



Handbuch

---

# Protokolle

---

Modbus

---



Alle in diesem Handbuch genannten HIMA Produkte sind mit dem Warenzeichen geschützt. Dies gilt ebenfalls, soweit nicht anders vermerkt, für weitere genannte Hersteller und deren Produkte.

HIQuad®, HIQuad®X, HIMax®, HIMatrix®, SILworX®, XMR®, HICore® und FlexSILon® sind eingetragene Warenzeichen der HIMA Paul Hildebrandt GmbH.

Alle technischen Angaben und Hinweise in diesem Handbuch wurden mit größter Sorgfalt erarbeitet und unter Einschaltung wirksamer Kontrollmaßnahmen zusammengestellt. Bei Fragen bitte direkt an HIMA wenden. Für Anregungen, z. B. welche Informationen noch in das Handbuch aufgenommen werden sollen, ist HIMA dankbar.

Technische Änderungen vorbehalten. Ferner behält sich HIMA vor, Aktualisierungen des schriftlichen Materials ohne vorherige Ankündigungen vorzunehmen.

Alle aktuellen Handbücher können über die E-Mail-Adresse [documentation@hima.com](mailto:documentation@hima.com) angefragt werden.

© Copyright 2019, HIMA Paul Hildebrandt GmbH

Alle Rechte vorbehalten.

## Kontakt

HIMA Paul Hildebrandt GmbH

Postfach 1261

68777 Brühl

Tel.: +49 6202 709-0

Fax: +49 6202 709-107

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

Revisions- index	Änderungen	Art der Änderung	
		technisch	redaktionell
11.00	Neu erstellt für SILworX V11		

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Aufbau und Gebrauch des Handbuchs	5
1.2	Zielgruppe	6
1.3	Darstellungskonventionen	7
1.3.1	Sicherheitshinweise	7
1.3.2	Gebrauchshinweise	8
1.4	Safety Lifecycle Services	9
<b>2</b>	<b>Sicherheit</b>	<b>10</b>
2.1	Bestimmungsgemäßer Einsatz	10
2.2	Restrisiken	10
2.3	Sicherheitsvorkehrungen	10
2.4	Notfallinformationen	10
2.5	Automation Security bei HIMA Systemen	10
<b>3</b>	<b>Modbus</b>	<b>12</b>
<b>4</b>	<b>Modbus Master</b>	<b>13</b>
4.1	Modbus Beispiel	14
4.1.1	Konfiguration des Modbus TCP Slave	15
4.1.2	Konfiguration des Modbus TCP Master	16
4.2	Beispiel zur Alternativen Register/Bit-Adressierung	19
4.3	Menüfunktionen des Modbus Master	20
4.3.1	Edit	20
4.3.2	Eigenschaften	20
4.4	Modbus Funktionscodes des Masters	23
4.4.1	Modbus Standard Funktionscodes	23
4.4.2	HIMA spezifische Funktionscodes	24
4.4.3	Format der Request und Response Header	25
4.4.4	Anforderungstelegramme zum Lesen	26
4.4.5	Anforderungstelegramm zum Lesen und Schreiben	27
4.4.6	Anforderungstelegramm zum Schreiben	27
4.5	Ethernet Slaves (TCP/UDP-Slaves)	29
4.5.1	Systemvariablen der TCP/UDP-Slaves	30
4.5.2	Eigenschaften TCP/UDP-Slaves	31
4.6	Modbus Gateway (TCP/UDP Gateway)	32
4.6.1	Eigenschaften Gateway	34
4.6.2	Systemvariablen Gateway-Slave	34
4.6.3	Eigenschaften Gateway-Slave	34
4.7	Serieller Modbus	35
4.7.1	Eigenschaften Serieller Modbus	36
4.7.2	Systemvariablen Modbus Slave	37
4.7.3	Eigenschaften Modbus Slave	37
4.8	Control-Panel (Modbus Master)	38
4.8.1	Kontextmenü (Modbus Master)	38
4.8.2	Anzeigefeld (Modbus Master)	38
4.9	Control-Panel (Modbus Master->Slave)	38

<b>4.10</b>	<b>Funktion der FBx LED beim Modbus Master</b>	<b>39</b>
<b>4.11</b>	<b>Funktion der FAULT LED beim Modbus Master (Nur H1Max)</b>	<b>39</b>
<b>5</b>	<b>Modbus Slave</b>	<b>40</b>
<b>5.1</b>	<b>Benötigte Ausstattung und Systemanforderungen</b>	<b>40</b>
<b>5.2</b>	<b>Modbus Slave (Eigenschaften)</b>	<b>41</b>
<b>5.3</b>	<b>Konfiguration des Modbus TCP Slave</b>	<b>41</b>
<b>5.4</b>	<b>Konfiguration des redundanten Modbus TCP Slave</b>	<b>42</b>
<b>5.5</b>	<b>Regeln für den redundanten Modbus TCP Slave</b>	<b>43</b>
5.5.1	Prozessdatenmenge für den redundanten Modbus Slave	43
5.5.2	Zulässige Steckplätze der redundanten Modbus Slave H1Max COM-Module	43
5.5.3	Redundante Modbus Slave COM-Module in unterschiedlichen Basisträgern	43
<b>5.6</b>	<b>Menüfunktionen des Modbus Slave Set</b>	<b>44</b>
5.6.1	Eigenschaften Modbus Slave Set	44
5.6.2	Registervariablen	46
5.6.3	Bitvariablen	47
<b>5.7</b>	<b>Systemvariablen Modbus-Slave-Set</b>	<b>47</b>
5.7.1	Modbus Slave und Modbus Slave Redundant	48
<b>5.8</b>	<b>Modbus Funktionscodes des Modbus Slaves</b>	<b>51</b>
5.8.1	Modbus-Funktionscodes	51
<b>5.9</b>	<b>H1A spezifischen Funktionscodes</b>	<b>53</b>
5.9.1	Format der Request und Response Header	54
<b>5.10</b>	<b>Modbus Adressierung durch Bit und Register</b>	<b>55</b>
5.10.1	Register Bereich	55
5.10.2	Bit Bereich	57
<b>5.11</b>	<b>Offsets für alternative Modbus Adressierung</b>	<b>59</b>
5.11.1	Zugriff auf die Registervariablen im Bit Bereich des Modbus Slave	60
5.11.2	Zugriff auf die Bitvariablen im Registerbereich des Modbus Slave	62
<b>5.12</b>	<b>Control-Panel (Modbus Slave)</b>	<b>64</b>
5.12.1	Kontextmenü (Modbus Slave)	64
5.12.2	Anzeigefeld (Modbus Slave)	64
5.12.3	Anzeigefeld (Masterdaten)	64
<b>5.13</b>	<b>Funktion der FBx LED beim Modbus Slave</b>	<b>65</b>
<b>5.14</b>	<b>Funktion der FAULT LED beim Modbus Slave (Nur H1Max)</b>	<b>65</b>
<b>6</b>	<b>Allgemein</b>	<b>66</b>
<b>6.1</b>	<b>Maximale Kommunikationszeitscheibe</b>	<b>66</b>
6.1.1	Ermitteln der maximalen Dauer der Kommunikationszeitscheibe	66
<b>6.2</b>	<b>Lastbegrenzung</b>	<b>66</b>
	<b>Anhang</b>	<b>69</b>
	<b>Glossar</b>	<b>69</b>
	<b>Abbildungsverzeichnis</b>	<b>70</b>
	<b>Tabellenverzeichnis</b>	<b>70</b>

# 1 Einleitung

Das Modbus Handbuch beschreibt die Eigenschaften und die Konfiguration des Modbus Protokolls für die sicherheitsbezogenen HIMA Steuerungssysteme mit dem Programmierwerkzeug SILworX.

Die Kenntnis von Vorschriften und das technisch einwandfreie Umsetzen der in diesem Handbuch enthaltenen Hinweise durch qualifiziertes Personal sind Voraussetzung für die Planung, Projektierung, Programmierung, Installation, Inbetriebnahme, Betrieb und Instandhaltung der HIMA Steuerungen.

Bei nicht qualifizierten Eingriffen in die Geräte, bei Abschalten oder Umgehen (Bypass) von Sicherheitsfunktionen oder bei Nichtbeachtung von Hinweisen dieses Handbuchs (und dadurch verursachten Störungen oder Beeinträchtigungen von Sicherheitsfunktionen) können schwere Personen-, Sach- oder Umweltschäden eintreten, für die HIMA keine Haftung übernehmen kann.

HIMA Automatisierungsgeräte werden unter Beachtung der einschlägigen Sicherheitsnormen entwickelt, gefertigt und geprüft. Nur für die in den Beschreibungen vorgesehenen Einsatzfälle mit den spezifizierten Umgebungsbedingungen verwenden.

## 1.1 Aufbau und Gebrauch des Handbuchs

Das Handbuch enthält die folgenden Hauptkapitel:

- Einleitung
- Sicherheit
- Produktbeschreibung
- Modbus Master
- Modbus Slave
- Allgemein

Zusätzlich sind die folgenden Dokumente zu beachten:

Name	Inhalt	Dokumenten-Nr.
HIMax Systemhandbuch	Hardware-Beschreibung HIMax System	HI 801 000 D
HIMax Sicherheitshandbuch	Sicherheitsfunktionen HIMax Systems	HI 801 002 D
HIMatrix Sicherheitshandbuch	Sicherheitsfunktionen HIMatrix Systems	HI 800 022 D
HIMatrix Kompakt Systemhandbuch	Hardware-Beschreibung HIMatrix Kompakt System	HI 800 140 D
HIMatrix Modular Systemhandbuch	Hardware-Beschreibung HIMatrix Modular System F 60	HI 800 190 D
HIQuad X Systemhandbuch	Hardware-Beschreibung HIQuad X System	HI 803 210 D
HIQuad X Sicherheitshandbuch	Sicherheitsfunktionen HIQuad X System	HI 803 208 D
Automation Security Handbuch	Beschreibung von Automation Security Aspekten bei HIMA Systemen	HI 801 372 D
SILworX Erste Schritte	Einführung in SILworX	HI 801 102 D

Tabelle 1: Zusätzlich geltende Handbücher

Alle aktuellen Handbücher können über die E-Mail-Adresse [documentation@hima.com](mailto:documentation@hima.com) angefragt werden. Für registrierte Kunden stellt HIMA die Dokumentationen im Download-Bereich <https://www.hima.com/de/downloads/> zur Verfügung.

## 1.2 Zielgruppe

Dieses Dokument wendet sich an Planer, Projektoren, Programmierer und Personen, die zur Inbetriebnahme, zur Wartung und zum Betreiben von Automatisierungsanlagen berechtigt sind. Vorausgesetzt werden spezielle Kenntnisse auf dem Gebiet der sicherheitsbezogenen Automatisierungssysteme.

## 1.3 Darstellungskonventionen

Zur besseren Lesbarkeit und zur Verdeutlichung gelten in diesem Dokument folgende Schreibweisen:

<b>Fett</b>	Hervorhebung wichtiger Textteile. Bezeichnungen von Schaltflächen, Menüpunkten und Registern im Programmierwerkzeug, die angeklickt werden können.
<i>Kursiv</i>	Parameter und Systemvariablen, Referenzen.
<code>Courier</code>	Wörtliche Benutzereingaben.
<b>RUN</b>	Bezeichnungen von Betriebszuständen (Großbuchstaben).
Kap. 1.2.3	Querverweise sind Hyperlinks, auch wenn sie nicht besonders gekennzeichnet sind. Im elektronischen Dokument (PDF): Wird der Mauszeiger auf einen Hyperlink positioniert, verändert er seine Gestalt. Bei einem Klick springt das Dokument zur betreffenden Stelle.

Sicherheits- und Gebrauchshinweise sind besonders gekennzeichnet.

### 1.3.1 Sicherheitshinweise

Um ein möglichst geringes Risiko zu gewährleisten, sind die Sicherheitshinweise unbedingt zu befolgen.

Die Sicherheitshinweise im Dokument sind wie folgt dargestellt.

- Signalwort: Warnung, Vorsicht, Hinweis.
- Art und Quelle des Risikos.
- Folgen bei Nichtbeachtung.
- Vermeidung des Risikos.

Die Bedeutung der Signalworte ist:

- Warnung: Bei Missachtung droht schwere Körperverletzung bis Tod.
- Vorsicht: Bei Missachtung droht leichte Körperverletzung.
- Hinweis: Bei Missachtung droht Sachschaden.

#### **SIGNALWORT**



**Art und Quelle des Risikos!**  
**Folgen bei Nichtbeachtung.**  
**Vermeidung des Risikos.**

---

#### **HINWEIS**



**Art und Quelle des Schadens!**  
**Vermeidung des Schadens.**

---

### 1.3.2 Gebrauchshinweise

Zusatzinformationen sind nach folgendem Beispiel aufgebaut:

---

**i**

An dieser Stelle steht der Text der Zusatzinformation.

---

Nützliche Tipps und Tricks erscheinen in der Form:

---

**TIPP**

An dieser Stelle steht der Text des Tipps.

---



## 1.4 Safety Lifecycle Services

HIMA unterstützt Sie in allen Phasen des Sicherheitslebenszyklus der Anlage: Von der Planung, der Projektierung, über die Inbetriebnahme, bis zur Aufrechterhaltung der Sicherheit.

Für Informationen und Fragen zu unseren Produkten, zu Funktionaler Sicherheit und zu Automation Security stehen Ihnen die Experten des HIMA Support zur Verfügung.

Für die geforderte Qualifizierung gemäß Sicherheitsstandards, führt HIMA produkt- oder kundenspezifische Seminare in eigenen Trainingszentren, oder bei Ihnen vor Ort durch. Das aktuelle Seminarangebot zu Funktionaler Sicherheit, Automation Security und zu HIMA Produkten finden Sie auf der HIMA Webseite.

### Safety Lifecycle Services:

<b>Onsite+ / Vor-Ort-Engineering</b>	In enger Abstimmung mit Ihnen führt HIMA vor Ort Änderungen oder Erweiterungen durch.
<b>Startup+ / Vorbeugende Wartung</b>	HIMA ist verantwortlich für die Planung und Durchführung der vorbeugenden Wartung. Wartungsarbeiten erfolgen gemäß der Herstellervorgabe und werden für den Kunden dokumentiert.
<b>Lifecycle+ / Lifecycle-Management</b>	Im Rahmen des Lifecycle-Managements analysiert HIMA den aktuellen Status aller installierten Systeme und erstellt konkrete Empfehlungen zu Wartung, Upgrade und Migration.
<b>Hotline+ / 24-h-Hotline</b>	HIMA Sicherheitsingenieure stehen Ihnen für Problemlösung rund um die Uhr telefonisch zur Verfügung.
<b>Standby+ / 24-h-Rufbereitschaft</b>	Fehler, die nicht telefonisch gelöst werden können, werden von HIMA Spezialisten innerhalb vertraglich festgelegter Zeitfenster bearbeitet.
<b>Logistic+/ 24-h-Ersatzteilservice</b>	HIMA hält notwendige Ersatzteile vor und garantiert eine schnelle und langfristige Verfügbarkeit.

### Ansprechpartner:

<b>Safety Lifecycle Services</b>	<a href="https://www.hima.com/de/unternehmen/ansprechpartner-weltweit/">https://www.hima.com/de/unternehmen/ansprechpartner-weltweit/</a>
<b>Technischer Support</b>	<a href="https://www.hima.com/de/produkte-services/support/">https://www.hima.com/de/produkte-services/support/</a>
<b>Seminarangebot</b>	<a href="https://www.hima.com/de/produkte-services/seminarangebot/">https://www.hima.com/de/produkte-services/seminarangebot/</a>

## 2 Sicherheit

Sicherheitsinformationen, Hinweise und Anweisungen in diesem Dokument unbedingt lesen. Das Produkt nur unter Beachtung aller Richtlinien und Sicherheitsrichtlinien einsetzen.

Dieses Produkt wird mit SELV oder PELV betrieben. Vom Produkt selbst geht kein Risiko aus. Einsatz im Ex-Bereich nur mit zusätzlichen Maßnahmen erlaubt.

### 2.1 Bestimmungsgemäßer Einsatz

Für den Einsatz von HIMA Steuerungen, sind die jeweiligen Bedingungen einzuhalten, siehe zusätzlich geltende Handbücher, siehe Tabelle 1.

### 2.2 Restrisiken

Von einem HIMA System selbst geht kein Risiko aus.

Restrisiken können ausgehen von:

- Fehlern in der Projektierung.
- Fehlern im Anwenderprogramm.
- Fehlern in der Verdrahtung.

### 2.3 Sicherheitsvorkehrungen

Am Einsatzort geltende Sicherheitsbestimmungen beachten und vorgeschriebene Schutzausrüstung tragen.

### 2.4 Notfallinformationen

Ein HIMA System ist Teil der Sicherheitstechnik einer Anlage. Der Ausfall einer Steuerung bringt die Anlage in den sicheren Zustand.

Im Notfall ist jeder Eingriff, der die Sicherheitsfunktion des HIMA Systems verhindert, verboten.

### 2.5 Automation Security bei HIMA Systemen

Automation Security hat die Sicherheitsziele Vertraulichkeit, Integrität und Verfügbarkeit von Daten. In Bezug auf Automation Security muss von gezielten Angriffen ausgegangen werden. Insbesondere an Schnittstellen, wie sie in diesem Handbuch beschrieben werden, ist von möglichen Angriffen auszugehen.

#### **WARNUNG**



**Personenschaden durch unbefugte Manipulation an der Steuerung möglich!**

**Die Steuerung ist gegen unbefugte Zugriffe zu schützen!**

**Die für eine Anlage geeignete Umsetzung der benötigten Maßnahmen liegt in der Verantwortung des Anwenders!**

Sorgfältige Planung sollte die zu ergreifenden Maßnahmen nennen. Nach erfolgter Risikoanalyse sind die benötigten Maßnahmen zu ergreifen. Solche Maßnahmen sind beispielsweise:

- Sinnvolle Einteilung von Benutzergruppen.
- Gepflegte Netzwerklösungen helfen sicherzustellen, dass sichere Netzwerke dauerhaft von öffentlichen Netzwerken getrennt sind und, falls nötig, nur ein definierter Übergang (z. B. über eine Firewall oder eine DMZ) besteht.

- Verwendung geeigneter Passwörter.

Ein regelmäßiges Review (z. B. jährlich) der Security-Maßnahmen ist ratsam.

Weitere Einzelheiten siehe HIMA Automation Security Handbuch HI 801 372 D.

### 3 Modbus

Die Modbus Kopplung der HIMA Systeme kann an nahezu alle Prozessleit- und Visualisierungssysteme entweder über die RS485-Schnittstellen oder über die Ethernet-Schnittstellen der Steuerungen erfolgen. HIMA Systeme können sowohl als Master, als auch als Slave betrieben werden.

Die Modbusfunktionalität erleichtert vor allem die Anbindung von Bedienpanels oder anderen Steuerungen. Durch die intensive Verbreitung und den weltweiten Projekteinsatz ist Modbus vielfach praxiserprobt.

Modbus Master (siehe Kapitel 4)

Die Redundanz des Modbus Masters muss im Anwenderprogramm so konfiguriert werden, dass die redundanten Transportwege überwacht und die redundant übertragenen Prozessdaten dem jeweiligen Transportweg zugeordnet werden.

Modbus Slave (siehe Kapitel 5)

Der Modbus Slave kann redundant konfiguriert werden.

---

#### i

HIMA empfiehlt für ein neues Projekt, das Modbus Slave V2 Protokoll zu verwenden. Dieses beruht auf keiner neuen Modbus-Spezifikation, sondern ist eine erweiterte HIMA Variante, welche die interne Verarbeitung der Protokolldaten auf HIMA Steuerungen betrifft. Die Standard Modbus-Funktionscodes bleiben gleich, die HIMA spezifischen Modbus-Funktionscodes entfallen. Für weitere Informationen, siehe Modbus Slave V2 Handbuch HI 801 474 D.

---

## 4 Modbus Master

Die Datenübertragung zwischen dem Modbus Master und den Modbus Slaves kann über die serielle Schnittstelle (RS485) als auch über TCP/UDP (Ethernet) erfolgen. Zudem kann der Modbus Master auch als Gateway (Modbus von TCP/UDP -> RS485) eingesetzt werden.

### Benötigte Ausstattung und Systemanforderungen

Element	Beschreibung
HIMA Steuerung	HIMax mit X-COM 01 Modul HIQuad X mit F-COM 01 Modul HIMatrix
CPU-Modul	Die Ethernet-Schnittstellen des Prozessormoduls können für Modbus TCP nicht verwendet werden.
COM-Modul	Ethernet 10/100BaseT D-Sub Anschlüsse FB1 und FB2 Wird Modbus RTU verwendet, muss das COM-Modul an den verwendeten seriellen Feldbus-Schnittstellen (FB1 und/oder FB2) mit jeweils einem optionalen HIMA RS485 Submodul ausgerüstet sein. Schnittstellenbelegung, siehe Kommunikationshandbuch HI 801 100 D.
Aktivierung	Jede der beiden Modbus Master Funktionen muss einzeln frei geschaltet werden, siehe Kommunikationshandbuch HI 801 100 D. Modbus Master RTU (RS485) und Modbus Master TCP. Der Modbus Gateway benötigt die Modbus Master RTU Lizenz.

Tabelle 2: Ausstattung und Systemanforderung Modbus Master

### Modbus Master Eigenschaften

Eigenschaft	Beschreibung
Modbus Master	Es kann pro COM-Modul oder pro HIMatrix Steuerung ein Modbus Master konfiguriert werden. Der Modbus Master kann simultan <ul style="list-style-type: none"><li>– mit TCP/UDP-Slaves Daten austauschen.</li><li>– mit seriellen Slaves Daten austauschen.</li><li>– und als Gateway von Modbus TCP auf Modbus RTU dienen.</li></ul>
Max. Anzahl Modbus Slaves	Ein Modbus Master kann bis zu 247 Slaves bedienen. <ul style="list-style-type: none"><li>– 121 Modbus Slaves pro serielle Schnittstelle (FB1, FB2).</li><li>– 64 TCP Slaves über TCP/IP-Verbindung.</li><li>– 247 UDP Slaves über UDP/IP-Verbindung.</li></ul> Die maximale Anzahl UDP Slaves ist limitiert, da die Slaves auf der Master Seite verwaltet werden müssen.
Max. Anzahl Anforderungstelegramme	Es können bis zu 988 Anforderungstelegramme pro Modbus Master konfiguriert werden.
Max. Prozessdatenlänge pro Anforderungstelegramm	Die maximale Prozessdatenlänge ist abhängig vom Anforderungstelegrammen, siehe Kapitel 4.4.
Max. Größe der Sendedaten	<div><div>•</div><div>1</div><div>Die Statusbytes des Masters und die Statusbytes von jedem zugeordneten Slave müssen von der max. Größe der Sendedaten subtrahiert werden.</div></div>
Max. Größe der Empfangsdaten	
Siehe Kommunikationshandbuch HI 801 100 D.	

Eigenschaft	Beschreibung				
Darstellungsformat der Modbus-Daten	Die HIMA Steuerungen verwenden das Big Endian Format. Beispiel 32 Bit Daten (z. B. DWORD, DINT):				
	32 Bit Daten (hex)		0x12345678		
	Bytenummer (von links)	0	1	2	3
	Big Endian (HIMax, HIQuad X, HIMatrix)	12	34	56	78
	Middle Endian (H51q)	56	78	12	34
	Little Endian	78	56	34	12

Tabelle 3: Eigenschaften Modbus Master

Nach der Norm sind insgesamt drei Repeater zulässig, sodass maximal 121 Slaves pro serielle Schnittstelle eines Masters möglich sind.

## 4.1 Modbus Beispiel

In diesem Beispiel tauscht ein Modbus Master Daten mit einem Modbus Slave über Modbus TCP aus. Die beiden Steuerungen werden über die Ethernet-Schnittstellen der Kommunikationsmodule verbunden.

**i**

Befinden sich der Modbus Master und der Modbus Slave in verschiedenen Subnetzen, müssen in der Routing-Tabelle die entsprechenden benutzerdefinierten Routen eingetragen werden.

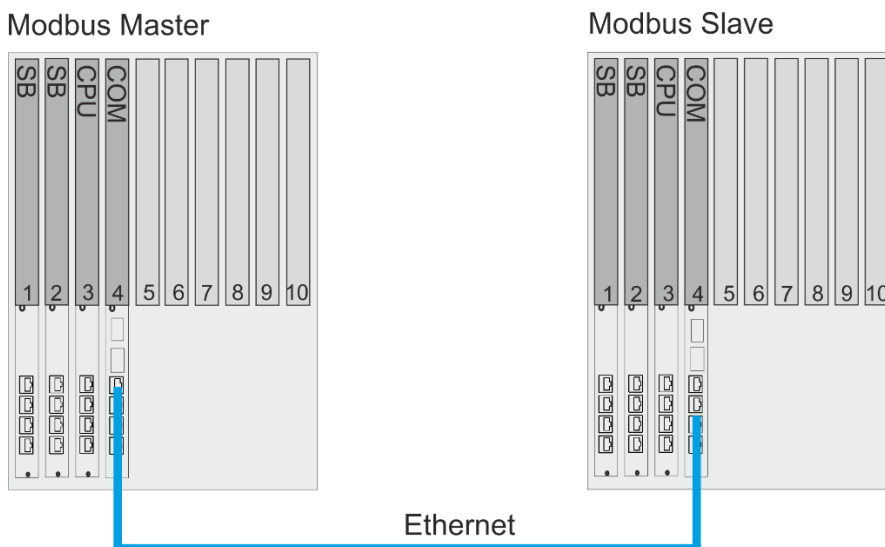


Bild 1: Kommunikation über Modbus TCP

Folgende Globale Variablen müssen für dieses Beispiel in SILworX angelegt werden:

Globale Variablen	Typ
Master->Slave_BOOL_00	BOOL
Master->Slave_BOOL_01	BOOL
Master->Slave_BOOL_02	BOOL
Master->Slave_WORD_00	WORD
Master->Slave_WORD_01	WORD
Slave->Master_WORD_00	WORD
Slave->Master_WORD_01	WORD

#### 4.1.1 Konfiguration des Modbus TCP Slave

##### Einen neuen Modbus Slave anlegen

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle** öffnen.
2. Im Kontextmenü von Protokolle **Neu, Modbus-Slave-Set** wählen, um einen neuen Modbus-Slave-Set hinzuzufügen.
3. Im Kontextmenü des Modbus-Slave-Set **Edit** wählen und **Eigenschaften Modbus-Slave-Set** öffnen, Standardwerte beibehalten.
4. Register **Modbus-Slave** wählen und folgende Einstellungen vornehmen:
  - **COM Modul** wählen
  - **TCP aktivieren** aktivieren
  - Die restlichen Parameter behalten die Standardwerte.

##### Bit Eingangsvariablen des Modbus Slave konfigurieren



Im Register Bitvariablen sind die boolschen Variablen einzutragen, die der Master Bitweise adressiert (Funktionscode 1, 2, 5, 15).

---

1. Im Kontextmenü des Modbus Slave **Edit, Bitvariablen** wählen.
2. In der **Objektauswahl** die folgenden Globalen Variablen auswählen und diese per Drag&Drop in den Bereich **Bit Eingänge** ziehen.

Bit Adresse	Bit Variable	Typ
0	Master->Slave_BOOL_00	BOOL
1	Master->Slave_BOOL_01	BOOL
2	Master->Slave_BOOL_02	BOOL

3. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Register Eingänge** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

### Register Eingangsvariablen des Modbus Slave konfigurieren

i

Im Register Registervariablen sind die Variablen einzutragen, die der Master Register-weise adressiert (Funktionscode 3, 4, 6, 16 ,23).

1. Im Kontextmenü des Modbus Slave **Edit, Registervariablen** wählen.
2. In der **Objektauswahl** die folgenden Variablen auswählen und diese per Drag&Drop in den Bereich **Register Eingänge** ziehen.

Register Adresse	Register Variablen	Typ
0	Master->Slave_WORD_00	WORD
1	Master->Slave_WORD_01	WORD

3. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Register Eingänge** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

### Register Ausgangsvariablen des Modbus Slave konfigurieren

1. Im Kontextmenü des Modbus Slave **Edit, Registervariablen** wählen.
2. In der **Objektauswahl** die folgenden Variablen auswählen und diese per Drag&Drop in den Bereich **Register Ausgänge** ziehen.

Register Adresse	Register Variablen	Typ
0	Slave->Master_WORD_00	WORD
1	Slave->Master_WORD_01	WORD

3. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Register Ausgänge** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

### Prüfen der Modbus TCP Slave Konfiguration

1. Kontextmenü des Modbus TCP Slave öffnen und **Verifikation** wählen.
2. Einträge im Logbuch sorgfältig überprüfen, gegebenenfalls korrigieren.

## 4.1.2 Konfiguration des Modbus TCP Master

### Modbus Master anlegen

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle** öffnen.
2. Im Kontextmenü von Protokolle **Neu, Modbus Master** wählen um einen neuen Modbus Master hinzuzufügen.
3. Im Kontextmenü vom Modbus Master **Eigenschaften, Allgemein** wählen.
4. **COM Modul** auswählen.  
Die restlichen Parameter behalten die Standardwerte.

### Im Modbus Master die Verbindung zu dem Modbus TCP Slave erstellen

1. Im Strukturbaum **Ressource, Protokolle, Modbus Master, Ethernet Slaves** öffnen.
2. Rechtsklick auf **Ethernet Slaves** und im Kontextmenü **Neu** wählen.
3. Aus der Liste **TCP/UDP-Slave** wählen und mit **OK** bestätigen.
4. Konfiguration des TCP/UDP-Slave im Modbus Master:
  - **Edit** zum Zuweisen der Systemvariablen wählen, siehe Kapitel 4.6.2.
  - **Eigenschaften** zum Konfigurieren der Eigenschaften wählen, siehe Kapitel 4.6.3.  
In den Eigenschaften des Slaves die **IP Adresse** des TCP/UDP-Slaves eintragen.

Die restlichen Parameter behalten die Standardwerte.



**Anforderungstelegramm zum Schreiben der Bit Ausgangsvariablen konfigurieren**

1. Rechtsklick auf **TCP/UDP-Slave** und im Kontextmenü **Neu** wählen.
2. Aus der Liste das Anforderungstelegramm **Write Multiple Coils (15)** wählen.
3. Rechtsklick auf das Anforderungstelegramm **Write Multiple Coils (15)** und im Kontextmenü **Eigenschaften** wählen.
  - **Startadresse des Schreibbereichs, 0** eintragen.
4. Rechtsklick auf das Anforderungstelegramm **Write Multiple Coils (15)** und im Kontextmenü **Edit** wählen.
5. In der **Objektauswahl** die folgenden Variablen auswählen und diese per Drag&Drop in das Register **Ausgangsvariablen** ziehen.

Offset	Bit Variablen	Typ
0	Master->Slave_BOOL_00	BOOL
1	Master->Slave_BOOL_01	BOOL
2	Master->Slave_BOOL_02	BOOL

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Ausgangsvariablen** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

**Anforderungstelegramm zum Schreiben der Register Ausgangsvariablen konfigurieren**

1. Rechtsklick auf **TCP/UDP-Slave** und im Kontextmenü **Neu** wählen.
2. Aus der Liste das Anforderungstelegramm **Write Multiple Registers (16)** wählen.
3. Rechtsklick auf das Anforderungstelegramm **Write Multiple Registers (16)** und im Kontextmenü **Eigenschaften** wählen.
  - **Startadresse des Schreibbereichs, 0** eintragen.
4. Rechtsklick auf das Anforderungstelegramm **Write Multiple Registers (16)** und im Kontextmenü **Edit** wählen.
5. In der **Objektauswahl** die folgenden Variablen auswählen und diese per Drag&Drop in das Register **Ausgangsvariablen** ziehen.

Offset	Register Variablen	Typ
0	Master->Slave_WORD_00	WORD
1	Master->Slave_WORD_01	WORD

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Ausgangsvariablen** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

**Im Modbus Master das Anforderungstelegramm zum lesen der Eingangsvariablen definieren**

1. Rechtsklick auf **TCP/UDP-Slave** und im Kontextmenü **Neu** wählen.
2. Aus der Liste das Anforderungstelegramm **Read Holding Registers (03)** wählen.
3. Rechtsklick auf das Anforderungstelegramm **Read Holding Registers (03)** und im Kontextmenü **Eigenschaften** wählen.
  - **Startadresse des Lesebereichs, 0** eintragen.
4. Rechtsklick auf das Anforderungstelegramm **Read Holding Registers (03)** und im Kontextmenü **Edit** wählen.
5. In der **Objektauswahl** die folgenden Variablen auswählen und diese per Drag&Drop in das Register **Eingangsvariablen** ziehen.

Offset	Register Variablen	Typ
0	Slave->Master_WORD_00	WORD
1	Slave->Master_WORD_01	WORD

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Eingangsvariablen** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

**Prüfen der Modbus TCP Master Konfiguration**

1. Kontextmenü des Modbus TCP Masters öffnen und **Verifikation** wählen.
2. Einträge im Logbuch sorgfältig überprüfen, gegebenenfalls korrigieren.

**Erstellen der Codes für die Steuerungen**

1. Code Generator der Master und der Slave Ressource starten.
2. Sicherstellen, dass die Codes fehlerfrei generiert wurden.
3. Jeweilige Codes in die Master und Slave Steuerungen laden.

## 4.2 Beispiel zur Alternativen Register/Bit-Adressierung

In diesem Beispiel wird die Konfiguration aus dem Kapitel 4.1 um 16 boolsche Variablen im Registerbereich erweitert. Die 16 boolsche Variablen werden mit dem Anforderungs-telegramm **Write Multiple Coils (15)** ausgelesen, siehe auch Kapitel 5.11.

### Konfiguration der Eingangsvariablen im Modbus Slave

1. Im Kontextmenü des Modbus Slave **Edit, Registervariablen** wählen.
2. In der **Objektauswahl** die 16 neuen boolsche Variablen auswählen und diese per Drag&Drop in den Bereich **Register Eingänge** ziehen.

Register Adresse	Register Variablen	Typ	
0	Master->Slave_WORD_00	WORD	
1	Master->Slave_WORD_01	WORD	
2	Master->Slave_BOOL_03 ..._18	BOOL	Neu hinzufügen

3. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Register Eingänge** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

### Im Modbus Slave die Alternative Register/Bit-Adressierung konfigurieren

1. Im Kontextmenü des Modbus Slave **Edit, Offsets** wählen und **Alternative Register/Bit-Adressierung verwenden** aktivieren.
2. Für dieses Beispiel die folgenden Offsets für die alternativen Bereiche verwenden:

Register Bereich Offset Bits Input	1000
Register Bereich Offset Bits Output	1000
Bit Bereich Offset Register Input	8000
Bit Bereich Offset Register Output	8000

### i

Um mit dem Modbus Anforderungstelegramm **Write Multiple Coils (15)** auf boolsche Variablen im Bereich **Register Variablen** zuzugreifen, müssen die Variablen in den Bereich **Bitvariablen** gespiegelt sein.

### Im Modbus Master das Anforderungstelegramm zum Schreiben der Ausgangsvariablen (BOOL) konfigurieren

1. Rechtsklick auf **TCP/UDP-Slave** und im Kontextmenü **Neu** wählen.
2. Aus der Liste das Anforderungstelegramm **Write Multiple Coils (15)** wählen.
3. Rechtsklick auf das Anforderungstelegramm **Write Multiple Coils (15)** und im Kontextmenü **Eigenschaften** wählen.
  - **Startadresse des Schreibbereichs, 8032** eintragen.
4. Rechtsklick auf das Anforderungstelegramm **Write Multiple Coils (15)** und im Kontextmenü **Edit** wählen.
5. In der **Objektauswahl** die folgenden Variablen auswählen und diese per Drag&Drop in das Register **Ausgangsvariablen** ziehen.

Offset	Gespiegelte Register Variable	Typ
0 bis 15	Master->Slave_BOOL_03..._18	BOOL

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Ausgangsvariablen** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

### 4.3 Menüfunktionen des Modbus Master

#### 4.3.1 Edit

Das Dialogfenster **Edit** des Modbus Master enthält das Register **Systemvariablen**.

Das Register **Systemvariablen** stellt Systemvariablen bereit, die es erlauben, den Zustand des Modbus Master im Anwenderprogramm auszuwerten und den Modbus Master zu steuern.

Element	Beschreibung
Anzahl fehlerhafte Slave-Verbindungen	Anzahl der fehlerhaften Verbindungen mit Modbus Slaves, die im Zustand Aktiviert sind. Deaktivierte Modbus Slaves werden hier nicht berücksichtigt.
Modbus-Master Aktivierungssteuerung	Hiermit kann der Modbus Master vom Anwenderprogramm gestoppt oder gestartet werden. 0: Aktivieren 1: Deaktivieren (Flankengetriggert! Modbus Master kann über PADT auch dann aktiviert werden wenn <i>Modbus-Master Aktivierungssteuerung</i> = 1.)
Modbus-Master Busfehler	Busfehler auf RS485, z. B. Telegrammfehler (unbekannte Codes etc), Längenfehler.
Modbus-Master Zustand	Der Modbus Master Zustand zeigt den momentanen Protokollzustand an: 1: OPERATE 0: OFFLINE
Reset aller Slave-Fehler	Mit einem Wechsel von FALSE->TRUE werden alle Slave-Fehler und Busfehler zurückgesetzt.

Tabelle 4: Systemvariablen Modbus Master

#### 4.3.2 Eigenschaften

Die Menüfunktion **Eigenschaften** aus dem Kontextmenü des Modbus Master öffnet den Dialog *Eigenschaften*.

Der Dialog enthält die Register **Allgemein** und **CPU/COM**.

##### 4.3.2.1 Allgemein

Im Register **Allgemein** werden der Name und die Beschreibung für den Modbus Master eingegeben. Zudem werden hier die Parameter eingestellt, wenn der Modbus Master zusätzlich als TCP und/oder UDP Gateway arbeiten soll.

Element	Beschreibung
Typ	Modbus Master.
Name	Name für den Modbus Master.
Beschreibung	Beschreibung für den Modbus Master.
Modul	Auswahl des COM-Moduls, auf dem dieses Protokoll abgearbeitet wird.
Max. µP-Budget aktivieren	Aktiviert: Limit des µP-Budget aus dem Feld <i>Max. µP-Budget in [%]</i> übernehmen.  Deaktiviert: Kein Limit des µP-Budget für dieses Protokoll verwenden.
Max. µP-Budget in [%]	Maximales µP-Budget des Moduls, welches bei der Abarbeitung des Protokolls produziert werden darf.  Wertebereich: 1 ... 100 % Standardwert: 30 %

Element	Beschreibung				
Verhalten bei CPU/COM Verbindungsverlust	<p>Bei Verbindungsverlust des Prozessormoduls zum Kommunikationsmodul werden in Abhängigkeit dieses Parameters die Eingangsvariablen entweder initialisiert oder unverändert im Prozessormodul verwendet. (z. B. wenn Kommunikationsmodul bei laufender Kommunikation gezogen wird).</p> <table> <tr> <td>Initialdaten annehmen</td><td>Eingangsvariablen werden auf die Initialwerte zurückgesetzt.</td></tr> <tr> <td>Letzten Wert beibehalten</td><td>Eingangsvariablen behalten den letzten Wert.</td></tr> </table>	Initialdaten annehmen	Eingangsvariablen werden auf die Initialwerte zurückgesetzt.	Letzten Wert beibehalten	Eingangsvariablen behalten den letzten Wert.
Initialdaten annehmen	Eingangsvariablen werden auf die Initialwerte zurückgesetzt.				
Letzten Wert beibehalten	Eingangsvariablen behalten den letzten Wert.				
TCP-Gateway aktivieren	Ist das TCP Modbus Gateway aktiviert, muss mindestens eine Modbus-RS485 Schnittstelle konfiguriert werden.				
TCP-Server-Port	<p>Standard: 502</p> <p>Es können auch andere TCP-Ports konfiguriert werden. Dabei ist die Port-Belegung bei der <i>Internet Corporation for Assigned Names and Numbers</i> (ICANN) zu beachten.</p>				
Maximale Anzahl TCP-Verbindungen als Server	<p>Maximale Anzahl gleichzeitig offener TCP-Verbindungen als Server.</p> <p>Wertebereich: 1 ... 64 Standardwert: 20</p>				
UDP-Gateway aktivieren	Ist das UDP Modbus Gateway aktiviert, muss mindestens eine Modbus-RS485 Schnittstelle konfiguriert werden.				
UDP-Port	<p>Standard: 502</p> <p>Es können auch andere UDP-Ports konfiguriert werden. Dabei ist die Port-Belegung bei der Internet Corporation for Assigned Names and Numbers (ICANN) zu beachten.</p>				
Maximale Länge der Queue	<p>Länge der Gateway-Warteschlange für noch nicht beantwortete Anforderungstelegramme von anderen Mastern. Dies wird nur beachtet, wenn ein Gateway aktiviert ist.</p> <p>Wertebereich: 1 ... 20 Standardwert: 3</p>				

Tabelle 5: Allgemeine Eigenschaften Modbus Master

#### 4.3.2.2 CPU/COM

Die Standardwerte für die Parameter sorgen für den schnellstmöglichen Datenaustausch der Modbus-Daten zwischen dem COM-Modul und dem CPU-Modul in der HIMA Steuerung.

Diese Parameter sollten nur dann geändert werden, wenn eine Reduzierung der COM- und/oder CPU-Auslastung für eine Anwendung erforderlich ist und der Prozess dies zulässt.

**i**

Die Änderung der Parameter wird nur dem erfahrenen Programmierer empfohlen.

Eine Erhöhung der COM und CPU Aktualisierungszeit bedeutet auch, dass die tatsächliche Aktualisierungszeit der Modbus-Daten erhöht wird. Die Zeitanforderungen der Anlage sind zu prüfen.

Element	Beschreibung
Aktualisierungsintervall der Prozessdaten [ms]	<p>Aktualisierungszeit in Millisekunden, mit der die Daten des Protokolls zwischen COM und CPU ausgetauscht werden.</p> <p>Ist das <i>Aktualisierungsintervall der Prozessdaten</i> Null oder kleiner als die Zykluszeit der Steuerung, dann erfolgt der Datenaustausch so schnell wie möglich.</p> <p>Wertebereich: 0 ... (<math>2^{31}-1</math>)</p> <p>Standardwert: 0</p>
Mehrere Fragmente pro Zyklus zulassen	<p>Aktiviert: Transfer der gesamten Daten des Protokolls von der CPU zur COM innerhalb eines Zyklus der CPU.</p> <p>Deaktiviert: Transfer der gesamten Daten des Protokolls von der CPU zur COM, verteilt über mehrere CPU Zyklen zu je 1100 Byte pro Datenrichtung.</p> <p>Damit kann eventuell auch die Zykluszeit der Steuerung reduziert werden.</p> <p>Standardwert: Aktiviert</p>

Tabelle 6: Parameter COM/CPU

## 4.4 Modbus Funktionscodes des Masters

Mit den Modbus Funktionscodes (Anforderungstelegrammen) können einzelne Variablen oder mehrere aufeinander folgende Variablen gelesen oder geschrieben werden.

### 4.4.1 Modbus Standard Funktionscodes

Folgende Modbus Standard Funktionscodes werden vom Modbus Master unterstützt.

Element	Code	Typ	Bedeutung
READ COILS	01	BOOL	Lesen mehrerer Variablen (BOOL) aus dem Slave. Max. Länge der Prozessdaten: 250 Bytes (2000 Coils).
READ DISCRETE INPUTS	02	BOOL	Lesen mehrerer Variablen (BOOL) aus dem Slave. Max. Länge der Prozessdaten: 250 Bytes (2000 Coils).
READ HOLDING REGISTERS	03	WORD	Lesen mehrerer Variablen beliebigen Typs aus dem Slave. Max. Länge der Prozessdaten: 250 Bytes.
READ INPUT REGISTERS	04	WORD	Lesen mehrerer Variablen beliebigen Typs aus dem Slave. Max. Länge der Prozessdaten: 250 Bytes.
WRITE SINGLE COIL	05	BOOL	Schreiben eines einzelnen Signals (BOOL) in den Slave. Max. Länge der Prozessdaten: 1 Byte (1 Coil)
WRITE SINGLE REGISTER	06	WORD	Schreiben eines einzelnen Signals (WORD) in den Slave. Max. Länge der Prozessdaten: 2 Bytes
WRITE MULTIPLE COILS	15	BOOL	Schreiben mehrerer Variablen (BOOL) in den Slave. Max. Länge der Prozessdaten: 246 Bytes (1968 Coils)
WRITE MULTIPLE REGISTERS	16	WORD	Schreiben mehrerer Variablen beliebigen Typs in den Slave. Max. Länge der Prozessdaten: 246 Bytes.
READ WRITE HOLDING REGISTERS	23	WORD	Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Slave. Max. Länge der Prozessdaten: 242 Bytes (Anforderungstelegramm des Modbus Masters) 250 Bytes (Antworttelegramm des Slaves).

Tabelle 7: Modbus-Funktionscodes



Weitere Informationen zu Modbus sind in der Spezifikation *Modbus Application Protocol Specification* [www.modbus.org](http://www.modbus.org) zu finden.

---

#### 4.4.2 HIMA spezifische Funktionscodes

Die HIMA spezifischen Funktionscodes entsprechen den Standard Modbus Funktionscodes. Die zwei Unterschiede sind die maximal zulässige Prozessdatenlänge von 1100 Bytes und das Format von Request und Response-Header.

Element	Code	Typ	Bedeutung
Read Coils Extended	100 (0x64)	BOOL	Entspricht dem Functioncode 01 Lesen mehrerer Variablen (BOOL) aus dem Import- oder Export <sup>1)</sup> -Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes.
Read Discrete Inputs Extended	101 (0x65)	BOOL	Entspricht dem Functioncode 02 Lesen mehrerer Variablen (BOOL) aus dem Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes.
Read Holding Registers Extended	102 (0x66)	WORD	Entspricht dem Functioncode 03 Lesen mehrerer Variablen beliebigen Typs aus dem Import- oder Export <sup>1)</sup> -Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes.
Read Input Registers Extended	103 (0x67)	WORD	Entspricht dem Functioncode 04 Lesen mehrerer Variablen beliebigen Typs aus dem Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes.
Write Multiple Coils Extended	104 (0x68)	BOOL	Entspricht dem Functioncode 15 Schreiben mehrerer Variablen (BOOL) in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes.
Write Multiple Registers Extended	105 (0x69)	WORD	Entspricht dem Functioncode 16 Schreiben mehrerer Variablen beliebigen Typs in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes.
Read/Write Multiple Registers Extended	106 (0x6A)	WORD	Entspricht dem Functioncode 23 Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Import-Bereich oder Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes (Anforderungstelegramm des Modbus Masters) 1100 Bytes (Antworttelegramm des Slaves).
<sup>1)</sup> Export Bereich kann nur bei HIMA Slaves gewählt werden.			

Tabelle 8: HIMA spezifische Funktionscodes



#### 4.4.3 Format der Request und Response Header

Die Request und Response Header der HIMA spezifischen Modbus-Funktionscodes sind wie folgt aufgebaut:

Code	Request	Response
100 (0x64)	1 Byte Funktionscode 0x64 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 ... 8800 (0x2260)	1 Byte Funktionscode 0x64 2 Bytes Anzahl von Bytes = N N Bytes Coil-Daten (8 Coils werden in ein Byte gepackt)
101 (0x65)	1 Byte Funktionscode 0x65 2 Bytes Startadresse 2 Bytes Anzahl von Discrete Inputs 1 ... 8800 (0x2260)	1 Byte Funktionscode 0x65 2 Bytes Anzahl von Bytes = N N Bytes Discrete Inputs -Daten (8 Discrete Inputs werden in ein Byte gepackt)
102 (0x66)	1 Byte Funktionscode 0x66 2 Bytes Startadresse 2 Bytes Anzahl von Register 1 ... 550 (0x226)	1 Byte Funktionscode 0x66 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten
103 (0x67)	1 Bytes Funktionscode 0x67 2 Bytes Startadresse 2 Bytes Anzahl von Register 1 ... 550 (0x226)	1 Byte Funktionscode 0x67 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten
104 (0x68)	1 Byte Funktionscode 0x68 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 ... 8800 (0x2260) 2 Bytes Anzahl von Bytes = N N Bytes Coil-Daten	1 Byte Funktionscode 0x68 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 ... 8800 (0x2260)
105 (0x69)	1 Byte Funktionscode 0x69 2 Bytes Startadresse 2 Bytes Anzahl von Registern 1 ... 550 (0x226) 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten	1 Byte Funktionscode 0x69 2 Bytes Startadresse 2 Bytes Anzahl von Registern 1 ... 550 (0x226)
106 (0x6A)	1 Byte Funktionscode 0x6a 2 Bytes Lese-Startadresse 2 Bytes Anzahl von Leseregistern 1 ... 550 (0x226) 2 Bytes Schreib-Startadresse 2 Bytes Anzahl von Schreibregistern 1 ... 550 (0x226) 2 Bytes Anzahl von Bytes zum Schreiben = N N Bytes Register-Daten	1 Byte Funktionscode 0x6a 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten
<sup>1)</sup> Export Bereich kann nur bei HIMA Slaves gewählt werden.		

Tabelle 9: Request und Response Header

#### 4.4.4 Anforderungstelegramme zum Lesen

Mit den Read-Funktionscodes können Variablen aus dem Slave gelesen werden.

Ein Anforderungstelegramm des Modbus Master enthält neben der Modbus Funktion die Startadresse des Lese-/Schreibbereichs.

Zum Lesen von Variablen sendet der Modbus Master ein *Anforderungstelegramm zum Lesen* an den Modbus Slave.

Der Modbus Slave sendet daraufhin ein Antworttelegramm mit den angeforderten Variablen an den Modbus Master zurück.

Die folgenden Anforderungstelegramme zum Lesen stehen zur Verfügung.

##### 4.4.4.1 Read Coils (01) und Extended (100)

Lesen mehrerer Variablen (BOOL) aus dem Slave.

Element	Bedeutung
Typ	Modbus-Funktion Read Coils.
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus –Funktion.
Startadresse des Lesebereichs	0 ... 65535

Tabelle 10: Anforderungstelegramm Read Coils

##### 4.4.4.2 Read Discrete Inputs (02) und Extended (101)

Lesen mehrerer Variablen (BOOL) aus dem Slave.

Element	Bedeutung
Typ	Modbus -Funktion Read Discrete Inputs.
Name	Beliebiger, eindeutiger Name, für die Modbus –Funktion.
Beschreibung	Beschreibung für die Modbus –Funktion.
Startadresse des Lesebereichs	0 ... 65535

Tabelle 11: Anforderungstelegramm Read Discrete Inputs

##### 4.4.4.3 Read Holding Registers (03) und Extended (102)

Lesen mehrerer Variablen beliebigen Typs aus dem Slave.

Element	Bedeutung
Typ	Modbus -Funktion Read Holding Registers.
Name	Beliebiger, eindeutiger Name, für die Modbus –Funktion.
Beschreibung	Beschreibung für die Modbus –Funktion.
Startadresse des Lesebereichs	0 ... 65535

Tabelle 12: Anforderungstelegramm Read Holding Registers

##### 4.4.4.4 Read Input Registers (04) und Extended (103)

Lesen mehrerer Variablen beliebigen Typs aus dem Slave

Element	Bedeutung
Typ	Modbus Funktion Read Input Registers.
Name	Beliebiger, eindeutiger Name, für die Modbus Funktion.
Beschreibung	Beschreibung für die Modbus Funktion.
Startadresse des Lesebereichs	0 ... 65535

Tabelle 13: Anforderungstelegramm Read Input Registers

#### 4.4.5 Anforderungstelegramm zum Lesen und Schreiben

Zum Lesen und Schreiben von Variablen sendet der Modbus Master ein *Anforderungstelegramm zum Lesen und Schreiben* an den Modbus Slave.

Zuerst schreibt der Modbus Master die definierten Schreibvariablen in den definierten Import-Bereich des Modbus Slave.

Anschließend liest der Modbus Master die definierten Lesevariablen aus dem definierten Export-Bereich des Modbus Slave.

Die folgenden Anforderungstelegramme zum Lesen und Schreiben stehen zur Verfügung

### i

Die Funktionen Schreiben und Lesen sind auch bei dem *Anforderungstelegramme zum Lesen und Schreiben* voneinander unabhängig, sie werden nur in einem gemeinsamen Anforderungstelegramm gesendet.

Eine häufige Anwendung für das *Anforderungstelegramme zum Lesen und Schreiben* ist jedoch, dass die geschriebenen Variablen des Modbus Master wieder zurückgelesen werden. Damit wird überprüft, ob die gesendeten Variablen korrekt geschrieben wurden.

#### 4.4.5.1 Read Write Holding Register (23) und Extended (106)

Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus-Funktion <i>Read Write Holding Registers</i> .
Name	Beliebiger, eindeutiger Name, für die Modbus-Funktion.
Beschreibung	Beschreibung für die Modbus-Funktion.
Startadresse des Lesebereichs	0 ... 65535
Startadresse des Schreibbereichs	0 ... 65535

Tabelle 14: Register Read Write Holding

#### 4.4.6 Anforderungstelegramm zum Schreiben

Mit den Write-Funktionscodes werden Variablen nur in den Importbereich eines Slaves geschrieben.

Ein Anforderungstelegramm des Modbus Master enthält neben der Modbus-Funktion die Startadresse des Lese-/Schreibbereichs.

Zum Schreiben von Variablen sendet der Modbus Master ein *Anforderungstelegramm zum Schreiben* an den Modbus Slave.

Der Modbus Slave schreibt die empfangenen Variablen in seinen Import-Bereich.

Im Dialog *Variablen zuweisen* eines *Anforderungstelegramms zum Schreiben* müssen die Variablen eingefügt werden, die der Modbus Master zum Modbus Slave schreibt.

Die folgenden Anforderungstelegramme zum Schreiben stehen zur Verfügung.

## 4.4.6.1 Write Multiple Coils (15) und Extended (104)

Schreiben mehrerer Variablen (BOOL) in den Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus Funktion <i>Write Multiple Coils</i> .
Name	Beliebiger, eindeutiger Name, für die Modbus Funktion.
Beschreibung	Beschreibung für die Modbus Funktion.
Startadresse des Schreibbereichs	0 ... 65535

Tabelle 15: Anforderungstelegramm Write Multiple Coils

## 4.4.6.2 Write Multiple Registers (16) und Extended (105)

Schreiben mehrerer Variablen beliebigen Typs in den Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus Funktion <i>Write Multiple Registers</i> .
Name	Beliebiger, eindeutiger Name, für die Modbus Funktion.
Beschreibung	Beschreibung für die Modbus Funktion.
Startadresse des Schreibbereichs	0 ... 65535.

Tabelle 16: Anforderungstelegramm Write Multiple Registers

## 4.4.6.3 Write Single Coil (05)

Schreiben einer einzelnen Variablen (BOOL) in den Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus Funktion <i>Write Single Coil</i> .
Name	Beliebiger, eindeutiger Name, für die Modbus Funktion.
Beschreibung	Beschreibung für die Modbus Funktion.
Startadresse des Schreibbereichs	0 ... 65535

Tabelle 17: Anforderungstelegramm Write Single Coil (05)

## 4.4.6.4 Write Single Register (06)

Schreiben einer einzelnen Variablen (WORD) in den Import-Bereich des Slaves.

Element	Bedeutung
Typ	Modbus Funktion <i>Write Single Register</i> .
Name	Beliebiger, eindeutiger Name, für die Modbus Funktion.
Beschreibung	Beschreibung für die Modbus Funktion.
Startadresse des Schreibbereichs	0 ... 65535

Tabelle 18: Anforderungstelegramm Write Single Register

## 4.5 Ethernet Slaves (TCP/UDP-Slaves)

Der Modbus Master kann mit bis zu 64 TCP/IP und 247 UDP/IP Slaves kommunizieren.

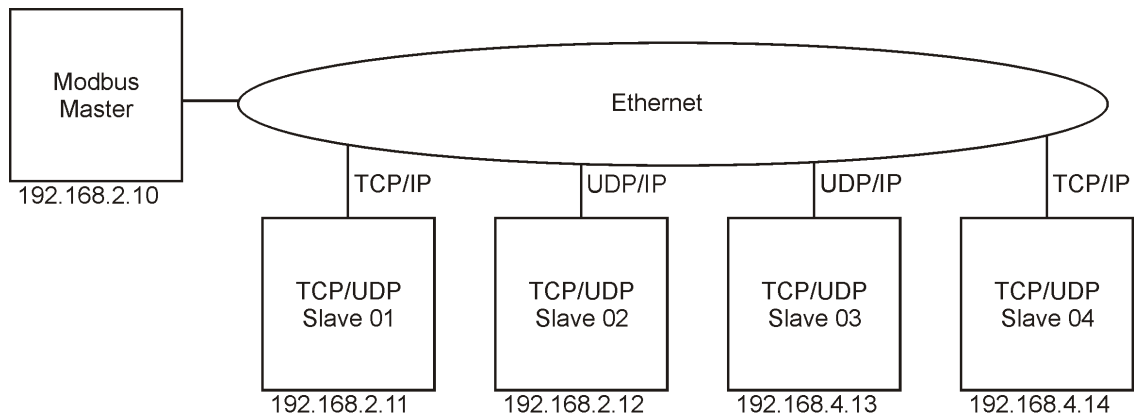


Bild 2: Modbus-Netzwerk

### Im Modbus Master eine neue Verbindung zu einem TCP/UDP Slave erstellen

1. Im Strukturbaum **Ressource, Protokolle, Modbus Master, Ethernet-Slaves** öffnen.
2. Rechtsklick auf **Ethernet-Slaves** und im Kontextmenü **Neu** wählen.
3. Aus der Liste **TCP/UDP-Slaves** wählen und mit **OK** bestätigen.
4. Konfiguration des TCP/UDP-Slave im Modbus Master::  
**Edit** zum Zuweisen der Systemvariablen, siehe Kapitel 4.5.1.  
**Eigenschaften** zum Konfigurieren der Eigenschaften, siehe Kapitel 4.5.2.

### i

Befinden sich die TCP/UDP-Slaves und der Modbus Master in verschiedenen Subnetzen, müssen in der Routing-Tabelle die entsprechenden benutzerdefinierten Routen eingetragen werden.

Der HIMA Modbus TCP Master sendet mit seinen Telegrammen an den Modbus TCP Slave zusätzlich zur IP-Adresse immer eine Modbus Slave Adresse (Unit Identifier) mit, diese ist immer FF (255).

#### 4.5.1 Systemvariablen der TCP/UDP-Slaves

Das Register *Systemvariablen* stellt Systemvariablen bereit, die es erlauben, den Zustand des TCP/UDP Slave im Anwenderprogramm auszuwerten und zu steuern.

Der Status des TCP/UDP Slave kann im Anwenderprogramm mit den folgenden Statusvariablen ausgewertet werden:

Element	Beschreibung
Modbus-Slave Aktivierungssteuerung	Hiermit kann der TCP/UDP Slave vom Anwenderprogramm deaktiviert oder aktiviert werden. 0: Aktivieren 1: Deaktivieren (Flankengetriggert! Modbus Slave kann über PADT auch dann aktiviert werden wenn <i>Modbus-Slave Aktivierungssteuerung</i> = 1.)
Modbus-Slave Fehler	Fehlercode Die Fehlercodes 0x01 ... 0x0b entsprechen den Exception Codes der Modbus-Protokollspezifikation. 0x00: Kein Fehler.  Exception Codes: 0x01: Ungültiger Funktionscode. 0x02: Ungültige Adressierung. 0x03: Ungültige Daten. 0x04: Nicht verwendet. 0x05: Nicht verwendet. 0x06: Device Busy (nur Gateway). 0x08: Nicht verwendet. 0x0a: Nicht verwendet. 0x0b: No Response from Slave (nur Gateway)  HIMA spezifische Codes: 0x10: Defekter Frame empfangen. 0x11: Frame mit falscher Transaktions ID empfangen. 0x12: Unerwartete Antwort empfangen. 0x13: Antwort über falsche Verbindung erhalten. 0x14: Falsche Antwort auf einen Schreib Auftrag.  0xff: Slave Timeout.
Modbus-Slave Zustand	Verbindungsstatus des TCP/UDP Slave: 0: Deaktiviert 1: Nicht verbunden. 2: Verbunden

Tabelle 19: Systemvariablen TCP/UDP-Slaves

#### 4.5.2 Eigenschaften TCP/UDP-Slaves

Zur Konfiguration der Verbindung zum TCP/UDP Slave müssen im Modbus Master die folgenden Parameter eingestellt werden.

Element	Beschreibung
Typ	TCP/UDP Slave.
Name	Beliebiger eindeutiger Name für den TCP/UDP Slave.
Beschreibung	Beliebige eindeutige Beschreibung für den TCP/UDP Slave.
Master-Slave Datenaustausch [ms]	Intervall für den Datenaustausch mit diesem Slave 1 bis ( $2^{31}-1$ ). Konnte der Slave nach <i>Maximale Anzahl Sendewiederholungen</i> nicht erreicht werden, wird das Intervall <i>Master-Slave Datenaustausch</i> um das Vierfache hochgesetzt.
TCP-Verbindung nur bei Bedarf	Wenn das Transportprotokoll TCP ist, wird hier eingestellt, ob die Verbindung zu diesem Slave nach jedem Datenaustausch automatisch abgebaut werden soll. TRUE: Die Verbindung abbauen. FALSE: Die Verbindung nicht abbauen. Standardwert: FALSE
Receive Timeout [ms]	Das Receive Timeout für diesen Slave [ms]. Nach dieser Zeit wird ein neuer Sendeversuch gestartet.
IP-Adresse	IP-Adresse des TCP/UDP Slave.
Port	Standard: 502 Es können auch andere TCP/UDP-Ports konfiguriert werden. Dabei ist die Port-Belegung bei der Internet Corporation for Assigned Names and Numbers (ICANN) zu beachten.
Kommunikationsart IP-Protokoll	TCP oder UDP. Standardwert: TCP
Maximale Anzahl Sendewiederholungen	Maximale Anzahl an Sendewiederholungen, falls Slave nicht antwortet. Die Anzahl der Sendewiederholungen kann beliebig eingestellt werden (0 ... 65535). Bei TCP/IP immer null, nicht änderbar. Empfohlen wird eine Anzahl von null bis acht Sendewiederholungen.

Tabelle 20: Konfigurationsparameter

## 4.6 Modbus Gateway (TCP/UDP Gateway)

Damit der Modbus Master als Modbus Gateway arbeiten kann wird die Modbus Master RTU Lizenz benötigt. In diesem Modus werden Master Requests, die das Gateway über Ethernet empfängt, an die an dem Gateway angeschlossenen RS485-Slaves weitergeleitet. Entsprechend werden die Antworten der Slaves über den Gateway an den Modbus Master weitergeleitet.

Bis zu 121 serielle Modbus Slaves können über die serielle Schnittstelle adressiert werden.

Der Bereich der Slave-Adressen ist 1 bis 247. Der Modbus Master 2 (Modbus Gateway) benötigt eine Modbus Master Lizenz, auch wenn nur der Modbus Gateway genutzt wird.

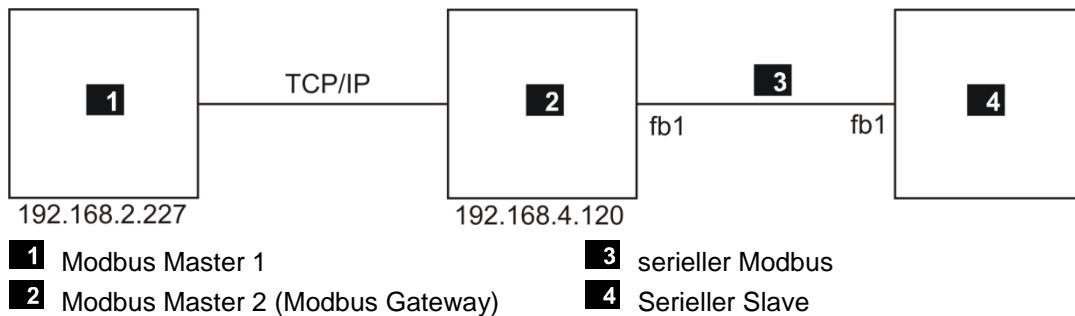


Bild 3: Modbus Gateway

**i**

Befinden sich der Modbus Gateway und der Modbus Master in verschiedenen Subnetzen, müssen in der Routing-Tabelle die entsprechenden benutzerdefinierten Routen eingetragen werden.

### Modbus Master 1

#### Im Modbus Master 1 die Verbindung zu dem Modbus Gateway erstellen

1. Im Strukturbaum **Ressource, Protokolle, Modbus Master** öffnen.
2. Rechtsklick auf **Modbus Master** und im Kontextmenü **Neu** wählen.
3. Aus der Liste **Modbus Gateway** wählen und mit **OK** bestätigen.
4. Konfiguration des Modbus Gateway im Modbus Master 1:  
**Eigenschaften** zum Konfigurieren der Eigenschaften, siehe Kapitel 4.7.3.  
In den Eigenschaften die **IP-Adresse** des Modbus Master 2 (Modbus Gateway) eintragen.

#### Im Modbus Master 1 die Verbindung zu dem Gateway-Slave erstellen

Im Modbus Master 1 muss der serielle Slave als Gateway-Slave angelegt werden.

1. Im Strukturbaum **Ressource, Protokolle, Modbus Master, Modbus Gateway** öffnen.
2. Rechtsklick auf **Modbus Gateway** und Kontextmenü **Neu** wählen.
3. Aus der Liste **Gateway-Slave** wählen und mit **OK** bestätigen.
4. Konfiguration des Gateway-Slave im Modbus Master 1:  
**Edit** zum Zuweisen der Systemvariablen, siehe Kapitel 4.6.2.  
**Eigenschaften** zum Konfigurieren der Eigenschaften, siehe Kapitel 4.6.3.  
In den Eigenschaften des Slaves die **Serielle Adresse** des Gateway-Slaves eintragen.



**Im Modbus Master 1 die Ein- und Ausgangsvariablen zu dem seriellen-Slave definieren**

1. Rechtsklick auf **Gateway-Slave** und im Kontextmenü **Neu** wählen.
2. Aus der Liste die benötigten **Anforderungstelegramme** wählen.
3. Rechtsklick auf das jeweilige **Anforderungstelegramm** und im Kontextmenü **Edit** wählen.  
In Register *Prozessvariablen* die Ein- oder Ausgangsvariablen eintragen.

**Modbus Master 2 (Modbus Gateway)**

In den Eigenschaften des Modbus Master 02 muss die Gateway Funktion aktiviert werden. Damit werden die im Master 01 konfigurierten Gateway-Slaves mit den seriellen Slaves verbunden.

**Im Modbus Master 2 die Gateway-Funktion aktivieren**

1. Im Strukturbaum **Ressource, Protokolle, Modbus Master** öffnen.
2. Rechtsklick auf **Modbus Master** und im Kontextmenü **Eigenschaften** wählen.
3. Parameter **TCP-Gateway aktivieren** einschalten damit der Modbus Master zusätzlich als TCP-Gateway arbeitet.
4. Parameter **UDP Gateway aktivieren** einschalten, damit der Modbus Master zusätzlich als UDP Gateway arbeitet.

**Im Modbus Master 2 den seriellen Modbus konfigurieren**

1. Im Strukturbaum **Ressource, Protokolle, Modbus Master** öffnen.
2. Rechtsklick auf **Modbus Master** und im Kontextmenü **Neu** wählen.
3. Aus der Liste **Serieller Modbus** wählen und mit **OK** bestätigen.
4. Konfiguration des **seriellen Modbus**  
**Eigenschaften** wählen und Schnittstelle, Baudrate etc. eintragen.

**Im Modbus Master 2 die Verbindung zu dem seriellen Slave konfigurieren**

1. Im Strukturbaum **Ressource, Protokolle, Modbus Master, Serieller Modbus** öffnen.
2. Rechtsklick auf **serieller Modbus** und im Kontextmenü **Neu** wählen.
3. Aus der Liste **Modbus Slave** wählen und mit **OK** bestätigen.
4. Konfiguration des **Modbus Slave**  
**Eigenschaften** wählen und **Slave-Adresse** des seriellen Slaves eintragen.

**Serieller Slave****Konfiguration des seriellen Modbus Slaves**

1. Im Strukturbaum **Ressource, Protokolle, Modbus Slave** öffnen.
2. Rechtsklick auf **Modbus Slave** und im Kontextmenü **Edit** wählen.
3. Konfiguration des **Modbus Slave**  
**Eigenschaften** und **Slave-Adresse** des seriellen Slaves eintragen.

#### 4.6.1 Eigenschaften Gateway

Der Modbus Master kommuniziert mit seinen Modbus Slaves über den Modbus Gateway.

Zur Konfiguration der Verbindung zum Modbus Gateway müssen im Modbus Master die folgenden Parameter eingestellt werden.

Element	Beschreibung
Typ	Modbus Gateway.
Name	Beliebiger, eindeutiger Name, für den Gateway.
Beschreibung	Beliebige, eindeutige Beschreibung für den TCP/UDP Slave.
Kommunikations IP-Protokoll	TCP oder UDP Standardwert: TCP
IP-Adresse	IP-Adresse des Gateways, über welches der Modbus Master mit seinen Modbus Slave kommunizieren soll. Standardwert: (0.0.0.0)
Port	Standardwert: 502

Tabelle 21: Verbindungsparameter Modbus Gateway

#### 4.6.2 Systemvariablen Gateway-Slave

Im Editor stehen die drei Statusvariablen bereit:

Element	Beschreibung
Modbus-Slave Aktivierungssteuerung	Hiermit kann der Gateway-Slave vom Anwenderprogramm deaktiviert oder aktiviert werden. 0:Aktivieren 1:Deaktivieren (Flankengetriggert! Modbus Slave kann über PADT auch dann aktiviert werden wenn <i>Modbus-Slave Aktivierungssteuerung</i> = 1).
Modbus-Slave Fehler	Parameter wie bei TCP/UDP Slave, siehe Kapitel 4.5.1.
Modbus-Slave Zustand	Verbindungsstatus des Gateway-Slave: 0: Deaktiviert 1: Nicht verbunden 2: Verbunden

Tabelle 22: Statusvariablen Gateway-Slave

#### 4.6.3 Eigenschaften Gateway-Slave

Zur Konfiguration der Verbindung zum Gateway-Slave müssen im Modbus Master die folgenden Parameter eingestellt werden.

Element	Beschreibung
Typ	Gateway-Slave
Name	Beliebiger eindeutiger Name für den Gateway-Slave.
Beschreibung	Beliebige eindeutige Beschreibung für den Gateway-Slave.
Slave-Adresse	1 ... 247
Restliche Parameter wie bei TCP/UDP Slave, siehe Kapitel 4.5.2	

Tabelle 23: Verbindungsparameter Gateway-Slave

## 4.7 Serieller Modbus

Der Modbus Master kann mit bis zu 247 seriellen Slaves kommunizieren. Nach der Norm sind insgesamt drei Repeater zulässig, sodass maximal 121 Busteilnehmer pro serielle Schnittstelle eines Masters möglich sind.

**i**

Die Pin-Belegung der D-Sub-Anschlüsse (fb1, fb2) des X-COM-Moduls, siehe Kommunikationshandbuch HI 801 100 D.

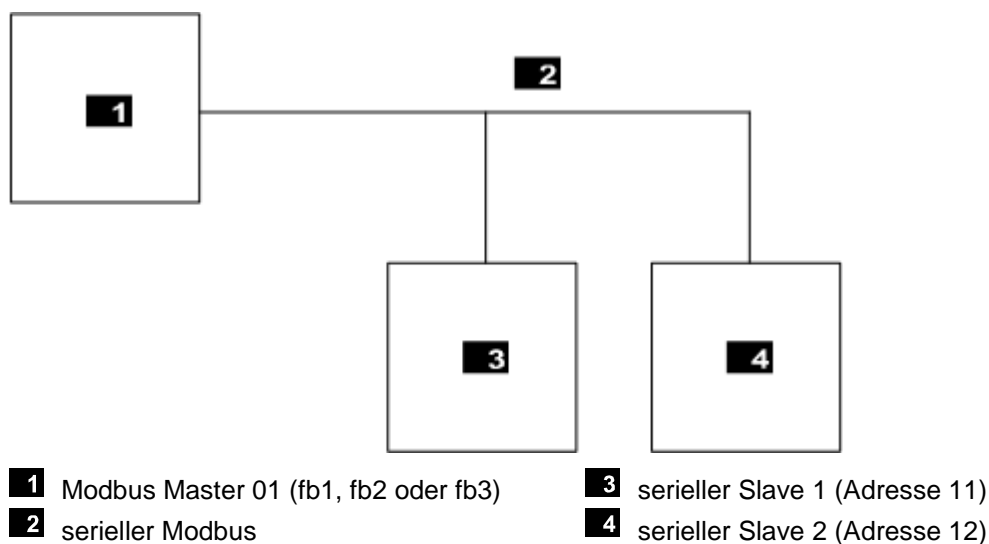


Bild 4: Serieller Modbus

Der Modbus Master unterstützt die Datenübertragung im RTU-Format (Remote Terminal Unit).

Der RTU Telegrammrahmen beginnt und endet mit den vom Anwender vorgegebenen Idle-Zeichen (Standardwert: 5 Idle-Zeichen).

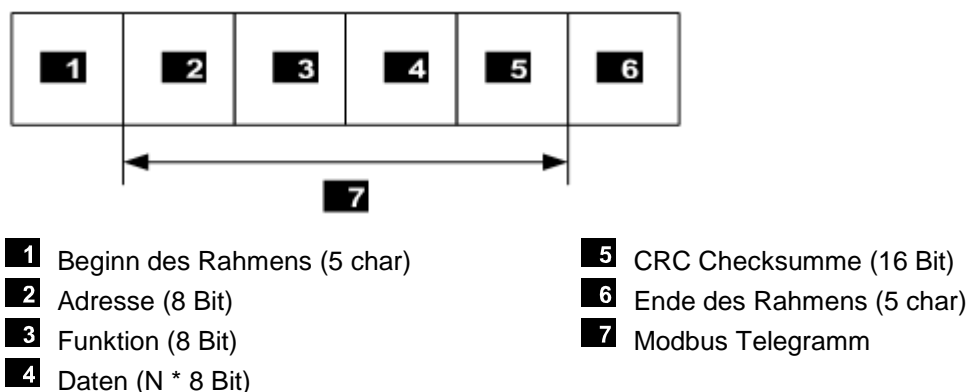


Bild 5: Modbus Telegramm

### Im Modbus Master einen seriellen Modbus erstellen

1. Im Strukturbaum **Ressource, Protokolle, Modbus Master, Serieller Modbus** öffnen.
2. Rechtsklick auf **seriellen Modbus** und im Kontextmenü **Neu** wählen.
3. Aus der Liste **Modbus Slave** wählen und mit **OK** bestätigen.
4. Konfiguration des Modbus Slaves im Modbus Master:  
**Edit** zum Zuweisen der Systemvariablen wählen, siehe 4.7.2.  
**Eigenschaften** zum Konfigurieren der Eigenschaften wählen, siehe Kapitel 4.7.3.

#### 4.7.1 Eigenschaften Serieller Modbus

Zur Konfiguration des seriellen Modbus im Modbus Master müssen die folgenden Parameter eingestellt werden.

Element	Beschreibung
Typ	Serieller Modbus
Name	Name des seriellen Modbus ist durch den Anwender wählbar.
Beschreibung	Beliebige, eindeutige Beschreibung für den seriellen Modbus.
Schnittstelle	Die Feldbus-Schnittstelle, die für den Modbus Master benutzt werden soll (fb1, fb2).
Baudrate [bps]	Übertragungsgeschwindigkeit für RS485 mögliche Werte: 300 Bit/s 600 Bit/s 1200 Bit/s 2400 Bit/s 4800 Bit/s 9600 Bit/s 19200 Bit/s 38400 Bit/s (maximale Baudrate HIMax vor V4) 57600 Bit/s (maximale Baudrate HIMax ab V4) 62500 Bit/s (HIMatrix/ HIQuad X) 76800 Bit/s (HIMatrix/ HIQuad X) 115000 Bit/s (HIMatrix/ HIQuad X)
Parität	keine ungerade gerade Standardwert: gerade
Stop-Bits	Standard (passt die Anzahl der Stop-Bits der Parität an: mit Parität = 1 Stop-Bit, keine Parität 2 Stop-Bit.) ein Stop-Bit zwei Stop-Bit Standardwert: Standard
Anzahl Idle-Zeichen	Die Anzahl der Idle-Zeichen am Beginn und Ende des RTU-Telegrammrahmens. Wertebereich: 0 ... 65535 Standardwert: 5 Zeichen

Tabelle 24: Parameter serieller Modbus Master

### 4.7.2 Systemvariablen Modbus Slave

Im Editor **Edit** stehen die drei Statusvariablen (Systemvariablen) bereit.

Element	Beschreibung
Modbus-Slave Aktivierungssteuerung	Modbus Slave im Anwenderprogramm aktivieren oder deaktivieren. 0: Aktivieren 1: Deaktivieren (Flankengetriggert! Modbus Slave kann über PADT auch dann aktiviert werden wenn <i>Modbus-Slave Aktivierungssteuerung</i> = 1).
Modbus-Slave Fehler	Parameter wie bei TCP/UDP Slave Kapitel 4.5.2.
Modbus-Slave Zustand	Verbindungsstatus des Modbus Slave: 0: Deaktiviert 1: Nicht verbunden 2: Verbunden

Tabelle 25: Systemvariablen Modbus Slave

### 4.7.3 Eigenschaften Modbus Slave

Zur Konfiguration der Verbindung zu den seriellen Slaves müssen im Modbus Master die folgenden Parameter eingestellt werden.

Element	Beschreibung
Typ	Modbus Slave
Name	Name des Modbus Slave; durch den Anwender wählbar.
Beschreibung	Beliebige, eindeutige Beschreibung für den Modbus Slave.
Slave-Adresse	1 ... 247
Restliche Parameter wie bei TCP/UDP Slave, siehe Kapitel 4.5.2.	

Tabelle 26: Verbindungsparameter Modbus Master

#### i

Das Receive Timeout ist beim seriellen Modbus Slave abhängig von der eingestellten Übertragungsgeschwindigkeit.

Ist die Baudrate 19200 [bit/s] oder höher, kann der Vorgabewert für das Receive Timeout verwendet werden. Bei niedrigeren Baudraten als 19200 [bit/s] muss das Receive Timeout erhöht werden.

## 4.8 Control-Panel (Modbus Master)

Im Control-Panel kann der Anwender die Einstellungen des Modbus Master überprüfen und steuern. Zudem werden aktuelle Statusinformationen (z. B. Master-Zustand usw.) des Masters angezeigt.

### Öffnen des Control Panels zur Überwachung des Modbus Master

1. Im Strukturbaum **Hardware** und im Kontextmeü **Online** wählen.
2. Im **System-Login**, Zugangsdaten eingeben um die Online Ansicht der Hardware zu öffnen.
3. Doppelklick auf **COM-Modul** und im Strukturbaum **Modbus Master** wählen.

### 4.8.1 Kontextmenü (Modbus Master)

Aus dem Kontextmenü des selektierte Modbus Master können die folgenden Kommandos gewählt werden:

#### Offline

Mit diesem Kommando wird der Modbus Master gestoppt.

#### Operate

Mit diesem Kommando wird der Modbus Master gestartet.

#### Statistik zurücksetzen

Setzt die statistischen Daten (z. B. Anzahl Busfehler, Zykluszeit min, max usw.) auf null zurück.

### 4.8.2 Anzeigefeld (Modbus Master)

In dem Anzeigefeld werden die folgenden Werte des selektierten Modbus Master angezeigt.

Element	Beschreibung
Name	Name des Modbus Masters.
Master-Zustand	Der Modbus Master Zustand zeigt den momentanen Protokollzustand an: OPERATE OFFLINE
Anzahl Busfehler	Zähler Anzahl der Busfehler.
Gestörte Verbindungen	Zähler Anzahl der gestörte Verbindungen.
µP-Last (projektierte)	Siehe Kapitel 4.3.2.
µP-Last (tatsächliche)	

Tabelle 27: Anzeigefeld Modbus Master

## 4.9 Control-Panel (Modbus Master->Slave)

Im Control-Panel kann der Anwender die Einstellungen der Kommunikationspartner des Modbus Master überprüfen und aktivieren/deaktivieren. Zudem werden aktuelle Statusinformationen (z. B. Slave-Zustand usw.) des Kommunikationspartners angezeigt.

### Öffnen des Control Panels zur Überwachung der Modbus Verbindung

1. Im Strukturbaum **Hardware** und im Kontextmeü **Online** wählen.
2. Im **System-Login**, Zugangsdaten eingeben um die Online Ansicht der Hardware zu öffnen.
3. Doppelklick auf **COM-Modul** und im Strukturbaum **Modbus Master, Slave** wählen.

#### 4.10 Funktion der FBx LED beim Modbus Master

Den Zustand des Modbus Protokolls signalisiert die FBx LED der zugeordneten Feldbus-Schnittstelle. Die Zustände der FBx LED sind in der folgenden Tabelle dargestellt:

LED	Farbe	Status	Bedeutung
FBx	Gelb	Blinken	Das Modbus Master Protokoll ist aktiv und befindet sich im Datenaustausch mit den Modbus Slaves.
		Aus	Das Modbus Master Protokoll ist nicht aktiv! D. h. die Steuerung ist im Zustand STOPP oder es ist kein Modbus Master konfiguriert.

Tabelle 28: LED FBx

#### 4.11 Funktion der FAULT LED beim Modbus Master (Nur HIMax)

Eine Störung des Modbus Protokolls signalisiert die FAULT LED der zugeordneten Feldbus-Schnittstelle. Die Zustände der FAULT LED sind in der folgenden Tabelle dargestellt:

LED	Farbe	Status	Bedeutung
Fault	Rot	Blinken <sup>1)</sup>	<p>Folgende Ereignisse führen zur Störung.</p> <ul style="list-style-type: none"> <li>▪ Fehlerhafte Antwort oder Fehlermeldung vom Slave empfangen.</li> <li>▪ Timeout zu einem oder mehreren Slaves.</li> <li>▪ Die Protokoll-Last erreicht das Maximum des konfigurierten Rechenzeitbudgets. Die Protokollausführung wird durch die Lastbegrenzung limitiert. Dies kann zu erheblichen Einschränkungen der Protokollfunktion führen. Es müssen entweder die Einstellungen des Protokolls oder die Einstellungen des Rechenzeitbudgets angepasst werden.</li> </ul> <p>Tritt länger als 5 s kein Fehlerereignis auf, so wechselt die Anzeige in den Zustand „Protokoll ohne Störung“.</p>
		Aus	Modbus Master Protokoll ist ohne Störung.

<sup>1)</sup> Blinkfrequenz: lang (600 ms) an, lang (600 ms) aus.

Tabelle 29: FAULT LED

## 5 Modbus Slave

Der Modbus Slave kann gleichzeitig über die serielle Schnittstelle (RS485) und über Ethernet (TCP/UDP) mehrere Modbus Master bedienen.

**i**

HIMA empfiehlt für ein neues Projekt, das Modbus Slave V2 Protokoll zu verwenden. Dieses beruht auf keiner neuen Modbus-Spezifikation, sondern ist eine erweiterte HIMA Variante, welche die interne Verarbeitung der Protokolldaten auf HIMA Steuerungen betrifft. Die Standard Modbus-Funktionscodes bleiben gleich, die HIMA spezifischen Modbus-Funktionscodes entfallen. Für weitere Informationen, siehe Modbus Slave V2 Handbuch HI 801 474 D.

### 5.1 Benötigte Ausstattung und Systemanforderungen

Element	Beschreibung
HIMA Steuerung	HIMax mit X-COM 01 Modul HIQuad X mit F-COM 01 Modul HIMatrix
CPU-Modul	Die Ethernet-Schnittstellen des Prozessormoduls können für Modbus TCP nicht verwendet werden.
COM-Modul	Ethernet 10/100BaseT D-Sub Anschlüsse FB1 und FB2 Wird Modbus RTU verwendet, muss das COM-Modul an den verwendeten seriellen Feldbus-Schnittstellen (FB1 und/oder FB2) mit jeweils einem HIMA RS485 Submodul ausgerüstet sein. Schnittstellenbelegung, siehe Kommunikationshandbuch HI 801 100 D.
Aktivierung	Jede der beiden Modbus Slave Funktionen muss einzeln freigeschaltet werden, siehe Kommunikationshandbuch HI 801 100 D. Modbus Slave RTU (RS485) Modbus Slave TCP Für den redundanten Modbus Slave werden 2 Lizenzen benötigt, eine für jedes Kommunikationsmodul. Eine redundante Modbus-Slave-Konfiguration ist nur auf den Steuerungen HIMax und HIQuad X möglich.

Tabelle 30: Ausstattung und Systemanforderung Modbus Slave



## 5.2 Modbus Slave (Eigenschaften)

Element	Beschreibung																									
Modbus Slave	Für jedes COM-Modul kann jeweils ein Modbus Slave TCP und Modbus Slave RTU (RS485) konfiguriert werden.																									
Redundanz	Anzahl der redundanten Modbus Slave Kommunikationsmodul-Paare, die in einem HIMA System betrieben werden können. HIMax:           Maximal 10 HIQuad X:       Maximal 5 Solange ein Modbus Slave Kommunikationsmodul-Paar redundant arbeitet, werden die gleichen Eingangs- und Ausgangsdaten über beide Kommunikationsmodule mit dem Modbus-Master ausgetauscht, siehe Kapitel 5.5.																									
Anzahl Masterzugriffe	RTU: Aufgrund der RS485 Übertragungstechnik kann jeweils nur ein Modbus Master auf eine installierte RS485 Schnittstelle zugreifen TCP: Maximal 20 Modbus Master können auf den Slave zugreifen. UDP: Unbegrenzte Anzahl von Modbus Master können auf den Slave zugreifen.																									
Max. Größe der Sendedaten	Siehe Kommunikationshandbuch HI 801 100 D.																									
Max. Größe der Empfangsdaten																										
Darstellungsformat der Modbus-Daten	Die HIMA Steuerungen verwenden das Big Endian Format. Beispiel 32 Bit Daten (z. B. DWORD, DINT): <table><tr><td>32 Bit Daten (hex)</td><td colspan="4">0x12345678</td></tr><tr><td>Bytenummer (von links)</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>Big Endian (HIMax, HIQuad X, HIMatrix)</td><td>12</td><td>34</td><td>56</td><td>78</td></tr><tr><td>Middle Endian (H51q)</td><td>56</td><td>78</td><td>12</td><td>34</td></tr><tr><td>Little Endian</td><td>78</td><td>56</td><td>34</td><td>12</td></tr></table>	32 Bit Daten (hex)	0x12345678				Bytenummer (von links)	0	1	2	3	Big Endian (HIMax, HIQuad X, HIMatrix)	12	34	56	78	Middle Endian (H51q)	56	78	12	34	Little Endian	78	56	34	12
32 Bit Daten (hex)	0x12345678																									
Bytenummer (von links)	0	1	2	3																						
Big Endian (HIMax, HIQuad X, HIMatrix)	12	34	56	78																						
Middle Endian (H51q)	56	78	12	34																						
Little Endian	78	56	34	12																						

Tabelle 31: Eigenschaften Modbus Slave

## 5.3 Konfiguration des Modbus TCP Slave

### Einen neuen Modbus Slave anlegen

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle** öffnen.
2. Im Kontextmenü von Protokolle **Neu, Modbus-Slave-Set** wählen, um einen neuen Modbus-Slave-Set hinzuzufügen.
3. Im Kontextmenü des Modbus-Slave-Set **Edit** wählen und **Eigenschaften Modbus-Slave-Set** öffnen, Standardwerte beibehalten.
4. Register **Modbus-Slave** wählen und folgende Einstellungen vornehmen:
  - **COM Modul** wählen
  - **TCP aktivieren** aktivieren
  - Die restlichen Parameter behalten die Standardwerte.

### i

Bei Verwendung der Ethernet-Schnittstellen als Transportkanal müssen sich die HIMA Steuerung und der Kommunikationspartner im selben Subnet befinden oder bei Verwendung eines Routers die entsprechenden Routingeinträge besitzen.

Ein Konfigurationsbeispiel zur Verbindung eines HIMA Modbus TCP Slave mit einem HIMA Modbus TCP Master ist in Kapitel 4.1 beschrieben.

## 5.4 Konfiguration des redundanten Modbus TCP Slave

### Einen redundanten Modbus Slave anlegen

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle, Modbus-Slave-Set** öffnen.
2. Im Kontextmenü des Modbus-Slave-Set **Edit** wählen und **Eigenschaften Modbus-Slave-Set** öffnen und folgende Einstellung vornehmen:
  - **Redundanzbetrieb aktivieren** aktivieren.
  - ☒ Das Register **Modbus-Slave-Redundant** wird automatisch hinzugefügt
3. Register **Modbus-Slave-Redundant** wählen und folgende Einstellungen vornehmen:
  - **COM Modul** wählen
  - **TCP aktivieren** aktivieren

Die restlichen Parameter behalten die Standardwerte.

---

**i**

Die im Modbus-Slave-Set zugewiesenen Sende- und Empfangsvariablen gelten für beide Modbus-Slaves.

---

## 5.5 Regeln für den redundanten Modbus TCP Slave

Für den redundanten Betrieb von HIMax oder HIQuad X Modbus Slave Kommunikationsmodulen wird eine redundante Systemkonfiguration empfohlen, siehe Systemhandbuch HI 801 000 D.

Ansonsten kann bereits beim Auftreten eines ersten Fehlers im HIMax oder HIQuad X System nicht sichergestellt werden, dass sich die Modbus Slave Kommunikationsmodul-Paare gegenüber ihrem externen Partner (Modbus Master) konsistent verhalten.

### 5.5.1 Prozessdatenmenge für den redundanten Modbus Slave

Die Prozessdatenmenge, die der Anwender konfigurieren kann reduziert sich um die Hälfte. Bei HIMax von 128 kB auf 64 kB und bei HIQuad X von 64 kB auf 32 kB pro Richtung (Input und Output). Siehe auch System Mengengerüst für Protokolle im Kommunikationshandbuch HI 801 100 D.

### 5.5.2 Zulässige Steckplätze der redundanten Modbus Slave HIMax COM-Module

Um mögliche Kollisionen auf dem HIMax Systembus zu minimieren, müssen die Systembus-Segmente (1 ... 3) auf dem Basisträger berücksichtigt werden. Daher sollten die redundanten Modbus Slave Kommunikationsmodule jeweils nur im gleichen Segment eines Basisträgers auf den folgenden Steckplätzen gesteckt werden:

Segment	Steckplatz
1	3 ... 6 (sofern hier keine Prozessormodule vorgesehen sind)
2	7 ... 14
3	15 ... 18

Tabelle 32: Zulässige Steckplätze der redundanten Modbus Slave HIMax COM-Module

### 5.5.3 Redundante Modbus Slave COM-Module in unterschiedlichen Basisträgern

Es dürfen nicht mehr als zwei redundante Modbus Slave Kommunikationsmodul-Paare betrieben werden, deren redundanten Modbus Slave Kommunikationsmodule sich in unterschiedlichen Basisträgern (0 ... 15) befinden.

Die redundanten Modbus Slave Kommunikationsmodule dürfen sich dann auch nur im jeweils benachbarten Basisträger befinden.

Zusätzlich dürfen weitere 8 Modbus Slave Kommunikationsmodul-Paare im gleichen HIMax-System nach den Regeln aus Kapitel 5.5.2 betrieben werden.

## HINWEIS



**Betriebsstörung möglich!**

**Steckplätze für redundante Modbus Slave Kommunikationsmodule nur entsprechend den genannten Regeln verwenden!**

**Zwischen einem X-COM Modul und den X-CPU Modulen ist eine zusätzliche Laufzeit (durch Länge der Kabel, Switches) von maximal 50 µs erlaubt.**

**Empfehlung: X-COM Module möglichst nahe zu den X-CPU Modulen betreiben (z. B. Rack 0, Rack 1)**

## 5.6 Menüfunktionen des Modbus Slave Set

Die Menüfunktion Edit aus dem Kontextmenü des Modbus Slave Set öffnet den Dialog *Modbus-Slave-Set*. Das Dialogfenster enthält die folgenden Register:

### 5.6.1 Eigenschaften Modbus Slave Set

Im Register **Eigenschaften Modbus Slave Set** werden die folgenden Parameter für den Modbus Slave eingestellt.

Element	Beschreibung
Name	Name des Modbus Slave Set.
Max. $\mu$ P-Budget aktivieren	Aktiviert: Limit des $\mu$ P-Budget aus dem Feld <i>Max. <math>\mu</math>P-Budget in [%]</i> übernehmen.  Deaktiviert: Kein Limit des $\mu$ P-Budget für dieses Protokoll verwenden.
Max. $\mu$ P-Budget in [%]	Maximale $\mu$ P-Last des COM-Moduls, welche bei der Abarbeitung des Protokolls produziert werden darf. Wertebereich: 1 ... 100 % Standardwert: 30 %
Redundanzbetrieb aktivieren	Aktiviert: Redundanzbetrieb Register <i>Modbus-Slave-Redundant</i> ist aktiviert.  Deaktiviert: Monobetrieb Register <i>Modbus-Slave-Redundant</i> ist deaktiviert.  Standardwert: Deaktiviert
Maximale Antwortzeit-Verzögerungszeit [ms]	Zeitspanne nach dem Empfang einer Anfrage, innerhalb welcher der Modbus Slave noch antworten darf. Wertebereich: 0 ... $(2^{31}-1)$ [ms] Standardwert: 5000 ms (0 = keine Limitierung)
Bereich zum Lesen der Funktionscodes 1, 3, 100, 102	Der Parameter bestimmt, von welchem Datenbereich die Daten für den Funktionscode 1, 3, 100, 102 gelesen werden sollen. Wertebereich: <ul style="list-style-type: none"> <li>Importbereich</li> <li>Exportbereich (kompatibel zu H51q)</li> </ul>
Bereich zum Lesen bei Funktionscode 23, 106	Hier kann der Anwender den Bereich des Modbus Slave festlegen aus dem der Funktionscode 23, lesen soll. Importbereich: Der Master greift auf den Importbereich des Slaves lesend und schreibend zu. Exportbereich: Der Master liest vom Exportbereich des Slaves und schreibt auf den Importbereich des Slaves. Hinweis: Das Lesen und Schreiben erfolgt innerhalb von einem CPU-Zyklus. D. h. die gelesenen Daten wurden vom <u>letzten</u> CPU-Zyklus bereitgestellt.
COM: Werte bei Verbindungsverlust zu Master	Bei Verbindungsverlust des Kommunikationsmoduls zum Modbus Master werden in Abhängigkeit dieses Parameters die Eingangsvariablen entweder initialisiert oder unverändert an das Prozessormodul weitergeleitet. Initialdaten annehmen      Eingangsvariablen werden auf die Initialwerte zurückgesetzt. Letzten Wert beibehalten      Eingangsvariablen behalten den letzten Wert.

Element	Beschreibung
CPU: Werte bei Verbindungsverlust zu COM	<p>Bei Verbindungsverlust des Prozessormoduls zum Kommunikationsmodul werden in Abhängigkeit dieses Parameters die Eingangsvariablen entweder initialisiert oder unverändert im Prozessormodul verwendet. (z. B. wenn Kommunikationsmodul bei laufender Kommunikation gezogen wird).</p> <p>Gleiches Verhalten wie COM zu Master      Siehe Einstellungen in Parameter <i>COM: Werte bei Verbindungsverlust zu Master</i>.</p> <p>Letzten Wert beibehalten      Eingangsvariablen behalten den letzten Wert.</p> <p>Standardwert: Gleiches Verhalten wie COM zu Master.</p>
Alternative Register-/Bit-Adressierung verwenden	<p>Aktiviert      Alternative Adressierung verwenden.</p> <p>Deaktiviert      Alternative Adressierung nicht verwenden.</p> <p>Standardwert: Deaktiviert, siehe Kapitel 5.11</p>
Register Bereich Offset Bits (Input)	<p>Wertebereich: 0 ... 65535</p> <p>Standardwert: 0</p>
Register Bereich Offset Bits (Output)	<p>Wertebereich: 0 ... 65535</p> <p>Standardwert: 0</p>
Bit Bereich Offset Register (Input)	<p>Wertebereich: 0 ... 65535</p> <p>Standardwert: 0</p>
Bit Bereich Offset Register (Output)	<p>Wertebereich: 0 ... 65535</p> <p>Standardwert: 0</p>
Aktualisierungsintervall der Prozessdaten [ms]	<p>Aktualisierungszeit in Millisekunden, mit der die Daten des Protokolls zwischen COM und CPU ausgetauscht werden.</p> <p>Ist die <i>Refresh Rate</i> Null oder kleiner als die Zykluszeit der Steuerung, dann erfolgt der Datenaustausch so schnell wie möglich.</p> <p>Wertebereich: 0 ... (2<sup>31</sup>-1)</p> <p>Standardwert: 0</p>
Mehrere Fragmente pro Zyklus zulassen	<p>Aktiviert:      Transfer der gesamten Daten des Protokolls von der CPU zur COM innerhalb eines Zyklus der CPU.</p> <p>Deaktiviert:      Transfer der gesamten Daten des Protokolls von der CPU zur COM, verteilt über mehrere CPU Zyklen zu je 1100 Byte pro Datenrichtung.</p> <p>Damit kann eventuell auch die Zykluszeit der Steuerung reduziert werden.</p> <p>Standardwert: Aktiviert</p>

Tabelle 33: Register Eigenschaften Modbus Slave Set

### 5.6.2 Registervariablen

Im Register **Registervariablen** sind die Variablen einzutragen, die der Master registerweise (16 bit) adressiert (Funktionscode 3, 4, 6, 16, 23, 102, 103, 105, 106).

Die Eingangs- und Ausgangsvariablen werden aus dem Objektbereich in die Felder *Register-Eingänge* oder *Register-Ausgänge* per Drag&Drop gezogen.

- Im Feld *Register-Eingänge* werden alle Variablen, die der Modbus Slave vom Modbus Master empfängt, zugewiesen.
- Im Feld *Register-Ausgänge* werden alle Variablen, die der Modbus Slave an den Modbus Master sendet, zugewiesen.

#### Neue Offsets generieren

1. Rechtsklick auf das Feld *Register-Eingänge* oder *Register-Ausgänge* und aus dem Kontextmenü *Neue Offsets* wählen, um das Dialogfenster *Neue Offsets* zu öffnen.
2. Gewünschte Einstellungen eintragen, um die Offsets der Variablen neu zu nummerieren, siehe Tabelle 34.

Der Dialog dient zur automatischen Offsetvergabe von Variablen. Das PADT vergibt eine neue Offset-Sortierung in Abhängigkeit von der Reihenfolge der Variablen in der selektierten Liste.

Element	Beschreibung	
Start-Register	Startadresse des ersten Registers.	
Modus	Element	▪ Beschreibung
	Kompakt (alle Bytes belegen)	<ul style="list-style-type: none"> <li>▪ Gepackte BOOLS auf Bit-Grenzen.</li> <li>▪ Variablen ab der Größe 1 Byte auf ganzen Bytes.</li> </ul>
	Registergrenzen einhalten (bis auf Bits)	<ul style="list-style-type: none"> <li>▪ Gepackte BOOLS auf Bit-Grenzen.</li> <li>▪ Sonst auf ganzzahligen Register-Adressen. Dabei setzt das PADT Variablen der Größen 2 ... 8 Bit auf Register-Bit 0.</li> </ul>
	Registergrenzen einhalten (bis auf Bit bis 1-Byte)	<ul style="list-style-type: none"> <li>▪ Gepackte BOOLS auf Bit-Grenzen.</li> <li>▪ Variablen der Größen 2-8 Bit auf ganzen Byteadressen.</li> <li>▪ Variablen größer 1 Byte auf ganzzahligen Register-Adressen.</li> </ul>
Den Dialog bei Erfolg automatisch schließen.	Aktiviert: Den Dialog bei Erfolg automatisch schließen. Deaktiviert: Der Dialog durch klicken der Schaltfläche <b>OK</b> schließen.	

Tabelle 34: Dialogfenster Neue Offsets

### 5.6.3 Bitvariablen

Im Register **Bitvariablen** sind die Variablen einzutragen, die der Master 1 bitweise adressiert werden (Funktionscode 1, 2, 5, 15, 100, 101, 104).

Die Eingangs- und Ausgangsvariablen werden aus dem Objektbereich in die Felder *Bit-Eingänge* oder *Bit-Ausgänge* per Drag&Drop gezogen.

- Im Feld *Bit-Eingänge* werden alle Variablen, die der Modbus Slave vom Modbus Master empfängt, zugewiesen.
- Im Feld *Bit-Ausgänge* werden alle Variablen, die der Modbus Slave an den Modbus Master sendet, zugewiesen.

#### Neue Offsets generieren

1. Rechtsklick auf das Feld *Bit-Eingänge* oder *Bit-Ausgänge* und aus dem Kontextmenü *Neue Offsets* wählen, um das Dialogfenster *Neue Offsets* zu öffnen.
2. Gewünschte Einstellungen eintragen, um die Offsets der Variablen neu zu nummerieren, siehe Tabelle 35.

Der Dialog dient zur automatischen Offsetvergabe von Variablen. Das PADT vergibt eine neue Offset-Sortierung in Abhängigkeit von der Reihenfolge der Variablen in der selektierten Liste.

Element	Beschreibung	
Start-Bit	Startadresse des ersten Bits.	
Modus	Element	▪ Beschreibung
	Kompakt (alle Bytes belegen)	<ul style="list-style-type: none"> <li>▪ Gepackte BOOLS auf Bit-Grenzen.</li> <li>▪ Variablen ab der Größe 1 Byte auf ganzen Bytes.</li> </ul>
	Registergrenzen einhalten (bis auf Bits)	<ul style="list-style-type: none"> <li>▪ Gepackte BOOLS auf Bit-Grenzen.</li> <li>▪ Sonst auf ganzzahligen Register-Adressen. Dabei setzt das PADT Variablen der Größen 2 ... 8 Bit auf Register-Bit 0.</li> </ul>
	Registergrenzen einhalten (bis auf Bit bis 1-Byte)	<ul style="list-style-type: none"> <li>▪ Gepackte BOOLS auf Bit-Grenzen.</li> <li>▪ Variablen der Größen 2-8 Bit auf ganzen Byteadressen.</li> <li>▪ Variablen größer 1 Byte auf ganzzahligen Register-Adressen.</li> </ul>
Den Dialog bei Erfolg automatisch schließen.	Aktiviert: Den Dialog bei Erfolg automatisch schließen. Deaktiviert: Der Dialog durch klicken der Schaltfläche <b>OK</b> schließen.	

Tabelle 35: Dialogfenster Neue Offsets

## 5.7 Systemvariablen Modbus-Slave-Set

Das Register **Systemvariablen Modbus-Slave-Set** stellt die folgende Systemvariable bereit.

Element	Beschreibung
Redundanz-Zustand	Dieser Parameter beschreibt den Redundanz-Zustand des redundanten Modbus Slave Kommunikationsmodul-Paars. 0: redundante Modbus Slave COM-Module aktiv. 1: Erstes Modbus Slave COM-Modul nicht aktiv. 2: Redundantes Modbus Slave COM-Modul nicht aktiv. 3: beide Modbus Slave COM-Module nicht aktiv.

Tabelle 36: Register Modbus Slave Set

### 5.7.1 Modbus Slave und Modbus Slave Redundant

Im Register *Modbus Slave* befinden sich die beiden Register *Eigenschaften* und *Systemvariablen*.

Das Register *Modbus-Slave-Redundant* ist im Dialog *Modbus-Slave-Set* vorhanden, wenn im Register *Eigenschaften Modbus Slave Set* der Parameter *Redundanzbetrieb aktivieren* gesetzt wurde.



Die Pin-Belegung der D-Sub-Anschlüsse (fb1, fb2) wird im Kommunikationshandbuch HI 801 100 D beschrieben.

#### 5.7.1.1 Eigenschaften

Element	Beschreibung
Modul	Auswahl des COM-Moduls, auf dem dieses Protokoll abgearbeitet wird.
Master Überwachungszeit [ms]	<p>Zeitliche Überwachung des oder der Modbus-Master. Falls die Master Überwachungszeit nicht 0 ist, wird überprüft, ob innerhalb dieser Zeitspanne eine gültige Anfrage von einem beliebigem Modbus Master bezüglich der Modbus Slave Datenbereiche empfangen wurde. Ist dies der Fall, so wird die Systemvariable <i>Master-Verbindungszustand</i> auf den Wert TRUE gesetzt.</p> <p>Bei Verbindungsverlust des Kommunikationsmoduls zum Modbus Master werden in Abhängigkeit des Parameters <i>COM: Werte bei Verbindungsverlust zu Master</i> die Eingangsvariablen entweder initialisiert oder unverändert an das Prozessormodul weitergeleitet.</p> <p>Hinweis: Bei Verwendung mehrerer Master gleichen Typs (Eth oder RS485) kann die Masterüberwachung nicht zwischen den Master unterscheiden. Es kann also sein, dass ein Master unbemerkt ausfällt.</p> <p>Siehe Kapitel 5.6.1. Wertebereich: 1 ... (2<sup>31</sup>-1) [ms] Standardwert: 0 ms (keine Limitierung)</p>
Parameter der Ethernet Schnittstelle	
TCP aktivieren	<p>Aktiviert TCP/IP-Verbindung aktiviert. Deaktiviert TCP/IP-Verbindung deaktiviert. Standardwert: Deaktiviert</p>
TCP-Port	Standardwert: 502
Maximale Anzahl TCP-Verbindungen	<p>Maximale Anzahl gleichzeitig offener TCP-Verbindungen als Server. Wertebereich: 1 ... 20 Standardwert: 20</p>
UDP aktivieren	<p>Aktiviert UDP/IP-Verbindung aktiviert. Deaktiviert UDP/IP-Verbindung deaktiviert. Standardwert: Deaktiviert</p>
UDP Port	Standardwert: 502
Parameter der seriellen Schnittstelle	
Name	Name der seriellen Schnittstelle
Schnittstelle	Auswahl der verfügbaren Feldbus-Schnittstellen, die für den Modbus Slave benutzt werden können (keine, fb1 und/oder fb2).
Slave-Adresse	<p>Busadresse des Slave Wertebereich: 1 ... 247</p>



Baudrate [bps]	Übertragungsgeschwindigkeit für RS485 mögliche Werte: 300 Bit/s 600 Bit/s 1200 Bit/s 2400 Bit/s 4800 Bit/s 9600 Bit/s 19200 Bit/s 38400 Bit/s (maximale Baudrate HIMax vor V4) 57600 Bit/s (maximale Baudrate HIMax ab V4) 62500 Bit/s (HIMatrix/ HIQuad X) 76800 Bit/s (HIMatrix/ HIQuad X) 115000 Bit/s (HIMatrix/ HIQuad X)
Parität	Wertebereich: <ul style="list-style-type: none"> <li>▪ keine</li> <li>▪ ungerade</li> <li>▪ gerade</li> </ul> Standardwert: gerade
Stop-Bits	Wertebereich: Standard (passt die Anzahl der Stop-Bits der Parität an: mit Parität = 1 Stop-Bit, keine Parität 2 Stop-Bit.) ein Stop-Bit zwei Stop-Bit Standardwert: Standard
Anzahl Idle-Zeichen	Die Anzahl der Idle-Zeichen am Beginn und Ende des RTU-Telegrammrahmens. Wertebereich: 1 ... 65535 Standardwert: 5 Zeichen

Tabelle 37: Register TCP- und UDP-Ports für Modbus Slave

### 5.7.1.2 Systemvariablen

Das Register **Systemvariablen** stellt Systemvariablen bereit, die es erlauben, den Zustand des Modbus Slave im Anwenderprogramm auszuwerten und den Modbus Slave zu steuern

Element	Beschreibung
Durchschnittlicher Pufferfüllstand für Anfragen	Durchschnittliche Anzahl gleichzeitiger Master Anfragen.
Gültige Master-Anfragen	Anzahl der gültigen Master Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten.
Master-Anfragen	Gesamtanzahl aller Master Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten.
Master-Überwachungszeit [ms]	Zeitliche Überwachung des oder der Modbus-Master, siehe Kapitel 5.7.1.
Master-Verbindungszustand	FALSE: Nicht verbunden. TRUE: Verbunden
Maximaler Pufferfüllstand für Anfragen	Maximale Anzahl gleichzeitiger Master Anfragen.
Rücksetzen aller Zähler	Mit dieser Systemvariable können über das Anwenderprogramm alle Zähler zurückgesetzt werden. Ein Wechsel von 0 auf 1 löst die Reset-Funktion aus. Werte > 1 werden als 1 behandelt.

Element	Beschreibung
Ungültige Master-Anfragen	Anzahl ungültiger Master Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten. Ungültige Anfragen sind solche, die der Modbus Slave mit einem Fehlercode an den Modbus Master beantwortet. Fehlerhafte Sendungen, die bereits auf Treiberebene erkannt und ausgefiltert werden (Framing Errors, CRC-Fehler, Längenfehler) sind hier nicht mit enthalten, sondern werden nur über die Diagnose gemeldet.
Verworfenne Anfragen	Anzahl der verworfenen Master Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten.
Zeitüberschreitungen bei Antworten	Anzahl der <i>Zeitüberschreitungen bei Antworten</i> seit dem letzten Rücksetzen aller Zähler oder Einschalten. Die <i>Zeitüberschreitungen bei Antworten</i> ist die maximale Zeit, die bis zur Empfangsbestätigung einer Nachricht beim Absender vergehen darf.

Tabelle 38: Register Systemvariablen für Modbus Slave

## 5.8 Modbus Funktionscodes des Modbus Slaves

### 5.8.1 Modbus-Funktionscodes

Folgende Modbus-Funktionscodes werden vom Modbus Slave unterstützt.

Element	Code	Typ	Bedeutung
READ COILS	01	BOOL	Lesen mehrerer Variablen (BOOL) aus dem Import- oder Export <sup>1)</sup> -Bereich des Slaves. Max. Länge der Prozessdaten: 250 Bytes (2000 Coils).
READ DISCRETE INPUT	02	BOOL	Lesen mehrerer Variablen (BOOL) aus dem Export-Bereich des Slaves. Max. Länge der Prozessdaten: 250 Bytes (2000 Coils).
READ HOLDING REGISTER	03	WORD	Lesen mehrerer Variablen beliebigen Typs aus dem Import- oder Export <sup>1)</sup> -Bereich des Slaves. Max. Länge der Prozessdaten: 250 Bytes.
READ INPUT REGISTER	04	WORD	Lesen mehrerer Variablen beliebigen Typs aus dem Export-Bereich des Slaves. Max. Länge der Prozessdaten: 250 Bytes.
WRITE SINGLE COIL	05	BOOL	Schreiben eines einzelnen Signals (BOOL) in den Import-Bereich des Slaves. Max. Länge der Prozessdaten: 1 Byte (1 Coil).
WRITE SINGLE REGISTER	06	WORD	Schreiben eines einzelnen Signals (WORD) in den Import-Bereich des Slaves. Max. Länge der Prozessdaten: 2 Bytes.
Diagnostics	08	x	Nur Subcode 0: Loopback-Funktion des Slave
WRITE MULTIPLE COILS	15	BOOL	Schreiben mehrerer Variablen (BOOL) in den Import-Bereich des Slaves. Max. Länge der Prozessdaten: 246 Bytes (1968 Coils).
WRITE MULTIPLE REGISTER	16	WORD	Schreiben mehrerer Variablen beliebigen Typs in den Import-Bereich des Slaves. Max. Länge der Prozessdaten: 246 Bytes.
READ WRITE MULTIPLE REGISTER	23	WORD	Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Import-Bereich oder Export-Bereich des Slaves. Max. Länge der Prozessdaten: 242 Bytes (Anforderungstelegramm des Modbus Masters) 250 Bytes (Antworttelegramm des Slaves).
Read Device Identification	43	x	Liefern die Identifikationsdaten des Slaves an den Master.

<sup>1)</sup> Export Bereich muss so eingestellt werden wie der Master es benötigt.

Tabelle 39: Modbus-Funktionscodes des Modbus Slave

Die Funktionscodes 03, 04, 16 und 23 unterstützen neben dem Datentyp WORD (2 Byte) auch beliebige andere Datentypen.

Fehlercodes:

- Sendet der Master ein Telegramm mit unbekanntem Funktionscode, so antwortet die Steuerung mit Fehlercode 1 (Invalid Code).
- Stimmt das Telegramm des Masters nicht mit der Konfiguration der Modbus Slaves überein (d. h. z. B. das Anfragetelegramm endet nicht glatt an einer Variablengrenze), so antwortet der Slave mit Fehlercode 2 (Invalid Data).
- Sendet der Master ein Telegramm mit fehlerhaften Werten (z. B. Längenfeld), so antwortet der Slave mit Fehlercode 3 (Invalid Value).

Die Kommunikation erfolgt nur im Zustand RUN des COM Moduls. Master Anfragen in allen anderen Betriebszuständen des COM-Moduls werden nicht beantwortet.

#### **Hinweis zur Modbus Funktion: Read Device Identification (43)**

Der HIMax Modbus Slave liefert die Identifikationsdaten an den Master und unterstützt die folgenden Object-Ids:

Basic:

0x00 VendorName "HIMA Paul Hildebrandt GmbH"

0x01 ProductCode "<Modul Seriennummer>"

0x02 MajorMinorRevision "<COM Vx.y CRC>"

Regular:

0x03 VendorUrl "http://www.hima.de"

0x04 ProductName "HIMax"

0x05 ModelName "HIMax"

0x06 UserApplicationName "-----[S.R.S]"

Extended:

0x80 leer "-----"

0x81 leer "-----"

0x82 leer "-----"

0x83 leer "-----"

0x84 leer "-----"

0x85 leer "-----"

0x86 CRC der Datei modbus.config "<0x234adcef>"

(Konfigurationsdatei des Modbus-Slave-Protokolls im Dateisystem der CPU. Zu vergleichen mit den Angaben in SILworX unter Online/Versionsvergleich).

Es werden folgende ReadDevice ID Codes unterstützt:

- (1) Read Basic device identification (stream access)
- (2) Read regular device identification (stream access)
- (3) Read extended device identification (stream access)
- (4) Read one specific identification object (individual access)

Weitere Informationen zu Modbus sind in der Spezifikation *Modbus Application Protocol Specification* [www.modbus.org](http://www.modbus.org) zu finden.

## 5.9 HIMA spezifischen Funktionscodes

Die HIMA spezifischen Funktionscodes entsprechen den Standard Modbus Funktionscodes. Die einzigen Unterschiede sind die maximal zulässige Prozessdatenlänge von 1100 Bytes und das Format von Request und Response-Header:

Element	Code	Typ	Bedeutung
Read Coils Extended	100 (0x64)	BOOL	Entspricht dem Functioncode 01 Lesen mehrerer Variablen (BOOL) aus dem Import- oder Export <sup>1)</sup> -Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read Discrete Inputs Extended	101 (0x65)	BOOL	Entspricht dem Functioncode 02 Lesen mehrerer Variablen (BOOL) aus dem Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read Holding Registers Extended	102 (0x66)	WORD	Entspricht dem Functioncode 03 Lesen mehrerer Variablen beliebigen Typs aus dem Import- oder Export <sup>1)</sup> -Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read Input Registers Extended	103 (0x67)	WORD	Entspricht dem Functioncode 04 Lesen mehrerer Variablen beliebigen Typs aus dem Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Write Multiple Coils Extended	104 (0x68)	BOOL	Entspricht dem Functioncode 15 Schreiben mehrerer Variablen (BOOL) in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Write Multiple Registers Extended	105 (0x69)	WORD	Entspricht dem Functioncode 16 Schreiben mehrerer Variablen beliebigen Typs in den Import-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes
Read/Write Multiple Registers Extended	106 (0x6A)	WORD	Entspricht dem Functioncode 23 Schreiben und Lesen mehrerer Variablen beliebigen Typs in und aus dem Import-Bereich oder Export-Bereich des Slaves. Maximale Länge der Prozessdaten: 1100 Bytes (Anforderungstelegramm des Modbus Masters) 1100 Bytes (Antworttelegramm des Slaves).

Tabelle 40: HIMA spezifischen Funktionscodes

### 5.9.1 Format der Request und Response Header

Die Request und Response Header der HIMA spezifischen Modbus-Funktionscodes sind wie folgt aufgebaut:

Code	Request	Response
100 (0x64)	1 Byte Funktionscode 0x64 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 ... 8800 (0x2260)	1 Byte Funktionscode 0x64 2 Bytes Anzahl von Bytes = N N Bytes Coil-Daten (8 Coils werden in ein Byte gepackt)
101 (0x65)	1 Byte Funktionscode 0x65 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 ... 8800 (0x226)	1 Byte Funktionscode 0x65 2 Bytes Anzahl von Bytes = N N Bytes Coil-Daten (8 Coils werden in ein Byte gepackt)
102 (0x66)	1 Byte Funktionscode 0x66 2 Bytes Startadresse 2 Bytes Anzahl von Register 1 ... 550 (0x226)	1 Byte Funktionscode 0x66 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten
103 (0x67)	1 Bytes Funktionscode 0x67 2 Bytes Startadresse 2 Bytes Anzahl von Register 1 ... 550 (0x226)	1 Byte Funktionscode 0x67 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten
104 (0x68)	1 Byte Funktionscode 0x68 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 ... 8800 (0x2260) 2 Bytes Anzahl von Bytes = N N Bytes Coil-Daten	1 Byte Funktionscode 0x66 2 Bytes Startadresse 2 Bytes Anzahl von Coils 1 ... 8800 (0x2260)
105 (0x69)	1 Byte Funktionscode 0x69 2 Bytes Startadresse 2 Bytes Anzahl von Registern 1 ... 550 (0x226) 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten	1 Byte Funktionscode 0x69 2 Bytes Startadresse 2 Bytes Anzahl von Registern 1 ... 550 (0x226)
106 (0x6A)	1 Byte Funktionscode 0x6a 2 Bytes Lese-Startadresse 2 Bytes Anzahl von Leseregistern 1 ... 550 (0x226) 2 Bytes Schreib-Startadresse 2 Bytes Anzahl von Schreibregistern 1 ... 550(0x226) 2 Bytes Anzahl von Bytes zum Schreiben=N N Bytes Register-Daten	1 Byte Funktionscode 0x6a 2 Bytes Anzahl von Bytes = N N Bytes Register-Daten

Tabelle 41: Request und Response Header

## 5.10 Modbus Adressierung durch Bit und Register

Dieser Adressierungsmodus entspricht dem Standard der Modbus Adressierung und kennt nur die zwei Datenlängen Bit (1 Bit) und Register (16 Bit) mit denen alle zugelassenen Datentypen übertragen werden können.

Im Modbus Slave gibt es einen **Register Bereich** (Ein- und Ausgänge) und einen **Bit Bereich** (Ein- und Ausgänge). Beide Bereiche sind voneinander getrennt und können alle zugelassenen Datentypen aufnehmen. Der Unterschied dieser Bereiche besteht in den erlaubten Modbus-Funktionscodes, mit denen auf diese Bereiche zugegriffen werden kann.

**i**

Die Modbus Adressierung durch Bit und Register garantiert keine Variablen-Integrität, d. h. mit diesem Zugriff können beliebige Teile von Variablen gelesen/geschrieben werden. Variablen vom Typ BOOL werden gepackt abgelegt, d. h. jede Variable vom Typ BOOL ist als Bit innerhalb eines Byte abgelegt.

### 5.10.1 Register Bereich

In der Lasche **Registervariablen** werden die Variablen im Register Bereich angelegt. Weitere Informationen zur Zuweisung der Sende/Empfangsvariablen siehe Kapitel 5.6.2.

**i**

Um mit den Modbus-Funktionscodes 1, 2, 5, 15 auf die Variablen im Register Bereich zuzugreifen, müssen die Variablen in den Bit Bereich gespiegelt werden, siehe Kapitel 5.11.1.

Auf die Variablen im Register Bereich kann nur über die Modbus-Funktionscodes 3, 4, 6, 16, 23 zugegriffen werden. Dazu muss in den Eigenschaften des Funktionscodes die Startadresse der ersten Variable eingetragen werden.

Beispiel: Zugriff auf Variablen im Register Bereich des Modbus Slave

Register Variablen	Register.Bit	Bit
00_Register_Bereich_WORD	0.0	0
01_Register_Bereich_SINT	1.8	16
02_Register_Bereich_SINT	1.0	24
03_Register_Bereich_REAL	2.0	32
04_Register_Bereich_BOOL	4.8	64
05_Register_Bereich_BOOL	4.9	65
06_Register_Bereich_BOOL	4.10	66
07_Register_Bereich_BOOL	4.11	67
08_Register_Bereich_BOOL	4.12	68
09_Register_Bereich_BOOL	4.13	69
10_Register_Bereich_BOOL	4.14	70
11_Register_Bereich_BOOL	4.15	71
12_Register_Bereich_BOOL	4.0	72
13_Register_Bereich_BOOL	4.1	73
14_Register_Bereich_BOOL	4.2	74
15_Register_Bereich_BOOL	4.3	75
16_Register_Bereich_BOOL	4.4	76
17_Register_Bereich_BOOL	4.5	77
18_Register_Bereich_BOOL	4.6	78
19_Register_Bereich_BOOL	4.7	79

Tabelle 42: Register Variablen im Register Bereich des Modbus Slave

#### 5.10.1.1 Modbus Master Konfiguration des Anforderungstelegramms

**Im Modbus Master die Variablen 01\_Register\_Bereich\_SINT bis 03\_Register\_Bereich\_REAL einlesen**

1. Rechtsklick auf **TCP/UDP-Slave** und im Kontextmenü **Neu** wählen.
2. Aus der Liste **Read Holding Registers (3)** wählen.
3. Rechtsklick auf **Read Holding Registers (3)** und **Eigenschaften** wählen.
  - **Startadresse des Lesebereichs**, 1 eintragen.
4. Rechtsklick auf **Read Holding Registers (3)** und **Edit** wählen.
5. Aus der **Objektauswahl** die folgenden Variablen in das Register **Eingangsvariablen** ziehen.

Register Variablen	Offset
01_Register_Bereich_SINT	0
02_Register_Bereich_SINT	1
03_Register_Bereich_REAL	2

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Ausgangsvariablen** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.



### 5.10.2 Bit Bereich

Im Register **Bitvariablen** werden die Variablen im Bit Bereich angelegt. Weitere Informationen zur Zuweisung der Sende/Empfangsvariablen siehe Kapitel 5.6.3.

#### i

Um mit Modbus-Funktionscodes 3, 4, 6, 16, 23 auf die Variablen im Bit Bereich zuzugreifen, müssen die Variablen in den Register Bereich gespiegelt werden, siehe Kapitel 5.11.2.

Auf die Variablen im Bit Bereich kann nur über die Modbus-Funktionscodes 1, 2, 5, 15 zugegriffen werden. Dazu muss in den Eigenschaften des Funktionscodes die Startadresse der ersten Variable eingetragen werden.

Beispiel: Zugriff auf Variablen im Bit Bereich des Modbus Slave

Bit Variablen	Bit	Register.Bit
00_BIT_Bereich_WORD	0	0.0
01_BIT_Bereich_SINT	16	1.8
02_BIT_Bereich_SINT	24	1.0
03_BIT_Bereich_REAL	32	2.0
04_BIT_Bereich_BOOL	64	4.8
05_BIT_Bereich_BOOL	65	4.9
06_BIT_Bereich_BOOL	66	4.10
07_BIT_Bereich_BOOL	67	4.11
08_BIT_Bereich_BOOL	68	4.12
09_BIT_Bereich_BOOL	69	4.13
10_BIT_Bereich_BOOL	70	4.14
11_BIT_Bereich_BOOL	71	4.15
12_BIT_Bereich_BOOL	72	4.0
13_BIT_Bereich_BOOL	73	4.1
14_BIT_Bereich_BOOL	74	4.2
15_BIT_Bereich_BOOL	75	4.3
16_BIT_Bereich_BOOL	76	4.4
17_BIT_Bereich_BOOL	77	4.5
18_BIT_Bereich_BOOL	78	4.6
19_BIT_Bereich_BOOL	79	4.7

Tabelle 43: Bit Variablen im Bit Bereich des Modbus Slave

### 5.10.2.1 Modbus Master Konfiguration des Anforderungstelegramms

**Im Modbus Master die Variablen 04\_BIT\_Bereich\_BOOL bis 06\_Bereich\_BOOL einlesen**

1. Rechtsklick auf **TCP/UDP-Slave** und im Kontextmenü **Neu** wählen.
2. Aus der Liste **Read Coils (1)** wählen.
3. Rechtsklick auf **Read Coils (1)** und im Kontextmenü **Eigenschaften** wählen.
  - **Startadresse des Lesebereichs, 64** eintragen.
4. Rechtsklick auf **Read Coils (1)** und im Kontextmenü **Edit** wählen.
5. Aus der **Objektauswahl** die folgenden Variablen in das Register **Eingangsvariablen** ziehen.

Bit Variablen	Offset
04_BIT_Bereich_BOOL	0
05_BIT_Bereich_BOOL	1
06_BIT_Bereich_BOOL	2

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Ausgangsvariablen** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

## 5.11 Offsets für alternative Modbus Adressierung

Um mit den Modbus-Funktionscodes (Typ Register) auf Variablen im **Bit Bereich** und mit den Modbus-Funktionscodes (Typ Bit) auf Variablen im **Register Bereich** zuzugreifen, müssen die Variablen in den jeweils anderen Bereich gespiegelt werden.

Im Register **Eigenschaften Modbus-Slave-Set** werden die Offsets der gespiegelten Variablen eingetragen.

### Die Variablen in den Bit- und Register Bereich spiegeln

1. In der Register **Eigenschaften Modbus-Slave-Set** des Modbus Slave, **Alternative Register/Bit-Adressierung verwenden** aktivieren.  
☒ Damit werden die Variablen in den jeweils anderen Bereich gespiegelt.
2. Offset für die gespiegelten Variablen im Bit- und Register Bereich eintragen.

**i**

Die gespiegelten Variablen im Bit-/ Register Bereich und die vorhandenen Variablen im Bit-/ Register Bereich dürfen sich bzgl. der Modbus Adressen nicht überlappen.

Element	Beschreibung/ Wertebereich
Alternative Register/Bit-Adressierung verwenden	Aktiviert      Alternative Adressierung verwenden. Deaktiviert    Alternative Adressierung nicht verwenden. Standardwert: Deaktiviert
Register Bereich Offset Bits (Input)	0 ... 65535
Register Bereich Offset Bits (Output)	0 ... 65535
Bit Bereich Offset Register (Input)	0 ... 65535
Bit Bereich Offset Register (Output)	0 ... 65535

Tabelle 44: Lasche Eigenschaften Modbus-Slave-Set

### 5.11.1 Zugriff auf die Registervariablen im Bit Bereich des Modbus Slave

Um mit den Modbus-Funktionscodes (Typ Bit) 1, 2, 5, 15 auf die **Register Variablen** zuzugreifen, müssen die **Register Variablen** in den **Bit Bereich** gespiegelt werden. Im Register **Eigenschaften/Offsets** müssen die Offsets der gespiegelten **Register Variablen** eingetragen werden.

Beispiel:

Bit Bereich Offset/Register Eingänge 8000

Bit Bereich Offset/Register Ausgänge 8000

Hier liegen die aus dem Register Bereich in den Bit Bereich gespiegelten Variablen ab Bit Adresse 8000.

Gespiegelte Register Variablen	Bit
00_Register_Bereich_WORD	8000
01_Register_Bereich_SINT	8016
02_Register_Bereich_SINT	8024
03_Register_Bereich_REAL	8032
04_Register_Bereich_BOOL	8064
05_Register_Bereich_BOOL	8065
06_Register_Bereich_BOOL	8066
07_Register_Bereich_BOOL	8067
08_Register_Bereich_BOOL	8068
09_Register_Bereich_BOOL	8069
10_Register_Bereich_BOOL	8070
11_Register_Bereich_BOOL	8071
12_Register_Bereich_BOOL	8072
13_Register_Bereich_BOOL	8073
14_Register_Bereich_BOOL	8074
15_Register_Bereich_BOOL	8075
16_Register_Bereich_BOOL	8076
17_Register_Bereich_BOOL	8077
18_Register_Bereich_BOOL	8078
19_Register_Bereich_BOOL	8079

Tabelle 45: Aus dem Register Bereich in den Bit Bereich gespiegelte Variablen

#### 5.11.1.1 Modbus Master Konfiguration des Anforderungstelegramms

**Im Modbus Master die Variablen 04\_Register\_Bereich\_BOOL bis 06\_Register\_Bereich\_BOOL einlesen**

1. Rechtsklick auf **TCP/UDP-Slave** und im Kontextmenü **Neu** wählen.
2. Aus der Liste **Read Coils (1)** wählen.
3. Rechtsklick auf **Read Coils (1)** und im Kontextmenü **Eigenschaften** wählen.
  - **Startadresse des Lesebereichs, 8064** eintragen.
4. Rechtsklick auf **Read Coils (1)** und im Kontextmenü **Edit** wählen.
5. Aus der **Objektauswahl** die folgenden Variablen in das Register **Eingangsvariablen** ziehen.

Gespiegelte Register Variablen	Offset
04_Register_Bereich_BOOL	0
05_Register_Bereich_BOOL	1
06_Register_Bereich_BOOL	2

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Ausgangsvariablen** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

### 5.11.2 Zugriff auf die Bitvariablen im Registerbereich des Modbus Slave

Um mit den Modbus-Funktionscodes (Typ Register) 3, 4, 6, 16, 23 auf die **Bit Variablen** zuzugreifen, müssen die **Bit Variablen** in den **Register Bereich** gespiegelt werden. In der Lasche **Eigenschaften/Offsets** müssen die Offsets der gespiegelten **Bit Variablen** eingetragen werden.

Beispiel:

Register Bereich Offset/Bit Eingänge: 1000

Register Bereich Offset/Bit Ausgänge: 1000

Hier liegen die aus dem Bit Bereich in den Register Bereich gespiegelten Variablen ab Register Adresse 1000.

Gespiegelte Bit Variablen	Register.Bit
00_BIT_Bereich_WORD	1000.0
01_BIT_Bereich_SINT	1001.8
02_BIT_Bereich_SINT	1001.0
03_BIT_Bereich_REAL	1002.0
04_BIT_Bereich_BOOL	1004.8
05_BIT_Bereich_BOOL	1004.9
06_BIT_Bereich_BOOL	1004.10
07_BIT_Bereich_BOOL	1004.11
08_BIT_Bereich_BOOL	1004.12
09_BIT_Bereich_BOOL	1004.13
10_BIT_Bereich_BOOL	1004.14
11_BIT_Bereich_BOOL	1004.15
12_BIT_Bereich_BOOL	1004.0
13_BIT_Bereich_BOOL	1004.1
14_BIT_Bereich_BOOL	1004.2
15_BIT_Bereich_BOOL	1004.3
16_BIT_Bereich_BOOL	1004.4
17_BIT_Bereich_BOOL	1004.5
18_BIT_Bereich_BOOL	1004.6
19_BIT_Bereich_BOOL	1004.7

Tabelle 46: Aus dem Bit Bereich in den Register Bereich gespiegelte Variablen

### 5.11.2.1 Modbus Master Konfiguration des Anforderungstelegramms

**Im Modbus Master die Variablen 01\_BIT\_Bereich\_SINT bis 03\_BIT\_Bereich\_REAL einlesen**

1. Rechtsklick auf **TCP/UDP-Slave** und im Kontextmenü **Neu** wählen.
2. Aus der Liste **Read Holding Registers (3)** wählen.
3. Rechtsklick auf **Read Holding Registers (3)** und **Eigenschaften** wählen.
  - **Startadresse des Lesebereichs** 1001 eintragen.
4. Rechtsklick auf **Read Holding Registers (3)** und **Edit** wählen.
5. In der **Objektauswahl** die folgenden Variablen in das Register **Eingangsvariablen** ziehen.

Gespiegelte Bit Variablen	Offset
01_BIT_Bereich_SINT	0
02_BIT_Bereich_SINT	1
03_BIT_Bereich_REAL	2

6. Kontextmenü durch Rechtsklick auf eine leere Stelle im Bereich **Ausgangsvariablen** öffnen und **Neue Offsets** wählen, um die Offsets der Variablen neu zu nummerieren.

## 5.12 Control-Panel (Modbus Slave)

Im Control-Panel kann der Anwender die Einstellungen des Modbus Slave überprüfen und steuern. Zudem werden aktuelle Statusinformationen (z. B. Master-Zustand usw.) des Slaves angezeigt.

### Öffnen des Control Panels zur Überwachung des Modbus Slave

1. Im Strukturbaum **Hardware** und im Kontextmenü **Online** wählen.
2. Im **System-Login**, Zugangsdaten eingeben um die Online Ansicht der Hardware zu öffnen.
3. Doppelklick auf **COM-Modul** und im Strukturbaum **Modbus Slave** wählen.

### 5.12.1 Kontextmenü (Modbus Slave)

Aus dem Kontextmenü des selektierten Modbus Slave kann das folgende Kommando gewählt werden:

#### Zurücksetzen Statistik

Setzt die statistischen Daten (Zykluszeit min, max usw.) auf null zurück.

### 5.12.2 Anzeigefeld (Modbus Slave)

Im Anzeigefeld werden die folgenden Werte des selektierten Modbus Slaves angezeigt.

Element	Beschreibung
Name	Name des Modbus Slave.
Projektiertes $\mu$ P-Budget [%]	Siehe Kapitel 5.6.
Aktuelles $\mu$ P-Budget [%]	
SRS redundanter Baugruppe	SRS des redundanten COM-Moduls.
Antwortzeit [ms]	Zeitspanne nach dem Empfang einer Anfrage, innerhalb welcher der Modbus Slave antwortet.

Tabelle 47: Anzeigefeld Modbus Slave

### 5.12.3 Anzeigefeld (Masterdaten)

In dem Anzeigefeld Masterdaten werden die folgenden Werte angezeigt.

Element	Beschreibung
Name	Name der Masterdaten.
Anfragen	Gesamtanzahl aller Master Anfragen seit dem letzten Zähler Reset.
Gültige Anfragen	Anzahl gültiger Master Anfragen seit dem letzten Zähler Reset.
Ungültige Anfragen	Anzahl ungültiger Master Anfragen seit dem letzten Zähler Reset. Zu ungültigen Anfragen zählen nur Anfragen, die vom Master quittiert wurden. Fehlerhaft empfangene Anfragen mit CRC Fehler werden automatisch verworfen.
Master-Timeout [ms]	Timeoutzeit, innerhalb welcher der Slave mindestens einen Request von seinem Master erhalten haben muss. Empfängt der Slave innerhalb der Timeoutzeit keinen Request, so wird der <i>Master Verbindungsstatus</i> auf <i>nicht verbunden</i> gesetzt.
Verbindungszustand	0 = nicht überwacht. 1 = nicht verbunden. 2 = verbunden
Zeitüberschreitungen bei Antworten	Anzahl der <i>Zeitüberschreitungen bei Antworten</i> seit dem letzten Zurücksetzen aller Zähler oder Einschalten. Die <i>Zeitüberschreitungen bei Antworten</i> ist die maximale Zeit, die bis zur Empfangsbestätigung einer Nachricht beim Absender vergehen darf.



Element	Beschreibung
Verworfenene Anfragen	Anzahl der verworfenen Master Anfragen seit dem letzten Rücksetzen aller Zähler oder Einschalten.
Maximaler Pufferfüllstand für Anfragen	Maximale Anzahl gleichzeitiger Master Anfragen.
Durchschnittlicher Pufferfüllstand für Anfragen	Durchschnittliche Anzahl gleichzeitiger Master Anfragen.

Tabelle 48: Anzeigefeld Masterdaten

### 5.13 Funktion der FBx LED beim Modbus Slave

Den Zustand des Modbus Protokolls signalisiert die FBx LED der zugeordneten Feldbus-Schnittstelle. Die Zustände der FBx LED sind in der folgenden Tabelle dargestellt:

LED	Farbe	Status	Bedeutung
FBx	Gelb	Blinken	Das Modbus-Slave-Protokoll ist aktiv und befindet sich im Datenaustausch mit dem Modbus Master.
		Aus	Das Modbus-Slave-Protokoll ist nicht aktiv! Das heißt, die Steuerung ist im Zustand STOPP oder es ist kein Modbus Master konfiguriert.

Tabelle 49: LED FBx

### 5.14 Funktion der FAULT LED beim Modbus Slave (Nur HIMax)

Eine Störung des Modbus Protokolls signalisiert die FAULT LED der zugeordneten Feldbus-Schnittstelle. Die Zustände der FAULT LED sind in der folgenden Tabelle dargestellt:

LED	Farbe	Status	Bedeutung
Fault	Rot	Blinken <sup>1)</sup>	Modbus-Slave-Protokoll ist gestört. Folgende Ereignisse führen zur Störung: <ul style="list-style-type: none"> <li>▪ Unbekannter Funktionscode empfangen.</li> <li>▪ Anfrage mit falscher Adressierung empfangen.</li> <li>▪ Die Protokoll-Last erreicht das Maximum des konfigurierten Rechenzeitbudgets. Die Protokollausführung wird durch die Lastbegrenzung limitiert. Dies kann zu erheblichen Einschränkungen der Protokollfunktion führen. Es müssen entweder die Einstellungen des Protokolls oder die Einstellungen des Rechenzeitbudgets angepasst werden.</li> </ul> Tritt länger als 5 s kein Fehlerereignis auf, so wechselt die Anzeige in den Zustand <i>Protokoll ohne Störung</i> .
		Aus	Modbus-Slave-Protokoll ist ohne Störung.

<sup>1)</sup> Blinkfrequenz: lang (600 ms) an, lang (600 ms) aus.

Tabelle 50: FAULT LED

## 6 Allgemein

In diesem Kapitel sind Parameter gesammelt, die für alle Kommunikationsprotokolle relevant sind.

### 6.1 Maximale Kommunikationszeitscheibe

Die maximale Kommunikationszeitscheibe ist die zugeteilte Zeit in Millisekunden (ms) pro CPU-Zyklus, innerhalb der das Prozessormodul die Kommunikationsaufgaben abarbeitet. Wenn die Protokollverarbeitung innerhalb der Dauer einer Kommunikationszeitscheibe nicht beendet werden konnte, führt die CPU dennoch die sicherheitsrelevanten Überwachungen für alle Protokolle in einem CPU-Zyklus aus.

---

i

Wenn nicht alle in einem CPU-Zyklus anstehenden Kommunikationsaufgaben ausgeführt werden können, erfolgt die komplette Übertragung der Kommunikationsdaten über mehrere CPU-Zyklen. Die Anzahl der Kommunikationszeitscheiben ist dann größer 1.

Für die Berechnungen der zulässigen maximalen Reaktionszeiten gilt die Bedingung, dass die Anzahl der Kommunikationszeitscheiben genau 1 ist.

---

#### 6.1.1 Ermitteln der maximalen Dauer der Kommunikationszeitscheibe

Für eine erste Abschätzung der maximalen Dauer der Kommunikationszeitscheibe müssen die folgenden Zeiten aufsummiert und das Ergebnis in den Systemparameter *Max. Kom.-Zeitscheibe [ms]* in den Eigenschaften der Ressource eingetragen werden:

- Pro COM-Modul 3 ms.
- Pro redundante **safeethernet** Verbindung 1 ms.
- Pro nicht redundante **safeethernet** Verbindung 0,5 ms.
- Pro KByte Nutzdaten bei nichtsicheren Protokollen (z. B. Modbus) 1 ms.

HIMA empfiehlt, den abgeschätzten Wert *Max. Kom.-Zeitscheibe [ms]* mit dem im Control Panel angezeigten Wert zu vergleichen und gegebenenfalls in den Eigenschaften der Ressource zu korrigieren. Dies kann z. B. in einem FAT (Factory Acceptance Test) oder SAT (Site Acceptance Test) durchgeführt werden.

#### Ermitteln der tatsächlichen Dauer der maximalen Kommunikationszeitscheibe

1. Das HIMA System unter voller Last betreiben (FAT, SAT):  
Alle Kommunikationsprotokolle sind in Betrieb (**safeethernet** und Standardprotokolle).
2. Das **Control Panel** öffnen und im Strukturbaum das Verzeichnis **Kom.-Zeitscheibe** wählen.
3. Anzeige *Maximale Kom.-Zeitscheibe Dauer pro Zyklus [ms]* auszulesen.
4. Anzeige *Maximale Anzahl benötigter Kom.-Zeitscheibe Zyklen* auszulesen.

Die Dauer der Kommunikationszeitscheibe ist so hoch einzustellen, dass der CPU-Zyklus die vom Prozess vorgegebene Watchdog-Zeit nicht überschreiten kann, wenn er die eingestellte Kommunikationszeitscheibe ausnutzt.

### 6.2 Lastbegrenzung

Für jedes Kommunikationsprotokoll kann ein Rechenzeitbudget in % (*μP-Budget*) vorgegeben werden. So kann die verfügbare Rechenzeit zwischen den konfigurierten Protokollen verteilt werden. Die Summe der Rechenzeitbudgets aller parametrisierten Kommunikationsprotokolle eines CPU- oder COM-Moduls darf nicht größer als 100 % sein.

Die festgelegten Rechenzeitbudgets der einzelnen Kommunikationsprotokolle werden überwacht. Hat ein Kommunikationsprotokoll sein Rechenzeitbudget erreicht oder überschritten und es steht keine zusätzliche Rechenzeit als Reserve zur Verfügung, so wird das Kommunikationsprotokoll nicht komplett abgearbeitet.

Wenn noch genügend zusätzliche Rechenzeit vorhanden ist, wird diese verwendet, um ein Kommunikationsprotokoll, das sein Rechenzeitbudget erreicht oder überschritten hat noch abzuarbeiten. Dadurch kann es vorkommen, dass ein Kommunikationsprotokoll tatsächlich ein höheres Rechenzeitbudget verwendet als ihm zugeteilt wurde.

Eventuell werden über 100 % Rechenzeitbudget online angezeigt. Dies ist kein Fehler, das Rechenzeitbudget über 100 % ist die zusätzlich verwendete Rechenzeit.

---

**i**

Das zusätzliche Rechenzeitbudget ist keinesfalls eine Zusicherung für ein bestimmtes Kommunikationsprotokoll und kann jederzeit vom System zurückgenommen werden.

---



## Anhang

### Glossar

Begriff	Beschreibung
ARP	Address Resolution Protocol: Netzwerkprotokoll zur Zuordnung von Netzwerkadressen zu Hardwareadressen.
Bit-Variable	Variable, die bitweise adressiert wird.
CENELEC	Comité Européen de Normalisation Électrotechnique (Europäisches Komitee für elektrotechnische Normung)
Connector Board	Anschlusskarte für HIMax Modul.
COM	Kommunikationsmodul
CPU	Prozessormodul
CRC	Cyclic Redundancy Check, Prüfsumme
Dataview	Einer Dataview sind die Globalen Variablen für Eingangs- und Ausgangsdaten für den Zugriff durch Modbus-Quellen zugeordnet.
EN	Europäische Normen
Exportbereich	Als Exportbereich wird die Prozessdatenmenge bezeichnet, die vom System (aus einem Anwenderprogramm, HW-Eingang oder einem anderen Protokoll) geschrieben und vom Modbus-Master gelesen werden kann.
FB	Feldbus
FBS	Funktionsbausteinsprache
ICMP	Internet Control Message Protocol: Netzwerkprotokoll für Status- und Fehlermeldungen.
IEC	Internationale Normen für die Elektrotechnik.
Importbereich	Als Importbereich wird die Prozessdatenmenge bezeichnet, die vom Modbus-Master geschrieben wird und als Eingangsdaten für das System (in einem Anwenderprogramm, HW-Ausgang oder einem anderen Protokoll) verwendet werden kann.
KE	Kommunikationsendpunkt
MAC-Adresse	Hardware-Adresse eines Netzwerkanschlusses (Media Access Control).
NSIP	Nicht-sicherheitsbezogenes Protokoll.
PADT	Programming and Debugging Tool (nach IEC 61131-3), PC mit SILworX.
PE	Schutzerde
PELV	Protective Extra Low Voltage: Funktionskleinspannung mit sicherer Trennung.
PES	Programmierbares Elektronisches System
R	Read
Rack-ID	Identifikation eines Basisträgers (Nummer).
rückwirkungsfrei	Es seien zwei Eingangsschaltungen an dieselbe Quelle (z. B. Transmitter) angeschlossen. Dann wird eine Eingangsschaltung „rückwirkungsfrei“ genannt, wenn sie die Signale der anderen Eingangsschaltung nicht verfälscht.
R/W	Read/Write
Register-Variable	Variable, die wortweise adressiert wird.
SB	Systembusmodul
SFF	Safe Failure Fraction, Anteil der sicher beherrschbaren Fehler.
SIF	Sicherheitstechnische Funktion
SIL	Safety Integrity Level (nach IEC 61508)
SILworX	Programmiersoftware für HIMax, HIQuad X und HIMatrix.
SIP	Sicherheitsbezogenes Protokoll
SNTP	Simple Network Time Protocol (RFC 1769)
SRS	System.Rack.Slot
SW	Software

Begriff	Beschreibung
TMO	Timeout
W	Write
WD	Watchdog
WDZ	Watchdog-Zeit

## Abbildungsverzeichnis

<b>Bild 1:</b>	<b>Kommunikation über Modbus TCP</b>	<b>14</b>
<b>Bild 2:</b>	<b>Modbus-Netzwerk</b>	<b>29</b>
<b>Bild 3:</b>	<b>Modbus Gateway</b>	<b>32</b>
<b>Bild 4:</b>	<b>Serieller Modbus</b>	<b>35</b>
<b>Bild 5:</b>	<b>Modbus Telegramm</b>	<b>35</b>

## Tabellenverzeichnis

<b>Tabelle 1:</b>	<b>Zusätzlich geltende Handbücher</b>	<b>5</b>
<b>Tabelle 2:</b>	<b>Ausstattung und Systemanforderung Modbus Master</b>	<b>13</b>
<b>Tabelle 3:</b>	<b>Eigenschaften Modbus Master</b>	<b>14</b>
<b>Tabelle 4:</b>	<b>Systemvariablen Modbus Master</b>	<b>20</b>
<b>Tabelle 5:</b>	<b>Allgemeine Eigenschaften Modbus Master</b>	<b>21</b>
<b>Tabelle 6:</b>	<b>Parameter COM/CPU</b>	<b>22</b>
<b>Tabelle 7:</b>	<b>Modbus-Funktionscodes</b>	<b>23</b>
<b>Tabelle 8:</b>	<b>HIMA spezifische Funktionscodes</b>	<b>24</b>
<b>Tabelle 9:</b>	<b>Request und Response Header</b>	<b>25</b>
<b>Tabelle 10:</b>	<b>Anforderungstelegramm Read Coils</b>	<b>26</b>
<b>Tabelle 11:</b>	<b>Anforderungstelegramm Read Discrete Inputs</b>	<b>26</b>
<b>Tabelle 12:</b>	<b>Anforderungstelegramm Read Holding Registers</b>	<b>26</b>
<b>Tabelle 13:</b>	<b>Anforderungstelegramm Read Input Registers</b>	<b>26</b>
<b>Tabelle 14:</b>	<b>Register Read Write Holding</b>	<b>27</b>
<b>Tabelle 15:</b>	<b>Anforderungstelegramm Write Multiple Coils</b>	<b>28</b>
<b>Tabelle 16:</b>	<b>Anforderungstelegramm Write Multiple Registers</b>	<b>28</b>
<b>Tabelle 17:</b>	<b>Anforderungstelegramm Write Single Coil (05)</b>	<b>28</b>
<b>Tabelle 18:</b>	<b>Anforderungstelegramm Write Single Register</b>	<b>28</b>
<b>Tabelle 19:</b>	<b>Systemvariablen TCP/UDP-Slaves</b>	<b>30</b>
<b>Tabelle 20:</b>	<b>Konfigurationsparameter</b>	<b>31</b>
<b>Tabelle 21:</b>	<b>Verbindungsparameter Modbus Gateway</b>	<b>34</b>
<b>Tabelle 22:</b>	<b>Statusvariablen Gateway-Slave</b>	<b>34</b>
<b>Tabelle 23:</b>	<b>Verbindungsparameter Gateway-Slave</b>	<b>34</b>
<b>Tabelle 24:</b>	<b>Parameter serieller Modbus Master</b>	<b>36</b>
<b>Tabelle 25:</b>	<b>Systemvariablen Modbus Slave</b>	<b>37</b>

<b>Tabelle 26: Verbindungsparameter Modbus Master</b>	<b>37</b>
<b>Tabelle 27: Anzeigefeld Modbus Master</b>	<b>38</b>
<b>Tabelle 28: LED FBx</b>	<b>39</b>
<b>Tabelle 29: FAULT LED</b>	<b>39</b>
<b>Tabelle 30: Ausstattung und Systemanforderung Modbus Slave</b>	<b>40</b>
<b>Tabelle 31: Eigenschaften Modbus Slave</b>	<b>41</b>
<b>Tabelle 32: Zulässige Steckplätze der redundanten Modbus Slave HIMax COM-Module</b>	<b>43</b>
<b>Tabelle 33: Register Eigenschaften Modbus Slave Set</b>	<b>45</b>
<b>Tabelle 34: Dialogfenster Neue Offsets</b>	<b>46</b>
<b>Tabelle 35: Dialogfenster Neue Offsets</b>	<b>47</b>
<b>Tabelle 36: Register Modbus Slave Set</b>	<b>47</b>
<b>Tabelle 37: Register TCP- und UDP-Ports für Modbus Slave</b>	<b>49</b>
<b>Tabelle 38: Register Systemvariablen für Modbus Slave</b>	<b>50</b>
<b>Tabelle 39: Modbus-Funktionscodes des Modbus Slave</b>	<b>51</b>
<b>Tabelle 40: HIMA spezifischen Funktionscodes</b>	<b>53</b>
<b>Tabelle 41: Request und Response Header</b>	<b>54</b>
<b>Tabelle 42: Register Variablen im Register Bereich des Modbus Slave</b>	<b>55</b>
<b>Tabelle 43: Bit Variablen im Bit Bereich des Modbus Slave</b>	<b>57</b>
<b>Tabelle 44: Lasche Eigenschaften Modbus-Slave-Set</b>	<b>59</b>
<b>Tabelle 45: Aus dem Register Bereich in den Bit Bereich gespiegelte Variablen</b>	<b>60</b>
<b>Tabelle 46: Aus dem Bit Bereich in den Register Bereich gespiegelte Variablen</b>	<b>62</b>
<b>Tabelle 47: Anzeigefeld Modbus Slave</b>	<b>64</b>
<b>Tabelle 48: Anzeigefeld Masterdaten</b>	<b>65</b>
<b>Tabelle 49: LED FBx</b>	<b>65</b>
<b>Tabelle 50: FAULT LED</b>	<b>65</b>

Für weitere Informationen kontaktieren Sie:

**HIMA Paul Hildebrandt GmbH**

Albert-Bassermann-Str. 28  
68782 Brühl, Germany

Telefon +49 6202 709-0  
Fax +49 6202 709-107  
E-Mail [info@hima.com](mailto:info@hima.com)

Erfahren Sie online mehr über HIMA Lösungen:



[www.hima.com/de/](http://www.hima.com/de/)