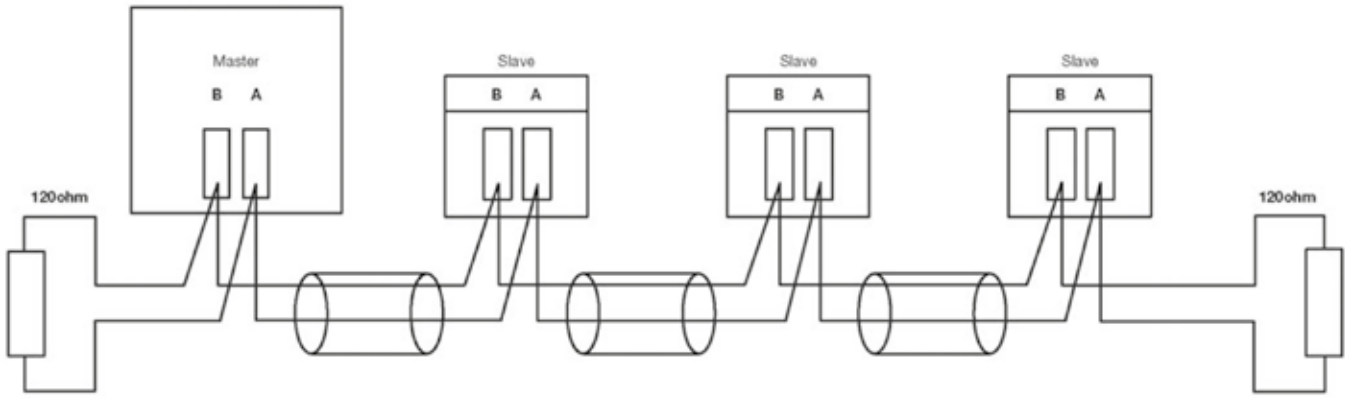
In einem RS485-Netzwerk beginnt der „Master“ die Unterhaltung mit einer „Abfrage“ (Query), die an einen spezifischen „Slave“ gerichtet ist. Der Master hört anschließend auf die Antwort des Slaves. Wenn der Slave nicht innerhalb eines vordefinierten Zeitraums antwortet (festgelegt durch die Steuerungssoftware im Master), bricht der Master die Unterhaltung ab.

2.2 Anschluss der Geräte

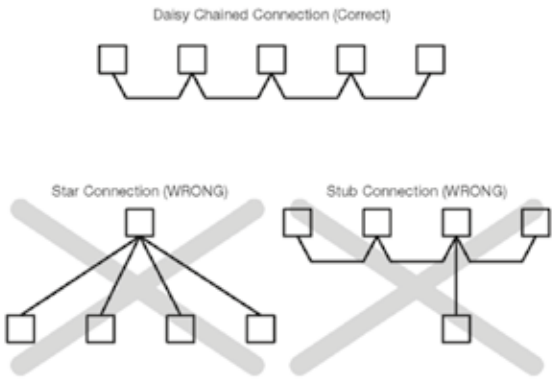
RS232-zu-485-Konverters mit einem USB-zu-RS232-Adapter in Erwägung gezogen wird. Nutzen Sie stattdessen entweder einen RS232-zu-RS485-Konverter, der direkt an eine geeignete RS232-Buchse am PC angeschlossen wird, einen reinen USB-zu-RS485-Konverter oder – bei Desktop-PCs – eine passende RS485-Einsteckkarte. (Viele 232:485-Konverter beziehen ihren Strom direkt aus der RS232-Buchse. Bei Verwendung eines USB-zu-RS232-Adapters liefert dieser möglicherweise nicht genügend Energie, um den 232:485-Konverter zu betreiben.)

Es sollte geschirmtes, verdrilltes Doppelleiterkabel (Screened Twisted Pair) verwendet werden. Bei längeren Kabelwegen oder in störungsanfälligen Umgebungen kann der Einsatz eines speziell für RS485 entwickelten Kabels notwendig sein, um eine optimale Leistung zu erzielen. Alle „A“-Anschlüsse müssen unter Verwendung eines Leiters des verdrillten Paares miteinander verbunden werden, und alle „B“-Anschlüsse entsprechend über den anderen Leiter des Paares.

Empfohlen wird ein Belden 9841 (einpaarig), 9842 (zweipaarig) oder ein ähnliches Kabel mit einem Wellenwiderstand von 120 Ohm. Das Kabel sollte an beiden Enden mit einem 120-Ohm-Widerstand (0,25 Watt oder höher) abgeschlossen werden. *Hinweis: Die Abbildung zeigt nur die Verdrahtungstopologie. Halten Sie sich stets an die Klemmenbezeichnung gemäß dem auf der Seite des 236-9293 angebrachten (gelaserten) Schaltplans.*

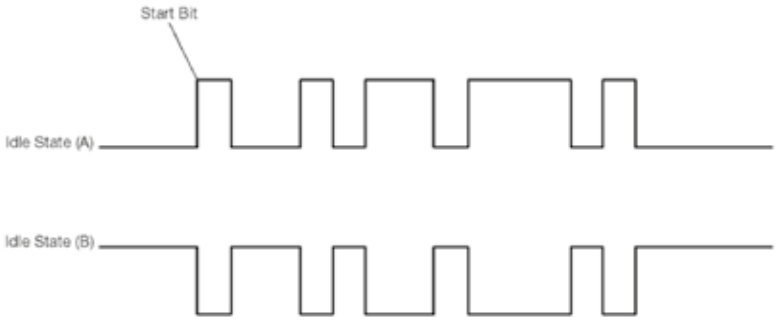


An jeder Klemme dürfen maximal zwei Drähte angeschlossen werden; dies stellt sicher, dass eine „Daisy Chain“-(Hintereinanderschaltung) oder eine „Straight Line“-Konfiguration (lineare Struktur) verwendet wird. Eine „Stern-Topologie“ oder ein Netzwerk mit „Stubs“ (Stichleitungen/Abzweigungen) wird nicht empfohlen, da Signalreflexionen innerhalb des Kabels zu Datenkorruption führen können.



2.3 A- und B-Anschlüsse

Die A- und B-Anschlüsse der 236-9293 Digitalmessgeräte können anhand der Signale identifiziert werden, die an ihnen anliegen, während Aktivität auf dem RS485-Bus herrscht:

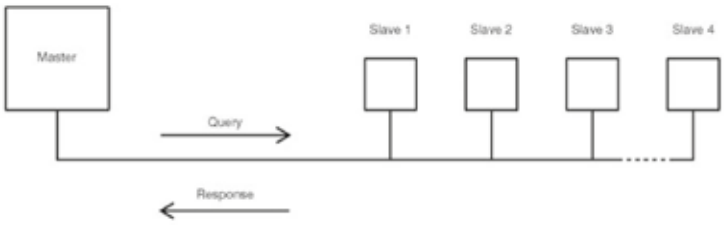


2.4 Fehlerbehebung (Troubleshooting)

- Beginnen Sie mit einem einfachen Netzwerk, bestehend aus einem Master und einem Slave. Beim Digitalmessgerät 236-9293 ist dies leicht zu bewerkstelligen, da das Netzwerk intakt bleiben kann, während einzelne Instrumente durch Abziehen des RS485-Anschlusses an der Geräterückseite getrennt werden.
- Überprüfen Sie, ob das Netzwerk korrekt verbunden ist. Das heißt, alle „A“-Anschlüsse sind miteinander verbunden und alle „B“-Anschlüsse sind miteinander verbunden. Stellen Sie sicher, dass die auf die RS485-Leitung „gesendeten“ Daten nicht über die RS232-Leitungen an den PC zurückgespiegelt (Echo) werden. (Diese Funktion ist bei Konvertern manchmal eine Steckbrücken-Option/Link-Option). Viele PC-basierte Softwarepakete scheinen nicht gut zu funktionieren, wenn sie ein Echo der Nachricht empfangen, die sie gerade selbst senden. (Software wie SpecView und PCView enthalten diese Funktion vermutlich).
- Bestätigen Sie, dass die Adresse des Instruments mit der Adresse übereinstimmt, die der „Master“ erwartet.
- Wenn das Netzwerk mit einem Instrument funktioniert, aber nicht mit mehreren, prüfen Sie, ob jedes Instrument eine eindeutige Adresse hat.
- Jede Datenabfrage muss auf maximal 40 Parameter beschränkt werden. Eine Verletzung dieser Anforderung beeinträchtigt die Leistung des Geräts und kann zu Antwortzeiten führen, die außerhalb der Spezifikation liegen.
- Überprüfen Sie, ob der Modbus-Protokoll-Modus (RTU oder ASCII) und die seriellen Parameter (Baudrate, Anzahl der Datenbits, Anzahl der Stoppbits und Parität) für alle Geräte im Netzwerk identisch sind.
- Stellen Sie sicher, dass der „Master“ Gleitkommavariablen (Floating-Point) abfragt (Registerpaare an Gleitkommagrenzen) und diese Variablen nicht „zerteilt“.
- Prüfen Sie, ob die vom „Master“ erwartete Byte-Reihenfolge der Gleitkommazahlen (Endianness) mit der des 236-9293 übereinstimmt. (Pakete wie PCView und Citect unterstützen verschiedene Formate, einschließlich des von diesem Messgerät verwendeten).
- Falls möglich, nutzen Sie einen zweiten RS232-zu-RS485-Konverter und verbinden Sie diesen mit dem RS485-Bus und einem zusätzlichen PC, auf dem eine Software installiert ist, die den Datenverkehr auf dem Bus anzeigen kann (Sniffer). Prüfen Sie so, ob gültige Abfragen vorhanden sind.

3. Modbus-Protokoll – Allgemeine Informationen

Die Kommunikation in einem Modbus-Protokoll-Netzwerk wird durch einen „Master“ initiiert (gestartet), der eine Abfrage (Query) an einen „Slave“ sendet. Der „Slave“, welcher das Netzwerk ständig auf an ihn gerichtete Abfragen überwacht, antwortet, indem er die angeforderte Aktion ausführt und eine Antwort an den „Master“ zurücksendet. Nur der „Master“ kann eine Abfrage initiieren.



Im Modbus-Protokoll kann der Master einzelne Slaves adressieren oder über eine spezielle „Broadcast“-Adresse eine Sammelmitteilung an alle Slaves senden. Das 236-9293 Digitalmessgerät unterstützt die Broadcast-Adresse nicht.

3.1 Modbus-Protokoll: Nachrichtenformat

Das Modbus-Protokoll definiert das Format für die Abfrage des Masters und die Antwort des Slaves.

Die Abfrage (Query) enthält die Geräteadresse (oder Broadcast-Adresse), einen Funktionscode, der die gewünschte Aktion definiert, alle zu sendenden Daten sowie ein Feld zur Fehlerüberprüfung.

Die Antwort (Response) enthält Felder zur Bestätigung der ausgeführten Aktion, alle zurückzugebenden Daten und ein Feld zur Fehlerüberprüfung. Wenn beim Empfang der Nachricht ein Fehler aufgetreten ist, wird die Nachricht ignoriert. Falls der Slave nicht in der Lage ist, die angeforderte Aktion auszuführen, erstellt er eine Fehlermeldung und sendet diese als Antwort zurück. Die vom 236-9293 verwendeten Modbus-Protokollfunktionen kopieren 16-Bit-Registerwerte zwischen Master und Slaves. Die vom Gerät verwendeten Daten liegen jedoch im 32-Bit-IEEE-754-Gleitkommaformat vor. Daher wird jeder Instrumentenparameter konzeptionell in zwei benachbarten Modbus-Protokoll-Registern gehalten. Abfrage (Query)

Das folgende Beispiel illustriert die Abfrage eines einzelnen Gleitkomma-Parameters (Floating Point), also von zwei 16-Bit-Modbus-Protokoll-Registern:

First Byte			Last Byte	
Slave Address	Function Code	Error Code	Error Check (Lo)	Error Check (Hi)

Slave-Adresse: Ein 8-Bit-Wert, der den adressierten Slave repräsentiert (1 bis 247); die 0 ist für die Broadcast-Adresse reserviert. Die Digitalmessgeräte unterstützen die Broadcast-Adresse nicht.

Funktionscode: Ein 8-Bit-Wert, der dem adressierten Slave mitteilt, welche Aktion ausgeführt werden soll. (Für das Digitalmessgerät sind die Werte 3, 4, 8 oder 16 gültig).

Startadresse (Hi): Die oberen (höchstwertigen) acht Bits einer 16-Bit-Zahl, welche die Startadresse der angeforderten Daten spezifiziert.

Startadresse (Lo): Die unteren (niederwertigen) acht Bits einer 16-Bit-Zahl, welche die Startadresse der angeforderten Daten spezifiziert. Da Register paarweise verwendet werden und bei Null beginnen, muss dies eine gerade Zahl sein.

Anzahl der Punkte (Hi): Die oberen (höchstwertigen) acht Bits einer 16-Bit-Zahl, welche die Anzahl der angeforderten Register spezifiziert.

Anzahl der Punkte (Lo): Die unteren (niederwertigen) acht Bits einer 16-Bit-Zahl, welche die Anzahl der angeforderten Register spezifiziert. Da Register paarweise verwendet werden, muss dies eine gerade Zahl sein.

Fehlerprüfung (Lo): Die unteren (niederwertigen) acht Bits einer 16-Bit-Zahl, die den Wert der Fehlerprüfung darstellt.

Fehlerprüfung (Hi): Die oberen (höchstwertigen) acht Bits einer 16-Bit-Zahl, die den Wert der Fehlerprüfung darstellt.

Antwort (Response)

Das Beispiel illustriert die normale Antwort auf eine Abfrage eines einzelnen Gleitkomma-Parameters, d. h. von zwei 16-Bit-Modbus-Protokoll-Registern.

First Byte			Last Byte	
Slave Address	Function Code	Error Code	Error Check (Lo)	Error Check (Hi)

Slave-Adresse: Ein 8-Bit-Wert, der die Adresse des Slaves darstellt, der gerade antwortet.
Funktionscode: Ein 8-Bit-Wert. Wenn dieser eine Kopie des Funktionscodes aus der Abfrage ist, zeigt dies an, dass der Slave die Abfrage erkannt und darauf geantwortet hat. (Siehe auch „Ausnahmeantwort“).
Byte-Zähler (Byte Count): Ein 8-Bit-Wert, der die Anzahl der in dieser Antwort enthaltenen Daten-Bytes angibt.
Erstes Register (Hi)*: Die oberen (höchstwertigen) acht Bits einer 16-Bit-Zahl, die das erste in der Abfrage angeforderte Register darstellt.
Erstes Register (Lo)*: Die unteren (niederwertigen) acht Bits einer 16-Bit-Zahl, die das erste in der Abfrage angeforderte Register darstellt.
Zweites Register (Hi)*: Die oberen (höchstwertigen) acht Bits einer 16-Bit-Zahl, die das zweite in der Abfrage angeforderte Register darstellt.
Zweites Register (Lo)*: Die unteren (niederwertigen) acht Bits einer 16-Bit-Zahl, die das zweite in der Abfrage angeforderte Register darstellt.
Fehlerprüfung (Lo): Die unteren (niederwertigen) acht Bits einer 16-Bit-Zahl, die den Wert der Fehlerprüfung darstellt.
Fehlerprüfung (Hi): Die oberen (höchstwertigen) acht Bits einer 16-Bit-Zahl, die den Wert der Fehlerprüfung darstellt.

**Diese vier Bytes zusammen ergeben den Wert des angeforderten Gleitkomma-Parameters.*

Ausnahmeantwort (Exception Response)

Falls ein Fehler im Inhalt der Abfrage festgestellt wird (ausgenommen Paritätsfehler und Unstimmigkeiten bei der Fehlerprüfung/CRC), wird eine Fehlermeldung (eine sogenannte **Ausnahmeantwort**) an den Master gesendet.

Die Ausnahmeantwort ist daran erkennbar, dass der Funktionscode eine Kopie des Abfrage-Funktionscodes ist, bei dem jedoch das **höchstwertige Bit (MSB) gesetzt** wurde. Die in einer Ausnahmeantwort enthaltenen Daten bestehen aus einem einzigen Byte, dem **Fehlercode (Error Code)**.

First Byte			Last Byte	
Slave Address	Function Code	Error Code	Error Check (Lo)	Error Check

Slave-Adresse: Ein 8-Bit-Wert, der die Adresse des antwortenden Slaves darstellt.
Funktionscode: Ein 8-Bit-Wert, der dem Funktionscode der Abfrage entspricht, jedoch mit 80 hex (ODER-verknüpft) kombiniert wurde. Dies zeigt an, dass der Slave die Abfrage entweder nicht erkennt oder die angeforderte Aktion nicht ausführen konnte.
Fehlercode (Error Code): Ein 8-Bit-Wert, der die Art der festgestellten Ausnahme angibt (siehe die spätere „Tabelle der Ausnahmecodes“).
Fehlerprüfung (Lo): Die unteren acht Bits der 16-Bit-Fehlerprüfung.
Fehlerprüfung (Hi): Die oberen acht Bits der 16-Bit-Fehlerprüfung.

3.2 Serielle Übertragungsmodi

Es gibt zwei serielle Modbus-Protokollmodi: ASCII und RTU. Das Digitalmessgerät 236-9293 unterstützt den ASCII-Modus nicht.
Im RTU-Modus (Remote Terminal Unit) wird jedes 8-Bit-Byte im vollen Binärbereich genutzt und ist nicht wie im ASCII-Modus auf ASCII-Zeichen beschränkt. Die höhere Datendichte ermöglicht einen besseren Datendurchsatz bei gleicher Baudrate; allerdings muss jede Nachricht in einem kontinuierlichen Datenstrom übertragen werden. Dies stellt für moderne Kommunikationsgeräte in der Regel kein Problem dar.
Kodierungssystem: Volle 8-Bit-Binärdaten pro Byte. In diesem Dokument wird der Wert jedes Bytes als zwei Hexadezimalzeichen (0–9 oder A–F) dargestellt.
Leitungsprotokoll: 1 Start-Bit, gefolgt von 8 Daten-Bits. Die 8 Daten-Bits werden mit dem niederwertigsten Bit zuerst (LSB first) gesendet.
Benutzeroptionen für Parität und Stopp-Bits:
Keine Parität (No Parity) und 2 Stopp-Bits
Keine Parität (No Parity) und 1 Stopp-Bit
Gerade Parität (Even Parity) und 1 Stopp-Bit
Ungerade Parität (Odd Parity) und 1 Stopp-Bit
Benutzeroptionen für die Baudrate:
4800, 9600, 19200, 38400 (Hinweis: Diese Digitalmessgeräte unterstützen 38400 nicht, bieten dafür aber 2400 an).
Die Baudrate, Parität und Stopp-Bits müssen so gewählt werden, dass sie exakt mit den Einstellungen des Masters übereinstimmen.

3.3 Modbus-Protokoll: Nachrichten-Timing (RTU-Modus)

Eine Nachricht im Modbus-Protokoll hat definierte Anfangs- und Endpunkte. Die empfangenden Geräte erkennen den Beginn der Nachricht, lesen die „Slave-Adresse“, um festzustellen, ob sie adressiert sind, und können nach Abschluss der Nachricht die Fehlerprüf-Bytes (CRC) und Paritätsbits verwenden, um die Integrität der Nachricht zu bestätigen. Wenn die Fehlerprüfung oder die Parität fehlschlägt, wird die Nachricht verworfen.

Start der Nachricht: Im RTU-Modus beginnen Nachrichten mit einem Ruheintervall von mindestens 3,5 Zeichenzeiten.

Adressprüfung: Das erste übertragene Byte einer Nachricht ist die Geräteadresse. Master- und Slave-Geräte überwachen das Netzwerk kontinuierlich, auch während der Ruheintervalle. Wenn das erste Byte empfangen wird, prüft jedes Gerät, ob es selbst adressiert ist. Ist dies der Fall, zeichnet es die gesamte Nachricht auf und handelt entsprechend; andernfalls überwacht es das Netzwerk weiter auf die nächste Nachricht.

Ende der Nachricht: Nach dem zuletzt übertragenen Byte markiert ein Ruheintervall von mindestens 3,5 Zeichenzeiten das Ende der Nachricht. Danach kann eine neue Nachricht beginnen.

Besonderheit beim 236-9293: Bei den Modellen 1000 und 2000 ist ein Ruheintervall von mindestens 60 ms erforderlich, um den erfolgreichen Empfang der nächsten Anfrage zu garantieren.

Kontinuität des Datenstroms: Die gesamte Nachricht muss als kontinuierlicher Datenstrom übertragen werden. Tritt vor Abschluss der Nachricht ein Ruheintervall von mehr als 1,5 Zeichenzeiten auf, verwirft das empfangende Gerät die unvollständige Nachricht und geht davon aus, dass das nächste Byte die Adresse einer neuen Nachricht ist.

Wenn eine neue Nachricht früher als 3,5 Zeichenzeiten nach einer vorherigen Nachricht beginnt, betrachtet das Empfangsgerät diese möglicherweise als Fortsetzung der vorangegangenen Nachricht. Dies führt zu einem Fehler, da der Wert im CRC-Feld für die kombinierten Nachrichten nicht gültig ist.

3.4 Wie Zeichen seriell übertragen werden

Wenn Nachrichten in Standard-Modbus-Seriennetzwerken übertragen werden, wird jedes Byte in dieser Reihenfolge gesendet (von links nach rechts): Übertragungszeichen = Start-Bit + Daten-Byte + Paritätsbit + 1 Stopp-Bit (insgesamt 11 Bits).

Die Übertragung erfolgt vom niederwertigsten Bit (LSB) zum höchstwertigsten Bit (MSB).

Start	1	2	3	4	5	6	7	8	Party	Stop

Timeouts: Übertragungszeichen = Start-Bit + Daten-Byte + 2 Stopp-Bits (insgesamt 11 Bits):

Start	1	2	3	4	5	6	7	8	Party	Stop

Das Gerät 236-9293 unterstützt zusätzlich die Konfiguration „Keine Parität, ein Stopp-Bit“). Übertragungszeichen = Start-Bit + Daten-Byte + 1 Stopp-Bit (insgesamt 10 Bits)

Start	1	2	3	4	5	6	7	8	Party	Stop

Der Master wird vom Benutzer so konfiguriert, dass er ein vorbestimmtes Timeout-Intervall abwartet. Der Master wartet diesen Zeitraum ab, bevor er entscheidet, dass der Slave nicht antworten wird und die Transaktion abgebrochen werden sollte. Bei der Festlegung der Timeout-Dauer müssen die Spezifikationen sowohl des Masters als auch der Slaves sorgfältig geprüft werden. Der Slave definiert die „Antwortzeit“ (Response Time) oft als den Zeitraum vom Empfang des letzten Bits der Abfrage bis zum Senden des ersten Bits der Antwort. Der Master definiert die „Antwortzeit“ hingegen oft als den Zeitraum zwischen dem Senden des ersten Bits der Abfrage bis zum Empfang des letzten Bits der Antwort. Daraus ergibt sich, dass die Übertragungszeit der Nachricht, welche von der Baudrate abhängt, zwingend in die Timeout-Berechnung einfließen muss.

Visualisierung des Zeitablaufs (Query/Response)

Die folgende Kette verdeutlicht den Prozess:

Beginn der Abfrage→(Übertragungszeit der Abfrage)→Abfrage vom Slave empfangen→(Verarbeitungszeit des Slaves)→Beginn der Antwort→(Übertragungszeit der Antwort)→Antwort vollständig empfangen

3.5 Fehlerprüfungsverfahren (Error Checking Methods)

Standardmäßige serielle Modbus-Netzwerke verwenden zwei Verfahren zur Fehlerprüfung: Die oben genannten Fehlerprüf-Bytes (CRC) prüfen die Integrität der gesamten Nachricht, während die Paritätsprüfung (gerade oder ungerade) auf jedes einzelne Byte in der Nachricht angewendet werden kann.

3.5.1 Paritätsprüfung (Parity Checking)

Wenn die Paritätsprüfung aktiviert ist – durch Auswahl von entweder „Even“ (gerade) oder „Odd“ (ungerade) Parität – wird die Anzahl der „1“-Bits im Datenteil jedes Übertragungszeichens gezählt. Das Paritätsbit wird dann auf 0 oder 1 gesetzt, um eine gerade oder ungerade Gesamtsumme an „1“-Bits zu erhalten.

Hinweis: Die Paritätsprüfung kann einen Fehler nur dann erkennen, wenn eine ungerade Anzahl von Bits während der Übertragung verfälscht wird (hinzugefügt oder verloren geht). Wenn beispielsweise zwei „1“-Bits zu „0“-Bits korrumpiert werden, findet die Paritätsprüfung den Fehler nicht. Falls „No Parity“ (keine Parität) spezifiziert ist

3.5.2 CRC-Prüfung (Cyclical Redundancy Check)

Die Fehlerprüf-Bytes der Modbus-Nachrichten enthalten einen CRC-Wert, der zur Überprüfung des Inhalts der gesamten Nachricht dient. Die Fehlerprüf-Bytes müssen immer vorhanden sein, um dem Modbus-Protokoll zu entsprechen; es gibt keine Option, diese zu deaktivieren.

Der CRC-Wert stellt einen 16-Bit-Binärwert dar, der vom sendenden Gerät berechnet wird. Das empfangende Gerät muss den CRC während des Empfangs der Nachricht neu berechnen und den berechneten Wert, mit dem in den Fehlerprüf-Bytes empfangenen Wert vergleichen. Wenn die beiden Werte nicht gleich sind, sollte die Nachricht verworfen werden.

Der Rechenvorgang:

1. Die Berechnung startet mit dem Vorladen eines 16-Bit-Registers mit lauter „1“-en (Hex FFFF).
2. Jedes aufeinanderfolgende 8-Bit-Byte der Nachricht wird auf den aktuellen Inhalt des Registers angewendet.
3. Wichtig: Nur die acht Datenbits jedes Zeichens werden für die CRC-Erzeugung verwendet. Start-Bits, Stopp-Bits und das Paritätsbit (falls vorhanden) fließen nicht in die Fehlerprüf-Bytes ein.

Der Algorithmus im Detail: Während der Erzeugung wird jedes 8-Bit-Byte der Nachricht mit der unteren Hälfte des 16-Bit-Registers Exklusiv-ODER (XOR) verknüpft. Das Register wird dann achtmal in Richtung des niederwertigsten Bits (LSB) verschoben, wobei das höchstwertigste Bit (MSB) mit einer Null aufgefüllt wird. Nach jedem Schiebevorgang wird das LSB (vor dem Verschieben) geprüft:

- Wenn das LSB eine 1 war, wird das Register mit einem fest vordefinierten Wert (Polynom) per XOR verknüpft.
- Wenn das LSB eine 0 war, findet keine XOR-Verknüpfung statt.

Dieser Vorgang wird wiederholt, bis alle acht Schiebevorgänge durchgeführt wurden. Danach wird das nächste 8-Bit-Byte der Nachricht mit der unteren Hälfte des Registers verknüpft und der Prozess wiederholt sich. Der endgültige Inhalt des Registers, nachdem alle Bytes der Nachricht verarbeitet wurden, ist der Fehlerprüfwert (CRC).

Pseudo-Code für die CRC-Berechnung

START

1. Fehler-Wort (Error Word) = Hex (FFFF)

    2. FÜR JEDES Byte in der Nachricht:

        3. Fehler-Wort = Fehler-Wort XOR (aktuelles Byte der Nachricht)
        4. FÜR JEDEN Bit im Byte (8 Durchläufe):
        5. LSB (niederwertigstes Bit) = Fehler-Wort AND Hex (0001)
        6. Fehler-Wort = Fehler-Wort um 1 Bit nach rechts verschieben *(Im Code dargestellt als: Wenn LSB=1, dann subtrahiere 1; danach teile durch 2)*
        7. WENN LSB = 1 DANN: Fehler-Wort = Fehler-Wort XOR Hex (A001)
        8. NÄCHSTES Bit

    3. NÄCHSTES Byte in der Nachricht ENDE

3.6 Funktionscodes

Der Funktionscode-Teil einer Modbus-Protokoll-Nachricht definiert die Aktion, die vom Slave ausgeführt werden soll. Das Digitalmessgerät 236-9293 unterstützt die folgenden Funktionscodes:

Code	MODBUS Protocol name	Description
03	Read Holding Registers	Read the contents of read/write location (4X references)
04	Read Input Registers	Read the contents of read only location (3X references)
08	Diagnostics	Only sub-function zero is supported. This returns the data element of the query unchanged.
15	Pre-set Multiple Registers	Set the contents of read/write location (4X references)

3.7 Das IEEE-Gleitkommaformat (Aufbau)

Das Modbus-Protokoll definiert 16-Bit-„Register“ für Datenvariablen. Eine 16-Bit-Zahl wäre jedoch zu restriktiv – zum Beispiel für Energieparameter –, da der maximale Bereich einer 16-Bit-Zahl bei 65.535 liegt.

Um diese Einschränkung zu überwinden, wurden verschiedene Ansätze entwickelt. Die 236-9293 Digitalmessgeräte verwenden zwei aufeinanderfolgende Register, um eine Gleitkommazahl (Floating Point Number) darzustellen. Dies erweitert den Bereich effektiv auf +/- 1 x 10 . Die von den Messgeräten erzeugten Werte können direkt verwendet werden, ohne dass eine Skalierung erforderlich ist. So ist die Einheit für Spannungsparameter beispielsweise Volt, für Leistungsparameter Watt usw.

Was ist eine Gleitkommazahl?

Eine Gleitkommazahl besteht aus zwei Teilen: einer Mantisse und einem Exponenten. Sie wird in der Form 1,234 x 10^5 geschrieben. Bei der Mantisse (in diesem Beispiel 1,234) muss das Dezimalkomma um die Anzahl an Stellen nach rechts verschoben werden, die durch den Exponenten vorgegeben ist (in diesem Beispiel 5 Stellen), also 1,234 x 10^5 = 123.400. Wenn der Exponent negativ ist, wird das Komma nach links verschoben.

Was ist eine Gleitkommazahl im IEEE-754-Format?

Eine IEEE-754-Gleitkommazahl ist das binäre Äquivalent zur oben gezeigten Dezimal-Gleitkommazahl. Der Hauptunterschied besteht darin, dass das höchstwertige Bit der Mantisse immer so angeordnet wird, dass es eine „1“ ist. Dadurch muss es in der Darstellung der Zahl nicht explizit gespeichert werden.

Dieser Prozess, bei dem das höchstwertige Bit als 1 festgelegt wird, nennt man Normalisierung. Die Mantisse wird daher als „normale Mantisse“ bezeichnet. Während der Normalisierung werden die Bits in der Mantisse so lange nach links verschoben, bis das höchstwertige Bit eine Eins ist, während gleichzeitig der Exponent entsprechend verringert wird. Im speziellen Fall, wenn die Zahl Null ist, sind sowohl die Mantisse als auch der Exponent Null.

Die Bits in einem IEEE-754-Format haben die folgende Bedeutung:

Data Hi Reg, Hi Byte.	Data Hi Reg, Lo Byte.	Data Lo Reg, Hi Byte.	Data Lo Reg, Lo Byte.
SEEE EEEE	EMMM MMMM	MMMM MMMM	MMMM MMMM

Wobei:

- S das Vorzeichen-Bit (Sign Bit) darstellt, wobei 1 negativ und 0 positiv ist.
- E der 8-Bit-Exponent mit einem Offset (Versatz) von 127 ist, d. h. ein Exponent von Null wird durch 127 dargestellt, ein Exponent von 1 durch 128 usw.
- M die normale 23-Bit-Mantisse ist. Das 24. Bit ist immer 1 und wird daher nicht gespeichert.

Unter Verwendung des oben genannten Formats wird die Gleitkommazahl 240.5 als 43708000 hex dargestellt:

Data Hi Reg, Hi Byte.	Data Hi Reg, Lo Byte.	Data Lo Reg, Hi Byte.	Data Lo Reg, Lo Byte.
43	70	80	00

Das folgende Beispiel demonstriert, wie IEEE-754-Gleitkommazahlen aus ihrer hexadezimalen Form in die dezimale Form umgewandelt werden. Für dieses Beispiel verwenden wir den oben gezeigten Wert für 240,5. Beachten Sie, dass das Speicherformat für Gleitkommazahlen kein intuitives Format ist. Um diesen Wert in eine Dezimalzahl umzuwandeln, müssen die Bits gemäß der oben gezeigten Tabelle für das Speicherformat von Gleitkommazahlen getrennt werden.
Zum Beispiel:

Data Hi Reg, Hi Byte.	Data Hi Reg, Lo Byte.	Data Lo Reg, Hi Byte.	Data Lo Reg, Lo Byte.
0100 0011	0111 0000	1000 0000	0000 0000

Daraus lassen sich die folgenden Informationen ableiten:

- Das Vorzeichen-Bit ist 0, was auf eine positive Zahl hinweist.
- Der Exponenten-Wert ist binär 10000110 oder dezimal 134. Zieht man 127 von 134 ab, bleibt 7 übrig – dies ist der tatsächliche Exponent.
- Die Mantisse erscheint als die Binärzahl 11100001000000000000000.

Links von der Mantisse befindet sich ein implizierter Binärpunkt, dem immer eine 1 vorangestellt ist. Dieses Bit wird nicht in der hexadezimalen Darstellung der Gleitkommazahl gespeichert. Fügt man die 1 und den Binärpunkt am Anfang der Mantisse hinzu, ergibt sich: 1.11100001000000000000000

Nun passen wir die Mantisse an den Exponenten an. Ein negativer Exponent verschiebt den Binärpunkt nach links, ein positiver Exponent verschiebt ihn nach rechts. Da der Exponent 7 ist, wird die Mantisse wie folgt angepasst: 11110000.1000000000000000

Schließlich haben wir eine binäre Gleitkommazahl. Binär-Bits links vom Punkt stellen die Zweierpotenz entsprechend ihrer Position dar. Zum Beispiel repräsentiert 11110000:

(1 x 27) + (1 x 26) + (1 x 25) + (1 x 24) + (0 x 23) + (0 x 22) + (0 x 21) + (0 x 20) = 240.

Binär-Bits rechts vom Punkt repräsentieren ebenfalls eine Zweierpotenz entsprechend ihrer Position. Da die Ziffern rechts vom Punkt stehen, sind die Potenzen negativ. Zum Beispiel repräsentiert .100:

(1 x 2-1} + (0 x 2-2) + (0 x 2-3) was gleich 0,5 ist.

Addiert man diese beiden Zahlen zusammen und berücksichtigt das Vorzeichen-Bit, ergibt sich die Zahl +240,5.

Für jeden angeforderten Gleitkommawert müssen zwei Modbus-Protokoll-Register (vier Bytes) angefordert werden. Die Empfangsreihenfolge und Bedeutung dieser vier Bytes für die 236-9293 Digitalmessgeräte ist unten dargestellt:

Data Hi Reg, Hi Byte.	Data Hi Reg, Lo Byte.	Data Lo Reg, Hi Byte.	Data Lo Reg, Lo Byte.

3.8 Unterstützte MODBUS-Protokollbefehle

Die Digitalmessgeräte der Serie 236-9293 unterstützen die Befehle „Read Input Register“ (3X-Register), „Read Holding Register“ (4X-Register) und „Pre-set Multiple Registers“ (Schreiben von 4X-Registern) des Modbus-RTU-Protokolls. Alle gespeicherten und zurückgegebenen Werte liegen im Gleitkommaformat nach IEEE 754 vor, wobei das höchstwertige Register zuerst (Most Significant Register First) übertragen wird.rst.

3.8.1 Read Input Registers (Eingangsregister lesen)

Der Modbus-Protokollcode 04 liest den Inhalt der 3X-Register

Beispiel: Abfrage von „Spannung 1“ (Volts 1)

Die folgende Abfrage fordert den Messwert für „Spannung 1“ von einem Gerät mit der Knotenadresse (Slave Address) 1 an:

Field Name	Example (Hex)
Slave Address	01
Function	04
Starting Address High	00
Starting Address Low	00
Number of Points High	00
Number of Points Low	02
Error Check Low	71
Error Check High	CB

Hinweis: Daten müssen zwingend in Registerpaaren angefordert werden. Das bedeutet, sowohl die „Startadresse“ als auch die „Anzahl der Punkte“ (Number of Points) müssen gerade Zahlen sein, um eine Gleitkomma-Variable korrekt abzufragen. Wenn die „Startadresse“ oder die „Anzahl der Punkte“ ungerade ist, würde die Abfrage mitten in eine Gleitkomma-Variable fallen – das Gerät wird in diesem Fall eine Fehlermeldung zurückgeben. Die folgende Antwort gibt den Inhalt von „Volts 1“ mit dem Wert 230,2 zurück (beachten Sie jedoch auch den späteren Abschnitt „Ausnahmeantwort“).

Field Name	Example (Hex)
Slave Address	01
Function	04
Byte Count	04
Data, High Reg, High Byte	43
Data, High Reg, Low Byte	66
Data, Low Reg, High Byte	33
Data, Low Reg, Low Byte	34
Error Check Low	1B
Error Check High	38

3.9 Holding-Register (Holding Registers)

Während die 3X-Register (Input Registers) meist für flüchtige Messwerte genutzt werden, dienen die 4X-Register (Holding Registers) in der Regel der Konfiguration oder der Abfrage von Parametern, die im Gerät gespeichert sind.

3.9.1 Read Holding Registers (Halte-Register lesen)

Der MODBUS-Protokollcode **03** liest den Inhalt der 4X-Register

Field Name	Example (Hex)
Slave Address	01
Function	03
Starting Address High	00
Starting Address Low	00
Number of Points High	00
Number of Points Low	02
Error Check Low	C4
Error Check High	0B

Hinweis: Daten müssen zwingend in Registerpaaren angefordert werden. Das bedeutet, sowohl die „Startadresse“ als auch die „Anzahl der Punkte“ müssen gerade Zahlen sein, um eine Gleitkomma-Variable abzufragen. Wenn die „Startadresse“ oder die „Anzahl der Punkte“ ungerade ist, würde die Abfrage mitten in eine Gleitkomma-Variable fallen – das Gerät wird in diesem Fall eine Fehlermeldung zurückgeben. Die folgende Antwort gibt den Inhalt der „Demand Time“ (Bedarfszeitraum) mit dem Wert 1 zurück (beachten Sie jedoch auch den späteren Abschnitt „Ausnahmeantwort“).

Field Name	Example (Hex)
Slave Address	01
Function	03
Byte Count	04
Data, High Reg, High Byte	3F
Data, High Reg, Low Byte	80
Data, Low Reg, High Byte	00
Data, Low Reg, Low Byte	00
Error Check Low	F7
Error Check High	CF

3.9.2 Halteregister schreiben (Write Holding Registers)

Der MODBUS-Protokollcode 10 (16 dezimal) schreibt die Inhalte der 4X-Register.

Beispiel Die folgende Abfrage setzt den Anforderungszeitraum (Demand Period) auf 60, was die Anforderungszeit (Demand Time) effektiv zurücksetzt:

Field Name	Example (Hex)
Slave Address	01
Function	10
Starting Address High	00
Starting Address Low	02
Number of Registers High	00
Number of Registers Low	02
Byte Count	04
Data, High Reg, High Byte	42
Data, High Reg, Low Byte	70
Data, Low Reg, High Byte	00
Data, Low Reg, Low Byte	00
Error Check Low	67
Error Check High	D5

Hinweis: Daten müssen in Registerpaaren geschrieben werden. Sowohl die „Startadresse“ als auch die „Anzahl der Punkte“ müssen gerade Zahlen sein, um eine Gleitkommavariable zu schreiben. Wenn die „Startadresse“ oder die „Anzahl der Punkte“ ungerade ist, liegt die Abfrage mitten in einer Gleitkommavariable. Das Produkt gibt dann eine Fehlermeldung zurück. Im Allgemeinen kann nur ein Gleitkommawert pro Abfrage geschrieben werden. Die folgende Antwort zeigt an, dass das Schreiben erfolgreich war. Siehe aber auch „Ausnahmeantwort“ weiter unten.

Field Name	Example (Hex)
Slave Address	01
Function	10
Starting Address High	00
Starting Address Low	02
Number of Registers High	00
Number of Registers Low	02
Error Check Low	E0
Error Check High	08

3.10 Ausnahmeantwort (Exception Response)

Falls der Slave im obigen Beispiel „Halte-Register schreiben“ die angeforderte Funktion nicht unterstützen würde, hätte er mit einer Ausnahmeantwort geantwortet, wie unten dargestellt. Der Ausnahme-Funktionscode ist der ursprüngliche Funktionscode der Abfrage, bei dem das höchstwertige Bit (MSB) gesetzt wurde – das heißt, es wurde logisch mit 80 Hex per ODER verknüpft. Der Ausnahmecode (Exception Code) gibt den Grund für die Ausnahme an. Wichtiger Hinweis zum Schweigen des Slaves: Der Slave antwortet überhaupt nicht, wenn ein Fehler in der Parität oder im CRC der Abfrage vorliegt (da die Nachricht dann als korrupt gilt). Wenn der Slave die Abfrage jedoch empfängt, aber nicht verarbeiten kann (z. B. falsche Adresse oder nicht unterstützter Befehl), antwortet er mit einer Ausnahme. In diesem speziellen Fall bedeutet ein Code 01, dass die angeforderte Funktion von diesem Slave nicht unterstützt wird.

Field Name	Example (Hex)
Slave Address	01
Function	10 OR 80 = 90
Exception Code	01
Error Check Low	8D
Error Check High	C0

3.11 Ausnahmecodes (Exception Codes)

3.11.1 Tabelle der Ausnahmecodes

Das Digitalmessgerät unterstützt die folgenden Ausnahmecodes, um den Grund eines Fehlers präzise zu benennen:

Exception Code	MODBUS Protocol name	Description
01	Illegal Function	The function code is not supported by the product
02	Illegal Data	Address Attempt to access an invalid address or an attempt to read or write part of a floating point value
03	Illegal Data	Value Attempt to set a floating point variable to an invalid value
05	Slave Device	Failure An error occurred when the instrument attempted to store an update to it’s configuration

3.12 Diagnose (Diagnostics)

Der MODBUS-Protokollcode 08 bietet eine Reihe von diagnostischen Unterfunktionen. Bei den Digitalmessgeräten der Serie 236-9293 wird ausschließlich die Unterfunktion „Abfragedaten zurückgeben“ (Sub-function 0, auch bekannt als Loopback-Test) unterstützt.

Diese Funktion dient dazu, die Kommunikation zwischen Master und Slave zu testen. Der Slave nimmt die empfangenen Daten und sendet sie exakt so wieder zurück.

Field Name	Example (Hex)
Slave Address	01
Function	08
Sub-Function High	00
Sub-Function Low	00
Data Byte 1	AA
Data Byte 2	55
Error Check Low	5E
Error Check High	94

Hinweis: Mit dieser Funktion muss genau ein Datenregister (zwei Bytes) gesendet werden. Die folgende Antwort stellt die korrekte Erwiderung auf die Abfrage dar – das heißt, es werden exakt dieselben Bytes wie in der Abfrage zurückgegeben:

Field Name	Example (Hex)
Slave Address	01
Function	08
Sub-Function High	00
Sub-Function Low	00
Data Byte 1	AA
Data Byte 2	55
Error Check Low	5E
Error Check High	94