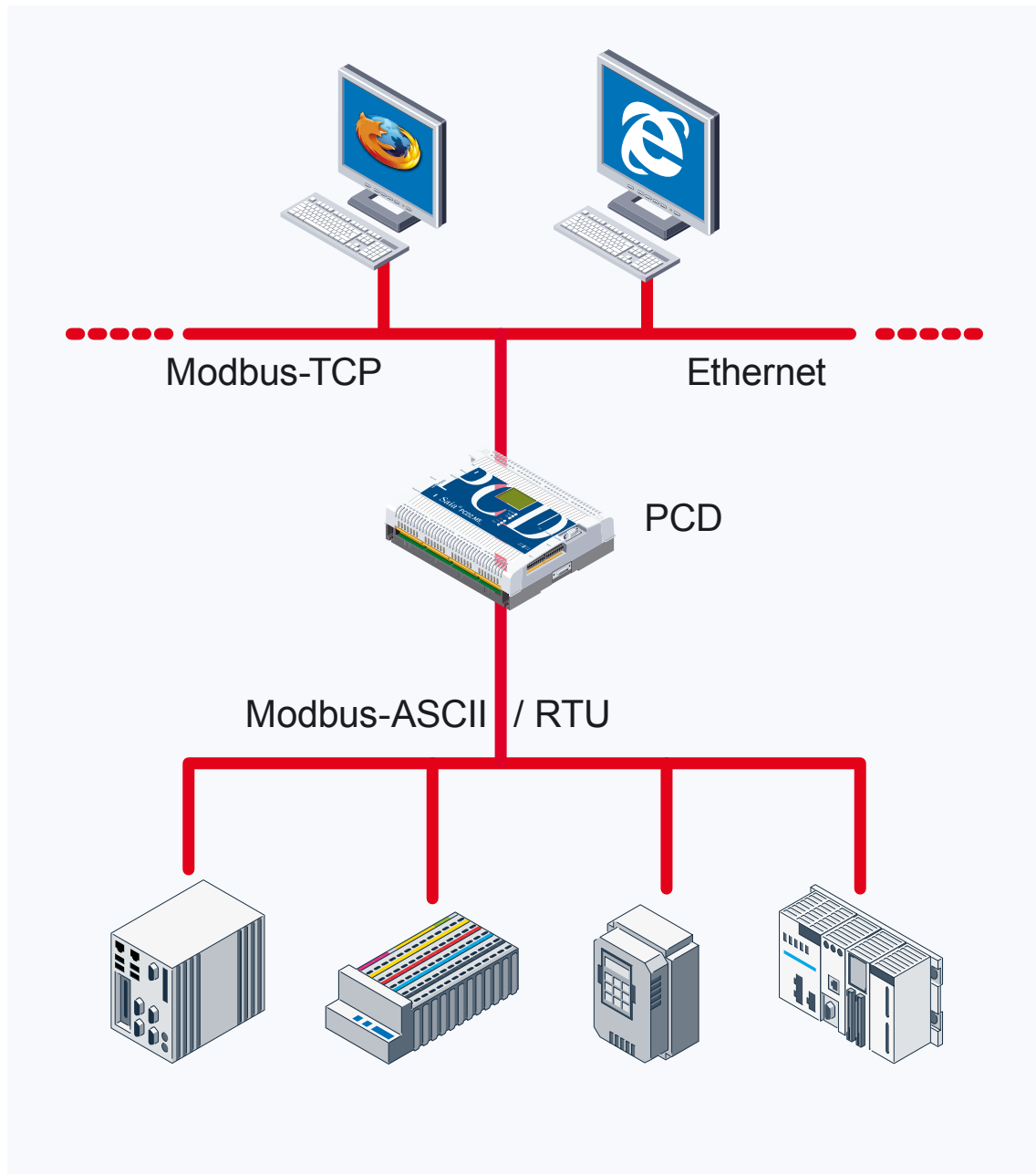




Modbus-Benutzerschnittstelle

Controls Division

0 Inhaltsverzeichnis

0.1	Dokument Historie	0-3
0.2	Warenzeichen	0-3

1 Das Modbus-Protokoll

1.1	ISO/OSI-Modell	1-1
1.2	Modbus-Kommunikation	1-1
1.3	Modbus-Medien	1-3

2 Saia® Modbus

2.1	Eigenschaften des Saia® Modbus	2-1
2.2	Diagnose	2-2
2.2.1	Client/Server-Diagnose-Flags	2-3
2.2.2	Diagnoseregister	2-3

3 Saia® Modbus-Server

3.1	Serverinstanz	3-1
3.2	Server-UID	3-2
3.3	Medienzuordnung	3-2

4 Saia® Modbus-Client

4.1	Client-Kanal	4-1
4.2	Modbus-Anforderungen	4-1

5 Modbus-FBox-Bibliothek**6 Modbus-Bibliothek der Systemfunktionen**

6.1	Einführung	6-1
6.2	Parameter	6-1
6.3	Fehlercodes	6-5
6.3.1	CSF-Fehlercodes	6-5
6.3.2	ModbusShell-Fehlercodes	6-6
6.3.3	Fehlercodes des ModbusDrivers	6-7
6.4	Funktionscodes	6-7
6.5	Protokolle	6-7
6.6	Modbus-Datentypen	6-7
6.7	Modbus-Bereichstypen (Server)	6-7
6.8	Modbus-Bereichszugriffstypen (Server)	6-8
6.9	Verarbeitungstypen (Client)	6-8
6.10	PCD-Medientypen	6-8
6.11	Ausnahmecodes	6-8

7 Modbus-CSF-Spezifikation

7.1	S.Modbus.InitSerialPort (Client / Server)	7-1
7.2	S.Modbus.OpenChannel (Client)	7-1
7.3	S.Modbus.SendReadRequest / S.Modbus.SendWriteRequest (Client)	7-2
7.4	S.Modbus.InitServer (Server)	7-4
7.5	S.Modbus.InitUID (Server)	7-4

7.6 S.Modbus.InitMap (Server) 7-5

A Anhang

A.1 Symbole A-1

A.2 Liste der Abkürzungen A-2

A.3 Adresse von Saia-Burgess A-3

0.1 Dokument Historie

Version	Geändert	Veröffentlicht	Anmerkungen
DE1		2009-06-18	Neue Ausgabe
DE1	2009-09-16	2009-09-21	Text in Zeichnung "Abbildung 7"
DE01	2010-02-10	2010-02-10	Wert in Tabelle Kapitel 7.2

0.2 Warenzeichen

Saia[®] and Saia[®]PCD sind registrierte Warenzeichen der Saia-Burgess Controls AG.

STEP7[®], SIMATIC[®], S7-300[®], S7-400[®] und Siemens[®] sind registrierte Warenzeichen der Siemens AG.

Technische Änderungen sind abhängig vom Stand der Technik.

Saia-Burgess Controls AG, 2009. © Alle Rechte vorbehalten.

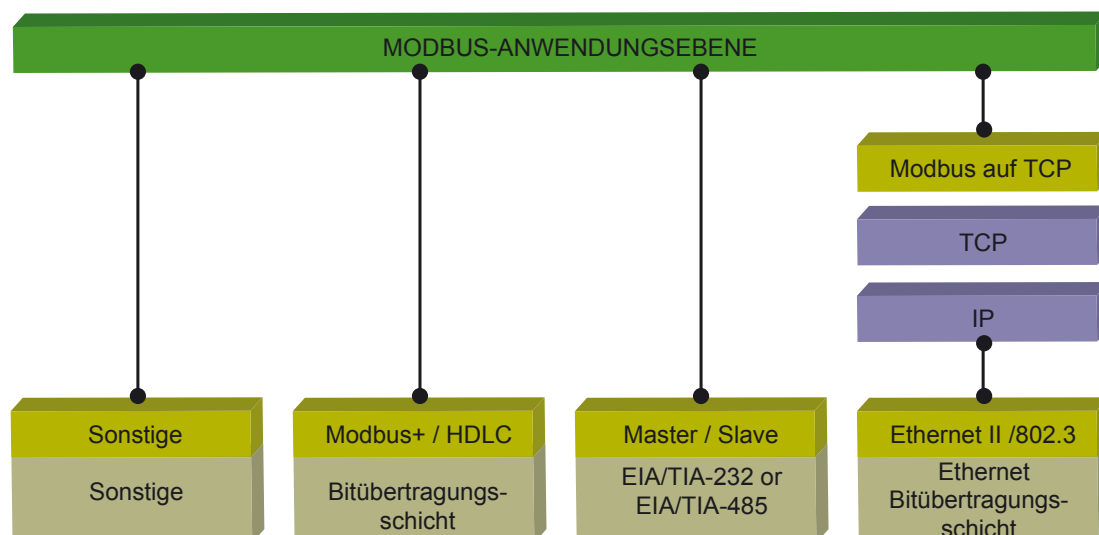
Veröffentlicht in der Schweiz

1 Das Modbus-Protokoll

1.1 ISO/OSI-Modell

1

Modbus ist ein Nachrichtenprotokoll auf Anwendungsebene für die Kommunikation zwischen einem Master und vielen Slaves. Es kann in RTU- und ASCII-Modi in seriellen Leitungen (RS 485, RS 232, RS 422) oder im Ethernet über TCP und UDP implementiert werden. Bei TCP/UDP ist der öffentliche Anschluss 502 für den Modbus reserviert. Die anderen Anschlüsse können jedoch festgelegt werden.



1.2 Modbus-Kommunikation

Modbus ist ein Frage/Antwort-Protokoll, das Dienste anbietet, die durch Funktionscodes angegeben sind. 127 öffentliche und benutzerdefinierte Funktionscodes stehen zur Verfügung, die für den Zugriff auf Daten (Lesen, Schreiben), die Diagnose und andere Dienste verwendet werden können.

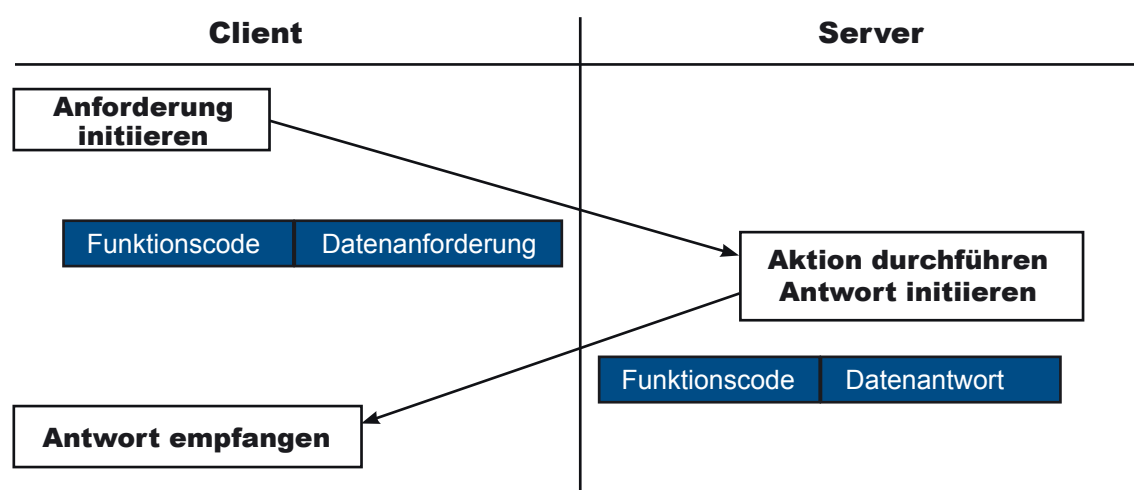
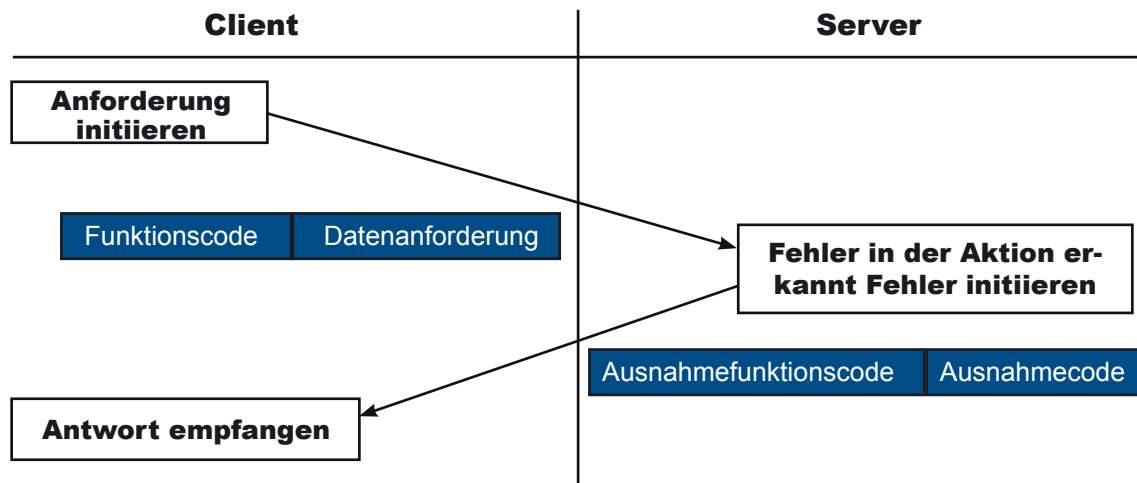


Abbildung 2: Modbus-Transaktion (fehlerfrei)



1

Abbildung 3: Modbus-Transaktion (Ausnahmereaktion)

Im Serienmodus können bis zu 247 Slaves im Bus vorhanden sein, jeder mit einer eindeutigen Adresse, Einheitsbezeichner (UID) genannt.

RTU ist der Standardmodus für einen Modbus in seriellen Leitungen. Dieser Modus ist zeitgesteuert: Intercharacter- und Interframe-Zeiten müssen berücksichtigt werden.

Für einen Modbus über eine serielle Leitung kann auch der ASCII-Modus verwendet werden. In diesem Fall wird die Kommunikation über bestimmte Start- und Stoppzeichen gesteuert. Da jedoch jedes Datenbyte mit 2 Zeichen gesendet wird, werden zweimal so viele Daten übertragen wie beim RTU.

Die Modbus-Kommunikation basiert auf bestimmten Frames, die mindestens einen Funktionscode und die benötigten Daten umfassen. Bei einem Modbus über eine serielle Leitung wird ein CRC und eine Adresse angehängt. Die Adresse kann auch für einen Modbus über IP verwendet.

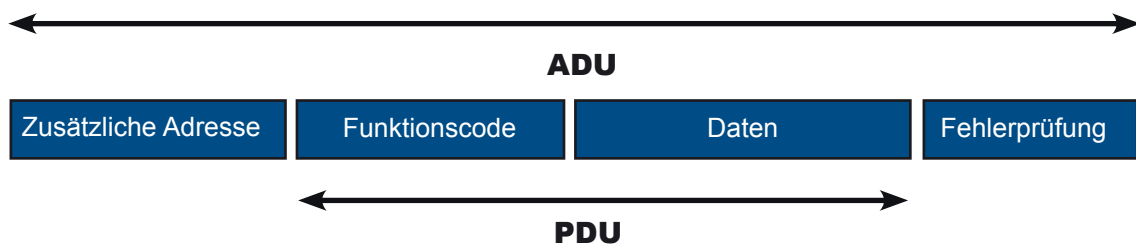


Abbildung 4: Allgemeiner Modbus-Frame

2 Anforderungsarten können gesendet werden:

- Leseanforderungen: Daten = Modbus-Adresse + Anzahl zu lesender Elemente
→ Antwort: Daten = Anzahl der Datenbytes + Daten
- Schreibanforderungen: Daten = Modbus-Adresse + Anzahl zu schreibender Elemente + zu schreibende Daten
→ Antwort: Daten = Anforderungsdaten
- Falls ein Fehler bezüglich der angeforderten Modbus-Funktion auftritt, sendet der Slave eine Ausnahmeantwort mit einem Ausnahmefunktionscode:
→ Ausnahmefunktionscode = Anforderungsfunktionscode + 0x80, Daten = Ausnahmecode

1.3 Modbus-Medien

Modbus legt 4 Medientypen fest:

- Haltereister (HR): 16 Bits - lesen/schreiben
- Eingaberegister (IR): 16 Bits - nur lesen
- Spulen: 1 Bit - lesen/schreiben
- Abstrakte Eingaben (DI): 1 Bit - nur lesen

1

Diese Modbus-Medien beziehen sich in jedem Gerät auf gerätespezifische Medien. Jedes Gerät kann über eine eigene Datenorganisation verfügen. Deshalb muss der Benutzer vor der Benutzung des Modbus einen Blick in die Dokumentation seines Gerätes werfen. Nachfolgend sehen Sie zwei Beispiele einer Datenorganisation.

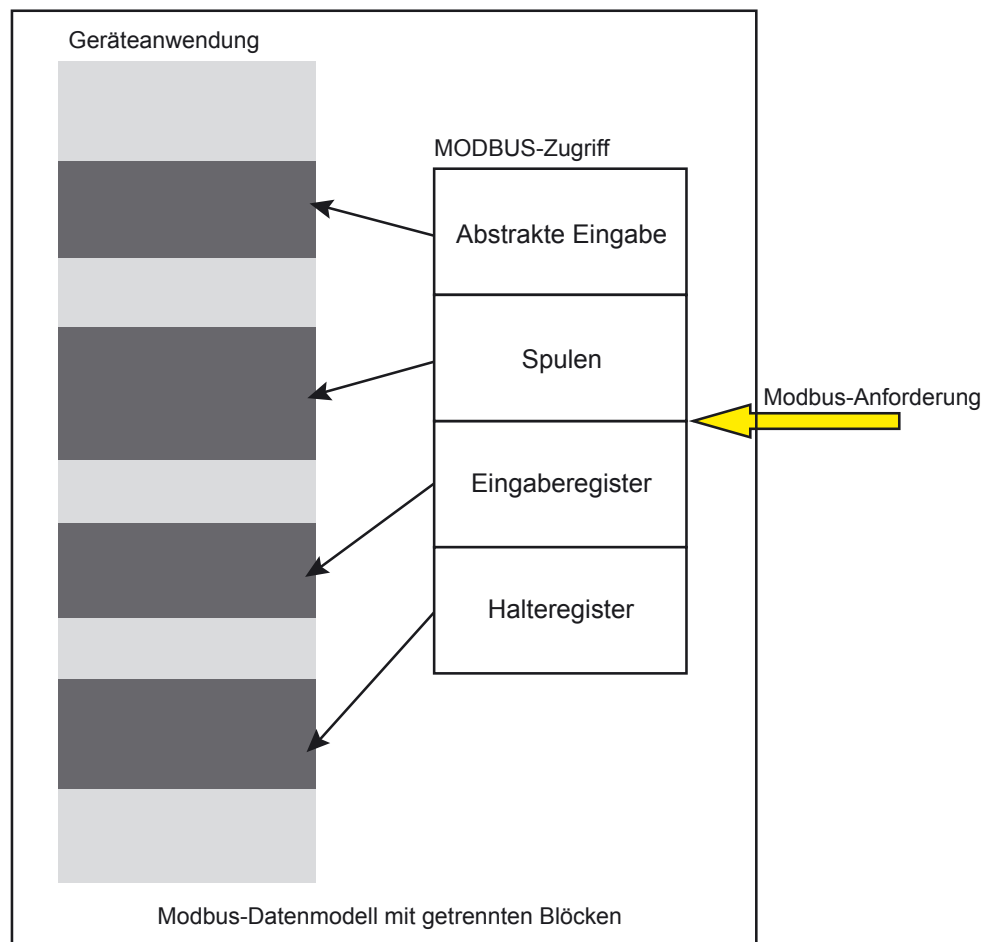


Abbildung 5: Modbus-Datenmodell mit getrennten Blöcken

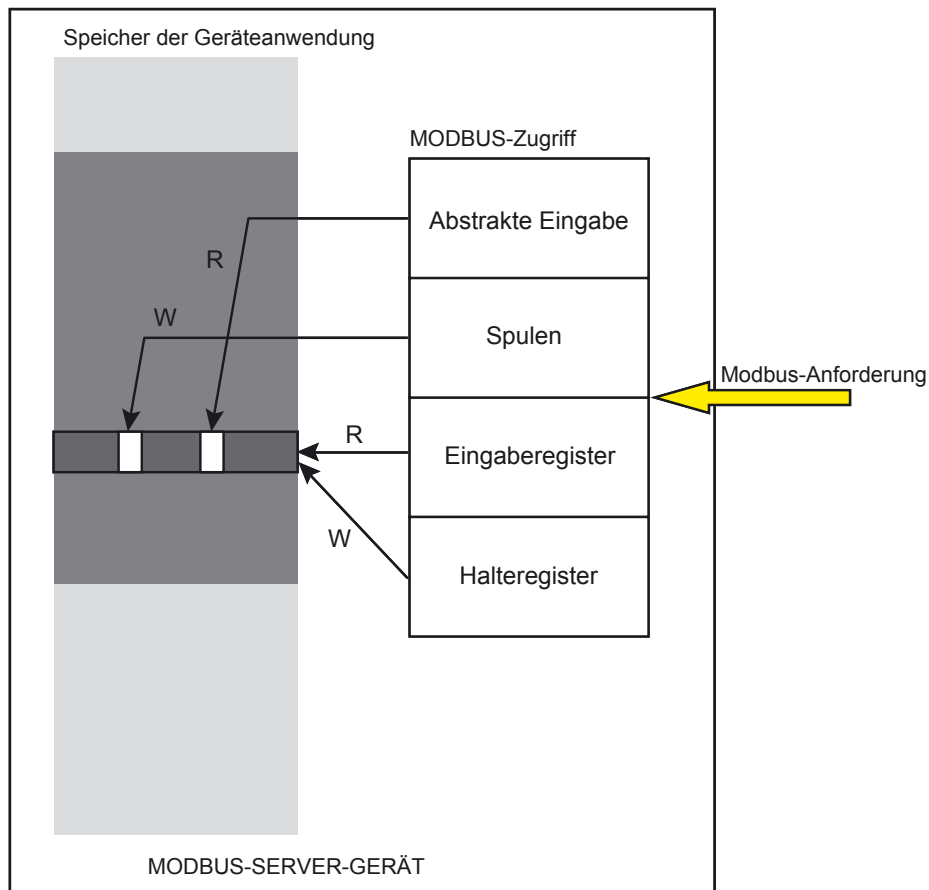
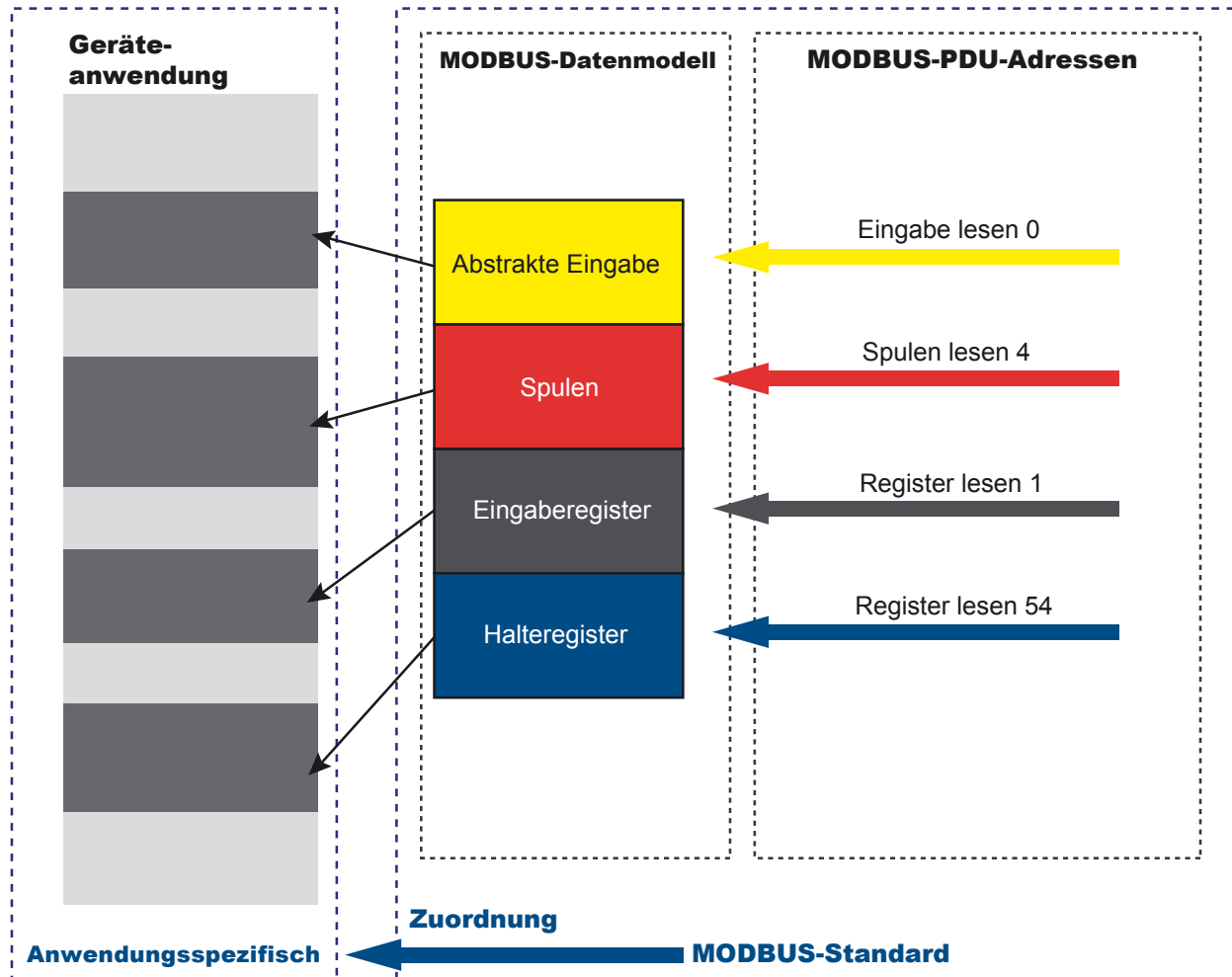


Abbildung 6: Modbus-Datenmodell mit nur einem Block

In einem Modbus-Frame kann auf alle Daten von 0 bis 65535 zugegriffen werden. Dabei greift das Modbus-Datenmodell auf jedes Element innerhalb eines Datenblocks von 1 bis n zu. Anschließend muss das Modbus-Datenmodell mit der Geräteanwendung verbunden werden. Die Zuordnung zwischen Modbus-Datenmodell und der Geräteanwendung ist je nach Anbieter unterschiedlich.

**Abbildung 7: Modbus-Zugriffsmodell**

Weitere Informationen zu Modbus finden Sie unter <http://www.modbus.org>.

2 Saia® Modbus

Dieser Teil beschreibt die Implementierung des Modbus-Protokolls bei Saia® PCD. In diesem Teil werden die Begriffe Client und Server entweder für TCP / UDP oder für RTU / ASCII statt Master und Slave verwendet.

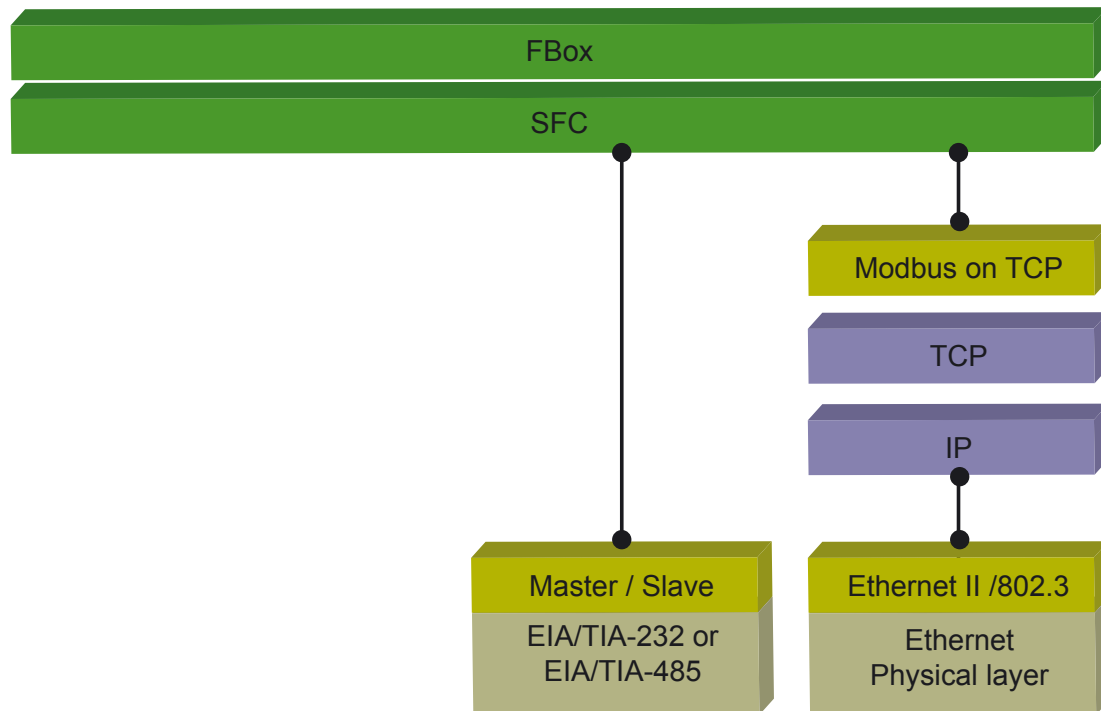


Abbildung 8: Saia® Modbus Kommunikations-Stack

Mit Hilfe der Saia-Implementierung des Modbus kann der Benutzer Modbus-Clients und -Server konfigurieren und nutzen. Dies erfolgt über eine zugewiesene Schnittstelle, die aus FBoxen und CSFs besteht.

2.1 Eigenschaften des Saia® Modbus

- **Client-** und Server-Funktionen
- **Modi:** IP (TCP und UDP) und seriell (RTU und ASCII)
- **Serieller Modbus:** an allen seriellen Anschlüssen verfügbar, von 1200 Bd bis 115,2 kBd (Zurzeit unterstützen die F2xx-Module den Modbus nicht: die Vorbereitung läuft jedoch)
- **8 Funktionscodes:**
 - Spulen lesen 1
 - Abstrakte Eingaben lesen 2
 - Haltereister lesen 3
 - Eingaberegister lesen 4
 - Einzelne Spule schreiben 5

- Mehrere Spulen schreiben 6
- Einzelnes Halteregeister schreiben 7
- Mehrere Halteregeister schreiben 8
- **Medienzuordnung:** benutzerdefinierbar
- **Datenverarbeitung:** viele verfügbar, um eine Kompatibilität mit Geräten sicherzustellen, die nicht von Saia stammen und einen 32 Bit Zugriff auf Register Gleitpunktformate, ... zu ermöglichen.
- **Server:** maximal 4 auf einem PCD-System (ein Server wird durch ein eindeutiges {Anschluss / Protokoll} Paar festgelegt) – Beispiel: 1 Server TCP-Anschluss 502, 1 Server TCP-Anschluss 503, 1 Server UDP-Anschluss 502 und 1 Server RTU am seriellen Anschluss 2.
- **Unit-IDs:** maximal 10 auf einem PCD-System: 10 = 9 + UID 255 (UID von 0 bis 254 sind berechtigt; UID 255 existiert standardmäßig auf dem Server, muss jedoch vor dem Zugriff konfiguriert werden).
- **Zuordnungsbereiche:** maximal 10 pro UID – Falls mehr als 10 Zuordnungsbereiche notwendig sind, kann eine zusätzliche UID festgelegt werden (Beispiel: definieren Sie eine UID für Spulen/Abstrakte Eingaben und eine weitere für Halte-/Eingaberegister).
- **Kanäle:** maximal 10 auf einem PCD-System (ein Kanal wird durch ein eindeutiges {Anschluss/Protokoll} Paar festgelegt) – Beispiel: 1 Kanal TCP an Serveranschluss 502, 1 Kanal TCP an Serveranschluss 503, 1 Kanal UDP an Serveranschluss 502 und 1 Kanal RTU der Anschluss 1 nutzt.
- **Verbindungen:** 26 Verbindungen können gleichzeitig geöffnet sein. Von diesen 26, können maximal 10 Verbindungen auf der Client-Seite geöffnet sein. Die verbleibenden Verbindungen können vom Server genutzt werden. Anmerkung: eine Verbindung wird durch eine eindeutige Dreiergruppe {Anschluss – Protokoll – IP-Adresse} im TCP/UDP und nur das Paar aus Anschluss/ Protokoll) im Serienmodus festgelegt). Die Verbindungen werden mit der ersten Modbus-Anforderung geöffnet. Im TCP/UDP werden die Verbindungen nach Ablauf einer vom Benutzer festgelegten Zeit automatisch geschlossen.

2.2 Diagnose

Diagnose-Flags und -register werden verwendet, um den Zustand der UIDs (UID-Diagnose) und Kanäle (Kanaldiagnose) zu berichten. Sie werden auch im Serienmodus zum Berichten von Fehlern verwendet, die in der seriellen Leitung auftreten (Server-Diagnose und Kanaldiagnose).

Im Diagnoseregister sind mit Ausnahme von „Zählung wiederholen“ und „Reaktionszeit überschritten“ alle Bits kumuliert (d.h., der Benutzer muss sie löschen).

2.2.1 Client/Server-Diagnose-Flags

Adresse	Name	Beschreibung
Fxxx	RBSY	Kanal: wird nicht verwendet UID: UID ausgelastet Server: wird nicht verwendet
Fxxx + 1	RFUL	Kanal: wird nicht verwendet UID: Anforderung mit gültiger UID empfangen und verarbeitet Server: wird nicht verwendet
Fxxx + 2	RDIA	Empfangsdiagnose (siehe Diagnoseregister)
Fxxx + 3	TBSY	Kanal: Kanal ausgelastet (wenn der Kanal ausgelastet ist, kann keine Anforderung an den Kanal gesendet werden) UID: wird nicht verwendet Server: wird nicht verwendet
Fxxx + 4	Reserviert	-
Fxxx + 5	TDIA	Übertragungsdiagnose (siehe Diagnoseregister)
Fxxx + 6	Reserviert	-
Fxxx + 7	NEXE	Client: die Anforderung wurde nicht erfolgreich verarbeitet UID: wird nicht verwendet Server: wird nicht verwendet

2.2.2 Diagnoseregister

	Bit	Bezeichnung	Beschreibung	Protokoll	Verwendung
E M P F A N G	0	Überlauffehler	Überlaufen des internen Empfangspuffers	Seriell	Kanal/Server
	1	Paritätsfehler	Fehler in der Parität des empfangenen Zeichens	Seriell	Kanal/Server
	2	Formfehler	Falsche Baud-Rate	Seriell	Kanal/Server
	3	Unterbrechungsfehler	Unterbrechung in der seriellen Leitung	Seriell	Kanal/Server
	4	CRC-Fehler	CRC falsch	Seriell	Kanal/Server
	5	-	-		
	6	-	-		
	7	-	-		
	8	Längenfehler	Telegramm mit unzulässiger Länge empfangen	Seriell	Kanal/Server
	9	Medienzugriffsfehler	Fehler beim Zugriff auf PCD-Medien	Alle	Kanal/UID
	10	-	-		
	11	Server-Startfehler	Server kann nicht starten	Alle	Server
	12	Bereichsfehler	Fehler in der Adresse oder Anzahl der Modbus-Elemente, auf die zugegriffen wird (siehe unten).	Alle	UID
	13	Wertfehler	Ein Wert im Telegramm ist ungültig (Medientyp, ...)	Alle	UID
	14	Bereichszugriffsfehler	Bereich nicht zugänglich (UID ausgelastet, Bereichszugriff nicht zulässig)	Alle	UID
	15	-	-		

Ü B E R E I C H S T R A G E N G	16	Zählung wiederholen	Anzahl durchgeführter Wiederholungen	Alle	Kanal
	17				
	18				
	19				
	20	Reaktionsausnahme	Reaktionsausnahme empfangen	Alle	Kanal
	21	Reaktionszeit überschritten	Nach der Zeitüberschreitung wurde keine Antwort empfangen	Alle	Kanal
	22	Reaktionsfehler	Ungültige Antwort empfangen	Alle	Kanal
	23	-	-		
	24	Versandfehler	Anforderung konnte nicht gesendet werden	Alle	Kanal
	25	-	-		
	26	Client-Startfehler	Client konnte nicht gestartet werden	Alle	Kanal
	27	-	-		
	28	Medienfehler	Fehler in den PCD-Medienparametern	Alle	Kanal
	29	-	-		
	30	Statusfehler	Fehler im Client-Status (interner Fehler)	Alle	Kanal
	31	Programmfehler	Fehler im Anwendungsprogramm (z. B.: Aufruf eines CSFs, während TBSY oder RBSY eingestellt sind)	Alle	Kanal, UID

Ein “**Bereichsfehler**” kann verschiedene Ursachen haben:

- Keine passende Zuordnung gefunden: In den benutzerdefinierten Zuordnungsbereichen (und auch in den Standardzuordnungsbereichen, wenn DEFMAP = 1) wurde kein Zuordnungsbereich gefunden, der zur Datenstartadresse und Anzahl der Modbus-Elemente passt.
- Anzahl der PCD-Medien, auf die zugegriffen werden soll, ist größer als der PCD-Bereich des Bereichs: Beispiel: Sie versuchen bei Verschiebung 2 auf 8 PCD-Register in einem Bereich zuzugreifen, der 8 PCD-Register enthält. Wenn Sie Löcher verwenden, müssen Sie auf die Anzahl der benötigten PCD-Register achten.
- Ungerade Zahl oder Verschiebung des HRs während des Zugriffs auf 32-Bit-Register: In einem 32-Bit-Bereich gehört 1 PCD-Register (32 Bits) zu 2 Modbus-Halteregistern. Deshalb ist es nicht zulässig, bei einer ungeraden Verschiebung auf PCD-Register oder mit einer ungeraden Zahl auf Halteregister zuzugreifen.

(siehe Abbildung 12: Zuordnung von Halteregistern (16 Bits) zu PCD-Registern (32 Bits) in einem 32-Bit-Bereich)

Ein “**Client-Startfehler**” kann verschiedene Ursachen haben:

- TCP/UDP: kann keine Verbindung zum Server herstellen
- RTU/ASCII: kann nicht mit dem Abhören der seriellen Leitung beginnen (ungültige Anschlussnummer → interner Fehler)

Ein “**Medienfehler**” kann auftreten, wenn die Anzahl von PCD-Medien, auf die zugegriffen werden soll, die in 7.2 (Parameter “Zählung”) festgelegte Grenze übersteigt oder wenn während des Zugriffs auf die PCD-Medien ein Fehler auftritt (PCD-Medien ungültig, Adresse der PCD-Medien ungültig...).

3 Saia® Modbus-Server

Auf der Server-Seite sind folgende Kommunikationsschritte notwendig:

1. Serverinstanz erstellen
2. Empfangene Anforderung verarbeiten
3. Antwort senden

Auf den Saia® Modbus-Server können Sie über TCP/UDP oder eine serielle Leitung (RTU/ASCII) zugreifen.

Saia® PCD

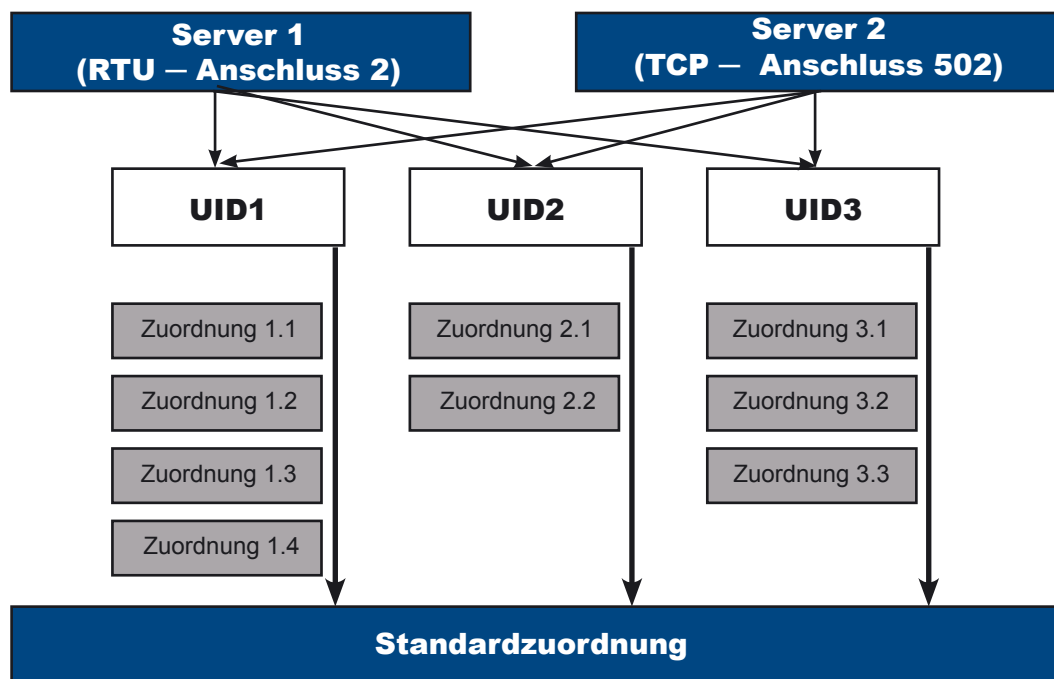


Abbildung 9: Allgemeine Modbus-Architektur einer PCD-Server-Station

3.1 Serverinstanz

Viele Serverinstanzen können auf einem Saia®PCD festgelegt werden und ermöglichen so mehrere Zugriffe über die serielle Leitung oder IP.

Ein Server wird als Paar definiert {Anschluss – Protokoll}. Im Serienmodus ist es nicht möglich, mehr als einen Server an einem Anschluss zu verwenden (RTU an Anschluss 1, ASCII an Anschluss 2, ...). Bei TCP/UDP ist der Anschluss ein logischer Anschluss. Mehrere Server können am gleichen Anschluss oder am gleichen Protokoll festgelegt werden (TCP an Anschluss 502, UDP an Anschluss 502, TCP an Anschluss 503, ...). Darüber hinaus ist es nicht möglich 2 identische Serverinstanzen festzulegen.

Der Zugriff auf die Server erfolgt über Verbindungen. Sowohl seriell als auch beim UDP wird eine Verbindung pro Serverinstanz verwendet. Beim TCP wird eine Verbindung pro verbundenem Client benötigt.

Für jeden Server stehen Diagnose-Flags und -register zur Verfügung (wird nur im Serienmodus verwendet).

3.2 Server-UID

Der Saia® Modbus-Server zeichnet sich auch durch einen oder mehrere Einheitsbezeichner (UID) aus. Alle Serverinstanzen der Station beziehen sich auf den UID. Ein UID ist auch mit einer bestimmten Medienzuordnung und einer bestimmten Datenverarbeitung verknüpft. Des Weiteren können Sie verschiedene UIDs auf einem Saia® PCD konfigurieren. Ein UID kann als Stationsnummer betrachtet werden. Der Modbus-Server verarbeitet alle Anforderungen, die an einen seiner UIDs gerichtet wurden.

Im Serienmodus antwortet der Server nur seinen festgelegten UIDs. Im Fall eines gesendeten UIDs (UID 0) verarbeitet der Server die Anforderung für alle festgelegten UIDs. Es ist nicht zulässig, den gleichen UID zweimal auf einem seriellen Bus festzulegen (ausgenommen 0).

Im TCP/UDP antwortet der Server allen Anforderungen entweder mit einem Ausnahmecode oder einer erfolgreichen Antwort. Es wird keine UID gesendet.

Dadurch, dass es mehrere UIDs gibt, können verschiedene Konfigurationen bei der Medienzuordnung mit verschiedenen Verarbeitungstypen verwendet werden oder die vorhandene Standardzuordnung kann erweitert werden.

Für jeden UID sind Diagnose-Flags und -register verfügbar.

3.3 Medienzuordnung

Modbus verwendet 4 Medientypen: Spulen, Abstrakte Eingaben, Haltereister, Eingaberegister.

Damit auf einem Modbus-Server auf die PCD-Medien zugegriffen werden kann, müssen diese Modbus-Medien PCD-Medien zugeordnet werden. Mithilfe der Zuordnungsbereiche und UIDs kann der Benutzer die Zuordnung der Modbus-Medien zu Saia® PCD-Medien konfigurieren. **Mindestens eine UID muss festgelegt werden.**

Spulen und Abstrakte Eingaben können nur in Flags oder E/As und ohne Verarbeitung zugeordnet werden.

Halteregister und Eingaberegister können nur in Zeitschaltern/Zählwerken/Registern/DB zugeordnet werden. Dabei muss jedoch auf die Datenverarbeitung geachtet werden. Zusätzlich können Register auf Client-Seite vom Server gelesen werden und in einem Text gespeichert werden oder ein Text kann gelesen und in Server-Register geschrieben werden. Dies geschieht mit den Funktionscodes für Halte- oder Eingaberegister.

Um eine Kompatibilität mit Nicht-Saia-Geräten sicherzustellen, können auf Client- oder Server-Seite unterschiedliche Verarbeitungstypen verwendet werden: 16 Bits/32 Bits, 16 Bits mit Vorzeichen/16 Bits ohne Vorzeichen, 32 Bits FFP/32 Bits IEEE-754, Verschiebung und bei 32-Bit-Registern Auslagerung und Löcher.

	Modbus-Medien			PCD-Medien			
Zugriffstyp	Medientyp	Start Adresse	Bereich	Medientyp	Start Adresse	Bereich	Feldtyp

Abbildung 10: Festlegung eines Zuordnungsbereichs

Der **Feldtyp** legt die Verarbeitung fest, die mit den empfangenen Daten durchgeführt wird:

- Für Spulen gibt es keine besondere Verarbeitung

- 16 Bits mit Vorzeichen (nur Schreibanforderungen): wenn HR/IR-Wert < 0, MSW am PCD-Register auf 0xFFFF einstellen
Beispiel: HR = 0x8001 pro PCD-Register = 0xFFFF 8001
- 16 Bits ohne Vorzeichen pro MSW des PCD-Registers mit Einstellung 0x0000
Beispiel: HR = 0x8001 pro PCD-Register = 0x0000 8001
- 32 Bits: 2 HR / IR = 1 PCD-Register, keine Verarbeitung
- 32 Bits IEEE: Konvertierung von FFP in IEEE wurde vor dem Versenden von Daten über den Bus durchgeführt (wird nur verwendet, wenn der Benutzer FFP-Werte hat, andernfalls 32 Bits verwenden), Konvertierung von IEEE in FFP wurde vor dem Schreiben der Daten in die PCD-Meiden durchgeführt (wird nur verwendet, wenn der Benutzer den Wert in FFP benötigt, andernfalls 32 Bits verwenden).

Das folgende Schema zeigt, wie die Daten zwischen PCD und Modbus-Frame verarbeitet werden.

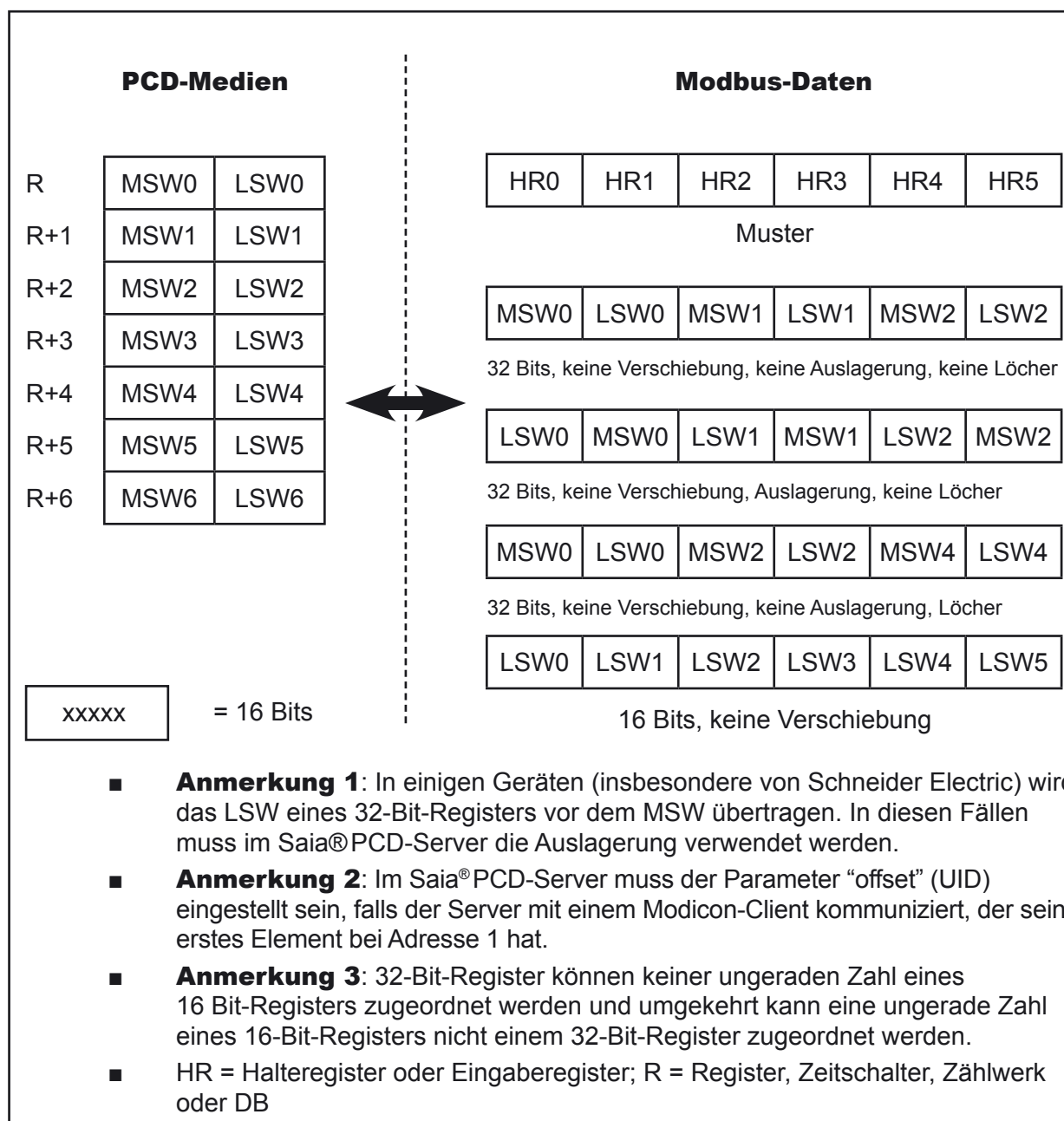


Abbildung 11: verschiedene Verarbeitungstypen für die Zuordnung von Modbus-Halteregistern zu PCD-Registern (Annahme: das Modbus-HR mit der Adresse 0 ist dem PCD-Register 0 zugeordnet)

Standardzuordnung

Zugriffs- typ	Modbus-Anforderung			PCD			Feldtyp
	Medientyp	Start- adr.	Be- reich	Medientyp	Start- adr.	Be- reich	
ReadWrite	MB_HOLDING_REG_MEDIA	1	10000	PCD_REG_MEDIA	0	10000	MB_AREA_16BITS_SIGNED
ReadWrite	MB_HOLDING_REG_MEDIA	10001	10000	PCD_REG_MEDIA	0	10000	MB_AREA_32BITS
ReadWrite	MB_HOLDING_REG_MEDIA	20001	10000	PCD_REG_MEDIA	0	10000	MB_AREA_32BITS_IEEE
ReadOnly	MB_INPUT_REG_MEDIA	1	10000	PCD_TC_MEDIA	0	10000	MB_AREA_16BITS_SIGNED
ReadOnly	MB_INPUT_REG_MEDIA	10001	10000	PCD_TC_MEDIA	0	10000	MB_AREA_32BITS
ReadWrite	MB_COILS_MEDIA	1	10000	PCD_FLAG_MEDIA	0	10000	MB_AREA_COILS
ReadOnly	MB_DISCR_INPUT_MEDIA	1	10000	PCD_IO_MEDIA	0	10000	MB_AREA_COILS

4 Saia® Modbus-Client

Die Modbus-Kommunikation wird vom Modbus-Client initialisiert. Beim Saia PCD-System erfolgt dies über Modbus-Kanäle.

Auf der Client-Seite sind folgende Kommunikationsschritte notwendig:

1. Client-Instanz erstellen
2. Verbindung zum Server herstellen (TCP)
3. Anforderung senden
4. Antwort verarbeiten

4

Mit dem Saia® Modbus-Client kann der Benutzer Modbus-Leseanforderungen und -Schreibanforderungen über TCP/UDP oder eine serielle Leitung (RTU/ASCII) senden.



Es ist nicht zulässig, 2 Modbus-Clients im gleichen seriellen Netzwerk zu verwenden. Wenn 2 Clients gleichzeitig über den Bus kommunizieren, kann nicht garantiert werden, dass sich das Netzwerk richtig verhält.

4.1 Client-Kanal

In einem Saia®PCD-System können viele Modbus-Kanäle festgelegt werden, so dass über eine serielle Leitung oder über IP mit verschiedenen Modbus-Servern kommuniziert werden kann.

Ein Kanal wird als Paar definiert {Anschluss–Protokoll}. Der Anschluss ist entweder die lokale Anschlussnummer im Serienmodus oder bei TCP/UDP der Anschluss am entfernten Server. Beispiel: {RTU–Anschluss 2}, {TCP–Anschluss 502}.

Für jeden Kanal sind Diagnose-Flags und -Register verfügbar. Darüber hinaus können 2 Zeitüberschreitungen festgelegt werden: eine Reaktionszeitüberschreitung, die die Zeit angibt, innerhalb der eine Antwort vom Server erwartet wird und eine Zeitüberschreitung zum Schließen der Verbindung, die die Zeit angibt, ab der die Verbindung geschlossen werden muss, wenn keine Aktivität vorhanden ist.

Der Kanal ist vom Senden der Anforderung bis zum Empfang der Antwort (oder bis zur Reaktionszeitüberschreitung) ausgelastet. Während dieses Zeitraums kann der Kanal nicht für andere Anforderungen verwendet werden.

Für jeden Kanal sind Diagnose-Flags und -Register verfügbar.

4.2 Modbus-Anforderungen

Eine Modbus-Anforderung wird über einen festgelegten Kanal gesendet und ist an einen bestimmten UID (und eine IP-Adresse für TCP/UDP) gerichtet. Sie können auch die Datenverarbeitung einstellen, die verwendet werden muss, um Daten senden und empfangen zu können.

Innerhalb eines angegebenen Zeitraums wird eine Antwort vom Server erwartet. Wenn innerhalb des Zeitfensters keine Antwort kommt, existiert der Server nicht (oder der UID ist im Serienmodus nicht festgelegt). Im Fall eines gesendeten UIDs (UID 0) über eine serielle Leitung wird keine Antwort vom Client erwartet. Sie müssen nur die festgelegte Zeit warten, bevor Sie die nächste Anforderung senden.

Beim Senden der Anforderung muss festgelegt werden, wie die Daten (vor dem Sen-

den oder nach dem Empfang) verarbeitet werden sollen:

- Coils_DI_Processing, keine besondere Verarbeitung
- 16 Bits mit Vorzeichen (nur Leseanforderungen): wenn HR/IR-Wert < 0, MSW am PCD-Register auf 0xFFFF einstellen

Beispiel: HR = 0x8001 pro PCD-Register = 0xFFFF 8000

- 16 Bits ohne Vorzeichen pro MSW des PCD-Registers mit Einstellung 0x0000

Beispiel: HR = 0x8001 pro PCD-Register = 0x0000 8001

- 32 Bits binär pro 2 HR / IR = 1 PCD-Register, keine Verarbeitung
- 32 Bits Auslagerung pro 2 HR / IR wortweise ausgelagert = 1 PCD-Register

Beispiel: HR0 = 0x1234, HR1 = 0x5678

PCD-Register = 0x1234 5678 (32 Bits binär)

PCD-Register = 0x5678 1234 (32 Bits Auslagerung)

- 32 Bits IEEE2FFP - 32 Bits FFP2IEEE

Konvertierung von FFP in IEEE wurde vor dem Versenden von Daten über den Bus durchgeführt (wird nur verwendet, wenn der Benutzer FFP-Werte hat, andernfalls 32 Bits verwenden),

Konvertierung von IEEE in FFP wurde vor dem Schreiben der Daten in die PCD-Medien durchgeführt (wird nur verwendet, wenn der Benutzer FFP-Werte benötigt, andernfalls 32 Bits verwenden),

- Auslagerung + 32 Bits IEEE2FFP / 32 Bits FFP2IEEE : es erfolgt eine wortweise Auslagerung direkt vor dem Senden der Daten / nach Empfang der Daten

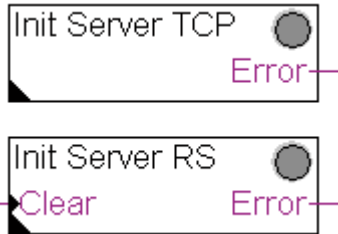
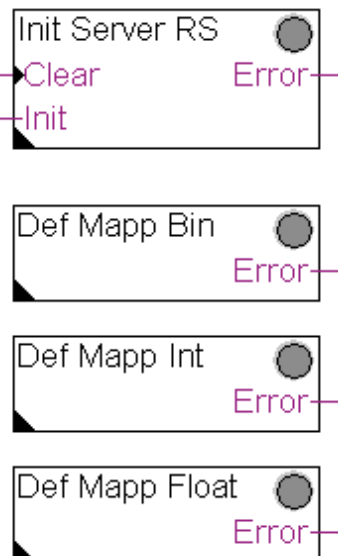
5 Modbus-FBox-Bibliothek

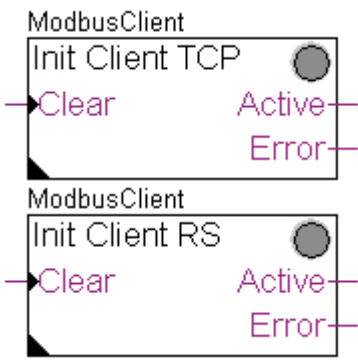
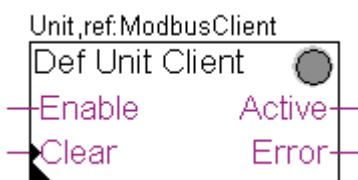
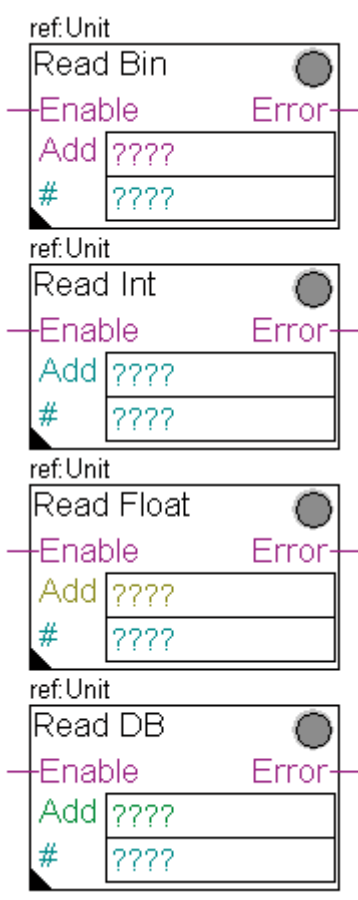
Mit der Modbus-FBox-Bibliothek können Sie das Modbus-Protokoll auf einem PCD nach bestimmten Funktionsblöcken verwenden.

Die Bibliothek verfügt über folgende Funktionen:

- Server-Erstellung
- Zuordnungen festlegen (Server)
- UUIDs festlegen (Server)
- Kanäle festlegen (Client)
- Lese- und Schreibanforderungen von einem Client senden

5

 The image shows two function blocks. The first is 'Init Server TCP' with an 'Error' output. The second is 'Init Server RS' with 'Clear' and 'Error' outputs.	Initialisiert die Serverinstanz: Anschlussnummer (+ Anschlussparameter für serielle Leitung) Modus: TCP, UDP, RTU, ASCII Es steht eine weitere Extended InitServerRS FBox zur Verfügung, um eine serielle Serverinstanz zerstören und den dazugehörigen seriellen Anschluss schließen zu können.
 The image shows the 'Def Unit Server' function block with 'Clear', 'Active', and 'Error' outputs.	UID-Definition mit bestimmten Parametern: Verschiebung, Löcher, Zuordnung, Auslagerung
 The image shows four function blocks. The first is 'Init Server RS' with 'Clear', 'Init', and 'Error' outputs. The second is 'Def Mapp Bin' with an 'Error' output. The third is 'Def Mapp Int' with an 'Error' output. The fourth is 'Def Mapp Float' with an 'Error' output.	Legt die Zuordnungsbereiche fest: Von der Zuordnung betroffener UID Zugeordnete Modbus-Medien und Adressen PCD-Medien und Adressen zu den Modbus-Daten Feldtyp

 ModbusClient Init Client TCP Clear Active Error ModbusClient Init Client RS Clear Active Error	Initialisiert die Kanalinstanz: Kanalnummer Anschlussnummer (+ Anschlussparameter für serielle Leitung) Modus: TCP, UDP, RTU, ASCII Schließzeit überschritten Reaktionszeit überschritten Wiederholungen Es steht eine weitere Extended InitClientRS FBox zur Verfügung, um einen seriellen Kanal zerstören und den dazugehörigen seriellen Anschluss schließen zu können.
 Unit,ref:ModbusClient Def Unit Client Enable Clear Active Error	Server festlegen, mit dem der Client kommuniziert: IP-Adresse für TCP / UDP UID
 ref:Unit Read Bin Enable Error Add # ref:Unit Read Int Enable Error Add # ref:Unit Read Float Enable Error Add # ref:Unit Read DB Enable Error Add #	Sendet eine Leseanforderung (Spulen lesen, Abstrakte Eingaben, Haltereister oder Eingaberegister). Empfangene Daten verarbeiten und in angegebenen PCD-Medien speichern. Es gibt auch zusätzliche FBoxes für eine indirekte Adressierung der PCD-Medien.

ref:Unit Write Bin ☐ Enable ☐ Error ☐ Add ???? # ???? ref:Unit Write Int ☐ Enable ☐ Error ☐ Add ???? # ???? ref:Unit Write Float ☐ Enable ☐ Error ☐ Add ???? # ???? ref:Unit Write DB ☐ Enable ☐ Error ☐ Add ???? # ????	Die angegebenen PCD-Medien lesen, Daten verarbeiten und als Schreibanforderung senden (Halteregister oder Spulen schreiben). Es gibt auch zusätzliche FBoxes für eine indirekte Adressierung der PCD-Medien.
--	---

Einzelheiten finden Sie in der Online-Hilfe der FBoxes.

6 Modbus-Bibliothek der Systemfunktionen

PCD-Komponenten sind wartungsfrei, mit Ausnahme einiger CPUs, bei denen die Batterie von Zeit zu Zeit ausgetauscht werden muss.

PCD-Komponenten enthalten keine Teile, die vom Benutzer ausgelagert werden können. Falls Hardware-Probleme auftreten, senden Sie die Komponenten bitte an Saia-Burgess zurück.

6.1 Einführung

Mit dem CSF-Befehl können Sie auf die Systemfunktionen eines Anwendungsprogramms zugreifen. Der CSF-Aufruf hat folgende allgemeine Form:

```
CSF  [cc] lib_number    ;Bibliotheksnummer 0-4095/4096-8191
      func_number      ;aufzurufende Funktionsnummer
      [parms...]      ;optionale Parameter
```

“lib_number” und “function_number” werden benötigt, um CSF-Befehle auszuführen. Wenn Sie die Datei “Modbus.inc” in PG5 einbeziehen, können Sie Stichwörter verwenden.

```
$INCLUDE "Modbus.inc"
COB      0
          0...
```

Der Zugriff auf Modbus-Bibliotheksaufrufe erfolgt über das Bibliotheksstichwort:

```
S.Modbus.Library
```

Die verfügbaren Funktionen werden in den nachfolgenden Kapiteln erklärt.

6.2 Parameter

Name	Verwendung	Beschreibung
AccessType	InitMap	Legt die Art und Weise fest, in der auf den Zuordnungsbereich zugegriffen werden kann (Lesen/Schreiben).
AreaType	InitMap	Legt die Standardverarbeitung fest, die für die Daten dieses Bereichs vor dem Zugriff auf die PCD-Medien (Schreiben) oder nach dem Zugriff auf die PCD-Medien (Lesen) ausgeführt werden muss.
ChannelID	OpenChannel SendReadRequest SendWriteRequest	Eine Client-Station kann einen oder mehrere logische Kanäle implementieren. Die ChannelID wird zum Erstellen oder Ändern eines Kanals verwendet oder um eine Anforderung an einen entfernten Partner zu senden. Zugelassener Bereich: 1-10
CloseTimeout	OpenChannel	Zeitüberschreitung zum Schließen einer Verbindung nach einer Leerlaufzeit (in Sekunden) - Wird nur im TCP/UDP verwendet.

Count	SendReadRequest SendWriteRequest	Anzahl, der zu lesenden/schreibenden Elemente (KEINE Modbus-Elemente: falls die Anforderung das Lesen eines 32-Bit-Registers auf einem Saia PCD betrifft, ist Count = 1) Max. Anzahl: - Spulen lesen / DI: 2000 - Spulen schreiben: 1968 - HR / IR lesen: 125 - HR schreiben: 123 bei einer 16-Bit-Anforderung (bzw. 62 und 61 bei einem 32-Bit-Zugriff)
Diag	OpenChannel InitUID	Dieser Parameter legt den Ort der Diagnose-Flags und -register fest. Format: "DIAG:Fxxx,Ryyy" Fxxx = Basisadresse 8 aufeinander folgender Flags Ryyy = Adresse des Diagnoseregisters Wir empfehlen Ihnen, alle Diagnosemedien zu löschen, die beim Programmstart verwendet wurden (XOB16).
ExceptCode	SendReadRequest SendWriteRequest	Ort, an dem der Ausnahmecode gespeichert werden muss, der in der Partnerantwort zurückgekommen ist.
Mode	InitUID	Legt den Betriebsmodus fest: 0 = automatisch: die Anforderungen werden automatisch verarbeitet und beantwortet 1 = transparent: die Anforderungen werden von der Benutzeranwendung verarbeitet und beantwortet (noch nicht implementiert)
PartnerRange	InitMap	Anzahl der zum Zuordnungsbereich gehörenden Modbus-Elemente .
PartnerStart	SendReadRequest SendWriteRequest InitMap	Erste Modbus-Adresse des Zuordnungsbereichs (InitMap) oder erste Modbus-Adresse der zu lesenden/schreibenden Daten (SendRequest). Achten Sie auf die Parameterlöcher (UID), um auf das richtige Register zuzugreifen.
PartnerType	InitMap	Dieser Parameter legt den Modbus-Datentyp fest, auf den der Client zugreift.
PCDMedia	SendReadRequest SendWriteRequest	Leseanforderung: legt fest, an welchen Ort auf dem PCD die angeforderten Daten kopiert werden. Schreibanforderung: legt fest, von welchem Ort auf dem PCD die Daten genommen werden, die gesendet werden
PCDMediaType	InitMap	PCD-Medientyp, auf dem die Modbus-Daten des Typs "PartnerType" zugeordnet werden.
PCDRange	InitMap	Anzahl der PCD-Elemente im Zuordnungsbereich. Beim DB die Anzahl der Elemente dieses DBs im Bereich. Die Adresse des 1. Elements ist 0.
PCDStart	InitMap	PCD-Startadresse, die zum Partner-Start gehört. Beim DB die Nummer des DBs, auf den sich der Bereich bezieht.

Port	OpenChannel	TCP/UDP: Nummer des logischen Anschlusses, an den die Anforderung gesendet wird RTU/ASCII: Nummer des seriellen Anschlusses, der zum Senden verwendet wird
PortParam	InitServer InitSerialPort	TCP/UDP: Nummer des logischen Anschlusses, den der Server verwendet RTU/ASCII: Text mit den Parametern des seriellen Anschlusses. Format: «PORT:portnumber; UART:baudrate,data,parity,stopbits; LINE:linetype» - PortNumber ist die Nummer des seriellen Anschlusses - Baudrate ist die am seriellen Anschluss verwendete Baud-Rate - Data, Parity, Stopbits: Es müssen 8 Daten sein, um der Modbus-Norm in RTU zu entsprechen und 7 in ASCII. Paritäts- und Stoppbits müssen außerdem so eingestellt sein, dass die Gesamtanzahl an Bits in einem Zeichen in RTU 11 und in ASCII 10 ist (d.h. N,2 oder E,1 oder O,1) - LineType kann folgendes sein: - SL0: RS 232 - SL1: RS 422 - SL2: RS 232, kein Handshake - SL3: RS 422, kein Handshake - SL4: RS 485, auto RTS Beim InitServer: dieser Text muss um "DIAG:Fxxx,Ryyy" erweitert werden, um anzuzeigen, wo die Diagnosen der Serverinstanz gespeichert werden müssen. Wir empfehlen Ihnen, die Diagnose-Flags und -register beim Programmstart zu löschen. Zum Schließen eines Anschlusses: Format: «PORT:portnumber;OFF»

Process	InitUID	Legt die Verarbeitung fest, die an den Daten durchgeführt wird. Format: "OFFSET=0, SWAP=1, DEFMAP=1, HOLES=0" OFFSET =1 für die Kommunikation mit einem Modicon-Client (erstes Element an Adresse 1); OFFSET =0 für andere Geräte mit erstem Register an Adresse 0 (Saia®PCD, ...). SWAP=1 falls eine wortweise Auslagerung an 32-Bit-Daten durchgeführt werden muss (cf. Abbildung 11) DEFMAP: legt fest, in welcher Zuordnungstabelle die Modbus-Adressen gesucht werden. DEFMAP=0: es darf nur eine benutzerdefinierte Zuordnung verwendet werden. DEFMAP=1: erst in der benutzerdefinierten Zuordnung suchen. Ohne Ergebnis, Suche in der Standardzuordnung fortsetzen (siehe DefaultMapping). HOLES: legt fest, wie die PCD-Register den HR/IR zugeordnet werden. HOLES=0: für 32-Bit-Zugriff, HR0-Anforderung aktualisiert HR0 und HR1 mit dem Inhalt von Reg0, HR2-Anforderung aktualisiert HR2 und HR3 mit dem Inhalt von Reg1. HOLES=1: für 32-Bit-Zugriff, HR0-Anforderung aktualisiert HR0 und HR1 mit dem Inhalt von Reg0, HR2-Anforderung aktualisiert HR2 und HR3 mit dem Inhalt von Reg2. (cf. Abbildung 11)
Processing	SendReadRequest SendWriteRequest	Leseanforderung: legt die Verarbeitung der empfangenen Daten fest, bevor diese in die PCD-Medien geschrieben werden Schreibanforderung: legt die Verarbeitung fest, die an den Daten durchgeführt, bevor diese in die Anforderung kopiert werden
Protocol	InitServer OpenChannel	Protokoll für die Modbus-Kommunikation.
RemoteIPAddr	SendReadRequest SendWriteRequest	IP-Adresse des Servers, an den die Anforderung gesendet wird (TCP/UDP)
ReqType	SendReadRequest SendWriteRequest	Modbus-Funktionscode. Dieser Parameter legt den Anforderungstyp fest (Lesen/Schreiben, Spulen/Halteregister, ...).
RespTimeout	OpenChannel	Zeit, innerhalb der eine Antwort eingehen soll (in Millisekunden).
Retries	OpenChannel	Anzahl der Wiederholungen im Fall einer Überschreitung der Antwortzeit (1. Versuch eingeschlossen).

Status	InitMap InitServer InitUID OpenChannel SendReadRequest SendWriteRequest	Status nach Ausführung des CSFs. Wenn der Wert kleiner als 0 ist, ist während des Aufrufs ein Fehler aufgerufen. Siehe Fehlercodeliste (§7.3) Wenn die Ausführung der Funktion erfolgreich war, ist der Status ≥ 0 .
UID	SendReadRequest SendWriteRequest InitUID InitMap	Der Einheitsbezeichner (UID) wird in die Anforderung eingefügt. Wenn eine Anforderung empfangen wird, wird der UID verwendet, um herauszufinden, welcher Modus verwendet werden muss und womit die Daten verarbeitet werden. Auf einer Server-Station können mehrere UIDs festgelegt werden. Alle Serverinstanzen auf der Station beziehen sich auf die UIDs. Wir empfehlen Ihnen, im Serienmodus nur eine UID auf einer Station zu verwenden. Wenn im Serienmodus eine Anforderung mit UID 0 (gesendete UID) empfangen wird, reagieren alle festgelegten UIDs mit Ihrer eigenen Verarbeitung und Zuordnung auf diese Anforderung. Im TCP/UDP beziehen sich UID 0 und UID 255 auf den gleichen UID. Um auf einem TCP/UDP-Server auf diesen UID zugreifen zu können, muss der Benutzer seine Parameter konfigurieren. Dies kann mit Hilfe von InitUID CSF erfolgen. Vermeiden Sie den Zugriff auf einen TCP/UDP-Server mit UID 0.

6.3 Fehlercodes

Im Fall einer erfolgreichen Verarbeitung geben alle CSFs entweder eine 0 (Null) oder einen positiven Wert aus. Ein negativer Wert zeigt einen Fehler an. Nachfolgend finden Sie eine Liste der Fehlercodes.

6.3.1 CSF-Fehlercodes

Die meisten dieser Fehler können auftreten, wenn ein oder mehrere CSF-Parameter falsch sind.

Code	Bezeichnung	Beschreibung
0	NO_ERROR	Kein Fehler
-4200	INVALID_MEDIATYPE	Ungültiger PCD-Medientyp (Sendeanforderung)
-4199	DIAGTEXT_ERROR	Diagnosetext nicht lesbar (Kanal / UID) (intern)
-4198	INVALID_DIAG_SYNTAX	Syntaxfehler im Diagnosetext (Kanal / UID)
-4197	INVALID_PROCESSING_PARAM	Mindestens 1 Parameter im Verarbeitungstext ist ungültig (UID)
-4196	INVALID_PROCESSING_SYNTAX	Syntaxfehler im Verarbeitungstext (UID)
-4195	PROCESSINGTEXT_ERROR	Verarbeitungstext nicht lesbar (UID) (intern)
-4194	CHANNEL_NOT_DEFINED	Kanal ist nicht definiert (Sendeanforderung)
-4193	CHANNEL_BUSY	Kanal wird bereits verwendet (Sendeanforderung)
-4192	SERVER_NOT_CREATED	Server kann nicht erstellt werden (Init Server): maximale Server-Anzahl ist bereits erreicht oder der Server wurde bereits definiert.
-4191	SERVER_NOT_STARTED	Server kann nicht starten (Init Server)
-4190	INVALID_FUNCTION_CODE	Funktionscode stimmt nicht mit dem Anforderungstyp überein (Lesen / Schreiben)

-4189	INVALID_PROCESSING	Verarbeitung stimmt nicht mit dem Funktionscode oder Anforderungstyp überein (Lesen / Schreiben)
-4188	INVALID_PROTOCOL	Protokoll ist ungültig (Server / Kanal)
-4187	INVALID_PARTNER_TYPE	Ungültiger Modbus-Medientyp (Init Map)
-4186	INVALID_PCDMEDIA_TYPE	Ungültiger PCD-Medientyp (Init Map)
-4185	INVALID_AREA_TYPE	Ungültiger Bereichstyp (Init Map)
-4184	INVALID_PCD_MEDIA_ADDR	Ungültige PCD-Medienadresse (Init Map)
-4183	INVALID_RANGE	Ungültiger Medienbereich (Init Map): Partnerbereich < PCD-Bereich oder Bereich = 0.
-4182	INVALID_AREA_ACCESS_TYPE	Ungültiger Zugriffstyp (Init Map): existiert nicht oder stimmt nicht mit den Modbus-Medien überein.
-4181	INVALID_NB_OF_MEDIA	Anzahl der Medien stimmt nicht mit dem Funktionscode überein (Sendeanforderung): > 1 für einzelne Spule oder einzelnes Register schreiben.
-4180	INVALID_UID	UID außerhalb des Bereichs 0-0xFF (Init UID, Sendeanforderung) oder UID = 0 für eine Leseanforderung im Serienmodus
-4179	INVALID_CHANNEL	Kanal außerhalb des Bereichs 1-10 (Offener Kanal, Sendeanforderung)
-4178	INCOMPATIBLE_PCDMEDIA	PCD-Medien sind nicht mit den Modbus-Medien (Init Map) oder dem Funktionscode (Sendeanforderung) kompatibel
-4177	INVALID_PORT_CONF	Anschlusskonfiguration (Text oder Nummer) stimmt nicht mit dem Protokoll überein (Init Server)
-4176	PORTCONFIGTEXT_ERROR	Kann den Text der Anschlusskonfiguration nicht im Serienmodus lesen (Init Server, Init SR Port) (intern)
-4175	PORTCONFIGPARAM_ERROR	Fehler im Text der Anschlusskonfiguration im Serienmodus (Init Server, Init SR Port)
-4174	SRPORT_START_ERROR	SR-Anschluss kann nicht offen sein (Init Server, Init SR Port). Im Serienmodus könnte der Anschluss bereits verwendet werden oder die Anschlussparameter sind ungültig.
-4173	INVALID_PORT_NUMBER	Ungültige SR-Anschlussnummer (Offener Kanal)
-4172	INVALID_RETRY_NB	Ungültige Anzahl an Wiederholungen (muss < 16 sein)
-4171	NO_PORT_TO_CLOSE	Der zu schließende serielle Anschluss wurde nicht geöffnet.

6.3.2 ModbusShell-Fehlercodes

Diese Fehler treten bei der CSF-Verarbeitung auf.

Code	Bezeichnung	Beschreibung
-4100	UID_TABLE_FULL	Maximale Anzahl an UIDs wurde bereits erreicht (Init UID)
-4099	UID_NOT_DEFINED	UID ist nicht definiert (Init Map)
-4098	UID_BUSY	UID wird verwendet (Init UID), lässt sich nicht ändern
-4097	CHANNEL_TABLE_FULL	Kanaltabelle ist voll, Kanal lässt sich nicht hinzufügen
-4096	CHANNEL_ALREADY_DEFINED	Es ist bereits ein Kanal mit dem gleichen Anschluss/Protokoll-Paar festgelegt (Offener Kanal)
-4095	CHANNEL_BUSY	Dieser Kanal wird verwendet und lässt sich nicht ändern
-4094	MAP_TABLE_FULL	Für den UID wurde die maximal Anzahl an Zuordnungen erreicht (Init Map)
-4093	MAP_AREA_OVERLAP	Mindestens zwei Zuordnungsbereiche überschneiden sich (Init Map)
-4092	CONNECTION_TABLE_FULL	Maximale Anzahl an "Verbindungen" ist erreicht (Sendeanforderung) TCP/UDP: Verbindung = {Anschluss-Protokoll-IP-Adresse} Seriell: Verbindung = {Anschluss-Protokoll}
-4091	CONNECTION_NOT_READY	Verbindung ist nicht versandbereit (Sendeanforderung) IP-Adresse ist nicht eingestellt, ...

6.3.3 Fehlercodes des ModbusDrivers

Die meisten Fehler zwischen -3980 und -4000 sind interne Fehler. Wenden Sie sich an Saia--Burgess, falls ein Fehler auftritt, der unten nicht aufgeführt ist.

Code	Bezeichnung	Beschreibung
-3990	CLII_DRIVER_NOT_READY	Treiber ist nicht bereit (Sende-anforderung)
-3987	QUANTITY_ERROR	Fehler in der Menge zu lesender / schreibender Medien (Sende-anforderung): überschreitet die Grenze oder < 0
-3983	CLIENT_NO_RESSOURCE	Keine Ressource verfügbar (Sende-anforderung)
-3982	PORT_ID_ERROR	Anschluss ist bereits initialisiert oder wird bereits verwendet
-3981	INVALID_PORT_STATE	Anschluss nicht bereit (seriell) (Sende-anforderung)
-3980	UNKNOWN_ERROR	Unbekannter Fehler

6.4 Funktionscodes

6

READ_COILS	0x01	
READ_DISC_INPUT	0x02	
READ_HOLD_REG	0x03	
READ_INPUT_REG	0x04	
WRITE_SINGLE_COIL	0x05	
WRITE_SINGLE_REG	0x06	Nur 16-Bit-Zugriff zulässig
WRITE_MULTIPLE_COILS	0x0F	

6.5 Protokolle

MODBUS_TCP	0
MODBUS_UDP	1
MODBUS_RTU	2
MODBUS_ASCII	3

6.6 Modbus-Datentypen

MB_NO_MEDIA	0
MB_COILS_MEDIA	1
MB_DISCR_INPUT_MEDIA	2
MB_HOLDING_REG_MEDIA	3
MB_INPUT_REG_MEDIA	4

6.7 Modbus-Bereichstypen (Server)

COILS	0	
16BITS_SIGNED	1	
16BITS_UNSIGNED	2	
32BITS	3	
32BITS_IEEE	4	Wird während des Client-Betriebs im IEEE-Modus verwendet

6.8 Modbus-Bereichszugriffstypen (Server)

READ_WRITE	0
READ_ONLY	1
WRITE_ONLY	2
NO_ACCESS	3

6.9 Verarbeitungstypen (Client)

COILS_DI_PROCESSING	0	(nur für Spulen oder Zugriff über abstrakte Eingaben)
R_16BITS_SIGNED	1	Leseanforderungen
RW_16BITS_UNSIGNED	2	
RW_32BITS_BINARY	3	
RW_32BITS_SWAP	4	
R_32BITS_IEEE2FFP	5	Leseanforderungen
R_32BITS_SWAP_IEEE2FFP	6	Leseanforderungen
W_32BITS_FFP2IEEE	7	Schreibanforderungen
W_32BITS_SWAP_FFP2IEEE	8	Schreibanforderungen

6.10 PCD-Medientypen

PCD_NO_MEDIA	0
PCD_IO_MEDIA	1
PCD_I_MEDIA	2
PCD_O_MEDIA	3
PCD_FLAG_MEDIA	4
PCD_TC_MEDIA	5
PCD_REG_MEDIA	6
PCD_DB_MEDIA	9

6.11 Ausnahmecodes

ILLEGAL_FUNCTION_CODE	1	Funktionscode ungültig oder entspricht nicht dem Modbus-Medientyp
ILLEGAL_DATA_ADDRESS	2	Kein Zugriff auf die Modbus-Medien an der Empfangsadresse möglich (UID-Zuordnung überprüfen / siehe Diagnose)
ILLEGAL_DATA_VALUE	3	Ungültige Anzahl an Daten in der Anforderung
SERVER_FAILURE	4	Kein Zugriff auf die PCD-Medien möglich (siehe Diagnose) oder UID ist nicht konfiguriert (insbesondere UID 255 in TCP/UDP)
ACKNOWLEDGE	5	
SERVER_BUSY	6	Server ist beim Zugriffsversuch bereits ausgelastet
GATEWAY_PATH_NOT_FOUND	10	UID nicht definiert.
GATEWAY_TARGET_PROBLEM	11	

7 Modbus-CSF-Spezifikation

7.1 S.Modbus.InitSerialPort (Client / Server)

S.Modbus.InitSerialPort		Diese Funktion öffnet einen seriellen Anschluss auf der Client-Seite. Auf der Server-Seite ist dies nicht nötig, da die Initialisierung des Anschlusses bei der Erstellung des Servers erfolgt. Diese Funktion kann für Client und Server auch zum Schließen eines seriellen Anschlusses sowie aller dazugehörigen Instanzen verwendet werden (Kanäle und Server). Zum Senden / Empfangen am geschlossenen seriellen Anschluss müssen die entsprechenden Initialisierungsfunktionen aufgerufen werden (InitServer, InitChannel, InitSerialPort) Eine mehrfache Verwendung des gleichen seriellen Anschlusses ist nicht zulässig.		
Parameter		8		
1	PortParam	X	IN	Parameter des zu öffnenden seriellen Anschlusses
2	Status	R	OUT	Antwortwert (siehe unten)
Antwortwert				
	Erfolg	= 0	Kanal erfolgreich erstellt oder geändert	
	Fehler	< 0	Siehe §7.3.	

Beispiel	Initialisierung eines seriellen Anschlusses			
CSF	<pre> S.Modbus.Library S.Modbus.InitSerialPort ConfigPort2 ; Text: "PORT:2; ART:9600,8,N,2;LINE:SL4" R_StatusOpenPort ; R 10101 </pre>			
Beispiel	Schließen eines seriellen Anschlusses			
CSF	<pre> S.Modbus.Library S.Modbus.InitSerialPort ConfigPort2 ; Text: "PORT:2; OFF" R_StatusOpenPort ; R 10101 </pre>			

7.2 S.Modbus.OpenChannel (Client)

S.Modbus.OpenChannel		Diese Funktion erstellt einen neuen Kanal oder ändert einen vorhandenen Kanal (wenn dieser gerade nicht ausgelastet ist). Ein Kanal wird zum Senden einer Anforderung an einen Partner oder zum Verarbeiten empfangener Daten verwendet. Ein Kanal kann über eine beliebige vorhandene Schnittstelle verwendet werden (Seriell / Ethernet). Es ist nicht zulässig, zwei Kanäle mit dem gleichen Anschluss/Protokoll-Paar zu öffnen. Wenn Sie versuchen, einen Kanal mit einer bereits festgelegten Kanal-ID zu öffnen, wird der "alte" Kanal überschrieben (wenn dieser gerade nicht verwendet wird, siehe TBSY). Es ist nicht zulässig, 2 Modbus-Clients im gleichen seriellen Netzwerk zu verwenden. Wenn 2 Clients gleichzeitig über den Bus kommunizieren, kann nicht garantiert werden, dass sich das Netzwerk richtig verhält.		
Parameter		8		
1	ChannelID	R	IN	Kanal-ID des zu erstellenden oder ändernden Kanals. Zugelassener Bereich: 1-10
		K		

S.Modbus.SendReadRequest / S.Modbus.SendWriteRequest

2	Port	R K	IN	TCP/UDP: Fernanschluss, an den die Anforderung gesendet wird RTU/ASCII: Nummer des seriellen Anschlusses, der zum Senden verwendet wird
3	Protocol	R K	IN	Protokoll für die Kommunikation auf diesem Kanal
4	Diag	X	IN	Diagnosedefinition. "DIAG:Fxxxx,Ryyyy"
5	CloseTimeout	R K	IN	Zeitüberschreitung, ab der die Verbindung nach einer Leerlaufzeit geschlossen wird
6	RespTimeout	R K	IN	Zeit, innerhalb der eine Antwort eingehen soll. Dieser Wert wird auch als "Sendezeitüberschreitung" verwendet.
7	Retries	R K	IN	Anzahl zulässiger Wiederholungen im Fall einer Antwortzeitüberschreitung auf diesem Kanal
8	Status	R	OUT	Antwortwert (siehe unten)
Antwortwert				
	Erfolg	= 0	Kanal erfolgreich erstellt oder geändert	
	Fehler	< 0		

Beispiel		
CSF	<pre> S.Modbus.Library S.Modbus.OpenChannel 2 ; Channel ID 502 ; Remote Port S.Modbus.Protocol.TCP ; Protocol Diagnostic ; «DIAG:F1000,R1000» 2000 ; CloseTimeout 1000 ; ResponseTimeout 0 ; Number of retries R_StatusOpenCh ; R 10105 </pre>	

7.3 S.Modbus.SendReadRequest / S.Modbus.SendWriteRequest (Client)

S.Modbus.SendReadRequest		Diese Funktion sendet eine Modbus-Leseanforderung über einen festgelegten Kanal.		
S.Modbus.SendWriteRequest		Diese Funktion sendet eine Modbus-Schreibanforderung über einen festgelegten Kanal.		
		Für Sendeanforderungen muss das TBSY-Flag des Kanals niedrig sein, um die Anforderung senden zu können. Andernfalls gibt das CSF einen Fehler aus.		
Parameter		12		
1	ChannelID	R K	IN	Kanal-ID (einer der bereits offenen Kanäle): zeigt an, auf welchem Kanal die Anforderung gesendet wird
2	RemoteIPAddr	R K	IN	IP-Adresse des Servers, an den die Anforderung gesendet wird (wird nur beim TCP/UDP verwendet)
3	UID	R K	IN	Einheitsbezeichner: wird verwendet, um festzulegen, welche Verarbeitung und welchen Modus der Server verwenden soll

4	ReqType	R K	IN	Funktionscode
5	ExceptCode	R	OUT	Ort, an den der Ausnahmecode der Partnerantwort kopiert wird
6	PCDMedia	R TC IO F X DB	IN	Leseanforderung: legt fest, auf welche PCD-Medien und an welchen Ort die angeforderten Daten kopiert werden Schreibanforderung: legt fest, von welchen PCD-Medien und von welchem Ort die Anforderungen genommen werden
7	PartnerStart	R K	IN	Legt fest, von welchen Server-Medien die Daten gelesen und auf welche Server-Medien die Daten geschrieben werden (abhängig von der Server-Zuordnung)
8	Count	R K	IN	Anzahl der beim Partner zu lesenden bzw. zu schreibenden Elemente (KEINE Modbus-Elemente: falls 32-Bit-Register angefordert werden, Anzahl der 32-Bit-Register)
9	Processing	R K	IN	Leseanforderung: legt fest, wie die empfangenen Daten verarbeitet werden, bevor diese in die PCD-Medien geschrieben werden Schreibanforderung: legt fest, wie die Daten verarbeitet werden, bevor diese in die Anforderung kopiert werden
10	Status	R	OUT	Antwortwert (siehe unten)
Antwortwert				
	Erfolg	= 0	Anforderung erfolgreich gesendet	
	Fehler	< 0		

Beispiel		
CSF	<pre> S.Modbus.Library S.Modbus.SendReadRequest ; Function = Send Read Request 2 ; ChannelID 0AC170281H ; IP address = 172.23.2.129 1 ; Remote UnitID S.Modbus.FunctionCode.ReadCoils R 100 ; Location to store Exception Code F 0 ; PCD Local Media 10000 ; PartnerStartAddress 10 ; Number of Media S.Modbus.ClientProcessing.Coils_DI_Processing R 101 ; Location to store CSF Status </pre>	

7.4 S.Modbus.InitServer (Server)

S.Modbus.InitServer		Mit dieser Funktion können Sie eine Serverinstanz erstellen. Beim TCP/UDP ist es nicht zulässig, einen Server mehrfach mit dem gleichen Anschluss/Protokoll-Paar zu verwenden (nur seriell darf der gleiche Anschluss verwendet werden).		
Parameter		3		
1	Protocol	R K	IN	Von der Serverinstanz verwendetes Kommunikationsprotokoll
2	PortParam	R K X	IN	TCP/UDP: Nummer des logischen Kommunikationsanschlusses, den der Server verwendet RTU/ASCII: Text mit den Parametern des seriellen Anschlusses
3	Status	R	OUT	Antwortwert (siehe unten)
Antwortwert				
	Erfolg	= 0		Server erfolgreich gestartet
	Fehler	< 0		

Beispiel (TCP/UDP)			
CSF	<pre> S.Modbus.Library S.Modbus.InitServer S.Modbus.Protocol.UDP 502 ; Port R 102 ; Location to store CSF status </pre>		

Beispiel (SERIELL)			
CSF	<pre> S.Modbus.Library S.Modbus.InitServer S.Modbus.Protocol.RTU ConfigPort1 ; Port configuration: "PORT:1; UART:9600,8,N,2;LINE:SL4 ;DIAG:F800,R800" R 102 ; Location to store CSF status </pre>		

7.5 S.Modbus.InitUID (Server)

S.Modbus.InitUID		Diese Funktion erstellt oder ändert einen Einheitsbezeichner, auf den sich alle Serverinstanzen beziehen. Wenn eine Anforderung empfangen wird, wird der UID verwendet, um herauszufinden, welcher Modus verwendet werden muss und womit die Daten verarbeitet werden. Auf einer Server-Station können mehrere UIDs festgelegt werden. Wenn Sie versuchen, eine UID mit einer bereits festgelegten UID-Nummer zu erstellen, wird die "alte" UID überschrieben (wenn diese gerade nicht verwendet wird, siehe RBSY).		
Parameter		5		
1	UID	R K	IN	Einheitsbezeichner

2	Mode	R K	IN	Server-Modus: 0 = auto, 1 = transparent (noch nicht implementiert)
3	Diag	X	IN	Diagnosedefinition. Format: "DIAG:Fxxx,Ryyy"
4	Process	X	IN	Definition der Datenverarbeitung. Format: "OFFSET=0, SWAP=1, DEFMAP=1, HOLES=0"
5	Status	R	OUT	Antwortwert (siehe unten)
Antwortwert				
	Erfolg	= 0	UID erfolgreich hinzugefügt	
	Fehler	< 0		

Beispiel		
CSF	<pre> S.Modbus.Library S.Modbus.InitUID 1 ; UID S.Modbus.ServerMode.Auto DiagnosticUID ; "DIAG:F2000,R2000" Processing ; "OFFSET=0, SWAP=0, DEFMAP=1, HOLES=0" R 103 ; Location to store CSF status </pre>	

7

7.6 S.Modbus.InitMap (Server)

S.Modbus.InitMap		Mit dieser Funktion können Sie Zuordnungsbereiche festlegen, z. B. durch Zuordnung eines Satzes Modbus-Medien an einer bestimmten Adresse zu einem Satz PCD-Medien an einer bestimmten Adresse. Es ist nicht zulässig Bereiche zu erstellen, die sich mit anderen überschneiden (gleicher Modbus-Medientyp und überschneidende Modbus-Adressbereiche)		
Parameter		10		
1	PartnerType	R K	IN	Modbus-Medientyp, der PCD-Medien zugeordnet wird
2	PartnerStart	R K	IN	Partner-Startadresse des Zuordnungsbereichs
3	PartnerRange	R K	IN	Anzahl der Elemente des Zuordnungsbereichs (kann nicht 0 sein)
4	PCDMediaType	R K	IN	PCD-Medientyp, dem die Modbus-Medien zugeordnet werden
5	PCDStart	R K	IN	PCD-Medien-Startadresse, die zur Partner-Startadresse gehört. Bei einer Zuordnung zu einem DB, Nummer des DBs, dem die Daten zugeordnet werden.

6	PCDRange	R K	IN	Anzahl der PCD-Elemente im Zuordnungsbereich (sollte der Anzahl an Partnerelementen entsprechen, achten Sie auf den verwendeten AreaType). Bei einem DB, Nummer des Elements dieses zum Bereich gehörenden DBs. Das 1. Element des Bereichs ist Element 0 des DBs. PCDRange kann nicht 0 sein. Um Fehler beim Zugriff auf den Bereich zu vermeiden, denken Sie daran, dass in einem 32-Bit-Bereich 1 PCD-Register zu 2 Haltereigistern gehört (siehe Abbildung 12 unten).
7	AreaType	R K	IN	Zuordnungsbereichstyp
8	AccessType	R K	IN	Zugriffstyp des Zuordnungsbereichs
9	UID	R K	IN	Von diesem Zuordnungsbereich betroffener UID
10	Status	R	OUT	Antwortwert (siehe unten)
Antwortwert				
	Erfolg	= 0	Zuordnungsbereich erfolgreich hinzugefügt	
	Fehler	< 0		

Beispiel		
CSF	<pre> S.Modbus.Library S.Modbus.InitMap S.Modbus.ModbusType.Coils 1000 ; Partner Start Address 2000 ; Partner Range S.Modbus.PCDMedia.Flag 100 ; PCD Start Address 2000 ; PCD Range S.Modbus.AreaType.Coils S.Modbus.AreaAccess.ReadWrite 1 ; UID concerned by this mapping R 104 ; Location to store CSF status </pre>	

In diesem Beispiel sind die Spulen mit Modbus-Adressen im Bereich 2000 bis 3999 entsprechend den PCD-Flags 100 bis 2199 zugeordnet (für UID 1).

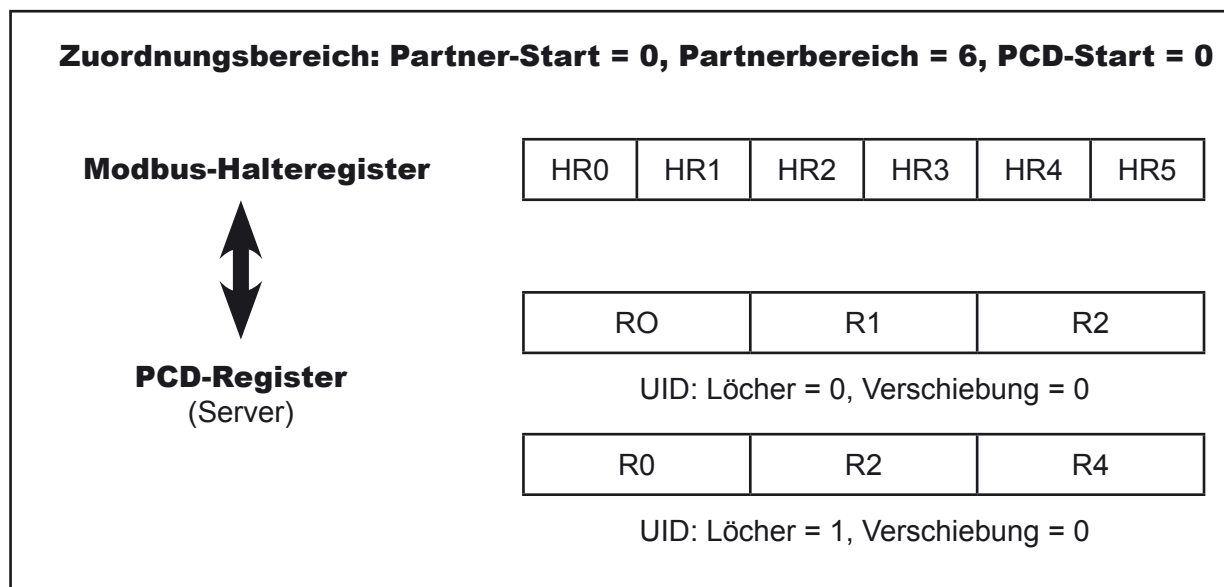


Abbildung 12: Zuordnung von Halteregistern (16 Bits) zu PCD-Registern (32 Bits) in einem 32-Bit-Bereich

In oben beschriebenem **32-Bit-Bereich** kann nicht auf Modbus-Element 1, 3 und 5 zugegriffen werden: Sie erhalten eine Fehlermeldung. Auch beim Zugriff auf HR2 mit 3 PCD-Registern (das entspricht 6 HR) erhalten Sie eine Fehlermeldung.

Um auf HR 0 mit 3 PCD-Registern (= 6 HR) zuzugreifen, muss der Zuordnungsbereich mit mindestens 6 HR festgelegt sein, beginnend mit Adresse 0 übereinstimmend mit:

- mindestens 3 PCD-Registern, beginnend bei Adresse 0 (1. Fall)
- mindestens 5 PCD-Registern, beginnend bei Adresse 0 (2. Fall)

A Anhang

A.1 Symbole

	In Betriebsanleitungen weist dieses Symbol den Leser auf weitere Informationen in dieser Anleitung oder in anderen Anleitungen oder technischen Dokumenten hin. Auf einen direkten Link zu solchen Dokumenten wird grundsätzlich verzichtet.
	Dieses Symbol warnt den Leser vor Komponenten, bei deren Berührung es zu einer elektrischen Entladung kommen kann. Empfehlung: Berühren Sie zumindest den Minuspol des Systems (Schaltschrank des PGU-Anschlusses), bevor Sie elektronische Teile berühren. Wir empfehlen jedoch einen Erdungshandtragsriemen, dessen Kabel am Minus des Systems angeschlossen ist.
	Anweisungen mit diesem Zeichen müssen immer befolgt werden.
	Die Erklärungen neben diesem Zeichen gelten nur für die Saia-Burgess PCD-Klassikserien.
	Die Erklärungen neben diesem Zeichen gelten nur für die Saia-Burgess PCD-xx7-Serien.

A.2 Liste der Abkürzungen

UID	Unit Identifier (Einheitsbezeichner)
HR	Holding Registers (Haltereister)
IR	Input Registers (Eingaberegister)
DI	Digital Inputs (Digitale Eingaben)
TCP	Transmission Control Protocol (Internetprotokoll)
UDP	User Datagram Protocol (Internetprotokoll)
FFP	Fast Floating Point (schnelles Gleitkomma)
IEEE	Institute of Electrical and Electronics Engineers (als Standardisierungs-Gremium fungierender Ingenieursverband in den USA)
LSW	Least Significant Word (Wort mit niedrigstem Stellenwert)
MSW	Most Significant Word (Wort mit höchstem Stellenwert)
DB	Data Block (Datenblock)
I/O	Inputs/Outputs (E/A Eingänge/Ausgänge)
SR	Seriell

A.3 Adresse von Saia-Burgess**Saia-Burgess Controls Ltd.**

Bahnhofstrasse 18
CH-3280 Murten / Schweiz

Telefon ++41 26 672 71 11

Telefax ++41 26 672 74 99

E-Mail: pcd@saia-burgess.com

Homepage: www.saia-pcd.com

Support: www.sbc-support.ch

**Adresse des Vertriebsbüros in der
Schweiz für Kundenretouren:****Saia-Burgess Controls AG**

Service Après-Vente
Bahnhofstrasse 18
CH-3280 Murten / Schweiz