



Protokoll Handbuch

# Modbus Slave V2



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
9.00	Erstausgabe des Handbuchs für SILworX V9		
11.00	Erweiterte Version des Handbuchs für SILworX V11	X	X

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Aufbau und Gebrauch des Handbuchs	7
1.2	Zielgruppe	8
1.3	Darstellungskonventionen	9
1.3.1	Sicherheitshinweise	9
1.3.2	Gebrauchshinweise	10
1.4	Safety Lifecycle Services	11
<b>2</b>	<b>Sicherheit</b>	<b>12</b>
2.1	Bestimmungsgemäßer Einsatz	12
2.2	Restrisiken	12
2.3	Sicherheitsvorkehrungen	12
2.4	Notfallinformationen	12
2.5	Automation-Security bei HIMA Systemen	12
<b>3</b>	<b>Produktbeschreibung</b>	<b>14</b>
3.1	Benötigte Ausstattung und Systemanforderung	14
3.2	System Mengengerüste für Modbus-Slave-V2	14
3.3	Unterschiede zwischen Modbus-Slave Set und Modbus-Slave V2	15
3.4	Allgemeine Benutzerauflagen	15
3.5	Konzept des Datenzugriffs auf globale Variablen	16
3.5.1	Durch Anwenderprogramm beschriebene Eingangsvariablen	17
3.5.2	Hinweise für Schreibanforderungen:	17
3.5.3	Dataview	17
3.5.4	Dataview-Link	17
3.5.5	Verhalten bei Verbindungsverlust von Dataviews	17
3.5.6	Retain-Verhalten von globalen Variablen	18
<b>4</b>	<b>Redundanz</b>	<b>19</b>
4.1	Logische Redundanz	19
4.2	COM-Redundanz zweier Kommunikationsmodule	20
4.2.1	Unterstützung bei der Konfiguration der redundanten Kommunikationsmodule	20
4.2.1.1	Einschränkung bei der Konfiguration (identische Master)	21
4.2.1.2	Kombination der Redundanzen	21
4.3	Hinweise zur Konfiguration einer Redundanz zweier Kommunikationsmodule auf einem HIMax System	21
4.3.1	Regeln für den redundanten HIMax Modbus-Slave-V2	21
4.3.2	Zulässige Steckplätze der redundanten HIMax Modbus-Slave-V2 COM-Module	21
4.3.3	Redundante Modbus-Slave COM-Module in unterschiedlichen Basisträgern	21
<b>5</b>	<b>Reload</b>	<b>23</b>
5.1	Änderungen der Modbus-Slave-V2-Konfiguration	23
<b>6</b>	<b>Anlegen des Modbus-Slave-V2 in SILworX</b>	<b>24</b>
6.1	Registrierung und Aktivierung des Protokolls	24
6.2	Anlegen des Modbus-Slave-Set-V2 Protokolls	24
6.3	Anlegen eines Dataview	25

<b>6.4</b>	<b>Konfiguration des Modbus-Slave</b>	<b>25</b>
6.4.1	Im Modbus-Slave die Verbindung zu dem Modbus-Master erstellen	25
6.4.1.1	Dataview-Link mit Dataview referenzieren	26
<b>6.5</b>	<b>Konfiguration des redundanten Modbus-Slave</b>	<b>26</b>
<b>7</b>	<b>Beschreibung der Modbus-Slave-V2-Editoren und Objekte</b>	<b>27</b>
<b>7.1</b>	<b>Modbus-Slave-Set-V2</b>	<b>27</b>
7.1.1	Set-Objekte-Tab	27
7.1.2	Eigenschaften-Tab	27
<b>7.2</b>	<b>Dataviews</b>	<b>28</b>
7.2.1	Dataview	28
7.2.1.1	Register-Variablen-Tab	28
7.2.1.2	Bit-Variablen-Tab	28
7.2.1.3	Eigenschaften-Tab	28
<b>7.3</b>	<b>Modbus-Slave</b>	<b>29</b>
7.3.1	Master-Objekte-Tab	29
7.3.2	Systemvariablen-Tab	29
7.3.3	Eigenschaften-Tab	29
<b>7.4</b>	<b>Modbus-Master</b>	<b>30</b>
7.4.1	Dataview-Links-Tab	30
7.4.2	Systemvariablen-Tab	30
7.4.3	Eigenschaften-Tab	30
<b>7.5</b>	<b>Dataview-Link</b>	<b>33</b>
7.5.1	Systemvariablen-Tab	33
7.5.2	Eigenschaften-Tab	34
<b>8</b>	<b>Modbus-Funktionscodes</b>	<b>35</b>
<b>8.1</b>	<b>Standard Modbus-Funktionscodes</b>	<b>35</b>
<b>8.2</b>	<b>Byte-Order</b>	<b>37</b>
<b>9</b>	<b>Anzeige und Diagnose im Control Panel</b>	<b>38</b>
<b>9.1</b>	<b>Online-Anzeige des Modbus</b>	<b>38</b>
<b>10</b>	<b>Versionsvergleich</b>	<b>40</b>
<b>10.1</b>	<b>Die für das Modbus-Slave-V2-Protokoll relevanten Konfigurationsdateien</b>	<b>40</b>
<b>10.2</b>	<b>Globale Multisource-Variable</b>	<b>41</b>
10.2.1	Identifikation der Globalen Multisource-Variablen im Versionsvergleich	41
<b>11</b>	<b>Modbus-Topologien mit Prozessleitsystemen</b>	<b>43</b>
<b>11.1</b>	<b>Einschränkung des Modbus-Slave durch externen Transportweg</b>	<b>43</b>
<b>11.2</b>	<b>Modbus über einen Transportweg</b>	<b>43</b>
11.2.1	Einzelne Linie ohne Redundanz	43
<b>11.3</b>	<b>Modbus über zwei Transportwege</b>	<b>44</b>
11.3.1	Beide Linien aktiv (beide Linien lesen und schreiben)	44
11.3.2	Beide Linien aktiv mit optional umschaltbaren Linien	45
11.3.3	Cold Standby (eine Linie aktiv, die zweite Linie dient als Backup)	45
11.3.4	Hot-Standby (beide Linien lesen und nur eine Linie schreibt)	46

<b>12</b>	<b>Allgemeines zur Datenübertragung</b>	<b>47</b>
<b>12.1</b>	<b>Erlaubte IP-Adressen Kombinationen des Masters</b>	<b>47</b>
12.1.1	Verwendete Netzwerkports für Ethernet-Kommunikation	47
<b>12.2</b>	<b>Serieller Modbus</b>	<b>47</b>



# 1 Einleitung

Das Modbus-Slave-V2 Handbuch beschreibt die Eigenschaften und die Konfiguration des Modbus-Slave-V2-Protokolls für die sicherheitsbezogenen Steuerungssysteme HIMax und HIMatrix 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
- Beschreibung der Konfiguration des Modbus-Slave-V2-Protokolls in SILworX

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 und Programmierer von Automatisierungsanlagen sowie Personen, die zu Inbetriebnahme, Betrieb und Wartung der Anlagen und Systeme 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 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

Industrielle Steuerungen müssen gegen IT-typische Problemquellen geschützt werden. Diese Problemquellen sind:

- Angreifer innerhalb und außerhalb der Kundenanlage
- Bedienungsfehler
- Software-Fehler

Die Anforderungen der Sicherheits- und Anwendungsnormen bezüglich des Schutzes vor Manipulationen sind zu beachten. Die Autorisierung von Personal und die notwendigen Schutzmaßnahmen unterliegen der Verantwortung des Betreibers.

#### **WARNUNG**



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

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

**Beispielsweise:**

- die Standardeinstellungen für Login und Passwort ändern.
- physischen Zugang zur Steuerung und zum PADT kontrollieren!

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 Netzwerkpläne helfen sicherzustellen, dass secure 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.

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

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

### 3 Produktbeschreibung

Das Modbus-Slave-V2-Protokoll 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.

Das Modbus-Slave-V2-Protokoll ermöglicht HIMA Steuerungen den Prozessdatenaustausch mit Fremdsystemen über die RS485-Schnittstellen oder über die Ethernet-Schnittstellen der Steuerungen. Modbus-Slave-V2 darf dabei nicht zur sicherheitsbezogenen Kommunikation verwendet werden.

#### 3.1 Benötigte Ausstattung und Systemanforderung

Die Ethernet-Schnittstellen des Prozessormoduls können für Modbus TCP nicht verwendet werden.

Element	Beschreibung
Steuerung und Betriebssystem	HIMax CPU-Betriebssysteme $\geq$ V11 COM-Betriebssysteme $\geq$ V11
	HIQuad X CPU-Betriebssysteme $\geq$ V11 COM-Betriebssysteme $\geq$ V11
	HIMatrix F CPU-Betriebssysteme $\geq$ V13 COM-Betriebssysteme $\geq$ V18
Programmierwerkzeug	SILworX $\geq$ V11
Aktivierung	Die Freischaltung erfolgt per Software-Freischaltcode, siehe Kapitel 6.1.

Tabelle 2: Nötige Systemanforderung zum Betreiben des Modbus-Slave-V2

#### 3.2 System Mengengerüste für Modbus-Slave-V2

Das HIMA System unterstützt für Modbus-Slave-V2 folgende Eigenschaften.

Eigenschaften	Beschreibung
Max. Anzahl Modbus-Slave-V2-Protokolle	HIMax: Maximal 20 Modbus-Slave-V2-Protokolle, 1 pro X-COM 01 Modul HIQuad X: Maximal 20 Modbus-Slave-V2-Protokolle, 1 pro F-COM 01 Modul HIMatrix: Maximal 1 Modbus-Slave-V2-Protokoll
Anzahl Modbus V2 Masterzugriffe	RTU: Aufgrund der RS485 Übertagungstechnik kann jeweils nur ein Modbus-Master auf eine installierte RS485 Schnittstelle zugreifen TCP/UDP: Maximal 20 Modbus-Master können auf den Slave zugreifen.
Max. Anzahl TCP-Verbindungen	Maximal 20 TCP-Verbindungen sind einstellbar. Der Parameter <i>Anzahl TCP-Verbindungen</i> muss für jeden Modbus-Master eingestellt werden. In der Summe sind maximal 20 TCP-Verbindungen zu den konfigurierten Modbus-Mastern erlaubt.
Max. Anzahl Dataviews	Maximal 20 Dataviews pro COM-Modul

Eigenschaften	Beschreibung
Max. Menge der Eingangsprozessdaten einer Dataview	8192 BOOLs oder 4096 Register. Eine BOOL-Variable muss bzgl. der Kalkulation der maximal zur Verfügung stehenden Prozessdatenmenge immer mit 1 Byte kalkuliert werden, da sie innerhalb des Systems ungepackt verarbeitet wird. Dies gilt unabhängig davon, ob eine BOOL-Variable über Modbus gepackt (als Bit) oder ungepackt (als Byte) transferiert wird.
Max. Menge der Prozessdaten je Richtung pro Steuerung über alle Dataviews	HIMax: 128 kB senden, 128 kB empfangen HIQuad X: 64 kB senden, 64 kB empfangen HIMatrix: 64 kB senden, 64 kB empfangen
Max. Anzahl Anforderungstelegramme pro COM-Modul	Modbus-Slave kann pro COM-Modul gleichzeitig bis zu 64 Anforderungstelegramme (Empfangspuffer) von Modbus-Mastern bearbeiten und beantworten. Weitere Anforderungstelegramme werden verworfen.

Tabelle 3: Systemeigenschaften für Modbus-Slave-V2

### 3.3 Unterschiede zwischen Modbus-Slave Set und Modbus-Slave V2

Eigenschaft	Modbus-Slave Set	Modbus-Slave V2
Schreibanforderung	Globale Variablen werden in jedem System-Zyklus an das Prozessormodul gesendet.	Globale Variablen werden nach einer Schreibanforderung einmalig beschrieben und an das Prozessormodul gesendet, siehe Kapitel 3.5.
Schreibantwort	Die Schreibantwort wird unmittelbar nach der Speicherung der Prozesswerte auf dem Prozessormodul an den Modbus-Master gesendet.	Die Schreibantwort wird erst nach Speicherung, Verarbeitung und Bestätigung der Prozesswerte durch das Prozessormodul an den Modbus-Master gesendet.
Leseantwort	Response Telegramm des Modbus-Slaves auf eine Leseanfrage. In redundanter Konfiguration senden beide Protokollvarianten die jeweils älteren Prozessdatenwerte der redundanten Partner.	
Reload	Bei Änderungen am Protokoll im Rahmen eines Reloads erfolgt auf der COM ein Cold Reload, d.h. sie wird zwischenzeitlich gestoppt.	Reloadbar, siehe Tabelle 6.
Funktionscode	Standard und HIMA spezifische	Standard
Master Identifikation	keine	Das System bestimmt die Identität eines Masters anhand dessen externen Transportwegs (Feldbusschnittstelle oder IP-Adresse)

Tabelle 4: Unterschiede von Modbus-Slave Set und Modbus-Slave Variante2

### 3.4 Allgemeine Benutzerauflagen

Der Anwender muss bei der Inbetriebnahme kontrollieren, dass die Zuordnung der über Modbus-Slave-V2 transportierten Variablen im Slave zu den Variablen im Master wie gewünscht erfolgt. Es ist nicht möglich, Modbus-Slave Set und Modbus-Slave-V2 gleichzeitig auf einem COM-Modul zu betreiben. Dies wird bereits bei der Codegenerierung durch das PADT abgelehnt.



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 Routinginträge besitzen.

### 3.5 Konzept des Datenzugriffs auf globale Variablen

Der schreibende Zugriff auf eine globale Variable erfolgt in Modbus-Slave-V2 aufgrund einer Schreibanforderung einmalig oder zyklisch durch ein Anwenderprogramm.

Die globalen Variablen werden aktualisiert, sobald eine entsprechende Schreibanforderung über das Modbus-Slave-V2-Protokoll empfangen wurde. Dabei bestimmt jeweils das zeitlich letzte Schreiben (einer Modbus-Quelle oder eines Anwenderprogramms) den Wert der globalen Variable. Dieses Verhalten kann z. B. auch zum Aufbau einer Redundanz genutzt werden, siehe Kapitel 4.

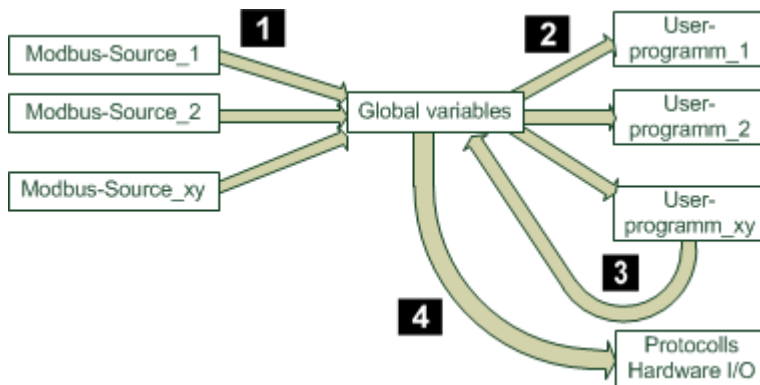


Bild 1: Konzept des Datenzugriffs auf eine globale Variable

Bild 1 zeigt die unterschiedlichen möglichen Zugriffe auf eine einzelne globale Variable bei Verwendung des Modbus-Slave-V2-Protokolls in deren zeitlichen Abfolge innerhalb eines Systemzyklus. Im Versionsvergleich werden diese globalen Variablen auch globale Multisource-Variablen genannt, siehe Kapitel 10.2.

- 1** Eine oder mehrere Modbus-Quellen können die globale Variable beschreiben, abhängig davon ob für die jeweilige Modbus-Quelle gerade eine Anforderung vorliegt oder nicht. Der Wert der globalen Variable wird durch die Modbus-Quelle bestimmt, die sie als letztes beschrieben hat.
- 2** Ein oder mehrere Anwenderprogramme können die globale Variable lesen, um sie in deren Programmlogik weiterzuverarbeiten. Das Lesen erfolgt nicht unbedingt in jedem Systemzyklus, falls sich die Ausführung des Anwenderprogramms über mehrere Systemzyklen erstreckt.
- 3** Genau 1 oder kein Anwenderprogramm kann die globale Variable nach der Verarbeitung der Programmlogik beschreiben. Das Beschreiben erfolgt nicht unbedingt in jedem Systemzyklus, falls sich die Ausführung des Anwenderprogramms über mehrere Systemzyklen erstreckt.
- 4** Verschiedene Verbraucher, wie Protokolle oder E/A-Module, können die globale Variable lesen, um sie weiterzuverarbeiten, d.h. i.d.R. nach extern auszugeben. Das Lesen erfolgt in jedem Systemzyklus.

## i

**Globale Multisource-Variablen dürfen nicht für die Sicherheitsfunktionen des Anwenderprogramms verwendet werden!**



### 3.5.1 Durch Anwenderprogramm beschriebene Eingangsvariablen

Wird eine Eingangsvariable außer von Modbus-Quellen auch von einem Anwenderprogramm beschrieben, wird die Eingangsvariable einmalig auf ihren Initialwert gesetzt sobald das Anwenderprogramm in einen der folgenden Zustände wechselt:

- STOPP
- FEHLERSTOPP
- Kaltstart des Anwenderprogramms

### 3.5.2 Hinweise für Schreibanforderungen:

- Bei Schreibanforderungen von mehreren Modbus-Quellen innerhalb eines Zyklus der Steuerung, ist die Reihenfolge der Bearbeitung der Schreibanforderungen nicht vorhersagbar. Auch aus der Reihenfolge der Schreibanworten ist diese Bearbeitung nicht ableitbar. Der Wert der letzten Schreibanforderung wird im Anwenderprogramm wirksam.
- Prinzipbedingt gibt es keine Garantie, dass ein konkreter Modbus-Master den Wert, den er selbst geschrieben hat, zuverlässig zurücklesen wird. Es sei denn, dieser Modbus-Master ist der einzige, der diesen Adressbereich beschreibt.
- Wird ein Anwenderprogramm nicht in einem Zyklus abgearbeitet (z. B. wegen einer entsprechenden Parametrierung der Priorität des Anwenderprogramms oder Testmodus), kann nicht sichergestellt werden, dass die Auswertung und Rückkopplung einer globalen Variable im gleichen Zyklus im Anwenderprogramm wirksam wird. Die Folge ist, dass der durch die Schreibanforderung geschriebene Wert einer globalen Variable erhalten bleibt (eben nicht durch das Anwenderprogramm überschrieben) und über die Ausgangsdaten wirksam wird.
- Ein durch eine Schreibanforderung geschriebener Wert einer globalen Variable, die auch vom Anwenderprogramm beschrieben wird, kann nach außen wirken, wenn das Anwenderprogramm gestoppt ist.
- Bei einem Reload kann sich die Antwortzeit auf Schreibanforderungen zeitweise erhöhen.
- Die minimale Antwortzeit auf eine Schreibanforderung (Demand) beträgt 2 CPU-Zyklen. Für möglichst geringe Antwortzeiten ist die *Schedule-Timeslice-Async* so hoch einzustellen, dass 1 CPU-Zyklus immer für die Bearbeitung aller Prozessdaten ausreicht.
- Durch die Bearbeitung von Schreibanforderungen ist mit zeitweisen Verlängerungen der CPU-Zykluszeiten zu rechnen, die bei der Bestimmung der *Schedule-Timeslice-Async* und der CPU-Watchdog-Zeit berücksichtigt werden müssen. Der CPU-Zyklus erhöht sich dabei um maximal 1 ms pro 1 kByte Prozessdaten. Die Erhöhung der CPU-Zykluszeit beträgt maximal 20 ms, d. h. bei mehr als 20 kByte Prozessdaten ist mit keiner weiteren Erhöhung der CPU-Zykluszeit zu rechnen.

### 3.5.3 Dataview

Einer Dataview sind globale Variablen für den Zugriff durch Modbus-Quellen zugeordnet. Die Dataview definieren Adressbereiche (Modbus-Offsets) auf dem System, siehe Kapitel 7.2.1. Auf die Eingangsvariablen einer Dataview wird mit entsprechenden Schreibanforderungen der Modbus-Quellen schreibend zugegriffen.

### 3.5.4 Dataview-Link

Dataview-Links werden im Modbus-Slave-V2 unterhalb des Master-Objekts angelegt. Eine Dataview kann über Dataview-Links einem oder mehreren Master-Objekten zugeordnet werden.

### 3.5.5 Verhalten bei Verbindungsverlust von Dataviews

Das Verhalten bei Verbindungsverlust zwischen einem Master und einem Dataview wird im jeweiligen Dataview des Modbus-Slaves konfiguriert. Hier kann das Verhalten zwischen *Initialwert annehmen* und *Letzten Wert behalten* gewählt werden. Dieses Verhalten gilt dann für alle Eingangsvariablen des jeweiligen Dataviews.

**i**

Werden globale Variablen von mehreren Modbus-Quellen beschrieben, kann im Dataview nur die Option *Letzten Wert beibehalten* verwendet werden, sonst wird die Codegenerierung abgelehnt.

---

### 3.5.6 Retain-Verhalten von globalen Variablen

Um globale Variablen, die mit Schreibanforderung beschrieben wurden, im nichtflüchtigen Speicher zu speichern sind die folgenden Voraussetzung nötig:

- Die globale Variablen müssen in einem Anwenderprogramm sowohl gelesen als auch beschrieben werden.
- Für die globalen Variablen muss die Eigenschaft Retain gesetzt sein.

Eine solche als Retain deklarierte Demand-Variable, die von einem Anwenderprogramm beschrieben wird, wird beim Warmstart des Anwenderprogramms mit ihrem Retain-Wert beschrieben.

## 4 Redundanz

Die redundante Konfiguration des Modbus-Slave-V2-Protokolls ist durch die Verwendung von Dataviews vereinfacht worden. Dieses Mehrfachbeschreiben aus unterschiedlichen Modbus-Quellen kann zum Aufbau einer Redundanz genutzt werden. Eine globale Variable eines Slaves kann über Dataviews von allen Modbus-Quellen beschrieben werden. Dabei kann ein Schreiben nur über eine Schreibanforderung einer Modbus-Quelle ausgelöst werden. Die jeweils zeitlich letzte Schreibanforderung bestimmt den Wert der Prozessdaten, wie bereits in Kapitel 3.5 beschrieben.

### 4.1 Logische Redundanz

Bei der Konfiguration einer logischen Redundanz unterstützt das System den Anwender dadurch, dass mehrere Modbus-Quellen eine globale Variable beschreiben können, siehe Kapitel 3.5.

Der Zugriff auf die globalen Variablen erfolgt hierbei über Datenviews, die einen Adressbereich (Modbus-Offset) auf dem jeweiligen Kommunikationsmodul definieren, in dem verschiedene globale Variablen adressierbar sind. Eine solche Dataview kann auch von mehreren Mastern verwendet werden. Diese können getrennten Transportwegen nutzen, müssen aber über jeweils dasselbe Kommunikationsmodul kommunizieren.

i

Die globale Variable kann von allen Mastern (1...xy) beschrieben werden, so dass diese je nach Anwendung alle als redundant betrachtet werden können.

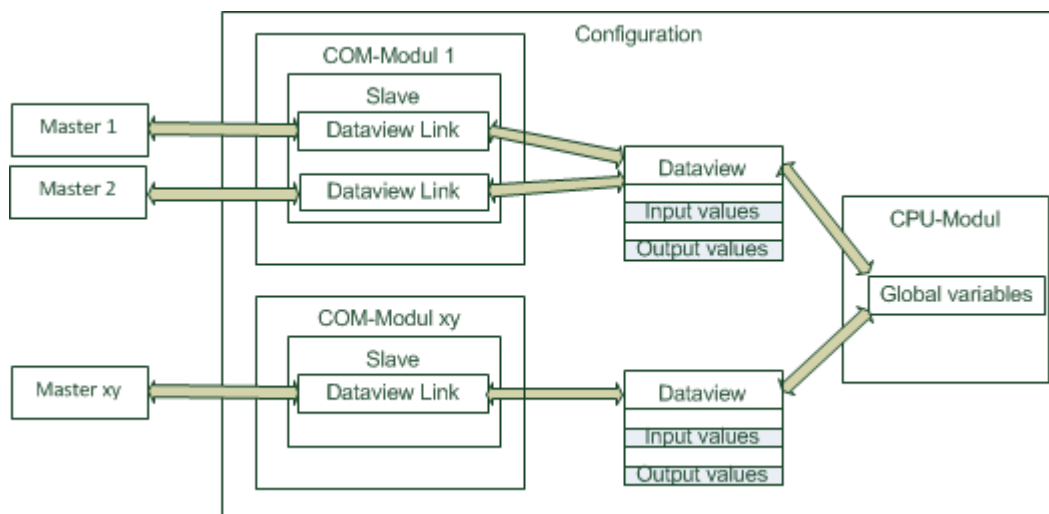


Bild 2: Redundante Master und mono Slave (HIMax und HIMatrix F\*03 Steuerung)

## 4.2 COM-Redundanz zweier Kommunikationsmodule

Das Beispiel in Bild 3 zeigt eine redundante Verbindung mit zwei durchgehend getrennten Transportwegen mit zwei Modbus-Slave-V2 auf zwei Kommunikationsmodulen.

Die COM-Redundanz zweier Kommunikationsmodule wird im Modbus-Slave-V2 automatisch unterstützt. Sobald im Modbus-Slave-Set-V2 zwei Modbus-Slave angelegt werden, kooperieren die beiden Kommunikationsmodule bei der Verarbeitung aller Schreib- und Lesezugriffe.

Für die reinen Schreibzugriffe auf globale Variablen hat dies in Hinblick auf die Wirkung und damit die Redundanz der Master verhältnismäßig wenig Unterschiede zur logischen Redundanz, die in Kapitel 4.2 beschrieben ist.

Für Lesezugriffe auf globale Variablen hat diese Form der Redundanz jedoch Vorteile:

- Die Lesezugriffe werden von den beiden Kommunikationsmodulen so koordiniert, dass die über beide Kommunikationsmodule gelesenen Daten in ihrer zeitlichen Abfolge konsistent sind. Dadurch ist es möglich, die Daten von beiden Kommunikationsmodulen miteinander in Bezug zu setzen und z. B. beim Verlust eines der Kommunikationsmodule nahtlos die Daten des verbleibenden Kommunikationsmoduls zu nutzen, ohne einen inkonsistenten zeitlichen Verlauf der gelesenen Prozessdaten fürchten zu müssen. Dies bezieht sich auch auf das Rücklesen der Daten aus dem Importbereich des Slaves.
- Dieses Verhalten bei Lesezugriffen ermöglicht einen redundanten Betrieb zweier Kommunikationsmodule, bei dem auch nach Ausfall eines der beiden Kommunikationsmodule, die Kommunikation stoßfrei weiterläuft.

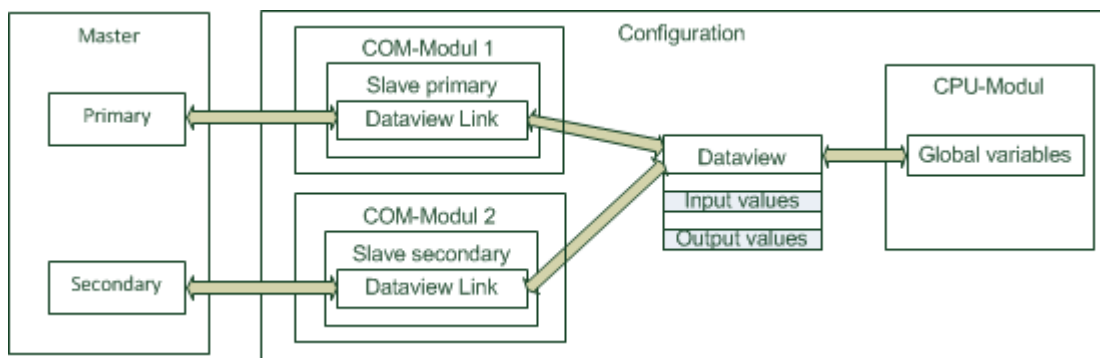


Bild 3: Redundante Master und redundanter Slave (HIMax Steuerung mit zwei COM-Modulen)

### 4.2.1 Unterstützung bei der Konfiguration der redundanten Kommunikationsmodule

Sobald im Modbus-Slave-V2 zwei Modbus-Slave angelegt werden, kooperieren die beiden Kommunikationsmodule und gleichen automatisch die gemeinsam verwendeten Dataviews ab.

Um den Anwender möglichst gut bei der fehlerfreien Konfiguration seiner Redundanz zu unterstützen, bietet SILworX die Möglichkeit zu definieren, welche Modbus-Master auf beiden Kommunikationsmodule redundant arbeiten sollen, um deren Gleichheit zu überprüfen.

Hierbei gibt es 2 Stufen der Gleichheitsprüfung:

- Alle relevanten Eigenschaften sind gleich, außer dem Transportweg des Masters  
In **Modbus-Slave-Set-V2, Modbus-Slave, Master** wählen und in den Eigenschaften den Systemparameter *Explizite Überprüfung redundanter Master* aktivieren.
- Alle relevanten Eigenschaften (inkl. Transportweg des Masters) sind gleich  
In **Modbus-Slave-Set-V2, Modbus-Slave, Master** wählen und in den Eigenschaften den Systemparameter *Redundanten Transportweg auf gleiche Einstellungen prüfen* aktivieren.

#### 4.2.1.1 Einschränkung bei der Konfiguration (identische Master)

- Das System bestimmt die Identität eines Masters anhand dessen externen Transportwegs (Feldbusschnittstelle oder IP-Adresse).
- Sind zwei Master auf den beiden redundant betriebenen Kommunikationsmodulen aufgrund der Konfiguration als identisch zu betrachten, müssen auch die Eigenschaften der beiden Master gleich sein.

#### 4.2.1.2 Kombination der Redundanzen

Die in diesem Kapitel beschriebenen redundanten Zugriffe auf globale Variablen lassen sich auch frei kombinieren. D. h. man kann auf globale Variablen, die von zwei redundanten Kommunikationsmodulen verwendet werden, auch von weiteren Kommunikationsmodulen aus zugreifen und 2 redundante Kommunikationsmodule können bzgl. der Dataviews und Master auch verschieden konfiguriert werden.

### 4.3 Hinweise zur Konfiguration einer Redundanz zweier Kommunikationsmodule auf einem HIMax System

#### 4.3.1 Regeln für den redundanten HIMax Modbus-Slave-V2

Für den redundanten Betrieb von HIMax Modbus-Slave Kommunikationsmodulen wird eine redundante HIMax Systemkonfiguration empfohlen, siehe Systemhandbuch HI 801 000 D. Ansonsten kann bereits beim Auftreten eines ersten Fehlers im HIMax System nicht sichergestellt werden, dass sich die Modbus-Slave Kommunikationsmodul-Paare gegenüber ihrem externen Partner (Modbus-Master) konsistent verhalten.

#### 4.3.2 Zulässige Steckplätze der redundanten HIMax Modbus-Slave-V2 COM-Module

Um mögliche Kollisionen auf dem HIMax Systembus zu minimieren, müssen die Systembus-Segmente (1 bis 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 5: Zulässige Steckplätze der redundanten Modbus-Slave COM-Module

#### 4.3.3 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 bis 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 4.3.2 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 Reload

Die Benutzung von Reload zum Ändern der Ressource-Konfiguration ist mit der zuständigen Prüfstelle abzustimmen! Weitere Informationen zum Reload siehe Sicherheitshandbuch der jeweiligen Systemfamilie.

### 5.1 Änderungen der Modbus-Slave-V2-Konfiguration

Das Hinzufügen eines Modbus-Slave-V2 per Reload setzt voraus, dass entweder kein Rechenzeitbudget konfiguriert ist oder genügend Reserve an Rechenzeitbudget über alle Protokolle vorhanden ist. Das Rechenzeitbudget bei anderen Standard-Protokollen kann nur durch Cold Reload geändert werden.

Die folgende Tabelle gibt eine Übersicht über Änderungen der Modbus-Slave-V2-Konfiguration und welche Auswirkungen diese auf den Modbus-Slave-V2-Reload haben.

Änderungen bei	HIMax X-COM 01	HIQuad X F-COM 01
Hinzufügen oder Löschen globaler Variablen aus Dataviews	•	•
Hinzufügen oder Löschen von Dataviews	•	•
Datentyp einer globalen Variable	-	-
Hinzufügen oder Löschen zugeordneter Modbus-Master	• <sup>1)</sup>	• <sup>1)</sup>
Master-Konfiguration zugeordneter Modbus-Master	• <sup>1)</sup>	• <sup>1)</sup>
Hinzufügen oder Löschen eines Modbus-Slave-V2	•	•
Rechenzeitbudgetierung des Modbus-Slave-V2	•	•
TCP-Einstellungen		
IP-Adresse ändern	•	•
Port ändern	• <sup>2)</sup>	• <sup>2)</sup>
Hinzufügen von Verbindungen	•	•
Löschen von Verbindungen	• <sup>3)</sup>	• <sup>3)</sup>
<ul style="list-style-type: none"> <li>• Modbus-Slave-V2-Reload möglich</li> <li>- Modbus-Slave-V2-Reload nicht möglich</li> </ul> n. a. nicht anwendbar <p><sup>1)</sup> Wird durch einen Reload die Identität des Masters geändert (IP-Adresse oder Feldbusschnittstelle), so werden alle dem Master zugeordneten Systemvariablen auf Initialwerte zurückgesetzt.</p> <p><sup>2)</sup> Bei der Änderung der Ports wird der Reload durchgeführt, alle Verbindungen zum Modbus-Master werden aber abgebrochen. Dies wird per Reload-Warnung angezeigt.</p> <p><sup>3)</sup> Bei Verringerung der Anzahl der TCP-Verbindungen werden alle Verbindungen zum Modbus-Master geschlossen, wenn die Anzahl der tatsächlich benutzten Verbindungen größer ist als die in der Reload-Konfiguration. Dies wird per Reload-Warnung angezeigt. Unabhängig davon warnt SILworX bei der Codegenerierung, wenn die Anzahl der TCP-Verbindungen in der Konfiguration verringert werden.</p> <p>Wird während des Reloads noch eine weitere Verbindung aufgebaut, kann es passieren, dass die Firmware keine Reload-Warnung wirft, die Verbindungen aber trotzdem geschlossen werden.</p>		

Tabelle 6: Modbus-Slave-V2-Reload nach Änderungen

## 6 Anlegen des Modbus-Slave-V2 in SILworX

### 6.1 Registrierung und Aktivierung des Protokolls

Der Software-Freischaltcode mit den benötigten Lizenzen wird auf der HIMA Webseite mit der System-ID (z. B. 60000) der Steuerung generiert. Den Anweisungen auf der HIMA Webseite folgen [www.hima.com](http://www.hima.com)-> *Produkte & Services*-> *Produkt-Registrierung*-> *Optionen SILworX*.

i

Die Lizenz ist untrennbar mit dieser System-ID verbunden. Eine Lizenz kann nur einmalig für eine bestimmte System-ID genutzt werden. Deshalb sollte die Freischaltung erst durchgeführt werden, wenn die System-ID eindeutig feststeht.

Ein Software-Freischaltcode kann maximal 32 Lizenzen enthalten. Es können auch mehrere Freischaltcodes in der Lizenzverwaltung eingetragen werden. In eine Steuerung können maximal 64 Lizenzen geladen werden.

#### Den Software-Freischaltcode in SILworX eintragen:

1. Im Strukturbaum **Konfiguration, Ressource, Lizenzverwaltung** wählen.
2. Rechtsklick auf **Lizenzverwaltung** und im Kontextmenü **Neu, Lizenzschlüssel** wählen.  
☒ Der Lizenzschlüssel wird neu hinzugefügt.
3. Rechtsklick auf **Lizenzschlüssel** und im Kontextmenü **Eigenschaften** wählen.
4. Im Feld **Freischaltcode** den generierten Software-Freischaltcode eintragen.

### 6.2 Anlegen des Modbus-Slave-Set-V2 Protokolls

Das Modbus-Slave-Set-V2 wird unterhalb des Protokolle-Ordners einer Ressource angelegt und dient der Konfiguration des Modbus-Slave-V2-Protokolls.

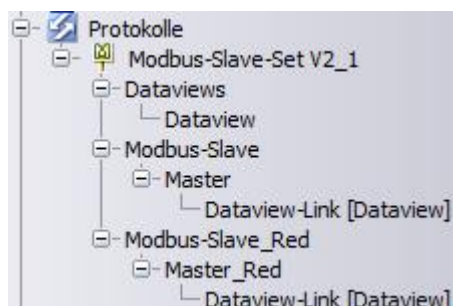


Bild 4: Modbus-Slave-Set-V2 im SILworX Strukturbaum

#### Modbus-Slave-Set-V2 anlegen:

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle** wählen.
2. Im Kontextmenü von Protokolle **Neu, Modbus-Slave-Set-V2** wählen, um ein neues Modbus-Slave-Set-V2 hinzuzufügen.
3. Im Kontextmenü von *Modbus-Slave-Set-V2* **Edit** wählen, um den *Modbus-Slave-Editor* zu öffnen. Dieser beinhaltet die Tabs **Set-Objekte** und **Eigenschaften**, siehe Kapitel 7.1.
4. Im *Modbus-Slave Set V2-Übersichtseditor* Tab **Eigenschaften** wählen.
  - **Name des Modbus-Slave** eintragen.

Die restlichen Eigenschaften können zunächst die Standardeinstellungen behalten.



### 6.3 Anlegen eines Dataview

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle, Modbus-Slave-Set-V2, Dataviews** wählen.
2. Im Kontextmenü von *Dataviews* **Neu, Dataview** wählen, um ein neues Dataview hinzuzufügen.
3. Im Kontextmenü von *Dataview* **Edit** wählen, um den *Dataview-Verbindungseeditor* zu öffnen. Dieser beinhaltet die Tabs **Registervariablen, Bitvariablen** und **Eigenschaften**.
4. Im *Dataview-Verbindungseeditor* Tab **Registervariablen** wählen.
  - Register Eingänge  
Zum Zuweisen der Empfangsvariablen, in der Objektauswahl die passenden globalen Variablen wählen und per Drag&Drop in das Feld **Register Eingänge** ziehen.
  - Register Ausgänge  
Zum Zuweisen der Sendevariablen, in der Objektauswahl die passenden globalen Variablen wählen und per Drag&Drop in das Feld **Register Ausgänge** ziehen.
5. Im *Dataview-Verbindungseeditor* Tab **Bitvariablen** wählen.
  - Bit Eingänge  
Zum Zuweisen der Empfangsvariablen, in der Objektauswahl die passenden globalen Variablen wählen und per Drag&Drop in das Feld **Bit Eingänge** ziehen.
  - Bit Ausgänge  
Zum Zuweisen der Sendevariablen, in der Objektauswahl die passenden globalen Variablen wählen und per Drag&Drop in das Feld **Bit Ausgänge** ziehen.
6. Im *Dataview-Verbindungseeditor* Tab **Eigenschaften** wählen.

Die Eigenschaften können die Standardeinstellungen behalten.

### 6.4 Konfiguration des Modbus-Slave

1. Im Strukturbaum **Konfiguration, Ressource, Protokolle, Modbus-Slave-Set-V2** wählen.
2. Im Kontextmenü von *Modbus-Slave-Set-V2* **Neu, Modbus-Slave** wählen um einen neuen Modbus-Slave hinzuzufügen.
3. Im Kontextmenü von *Modbus-Slave* **Edit** wählen, um den *Modbus-Slave-Editor* zu öffnen. Dieser beinhaltet die Tabs **Master-Objekte, Systemvariablen** und **Eigenschaften**.
4. Im *Modbus-Slave-Editor* Tab **Eigenschaften** wählen.
  - **COM-Modul** auswählen, auf dem dieser Modbus-Slave laufen soll.
  - Die restlichen Eigenschaften können zunächst die Standardeinstellungen behalten.

#### 6.4.1 Im Modbus-Slave die Verbindung zu dem Modbus-Master erstellen

1. Im Kontextmenü von *Modbus-Slave* **Neu, Modbus-Master** wählen um einen neuen Modbus-Master hinzuzufügen..
  2. Im Kontextmenü von *Modbus-Master* **Edit** wählen, um den *Modbus-Master-Editor* zu öffnen. Dieser beinhaltet die Tabs **Dataview-Links, Systemvariablen** und **Eigenschaften**.
  3. Im *Modbus-Slave-Editor* Tab **Eigenschaften** wählen.
    - **Master-IP-Adresse** eintragen, mit dem dieser Modbus-Slave verbunden sein soll.
- Die restlichen Eigenschaften können zunächst die Standardeinstellungen behalten.

#### 6.4.1.1 Dataview-Link mit Dataview referenzieren

Unterhalb des Master-Objekts ist das Dataview-Link-Objekt vorhanden.

1. Im Kontextmenü von *Dataview-Link* **Edit** wählen, um den *Dataview-Link-Editor* zu öffnen. Dieser beinhaltet die Tabs **Systemvariablen** und **Eigenschaften**.
2. Im *Dataview-Link-Editor* Tab **Eigenschaften** wählen.
  - Aus dem Dropdown Menü **Referenz zum Dataview-Objekt** die Dataview auswählen (z. B. Dataview, das in Kapitel 6.3 angelegt wurde), mit dem dieser Modbus-Master verbunden sein soll.

Die restlichen Eigenschaften können zunächst die Standardeinstellungen behalten.

### 6.5 Konfiguration des redundanten Modbus-Slave

Die COM-Redundanz zweier Kommunikationsmodule wird im Modbus-Slave-Set-V2 automatisch unterstützt. Sobald im Modbus-Slave-Set-V2 zwei Modbus-Slave angelegt werden, kooperieren die beiden Kommunikationsmodule bei der Verarbeitung aller Schreib- und Lesezugriffe, wenn diese auf die gleichen Dataviews (z. B. Dataview, das in Kapitel 6.3 angelegt wurde) zugreifen.

Die COM-Redundanz zweier Kommunikationsmodule wird im Modbus-Slave-Set-V2 automatisch unterstützt, siehe Kapitel 4.2.

## 7 Beschreibung der Modbus-Slave-V2-Editoren und Objekte

Dieses Kapitel beschreibt die Eigenschaften des HIMA Modbus-Slave-V2 sowie die Menüfunktionen und Dialoge in SILworX, die zur Konfiguration des HIMA Modbus-Slave-V2 benötigt werden.

### 7.1 Modbus-Slave-Set-V2

Modbus-Slave-Set-V2 wird im Ordner *Protokolle* angelegt und dient der Konfiguration des Modbus-Slave-V2-Protokolls.

Im Kontextmenü von Modbus-Slave-Set-V2 **Edit** wählen, um den Modbus-Slave-Set-V2 Editor zu öffnen. Dieser enthält die folgenden Tabs:

- Set-Objekte-Tab
- Eigenschaften-Tab

#### 7.1.1 Set-Objekte-Tab

Das Set-Objekte-Tab enthält die Auflistung aller Modbus-Slave-Set-V2 Objekte, die in diesem Modbus-Slave-Set-V2 angelegt wurden.

Set-Objekte	Beschreibung
Dataviews	Im Objekt Dataviews liegen alle Dataviews die für diesen Modbus-Slave-V2 angelegt wurden.
Modbus-Slave	Hier werden die Eigenschaften des Modbus-Slave und die Verbindungen zu den Master(n) eingestellt.

Tabelle 7: Set-Objekte eines Modbus-Slave-Set-V2

#### 7.1.2 Eigenschaften-Tab

Im Eigenschaften-Tab können die Parameter des Modbus-Slave-Set-V2 eingestellt werden.

Eigenschaften	Beschreibung
Typ	Modbus-Slave-Set-V2
Name	Änderbar Standard: Modbus-Slave-Set-V2
Max. $\mu$ P-Budget in [%]	Maximales $\mu$ P-Budget des Moduls, welches bei der Abarbeitung des Protokolls produziert werden darf. <ul style="list-style-type: none"> <li>▪ Keine Limitierung</li> <li>▪ Wertebereich: 1...100 %</li> <li>Standardwert: 30 %</li> </ul>
Warnung bei $\mu$ P-Budget-Überschreitung in [%]	Bei Überschreitung dieser Warnschwelle wird eine Warnmeldung generiert. <ul style="list-style-type: none"> <li>▪ Nicht verwenden</li> <li>▪ Wertebereich: 1...100 %</li> <li>Standardwert: 80 %</li> </ul>

Tabelle 8: Modbus-Slave-Set-V2 Eigenschaften

## 7.2 Dataviews

Das Dataviews-Objekt enthält die Auflistung aller Dataview-Objekte, die in diesem Modbus-Slave-V2-Set angelegt wurden.

### 7.2.1 Dataview

Im Dataview-Editor werden die Prozessvariablen für die jeweilige Kommunikationsverbindung logisch zugeordnet.

Im Kontextmenü von Dataview **Edit** wählen, um den Dataview-Editor zu öffnen. Dieser enthält die folgenden Tabs:

- Register-Variablen-Tab
- Bit-Variablen-Tab
- Eigenschaften-Tab

#### 7.2.1.1 Register-Variablen-Tab

Im **Register-Variablen-Tab** werden die globalen Variablen im Register **Bereich** angelegt.

Auf die globalen Variablen im Register **Bereich** kann nur über die Modbus-Funktionscodes 3, 4, 6, 16, 23 zugegriffen werden.

#### 7.2.1.2 Bit-Variablen-Tab

Im **Bit-Variablen-Tab** werden die globalen Variablen im Bit-Bereich angelegt.

Auf die Globalen Variablen im Bit-Bereich kann nur über die Modbus-Funktionscodes 1, 2, 5, 15 zugegriffen werden.

#### 7.2.1.3 Eigenschaften-Tab

Eigenschaften	Beschreibung
Typ	Dataview
Name	Name des Dataview, änderbar Standard: Dataview
Verhalten bei Verbindungsverlust	Bei Verbindungsverlust des Prozessormoduls zum Kommunikationsmodul oder Master werden in Abhängigkeit dieses Parameters die Eingangsvariablen entweder initialisiert oder unverändert im Prozessormodul verwendet. Werden globale Variablen von mehreren Modbus-Quellen, d. h. von mehreren Dataviews beschrieben, kann nur die Option <i>Letzten Wert beibehalten</i> verwendet werden, sonst wird die Codegenerierung abgelehnt. <div style="display: flex; justify-content: space-between;"> <div>Initialdaten annehmen</div> <div>Eingangsvariablen werden auf die Initialwerte zurückgesetzt.</div> </div> <div style="display: flex; justify-content: space-between;"> <div>Letzten Wert beibehalten</div> <div>Eingangsvariablen behalten den letzten Wert.</div> </div>

Tabelle 9: Dataviews Eigenschaften

### 7.3 Modbus-Slave

Im Kontextmenü von Modbus-Slave **Edit** wählen, um den Modbus-Slave-Editor zu öffnen. Dieser enthält die folgenden Tabs:

- Master-Objekte-Tab
- Systemvariablen-Tab
- Eigenschaften-Tab

#### 7.3.1 Master-Objekte-Tab

In Master-Objekte-Tab werden alle Master Objekte aufgelistet, welche diesem Modbus-Slave logisch zugeordnet sind.

#### 7.3.2 Systemvariablen-Tab

Eigenschaften	Datentyp	Beschreibung
Anzahl Zugriffe nicht konfigurierter Master auf das System	UDINT	Dataview
IP-Adresse eines nicht konfigurierten Masters	DWORD	Die IP-Adresse eines nicht konfigurierten Masters, der zuletzt über ein Modbus-Slave-V2-Protokoll auf die Steuerung zugegriffen hat, wird als Systemvariable zur Verfügung gestellt. Wertebereich: gültige Unicast-IP-Adresse Standard: 0
Redundanz-Zustand	BYTE	Für jedes COM-Modul wird der aktuelle Redundanzzustand zur Verfügung gestellt. Wertebereich: 0: Das COM-Modul ist passiv (P). 1: Das COM-Modul arbeitet exklusiv (E). 2: Das COM-Modul arbeitet redundant (R). Standardwert: 0
Reset-Statistik	BYTE	Rücksetzen der Statistik.

Tabelle 10: Systemvariablen

#### 7.3.3 Eigenschaften-Tab

Eigenschaften	Beschreibung
Typ	Dataview
Name	Änderbar Standard: Modbus-Slave
Modul	Das COM-Modul, auf dem der Modbus-Slave läuft.
TCP-Port	TCP-Port für die TCP-Verbindungen zu den Modbus-Mastern. Wertebereich: 1 ... 65535 Standard: 502 0, falls der Modbus-Slave nicht über TCP kommuniziert.
UDP-Port	UDP-Port für die UDP-Verbindungen zu den Modbus-Mastern. Wertebereich: 1 ... 65535 Standard: 502 0, falls der Modbus-Slave nicht über UDP kommuniziert.

Tabelle 11: Modbus-Slave Eigenschaften

## 7.4 Modbus-Master

Im Kontextmenü von Modbus-Master **Edit** wählen, um den Modbus-Master-Editor zu öffnen. Dieser enthält die folgenden Tabs:

- Dataview-Links-Tab
- Systemvariablen-Tab
- Eigenschaften-Tab

### 7.4.1 Dataview-Links-Tab

In Dataview-Links-Tab werden alle Dataview-Link-Objekte aufgelistet, die für die Verbindung zu diesem Modbus-Master-Objekt angelegt wurden. Ein Dataview-Link verweist auf einen Dataview, in dem die globalen Variablen für Eingangs- und Ausgangsvariablen angelegt sind und kann angepasst werden, siehe Kapitel 7.5.

### 7.4.2 Systemvariablen-Tab

Eigenschaften	Datentyp	Beschreibung
Aktueller Master Verbindungszustand eines konfigurierten Masters	BOOL	Aktueller Verbindungszustand: TRUE    Verbunden FALSE   Nicht verbunden

Tabelle 12: Systemvariablen

### 7.4.3 Eigenschaften-Tab

Element (Editor)	Beschreibung
Typ	Master
Name	Name ist änderbar. Standardwert: Master
Byte-Reihenfolge	Mögliche Einstellungen: <ul style="list-style-type: none"> <li>▪ BE (Big Endian)</li> <li>▪ BEB (Big Endian mit Byte-Swapping)</li> <li>▪ LE (Little Endian)</li> <li>▪ LEB (Little Endian mit Byte-Swapping)</li> </ul> Weitere Informationen hierzu siehe Tabelle 17. Standardwert: BE (Big Endian)
Lesebereich für Coil-, Holding Register- und Read/Write Multiple Register-Zugriffe	Mögliche Einstellungen: <ul style="list-style-type: none"> <li>▪ Exportbereich</li> <li>▪ Importbereich</li> </ul> Standardwert: Exportbereich
Maximale Antwort-Verzögerungszeit [ms]	Mögliche Einstellungen: <ul style="list-style-type: none"> <li>▪ Keine Limitierung</li> <li>▪ Maximale Antwort-Verzögerungszeit [ms]</li> </ul> Standardwert: 5000
Transportweg	Mögliche Einstellungen: <ul style="list-style-type: none"> <li>▪ TCP</li> <li>▪ UDP</li> <li>▪ RS485</li> </ul> Standardwert: TCP
Master-IP-Adresse	IP-Adresse des TCP/UDP Masters.

Element (Editor)	Beschreibung
Anzahl TCP-Verbindungen	<p>Maximal 20 TCP-Verbindungen sind einstellbar.</p> <p>Der Parameter <i>Anzahl TCP-Verbindungen</i> muss für jeden Modbus-Master eingestellt werden. In der Summe sind maximal 20 TCP-Verbindungen zu den konfigurierten Modbus-Mastern erlaubt.</p> <p>Der Parameter <i>Transportweg</i> muss auf TCP eingestellt sein, sonst ist das Eingabefeld gesperrt.</p> <p>Wertebereich: 1 ... 20</p> <p>Standardwert: 1</p>
Die folgenden Elemente können nur Bearbeitet werden, wenn Transportweg <i>RS485</i> gewählt wurde.	
Schnittstelle	<p>Die Feldbus-Schnittstelle, die für den Modbus-Slave benutzt werden soll (fb1, fb2, fb3, fb4).</p> <p>Wertebereich: Abhängig vom Typ der COM-Baugruppe</p>
Slave-Adresse	<p>Slave-Adresse für die Feldbus-Schnittstelle.</p> <p>Wertebereich: 1 ... 247</p> <p>Standardwert: 1</p>
Baudrate [bps]	<p>Übertragungsgeschwindigkeit für RS485 mögliche Werte:</p> <p>300 bit/s</p> <p>600 bit/s</p> <p>1200 bit/s</p> <p>2400 bit/s</p> <p>4800 bit/s</p> <p>9600 bit/s</p> <p>19200 bit/s</p> <p>38400 bit/s</p> <p>57600 bit/s (maximale Baudrate HIMax)</p> <p>62500 bit/s (HIMatrix)</p> <p>76800 bit/s (HIMatrix)</p> <p>115000 bit/s (HIMatrix)</p>
Parität	<p>Mögliche Einstellungen:</p> <ul style="list-style-type: none"> <li>▪ Keine</li> <li>▪ Ungerade</li> <li>▪ Gerade</li> </ul> <p>Standardwert: Gerade</p>
Stop-Bits	<p>Standard (passt die Anzahl der Stop-Bits der Parität an: mit Parität = 1 Stop-Bit, keine Parität 2 Stop-Bit.)</p> <p>Mögliche Einstellungen:</p> <ul style="list-style-type: none"> <li>▪ ein Stop-Bit</li> <li>▪ zwei Stop-Bits</li> </ul> <p>Standardwert: Standard</p>
Anzahl Idle-Zeichen	<p>Die Anzahl der Idle-Zeichen am Beginn und Ende des RTU-Telegrammrahmens.</p> <p>Wertebereich HIMax: 0 ... 65535</p> <p>HIMatrix F*03: 0 ... 93</p> <p>Standardwert: 5 Zeichen</p> <p>Für die HIMax mit RS485-Betrieb sollten nicht weniger als 4 Idle-Zeichen eingestellt werden, da die Hardware das RTS-Signal noch 3,1 Zeichenlängen nach dem letzten gesendeten Low-Datenbit anstehen lässt.</p> <p>Das RTS-Signal verhindert auf dem RS485-Modul, dass der Slave seine eigenen Datensendungen empfängt. D. h., der RS485-Receiver ist noch 3,1 Zeichenlängen nach dem Versenden des letzten Datenbits ausgeschaltet.</p>

Element (Editor)	Beschreibung
Explizite Überprüfung redundanter Master	<p>Um die Konsistenzhaltung von redundanten Mastern zu erleichtern, kann die Funktion zur expliziten Überprüfung redundanter Master auf identische Konfiguration genutzt werden.</p> <p>TRUE Die explizite Prüfung der redundanten Master ist aktiviert.</p> <p>FALSE Keine explizite Prüfung.</p>
Namen der redundanten Master unterscheiden sich vom eigenen	<p>Wenn der Parameter <i>Explizite Überprüfung redundanter Master</i> aktiviert ist, bietet das Master-Objekt die Eigenschaft <i>Namen der redundanten Master unterscheiden sich vom eigenen</i> an.</p> <p>TRUE Der Anwender muss die Namen der redundanten Master im darunterliegenden Feld <i>Redundante Master</i> eintragen.</p> <p>FALSE Der Name des redundanten Masters ist der gleiche wie der eigene.</p>
Redundante Master	<p>Ist die Checkbox <i>Namen der redundanten Master unterscheiden sich vom eigenen</i> auf TRUE gesetzt, dann ist das Feld <i>Redundante Master</i> aktiv. Die Namen der redundanten Master können mit Komma getrennt eingegeben werden.</p> <p>Die neben dem Feld liegende Schaltfläche dient zum Starten eines Dialogs. In diesem Dialog können die redundanten Master des Slaves ausgewählt werden.</p> <p>Sollte ein Master in der Eigenschaft nicht mehr als redundanter Master vorliegen, meldet dies der Dialog über eine Messagebox und ignoriert die Auswahl.</p>
Redundanten Transportweg auf gleiche Einstellung prüfen	<p>Wenn der Parameter <i>Explizite Überprüfung redundanter Master</i> aktiviert ist, bietet das Master-Objekt die Eigenschaft <i>Redundanten Transportweg auf gleiche Einstellung prüfen</i> an.</p> <p>Es wird geprüft, dass der redundante Master die gleiche Transportweg-Einstellungen (Baudrate, IP-Adresse, etc.) hat.</p>

Tabelle 13: Modbus-Master-Eigenschaften



## 7.5 Dataview-Link

Im Kontextmenü von Dataview-Link **Edit** wählen, um den Dataview-Link-Editor zu öffnen. Dieser enthält die folgenden Tabs:

- Systemvariablen-Tab
- Eigenschaften-Tab



Sind mehrere Dataview-Links für einen Master angelegt dürfen sich die Modbus-Adressen nicht überlappen!

### 7.5.1 Systemvariablen-Tab

Eigenschaften	Datentyp	Beschreibung
Verbindungs-Statusbyte	Byte	Das System stellt ein Verbindungs-Statusbyte für jede Verbindung zwischen einem Master und einem Dataview zur Verfügung. Das Verbindungs-Statusbyte hat den folgenden Wertebereich: Anzahl konfigurierter Dataviews als Bitfeld 0-0x07 [BYTE]
		Bit 0: Aktueller Verbindungszustand Master <-> Dataview.
		0: Master <-> Dataview sind aktuell nicht verbunden.
		1: Master <-> Dataview sind aktuell verbunden oder die Verbindung wird nicht überwacht.
		Bit 1: Historischer Verbindungszustand Master <-> Dataview.
		0: Die Überwachungszeit ist seit dem letzten Reset niemals abgelaufen bzw. Systemstart oder die Zugriffe des Masters auf diese Dataview werden nicht überwacht.
		1: Die Überwachungszeit für die Dataview ist seit dem letzten Reset mindestens einmal abgelaufen bzw. Systemstart.
		Bit 2: Ungültige Master-Requests.
		0: Seit dem Start oder letzten Reset der Systemvariablen gab es keine ungültigen Master-Requests.
		1: Es gab mindestens einen ungültigen Master-Request.

Tabelle 14: Dataview-Links Systemvariablen

## 7.5.2      Eigenschaften-Tab

Eigenschaften	Beschreibung
Typ	Dataview-Link
Name	Dataview-Link
Referenz zum Dataview-Objekt	Die Referenz ist der Name des Dataviews, der über den Dataview-Link mit diesem Master verknüpft ist (Auswahl aus verfügbaren Dataviews).
Master Überwachungszeit [ms]	Die für die Überwachung relevante Zeit ist, wann dieser Master zuletzt auf die Dataview über diesen Link per Lese-/Schreibanforderung zugegriffen hat, siehe Tabelle 14. Mögliche Einstellungen: <ul style="list-style-type: none"><li>▪ Keine Limitierung</li><li>▪ Master Überwachungszeit [ms] = 1...2147483647</li></ul> Standardwert: Keine Limitierung

Tabelle 15: Dataview-Links Eigenschaften

## 8 Modbus-Funktionscodes

### 8.1 Standard Modbus-Funktionscodes

Folgende Modbus-Funktionscodes werden vom HIMA Modbus-Slave-V2 Slave unterstützt.

Element	Code	Bedeutung
READ COILS	01	Lesen mehrerer Coils aus dem parametrierten Lesebereich <sup>1)</sup> des Modbus-Slaves. Maximale Länge der Prozessdaten pro Modbus-Telegramm: 250 Bytes (2000 Coils).
READ DISCRETE INPUT	02	Lesen mehrerer Discrete Inputs aus dem Exportbereich des Modbus-Slaves. Maximale Länge der Prozessdaten pro Modbus-Telegramm: 250 Bytes (2000 Coils).
READ HOLDING REGISTER	03	Lesen mehrerer Register aus dem parametrierten Lesebereich <sup>1)</sup> des Modbus-Slaves. Maximale Länge der Prozessdaten pro Modbus-Telegramm: 250 Bytes (125 Register).
READ INPUT REGISTER	04	Lesen mehrerer Register aus dem Exportbereich des Modbus-Slaves. Maximale Länge der Prozessdaten pro Modbus-Telegramm: 250 Bytes.
WRITE SINGLE COIL	05	Schreiben einer einzelnen Coil in den Importbereich des Modbus-Slaves. Maximale Länge der Prozessdaten pro Modbus-Telegramm: 1 Byte.
WRITE SINGLE REGISTER	06	Schreiben eines einzelnen Register in den Importbereich des Modbus-Slaves. Maximale Länge der Prozessdaten pro Modbus-Telegramm: 2 Bytes.
Diagnostics	08	Nur Subcode 0 (Loopback-Funktion) von FC08.
WRITE MULTIPLE COILS	15	Schreiben mehrerer Coils in den Importbereich des Modbus-Slaves. Maximale Länge der Prozessdaten pro Modbus-Telegramm: 246 Bytes (1968 Coils).
WRITE MULTIPLE REGISTER	16	Der FC 16 zum Schreiben mehrerer Register in den Importbereich des Modbus-Slaves. Maximale Länge der Prozessdaten pro Modbus-Telegramm: 246 Bytes (123 Register).
READ WRITE MULTIPLE REGISTER	23	Lesen mehrerer Register aus dem parametrierten Lesebereich <sup>1)</sup> des Modbus-Slaves. Schreiben mehrerer Register in den Importbereich des Modbus-Slaves. Maximale Länge der Prozessdaten pro Modbus-Telegramm: 242 Bytes (121 Register), vom Modbus-Master. 250 Bytes (125 Register), zum Modbus-Master.
Read Device Identification	43	Liefert die Identifikationsdaten des Slaves an den Modbus-Master.
<sup>1)</sup> Export- oder Importbereich		

Tabelle 16: Modbus-Funktionscodes des HIMA Modbus-Slave

Die Funktionscodes 01, 02, 05 und 15 unterstützen den Datentyp BOOL. Die Funktionscodes 03, 04, 16 und 23 unterstützen beliebige Datentypen mit der Länge 2 Bytes.

Für jeden Request muss die Startadresse der ersten zu übertragenden Variable und die Anzahl der Register/Bits der zu übertragenden Variablen eingetragen werden.

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 (Relevant sind das Datenlayout und die Offsets der verknüpften Dataviews), 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 HIMA 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 / COM Vx.y CRC>"

Regular:

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

0x04 ProductName "HIMax /HIMatrix"

0x05 ModelName "ModulType"

0x06 UserApplicationName "-----[SRS]"

Extended:

0x80 leer "-----"

0x81 leer "-----"

0x82 leer "-----"

0x83 leer "-----"

0x84 leer "-----"

0x85 leer "-----"

0x86 CRC der Datei modbus.config

(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.

## 8.2 Byte-Order

Die Byte-Order beschreibt das Endian-Format der ein- und ausgehenden Prozessdaten.

Die folgende Tabelle zeigt wie die Byte-Order den Datenstrom des Protokolls Modbus V2 beeinflusst, wenn ein Datentyp nicht an einer Registergrenze anfängt oder endet.

Register	Datenbyte	Datentyp	Little-Endian [LE]	Big-Endian [BE]	Little-Endian mit Byte Swapping [LEB]	Big-Endian mit Byte Swapping [BEB]
0	0	Byte A	Byte A	Byte A	Byte A	Byte A
	1	DWord	DWord[0]LSB	DWord[3]MSB	DWord[1]	DWord[2]
1	2		DWord[1]	DWord[2]	DWord[0]LSB	DWord[3]MSB
	3		DWord[2]	DWord[1]	DWord[3]MSB	DWord[0]LSB
2	4	Word	DWord[3]MSB	DWord[0]LSB	DWord[2]	DWord[1]
	5		Word[0]LSB	Word[1]MSB	Word[1]MSB	Word[0]LSB
3	6	Word	Word[1]MSB	Word[0]LSB	Word[0]LSB	Word[1]MSB
	7		Byte B	Byte B	Byte B	Byte B

Tabelle 17: Byte-Order

## 9 Anzeige und Diagnose im Control Panel

Nach dem erfolgreichen System-Login wird das Control Panel geöffnet.

Im Knoten *Modbus-Slave* sind die Informationen über den Betrieb des Protokolls *Modbus V2 über Ethernet* zusammengefasst.

### 9.1 Online-Anzeige des Modbus

Im Anzeigefeld werden die verwendeten Modbus-Verbindungen des *Modbus V2 Master* angezeigt:

Name	Name	SRS der redundanten Baugruppe	Konfiguriertes µP-Budget	Aktuelles µP-Budget	Maximaler Pufferfüllstand für Anfragen	Durchschnittlicher Pufferfüllstand für Anfragen
Modbus-Slave	Modbus-Slave	X.X.X	100	0	1	2

Name	Verbindungszustand	Gültige Anfragen	Ungültige Anfragen	Master-Timeout	Zeitüberschreitungen bei Antworten	Verworfenne Anfragen	Master-Identifikation	Aktuell geöffnete TCP-Verbindungen
1 Master 1	Nicht überwacht	3	4	5		6	7 FB255	5
2 View1	Ungültiger Wert (3)	3	4	5				

Bild 5: Modbus-Online-Anzeige

In der folgenden Tabelle sind alle Parameter der Modbus-Online-Anzeige nochmals dargestellt und stehen für die Online-Diagnose zur Verfügung.

Modbus-Slave (Online)	Beschreibung
Name	Modbus-Slave
SRS des redundanten Moduls	System.Rack.Slot
Konfiguriertes µP-Budget	Siehe Tabelle 8.
Aktuelles µP-Budget	Siehe Tabelle 8.
Maximaler Pufferfüllstand für Anfragen	Maximaler Füllstand des Anfragenpuffers für ein Modbus-Slave-V2-Protokoll.
Durchschnittlicher Pufferfüllstand für Anfragen	Durchschnittlicher Füllstand des Anfragenpuffers für ein Modbus-Slave-V2-Protokoll.
Modbus-Master (Online)	Beschreibung
Name	Name des Masters.
Verbindungszustand	Siehe Tabelle 12.
Gültige Abfragen	Anzahl korrekter Anfragen, die vom Modbus-Slave erfolgreich bearbeitet und beantwortet werden kann.
Ungültige Anfragen	Anzahl inkorrektter Anfragen, die vom Modbus-Slave mit einem Fehlercode (siehe Modbus Exception Codes) beantwortet wird.
Master-Timeout	Falls das Master-Request Timeout nicht 0 ist, muss pro Verbindung Master <-> Dataview überprüft werden, ob innerhalb dieses Timeouts eine gültige Anfrage von dem konfigurierten Master erfolgte. Als gültige Anfrage ist hierbei ein beliebiger Modbus-Funktionscode zu verstehen. Hat ein Master mehrere Verbindungen zu mehreren Dataviews und gehen mehr als eine Verbindung zu den Dataviews verloren, so wird dies nur als ein Kommunikationsfehler gemeldet und nicht als mehrere Kommunikationsfehler.
Zeitüberschreitungen bei Antworten	Anzahl Zeitüberschreitungen von Antworten des Modbus-Slaves. Das bedeutet, dass die Antwort nicht innerhalb der konfigurierten maximalen Antwortzeit erfolgt ist und daher verworfen wird. Zähler ist umlaufend.

Verworfenne Anfragen	Anzahl verworfener Anfragen eines konfigurierten Masters. Zähler ist umlaufend.
Master-Identifikation	<p>Für jeden konfigurierten Master auf einem Kommunikationsmodul wird dessen Identifikation zur Verfügung gestellt.</p> <ul style="list-style-type: none"> <li>▪ Bei Modbus-UDP/TCP-Mastern erfolgt die Identifikation durch die IP-Adresse.</li> <li>▪ Bei Modbus-RTU-Master erfolgt die Identifikation durch die Kennung der RS485-Schnittstelle, über welche das jeweilige Kommunikationsmodul angeschlossen ist.</li> </ul>
Aktuell geöffnete TCP-Verbindungen	Für jeden konfigurierten Master wird die Anzahl aktuell geöffneter TCP-Verbindungen angezeigt. Wertebereich: 0 ... 20
Dataview (Online)	Beschreibung
Name	Name des Dataview.
Verbindungszustand	Siehe Tabelle 12.
Gültige Abfragen	Parameter wie bei Modbus-Master (Online), jedoch bezogen auf den Offset-Bereich der Dataview.
Ungültige Anfragen	
Master-Timeout	

Tabelle 18: Elemente der Modbus Online-Anzeige

## 10 Versionsvergleich

Der Versionsvergleich erfolgt anhand der vom Codegenerator erstellten Prüfsummen (CRCs) des Projekts. Für weitere Informationen siehe Handbuch Versionsvergleich HI 801 285 D.

Beim Versionsvergleich werden verschiedene Ressourcenkonfigurationen miteinander verglichen und die Unterschiede zwischen den einzelnen Konfigurationsdateien angezeigt. Das Ergebnis des Versionsvergleichs hat SIL3-Qualität und beruht auf den Konfigurationsdateien, welche den ausführbaren Code beschreiben.

Versionsvergleich DL -> CG HIMax [2]							
Vergleich		Reload-Warnungen					
	Name	Beschreibung	CRC DL	Version DL	CRC CG	Version CG	Vergleich der CRCs
1	/addon.config	AddOn-Daten der Konfiguration	16#f9c7df91	V9	16#b415c285	V9	-
2	/root.config	Hauptdatei der Konfiguration	16#5f498e2f	V9	16#081f9b8e	V9	-
3	/000.00/root.config	Hauptdatei - CPU	16#abb2ac89	V3	16#f8764434	V3	-
4	/000.01/root.config	Hauptdatei - Kommunikationsmodul	16#b2444167	V9	16#d6c41a85	V9	-
5	/000.01/cpcnsip.config	Standardprotokoll	16#39ace0cd	V9	16#5a0a5c47	V9	-
6	/000.01/dpcom.config	Verteilung der Schreib-Anforderungen	16#cf3c4ad6	V2	16#c3789be7	V2	-
7	/000.01/hh.config	HIMA-HIMA-Kommunikation-Konfigura...	16#00000000		16#a4ce28b2	V8	--
8	/000.01/ke.config	COM-Datenlayout und Transport	16#49b930c5	V2	16#49b930c5	V2	ok
9	/000.01/mos.config	Modbus-Slave V2	16#4af207c9	V9	16#0aeadfcb	V9	-
10	/000.01/net.config	Netzwerkeinstellung	16#8e45fabe	V2	16#328dc994	V2	-
11	/sys/root.config	Hauptdatei - System	16#8ac4e51b	V9	16#7eb7a0c7	V9	-
12	/sys/cpc.config	Systemprotokolle Basis	16#cea77ad6	V2	16#f3edb36e	V2	-
13	/sys/cpcnsip.config	Standardprotokoll	16#bec69940	V9	16#b00b4793	V9	-
14	/sys/cpcnsip.config	Sicherheitsgerichtetes Protokoll	16#00000000		16#04880f3b	V6	--
15	/sys/cpu.config	Systemdaten	16#a2036579	V7	16#a2036579	V7	ok
16	/sys/dpcpu.config	Verteilung der Schreib-Anforderungen	16#2c2223f9	V9	16#c157db5c	V9	-
17	/sys/ia.config	IO-Konfiguration	16#fd1ceebb	V2	16#fd1ceebb	V2	ok
18	/sys/ke.config	Datenlayout und Transport	16#f85fda6f	V9	16#c4d20c51	V9	-

Bild 6: Modbus-Slave-V2 im Versionsvergleich

### 10.1 Die für das Modbus-Slave-V2-Protokoll relevanten Konfigurationsdateien

Unterhalb der Konfigurationsdatei `sys/root.config` befinden sich für das Modbus-Slave-V2-Protokoll die relevanten Konfigurationsdateien, die in Tabelle 19 aufgeführt sind.

Zeile	Konfigurationsdatei	Beschreibung						
4	/000.01/root.config	Hauptdatei des Kommunikationsmoduls. Diese Konfigurationsdatei referenziert auf untergeordnete Konfigurationsdateien und ändert sich immer, wenn sich eine untergeordnete Konfigurationsdatei ändert.						
6	/000.01/dpcom.config	Konfigurationsdatei des Modbus-Protokolls. Darin werden die geänderten Einstellungen von z. B. Anzahl Dataviews, Länge des Eingangsbereichs/ Ausgangsbereichs etc. aufgeführt.						
9	/000.01/mos.config	<p>Modbus-Slave-V2-Parameter, Eigenschaften der Verbindungen.</p> <p>Das PADT erzeugt in der Codegenerierung eine Konfigurationsdatei mos.config, abhängig vom Wert des Parameters <i>Anzahl TCP-Verbindungen</i>, mit den folgenden Versionen.</p> <table><tr><th>Wert Anzahl TCP-Verbindungen</th><th>mos.config Version</th></tr><tr><td>1</td><td>1.0</td></tr><tr><td>2 ... 20</td><td>2.0</td></tr></table>	Wert Anzahl TCP-Verbindungen	mos.config Version	1	1.0	2 ... 20	2.0
Wert Anzahl TCP-Verbindungen	mos.config Version							
1	1.0							
2 ... 20	2.0							
13	/sys/cpcnsip.config	In der Konfigurationsdatei werden die Einstellungen zur Übertragung der nicht sicheren Protokolldaten zwischen CPU-Modul und COM-Modul aufgeführt.						



Zeile	Konfigurationsdatei	Beschreibung
18	/sys/ke.config	Konfigurationsdatei für das Lesen und Schreiben globaler Variablen in Protokollen (ke = Kommunikations-Endpunkt = globale Variable). In dieser Datei befindet sich eine Tabelle mit der globale Variablen auf Pausibilität bzgl. der beabsichtigten Verwendung als Globale Multisource Variablen überprüft werden können, siehe Kapitel 10.1.

Tabelle 19: Modbus-Slave-V2-Konfigurationsdateien

## 10.2 Globale Multisource-Variable

Eine globale Variable gilt als Globale Multisource-Variable, sobald diese in einem Dataview verwendet wird. Dies gilt auch für den Fall eines abgeleiteten Datentyps einer globalen Variable (z. B. Array), wenn mindestens ein Unterelement in einem Dataview verwendet wird.

Globale Multisource-Variablen können neben ihrer Modbus-Quelle zusätzlich von einem Anwenderprogramm beschrieben werden. Dabei ist es im Fall des abgeleiteten Datentyps unerheblich, ob exakt die gleichen Unterelemente vom Anwenderprogramm und der Modbus-Quelle beschrieben werden oder unterschiedliche Unterelemente.

Wird eine Globale Multisource-Variable im Anwenderprogramm verwendet muss diese Schreib-/ Lesezugriff haben. Zuerst muss die Globale Multisource-Variable im Anwenderprogramm gelesen werden, bevor das Anwenderprogramm diese Variable beschreibt. Erfolgt der Zugriff auf eine Globale Multisource-Variable durch das Anwenderprogramm ausschließlich schreibend, wird nach der Codegenerierung im Logbuch die folgende Meldung generiert:

*Die globale Variable 'Global Variable\_5' hat mehr als eine Quelle, wird im Programm jedoch nur geschrieben. In diesem Fall muss sie im Programm auch gelesen werden.*

Abhilfe: globale Variable\_5 im Anwenderprogramm vor dem Beschreiben erst lesen!

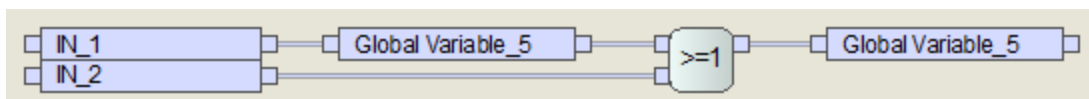


Bild 7: Lesen und Beschreiben einer Globale Multisource-Variable im Anwenderprogramm

### 10.2.1 Identifikation der Globalen Multisource-Variablen im Versionsvergleich

Mithilfe des Versionsvergleich können Globale Multisource-Variablen der Ressource bzgl. der Verwendung identifiziert werden.

#### 1

**Globale Multisource-Variablen dürfen nicht für die Sicherheitsfunktionen des Anwenderprogramms verwendet werden!**

#### Versionsvergleich für die zu prüfende Ressource öffnen

1. Im Strukturbaum die **Ressource** markieren, für welche der Versionsvergleich durchgeführt werden soll.
  2. In der Menüleiste **Extras, Versionsvergleich** wählen.
    - Der Dialog *Versionsübersicht* öffnet sich.
- Beim Versionsvergleich können verschiedene Ressource-Konfigurationen miteinander verglichen werden. Für weitere Informationen hierzu siehe Handbuch Versionsvergleich HI 801 285 D.

### Detailansicht der ke.config öffnen

1. Doppelklick auf die Zeile der Konfigurationsdatei *ke.config*.  
☒ Es wird die Detailansicht geöffnet.
  2. Tab **Globale Multisource-Variablen (CG)** wählen.  
☒ Hier werden alle Globalen Multisource-Variablen (CG) gelistet.
  3. Doppelklick auf die Zeile einer globalen Variablen.  
☒ Es öffnet sich der Globale-Variablen-Editor mit der Ansicht der Querverweise.
- Die Informationen werden im Klartext angezeigt. Das nachfolgende Bild 8 zeigt die Globalen Multisource-Variable *Globale\_Variable\_5* und ihre Verwendungsstellen im Anwenderprogramm und im Modbus-Slave-Set V2.

Name	Datentyp	Initialwert	Beschreibung	Zusatzkommentar	Technische Einheit	Retain	Konstant
5 Globale Variable_2	DWORD					<input type="checkbox"/>	<input type="checkbox"/>
6 Globale Variable_3	DWORD					<input type="checkbox"/>	<input type="checkbox"/>
7 Globale Variable_4	DWORD					<input type="checkbox"/>	<input type="checkbox"/>
8 <b>Globale Variable_5</b>	BOOL					<input type="checkbox"/>	<input type="checkbox"/>
9 Globale Variable_55555	Array_1					<input type="checkbox"/>	<input type="checkbox"/>

Querverweise			
Verwendung	Strukturinfo	Info	Strukturpfad
1 1x lesend/2x schreibend	Externe POE	Programm	/Konfiguration/HiMax/Programm
2 <b>schreibend</b>	Modbus-Slave-Set V2	Modbus-Slave-Set V2_1 [Globale Varia...	/Konfiguration/HiMax/Protokolle/Modbus-Slave-Set V2_1/Dataviews/Dataview/Register-Variablen/Register-Eingänge

Bild 8: Verwendungsstellen der Globalen Multisource-Variable *Globale\_Variable\_5*

## 11 Modbus-Topologien mit Prozessleitsystemen

Dieses Kapitel zeigt typische Kombinationen zur Integration einer HIMA Steuerung mit einem Prozessleitsystem über das Modbus-Slave-V2-Protokoll. Das Prozessleitsystem muss die Rolle des/ der Master übernehmen und initiiert hierbei alle Datentransfer-Prozesse mit dem HIMA Modbus-Slave-V2.

Die vom Modbus-Slave-V2 unterstützten Integrationsmöglichkeiten beschränken sich nicht auf die hier aufgeführten. Bei Fragen hierzu unterstützt Sie der HIMA Support.

Die Modbus-Kopplung der HIMA Steuerung kann an nahezu alle Prozessleit- und Visualisierungssysteme entweder über die RS485-Schnittstellen oder über die Ethernet-Schnittstellen der HIMA Steuerung erfolgen.

i

Bei Verwendung der Ethernet-Schnittstellen als Transportkanal müssen sich die HIMax/HIMatrix Steuerungen und das Prozessleitsystem im selben Subnet befinden oder bei Verwendung eines Routers die entsprechenden Routingeinträge besitzen.

### 11.1 Einschränkung des Modbus-Slave durch externen Transportweg

Das System bestimmt die Identität eines Masters anhand dessen externen Transportwegs (Feldbusschnittstelle oder IP-Adresse).

- Sind zwei Master auf den beiden redundant betriebenen Kommunikationsmodulen bezüglich des Transportwegs als identisch zu betrachten, müssen auch die in 4.3 genannten Eigenschaften der beiden Master gleich sein.
- Da der Transportweg als Identität eines Masters herangezogen wird, muss für jeden Transportweg ein eigener Master im Modbus-Slave-V2 angelegt werden, auch wenn sich logisch derselbe Master dahinter verbirgt.

### 11.2 Modbus über einen Transportweg

#### 11.2.1 Einzelne Linie ohne Redundanz

HIMA empfiehlt diese Variante nicht zu verwenden, wenn Ansprüche bzgl. der Verfügbarkeit bestehen. Für eine erhöhte Verfügbarkeit ist eine redundante Variante zu bevorzugen.

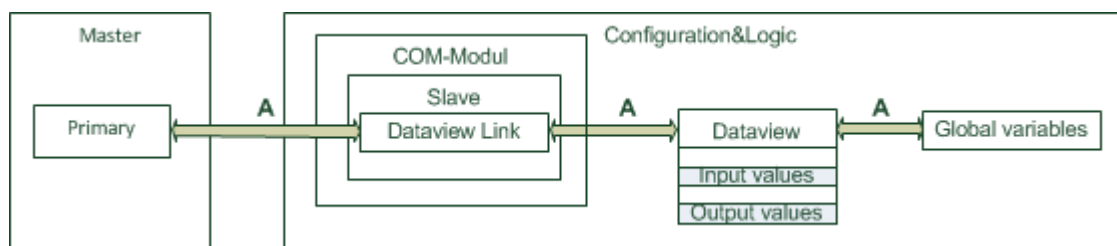


Bild 9: Eine Linie ohne Redundanz

### 11.3 Modbus über zwei Transportwege

In diesem Kapitel sind einige typische Kombinationen für eine redundante Modbus-Slave-V2 Verbindung dargestellt.

Diese Kombinationen sind mit einem oder zwei COM-Modulen realisierbar und bieten Redundanz über zwei Linien. Durch den Demand-Modus des Modbus-Slave-V2-Protokolls besteht kein weiterer Engineering-Aufwand im Anwenderprogramm, siehe Kapitel 4.

**i**

Gemäß der allgemein gültigen Regeln zur Erstellung von Ethernet-Netzwerken ist zu beachten, dass keine Netzschleifen entstehen. Datenpakete dürfen nur auf einem Weg zu einer Steuerung gelangen.

#### 11.3.1 Beide Linien aktiv (beide Linien lesen und schreiben)

Die globale Variable wird mit dem Wert der letzten Schreib Anforderung (A oder B) eines Masters beschrieben. Die Dataview-Links können in zwei redundanten COM-Modulen oder auf einem gemeinsamen COM-Modul konfiguriert werden.

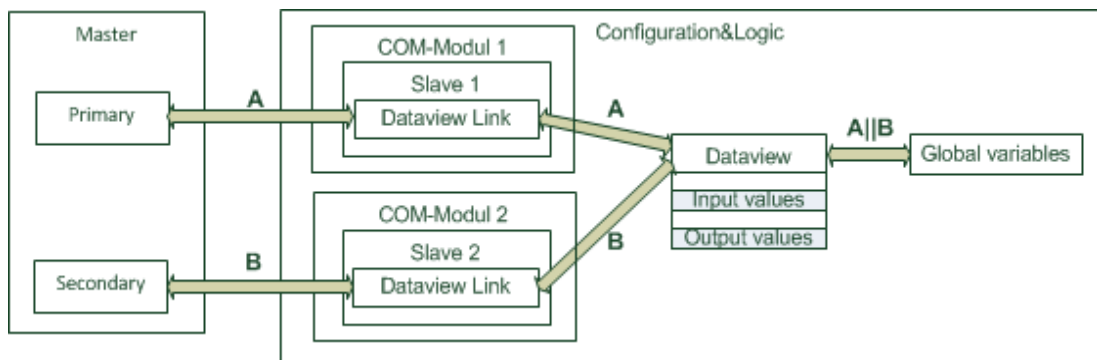


Bild 10: Zwei COM-Module, beide Linien aktiv (beide Linien lesen und schreiben)

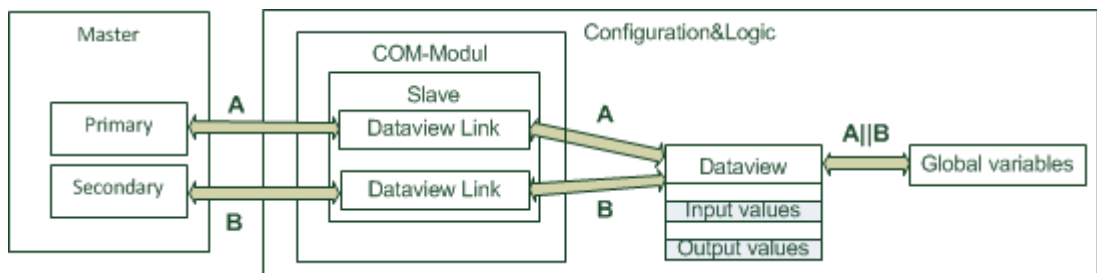


Bild 11: Ein COM-Modul, beide Linien aktiv (beide Linien lesen und schreiben)

### 11.3.2 Beide Linien aktiv mit optional umschaltbaren Linien

Wie beim Aufbau in Bild 10 wird die globale Variable mit dem Wert der letzten Schreibanforderung (A oder B) eines Masters beschrieben. Für die optionale Umschaltung der Master müssen im Slave jeweils ein zusätzlicher Dataview-Link in den zwei redundanten COM-Modulen konfiguriert werden.

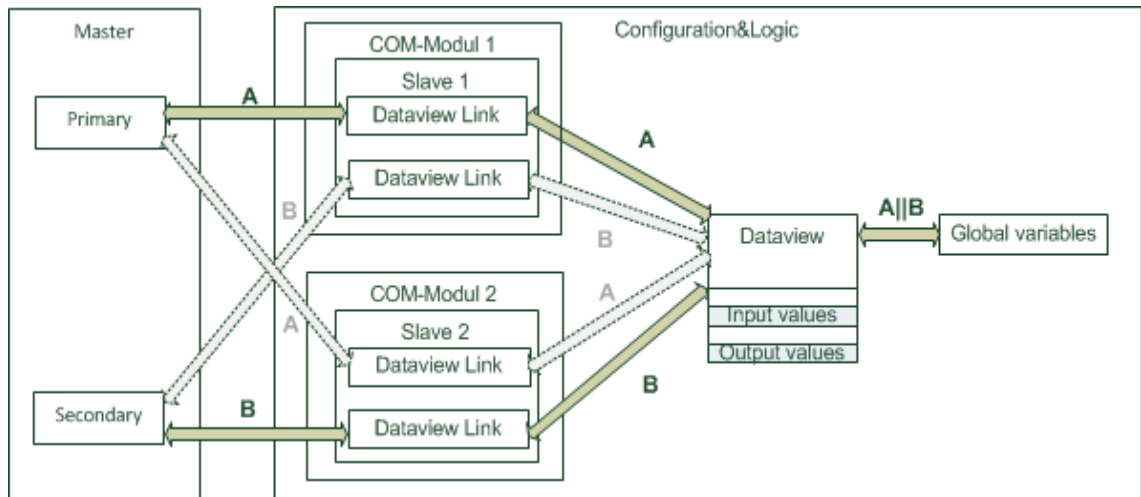


Bild 12: Zwei COM-Modul, beide Linien aktiv (beide Linien lesen und schreiben)

### 11.3.3 Cold Standby (eine Linie aktiv, die zweite Linie dient als Backup)

Die globale Variable wird mit dem Wert des jeweils aktiven Masters beschrieben. Bei Ausfall des aktiven Masters wird der Standby-Master aktiviert.

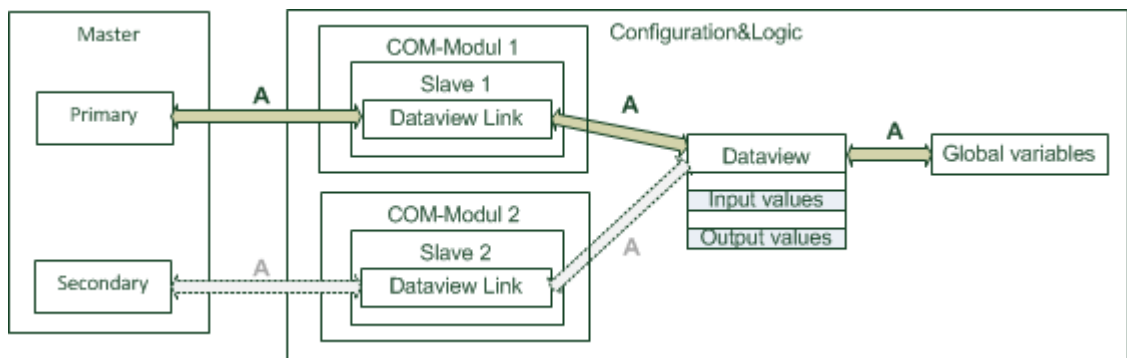


Bild 13: Cold Standby (eine Linie aktiv, die zweite Linie dient als Backup)

#### i

Die Dataview-Links können in zwei redundanten COM-Modulen oder auf einem gemeinsamen COM-Modul konfiguriert werden, wenn der redundante Master identische IP-Adressen bei einem redundanten Slave zulässt (simplex Slaves).

#### 11.3.4 Hot-Standby (beide Linien lesen und nur eine Linie schreibt)

Beide Master lesen, aber nur ein Master schreibt. Wenn der Schreib-Master ausfällt, bietet diese Variante eine sehr schnelle Umschaltung zwischen den Mastern.

Die globale Variable wird mit dem Wert des jeweils aktiven Schreib-Masters beschrieben. Bei Ausfall des aktiven Masters wird der Standby-Master aktiviert.

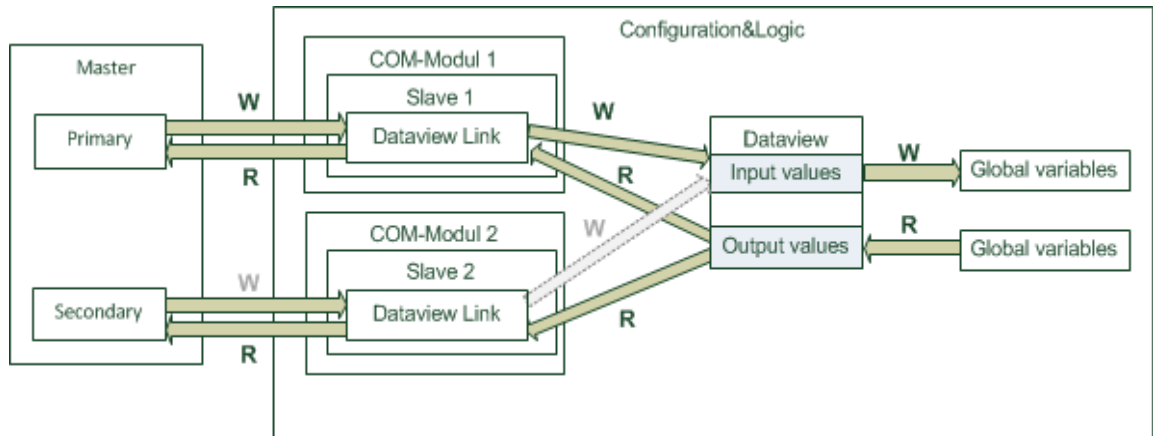


Bild 14: Hot-Standby (beide Linien lesen und nur eine Linie schreibt)

**i**

Die Dataview-Links können in zwei redundanten COM-Modulen oder auf einem gemeinsamen COM-Modul konfiguriert werden, wenn der redundante Master identische IP-Adressen bei einem redundanten Slave zulässt (simplex Slaves).

## 12 Allgemeines zur Datenübertragung

Weitere Informationen zur Ethernet- und Feldbuskommunikation, siehe SILworX Kommunikationshandbuch HI 801 100 D.

### 12.1 Erlaubte IP-Adressen Kombinationen des Masters

Allgemein gültige Regeln für die Vergabe von IP-Adressen und Subnet Mask müssen beachtet werden.

- 
- **1** Einige Leitsysteme erlauben es, mehreren Mastern die gleiche IP-Adresse zuzuweisen und eine hohe Verfügbarkeit bereitzustellen. In den Master-Objekten der Modbus-Slaves wird dann jeweils dieselbe IP-Adresse der beiden Master eingetragen.
- 

#### 12.1.1 Verwendete Netzwerkports für Ethernet-Kommunikation

Alle im folgenden aufgeführten Ports sind Destination Ports.

UDP-Ports / Verwendung

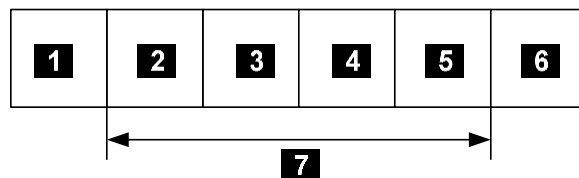
123	SNTP (Zeitsynchronisation zwischen PES und Remote I/O, sowie externen Geräten)
502	Modbus-Slave (vom Anwender änderbar)
6010	safeethernet und OPC
8001	Konfiguration der Remote I/O durch die PES
8000	Programmierung und Bedienung mit SILworX
34964	PROFINET Endpointmapper (für Verbindungsaufbau notwendig)
49152	PROFINET RPC-Server
49153	PROFINET RPC-Client

TCP Ports / Verwendung

502	Modbus-Slave (vom Anwender änderbar)
Xxx	TCP-SR durch Anwender vergeben

### 12.2 Serieller Modbus

Der HIMA Modbus-Slave-V2 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), siehe Tabelle 13.



<b>1</b> Beginn des Rahmens (Idle-Zeichen)	<b>5</b> CRC Checksumme (16 Bit)
<b>2</b> Adresse (8 Bit)	<b>6</b> Ende des Rahmens (Idle-Zeichen)
<b>3</b> Funktion (8 Bit)	<b>7</b> Modbus-Telegramm
<b>4</b> Daten (N * 8 Bit)	

Bild 15: Modbus-Telegramm im RTU-Format





## Anhang

### Glossar

Begriff	Beschreibung
ARP	Address Resolution Protocol: Netzwerkprotokoll zur Zuordnung von Netzwerkadressen zu Hardwareadressen
Bit-Variable	Variable, die bitweise adressiert wird
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)
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
SIL	Safety Integrity Level (nach IEC 61508)
SILworX	Programmiersoftware für HIMax, HIQuad X und HIMatrix
SNTP	Simple Network Time Protocol (RFC 1769)
SRS	System.Rack.Slot
SW	Software
TMO	Timeout
W	Write
WD	Watchdog
WDZ	Watchdog-Zeit

**Abbildungsverzeichnis**

Bild 1:	Konzept des Datenzugriffs auf eine globale Variable	16
Bild 2:	Redundante Master und mono Slave (HIMax und HIMatrix F*03 Steuerung)	19
Bild 3:	Redundante Master und redundanter Slave (HIMax Steuerung mit zwei COM-Modulen)	20
Bild 4:	Modbus-Slave-Set-V2 im SILworX Strukturbaum	24
Bild 5:	Modbus-Online-Anzeige	38
Bild 6:	Modbus-Slave-V2 im Versionsvergleich	40
Bild 7:	Lesen und Beschreiben einer Globale Multisource-Variable im Anwenderprogramm	41
Bild 8:	Verwendungsstellen der Globalen Multisource-Variable <i>Globale_Variable_5</i>	42
Bild 9:	Eine Linie ohne Redundanz	43
Bild 10:	Zwei COM-Module, beide Linien aktiv (beide Linien lesen und schreiben)	44
Bild 11:	Ein COM-Modul, beide Linien aktiv (beide Linien lesen und schreiben)	44
Bild 12:	Zwei COM-Modul, beide Linien aktiv (beide Linien lesen und schreiben)	45
Bild 13:	Cold Standby (eine Linie aktiv, die zweite Linie dient als Backup)	45
Bild 14:	Hot-Standby (beide Linien lesen und nur eine Linie Schreibt)	46
Bild 15:	Modbus-Telegramm im RTU-Format	47

**Tabellenverzeichnis**

Tabelle 1:	Zusätzlich geltende Handbücher	7
Tabelle 2:	Nötige Systemanforderung zum Betreiben des Modbus-Slave-V2	14
Tabelle 3:	Systemeigenschaften für Modbus-Slave-V2	15
Tabelle 4:	Unterschiede von Modbus-Slave Set und Modbus-Slave Variante2	15
Tabelle 5:	Zulässige Steckplätze der redundanten Modbus-Slave COM-Module	21
Tabelle 6:	Modbus-Slave-V2-Reload nach Änderungen	23
Tabelle 7:	Set-Objekte eines Modbus-Slave-Set-V2	27
Tabelle 8:	Modbus-Slave-Set-V2 Eigenschaften	27
Tabelle 9:	Dataviews Eigenschaften	28
Tabelle 10:	Systemvariablen	29
Tabelle 11:	Modbus-Slave Eigenschaften	29
Tabelle 12:	Systemvariablen	30
Tabelle 13:	Modbus-Master-Eigenschaften	32
Tabelle 14:	Dataview-Links Systemvariablen	33
Tabelle 15:	Dataview-Links Eigenschaften	34
Tabelle 16:	Modbus-Funktionscodes des HIMA Modbus-Slave	35
Tabelle 17:	Byte-Order	37
Tabelle 18:	Elemente der Modbus Online-Anzeige	39
Tabelle 19:	Modbus-Slave-V2-Konfigurationsdateien	41



HANDBUCH  
**Modbus V2**  
**HI 801 474 D**

---

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/)



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