

# VIPA SPEED7 Library

OPL\_SP7-LIB | SW90AS0MA V10.004 | Handbuch

HB00 | OPL\_SP7-LIB | SW90AS0MA V10.004 | de | 18-30

Baustein Bibliothek - Modbus Communication



VIPA GmbH  
Ohmstr. 4  
91074 Herzogenaurach  
Telefon: +49 9132 744-0  
Telefax: +49 9132 744-1864  
E-Mail: [info@vipa.com](mailto:info@vipa.com)  
Internet: [www.vipa.com](http://www.vipa.com)

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>4</b>
1.1	Copyright © VIPA GmbH	4
1.2	Über dieses Handbuch	5
<b>2</b>	<b>Wichtige Hinweise</b>	<b>6</b>
2.1	Allgemein	6
2.2	Intern verwendete Bausteine	6
<b>3</b>	<b>Bibliothek einbinden</b>	<b>7</b>
3.1	Einbinden in Siemens SIMATIC Manager	8
3.2	Einbinden in Siemens TIA Portal	9
<b>4</b>	<b>Modbus-Kommunikation - "Modbus Communication"</b>	<b>10</b>
4.1	TCP	10
4.1.1	FB 70 - TCP_MB_CLIENT - Modbus/TCP-Client	10
4.1.2	FB 71 - TCP_MB_SERVER - Modbus/TCP-Server	13
4.2	RTU	17
4.2.1	FB 72 - RTU_MB_MASTER - Modbus-RTU-Master	17
4.2.2	FB 73 - RTU_MB_SLAVE - Modbus-RTU-Slave	20
4.3	FKT Modbus-Funktionscodes - "FKT Codes"	24

# 1 Allgemeines

## 1.1 Copyright © VIPA GmbH

### All Rights Reserved

Dieses Dokument enthält geschützte Informationen von VIPA und darf außer in Übereinstimmung mit anwendbaren Vereinbarungen weder offengelegt noch benutzt werden.

Dieses Material ist durch Urheberrechtsgesetze geschützt. Ohne schriftliches Einverständnis von VIPA und dem Besitzer dieses Materials darf dieses Material weder reproduziert, verteilt, noch in keiner Form von keiner Einheit (sowohl VIPA-intern als auch -extern) geändert werden, es sei denn in Übereinstimmung mit anwendbaren Vereinbarungen, Verträgen oder Lizenzen.

Zur Genehmigung von Vervielfältigung oder Verteilung wenden Sie sich bitte an: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

E-Mail: [info@vipa.de](mailto:info@vipa.de)

<http://www.vipa.com>



*Es wurden alle Anstrengungen unternommen, um sicherzustellen, dass die in diesem Dokument enthaltenen Informationen zum Zeitpunkt der Veröffentlichung vollständig und richtig sind. Das Recht auf Änderungen der Informationen bleibt jedoch vorbehalten.*

*Die vorliegende Kundendokumentation beschreibt alle heute bekannten Hardware-Einheiten und Funktionen. Es ist möglich, dass Einheiten beschrieben sind, die beim Kunden nicht vorhanden sind. Der genaue Lieferumfang ist im jeweiligen Kaufvertrag beschrieben.*

### EG-Konformitätserklärung

Hiermit erklärt VIPA GmbH, dass die Produkte und Systeme mit den grundlegenden Anforderungen und den anderen relevanten Vorschriften übereinstimmen. Die Übereinstimmung ist durch CE-Zeichen gekennzeichnet.

### Informationen zur Konformitätserklärung

Für weitere Informationen zur CE-Kennzeichnung und Konformitätserklärung wenden Sie sich bitte an Ihre Landesvertretung der VIPA GmbH.

### Warenzeichen

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S und Commander Compact sind eingetragene Warenzeichen der VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 ist ein eingetragenes Warenzeichen der profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300, S7-400 und S7-1500 sind eingetragene Warenzeichen der Siemens AG.

Microsoft und Windows sind eingetragene Warenzeichen von Microsoft Inc., USA.

Portable Document Format (PDF) und Postscript sind eingetragene Warenzeichen von Adobe Systems, Inc.

Alle anderen erwähnten Firmennamen und Logos sowie Marken- oder Produktnamen sind Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer.

**Dokument-Support**

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Fehler anzeigen oder inhaltliche Fragen zu diesem Dokument stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744-1204

E-Mail: [documentation@vipa.de](mailto:documentation@vipa.de)

**Technischer Support**

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Probleme mit dem Produkt haben oder Fragen zum Produkt stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefon: +49 9132 744-1150 (Hotline)

E-Mail: [support@vipa.de](mailto:support@vipa.de)

## 1.2 Über dieses Handbuch

**Zielsetzung und Inhalt**

Das Handbuch beschreibt die Baustein-Bibliothek *"Modbus Communication"* von VIPA:

- Beschrieben wird Aufbau, Projektierung und Anwendung in verschiedenen Programmiersystemen.
- Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.
- Das Handbuch ist in elektronischer Form als PDF-Datei verfügbar. Hierzu ist der Adobe Acrobat Reader erforderlich.
- Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.
- Als Orientierungshilfe stehen im Handbuch zur Verfügung:
  - Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs
  - Verweise mit Seitenangabe

**Piktogramme Signalwörter**

Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

**GEFAHR!**

Unmittelbar drohende oder mögliche Gefahr. Personenschäden sind möglich.

**VORSICHT!**

Bei Nichtbefolgen sind Sachschäden möglich.



*Zusätzliche Informationen und nützliche Tipps.*

## 2 Wichtige Hinweise

### 2.1 Allgemein



*Nachfolgend finden Sie wichtige Hinweise, die grundsätzlich beim Einsatz der Bausteine zu beachten sind.*

### 2.2 Intern verwendete Bausteine



#### **VORSICHT!**

Folgende Bausteine werden intern verwendet und dürfen nicht überschrieben werden! Der direkte Aufruf eines internen Bausteins führt zu Fehler im entsprechenden Instanz-DB! Bitte verwenden Sie für den Aufruf immer die zugehörige Funktion.

FC/SFC	Bezeichnung	Beschreibung
FC/SFC 192	CP_S_R	wird intern für FB 7 und FB 8 verwendet
FC/SFC 196	AG_CNTRL	wird intern für FC 10 verwendet
FC/SFC 200	AG_GET	wird intern für FB/SFB 14 verwendet
FC/SFC 201	AG_PUT	wird intern für FB/SFB 15 verwendet
FC/SFC 202	AG_BSEND	wird intern für FB/SFB 12 verwendet
FC/SFC 203	AG_BRCV	wird intern für FB/SFB 13 verwendet
FC/SFC 204	IP_CONF	wird intern für FB 55 IP_CONF verwendet
FC/SFC 205	AG_SEND	wird intern für FC 5 AG_SEND verwendet
FC/SFC 206	AG_RECV	wird intern für FC 6 AG_RECV verwendet
FC/SFC 253	IBS_ACCESS	wird intern für SPEED-Bus-INTERBUS-Master verwendet
SFB 238	EC_RWOD	wird intern für EtherCAT-Kommunikation verwendet
SFB 239	FUNC	wird intern für FB 240, FB 241 verwendet

### 3 Bibliothek einbinden

#### Baustein-Bibliothek "Modbus Communication"

Die Baustein-Bibliothek finden Sie im "Service/Support"-Bereich auf [www.vipa.com](http://www.vipa.com) unter "Downloads → VIPA Lib" als "Baustein-Bibliothek Modbus Communication - SW90AS0MA" zum Download. Die Bibliothek liegt als gepackte zip-Dateien vor. Sobald Sie die Bausteine verwenden möchten, müssen Sie diese in Ihr Projekt importieren.



*Bitte verwenden Sie immer das zu Ihrer Bibliothek zugehörige Handbuch. Solange es keine beschreibungsrelevante Änderungen gibt, können im Handbuch die Versionsangaben der Bibliothek und der zugehörigen Dateien von denen der Bibliothek abweichen.*




#### Folgende Bausteinbibliotheken stehen zur Verfügung

Datei	Beschreibung
Modbus_S7_V0004.zip	<ul style="list-style-type: none"><li>■ Bausteinbibliothek für Siemens SIMATIC Manager.</li><li>■ Für den Einsatz in CPUs von VIPA bzw. S7-300 CPUs von Siemens.</li></ul>
Modbus_TIA_V14_V0005.zip	<ul style="list-style-type: none"><li>■ Bausteinbibliothek für Siemens TIA Portal V14.</li><li>■ Für den Einsatz in CPUs von VIPA bzw. S7-300 CPUs von Siemens.</li></ul>
Modbus_TIA_V11_V13_V0003.zip	<ul style="list-style-type: none"><li>■ Bausteinbibliothek für Siemens TIA Portal V11 und V13.</li><li>■ Für den Einsatz in CPUs von VIPA bzw. S7-300 CPUs von Siemens.</li></ul>


### 3.1 Einbinden in Siemens SIMATIC Manager

#### Übersicht






Die Einbindung in den Siemens SIMATIC Manager erfolgt nach folgenden Schritten:

1.  ZIP-Datei laden
2.  Bibliothek "dearchivieren"
3.  Bibliothek öffnen und Bausteine in Projekt übertragen



#### ZIP-Datei laden

-  Navigieren Sie auf der Webseite zu der gewünschten ZIP-Datei, laden und speichern Sie diese in Ihrem Arbeitsverzeichnis.

#### Bibliothek dearchivieren

1.  Starten Sie den Siemens SIMATIC Manager mit Ihrem Projekt.
2.  Öffnen Sie mit "*Datei → Dearchivieren*" das Dialogfenster zur Auswahl der ZIP-Datei.
3.  Wählen Sie die entsprechende ZIP-Datei an und klicken Sie auf [Öffnen].
4.  Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind.
5.  Starten Sie den Entpackvorgang mit [OK].

#### Bibliothek öffnen und Bausteine in Projekt übertragen

1.  Öffnen Sie die Bibliothek nach dem Entpackvorgang.
2.  Öffnen Sie Ihr Projekt und kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis "Bausteine" Ihres Projekts.
  - ⇒ Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die VIPA-spezifischen Bausteine.



*Werden anstelle der SFCs FCs verwendet, so werden diese von den VIPA CPUs ab Firmware 3.6.0 unterstützt.*



## 3.2 Einbinden in Siemens TIA Portal

### Übersicht

Die Einbindung in das Siemens TIA Portal erfolgt nach folgenden Schritten:

1. ZIP-Datei laden
2. ZIP-Datei entpacken
3. Bibliothek "dearchivieren"
4. Bibliothek öffnen und Bausteine in Projekt übertragen

### ZIP-Datei laden

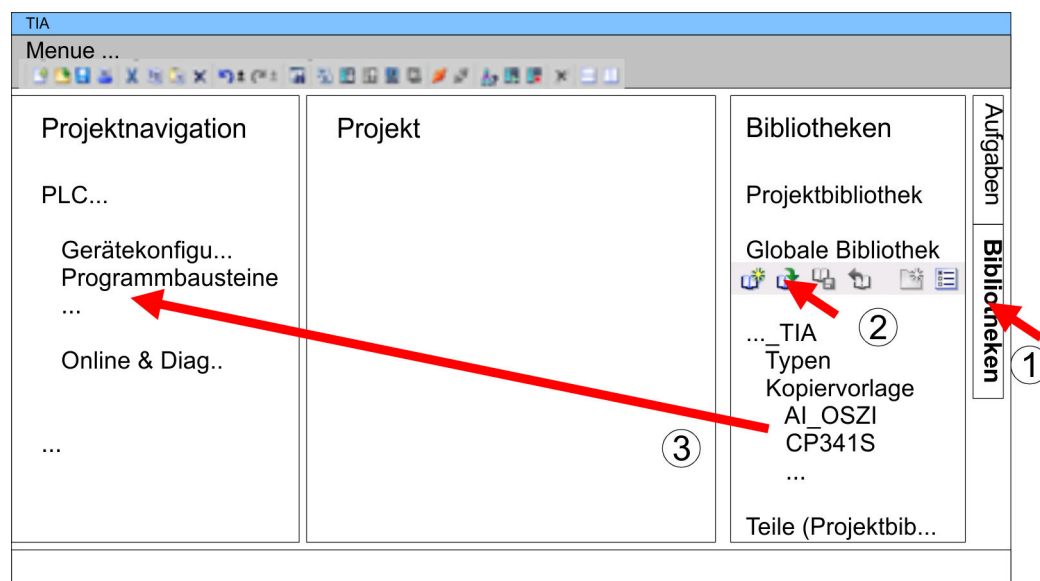
1. Navigieren Sie auf der Webseite zu der ZIP-Datei, welche zu Ihrer Programmversion passt.
2. Laden und speichern Sie diese in Ihrem Arbeitsverzeichnis.

### ZIP-Datei entpacken

- Entpacken Sie die ZIP-Datei mit Ihrem Entpackprogramm in ein Arbeitsverzeichnis für das Siemens TIA Portal.

### Bibliothek öffnen und Bausteine in Projekt übertragen

1. Starten Sie das Siemens TIA Portal mit Ihrem Projekt.
2. Wechseln Sie in die *Projektansicht*.
3. Wählen Sie auf der rechten Seite die Task-Card "Bibliotheken".
4. Klicken Sie auf "Globale Bibliothek".
5. Klicken Sie auf "Globale Bibliothek öffnen".
6. Navigieren Sie zu ihrem Arbeitsverzeichnis und laden Sie die Datei ...\_TIA.al1x.



7. Kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis "Programmbausteine" in der *Projektnavigation* Ihres Projekts. Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die VIPA-spezifischen Bausteine.

## 4 Modbus-Kommunikation - "Modbus Communication"

### 4.1 TCP

#### 4.1.1 FB 70 - TCP\_MB\_CLIENT - Modbus/TCP-Client

##### 4.1.1.1 Beschreibung

Dieser Funktionsbaustein ermöglicht den Betrieb einer Ethernet-Schnittstelle als Modbus/TCP-Client.

##### Aufrufparameter

Name	Deklaration	Typ	Beschreibung
REQ	IN	BOOL	Auftrag starten mit Flanke 0-1.
ID	IN	WORD	ID von TCON.
MB_FUNCTION	IN	BYTE	Modbus: <i>Funktions-Code</i> .
MB_DATA_ADDR	IN	WORD	Modbus: Startadresse oder <i>Sub-Funktions-Code</i> .
MB_DATA_LEN	IN	INT	Modbus: Anzahl der Register/Bits.
MB_DATA_PTR	IN	ANY	Modbus: Datenpuffer (nur Merkerbereich oder Datenbaustein vom Datentyp Byte zulässig) für den Zugriff mit <i>Funktions-Code 03h, 06h und 10h</i> .
DONE *	OUT	BOOL	Auftrag fertig ohne Fehler.
BUSY	OUT	BOOL	Auftrag ist in Bearbeitung.
ERROR *	OUT	BOOL	Auftrag fertig mit Fehler - Parameter STATUS enthält die Fehlerinformation.
STATUS *	OUT	WORD	Erweiterte Status- und Fehlerinformationen.

\*) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung

##### Parameter im Instanz-DB

Name	Deklaration	Typ	Beschreibung
PROTOCOL_TIMEOUT	STAT	INT	Sperrzeit bevor ein aktiver Auftrag vom Anwender abgebrochen werden kann. Default: 3s
RCV_TIMEOUT	STAT	INT	Überwachungszeit für einen Auftrag. Default: 2s
MB_TRANS_ID	STAT	WORD	Modbus: Startwert für den Transaktions Identifier. Default: 1
MB_UNIT_ID	STAT	BYTE	Modbus: Geräteidentifikation. Default: 255

Hierbei ist folgendes zu beachten:

- Die *Aufrufparameter* sind beim Baustein-Aufruf anzugeben. Neben den *Aufrufparametern* finden Sie alle Parameter im Instanz-DB.
- Die Kommunikationsverbindung muss zuvor über FB 65 (TCON) initialisiert werden.

- FB 63 (TSEND) und FB 64 (TRCV) sind für die Verwendung des Baustein erforderlich.
- Während einer Auftragsbearbeitung wird der Instanz-DB für andere Clients gesperrt.
- Während einer Auftragsbearbeitung werden Änderungen an den Eingangsparametern nicht ausgewertet.
- Unter einer der folgenden Bedingungen ist eine Auftragsbearbeitung abgeschlossen bzw. wird abgebrochen:
  - DONE = 1 bei Auftrag ohne Fehler
  - ERROR = 1 bei Auftrag mit Fehler
  - Ablauf von RCV\_TIMEOUT
  - REQ = FALSE nach Ablauf von PROTOCOL\_TIMEOUT
- Wird REQ zurückgesetzt bevor DONE oder ERROR gesetzt oder PROTOCOL\_TIMEOUT abgelaufen ist, wird STATUS 8200h geliefert. Hierbei wird der aktive Auftrag weiterhin bearbeitet.

**Status- und Fehleranzeige**

Der Funktionsbaustein liefert über STATUS die folgenden Status- und Fehlerinformationen.

STATUS	DONE	BUSY	ERROR	Beschreibung
0000h	1	0	0	Anweisung fehlerfrei ausgeführt.
7000h	0	0	0	Keine Verbindung aufgebaut oder Kommunikationsfehler (TCON).
7004h	0	0	0	Verbindung hergestellt und überwacht. Keine Auftragsbearbeitung aktiv.
7005h	0	1	0	Daten werden gesendet.
7006h	0	1	0	Daten werden empfangen.
8210h	0	0	1	Die Hardware ist inkompatibel zur Baustein-Bibliothek Modbus RTU/TCP.
8380h	0	0	1	Empfangenes Modbus-Frame hat nicht das richtige Format oder eine ungültige Länge.
8381h	0	0	1	Server liefert <i>Exception-Code 01h</i> .
8382h	0	0	1	Server liefert <i>Exception-Code 03h</i> oder falsche Startadresse.
8383h	0	0	1	Server liefert <i>Exception-Code 02h</i> .
8384h	0	0	1	Server liefert <i>Exception-Code 04h</i> .
8386h	0	0	1	Server liefert falschen <i>Funktions-Code</i> .
8387h	0	0	1	Verbindungs-ID (TCON) passt nicht zur Instanz oder Server liefert falsche Protokoll-ID.
8388h	0	0	1	Server liefert falschen Wert oder falsche Quantity.
80C8h	0	0	1	Keine Antwort des Servers im definierten Zeitraum (RCV_TIMEOUT).
8188h	0	0	1	MB_FUNCTION ungültig.
8189h	0	0	1	MB_DATA_ADDR ungültig.
818Ah	0	0	1	MB_DATA_LEN ungültig.
818Bh	0	0	1	MB_DATA_PTR ungültig.

TCP &gt; FB 70 - TCP\_MB\_CLIENT - Modbus/TCP-Client

STATUS	DONE	BUSY	ERROR	Beschreibung
818Ch	0	0	1	BLOCKED_PROC_TIMEOUT oder RCV_TIMEOUT ungültig.
818Dh	0	0	1	Server liefert falsche Transaktions-ID.
8200h	0	0	1	Eine andere Modbus-Anfrage wird zur Zeit über den Port verarbeitet (PROTOCOL_TIMEOUT).

#### 4.1.1.2 Beispiel

##### Aufgabenstellung

Von einem Modbus/TCP-Server sollen mit *Funktions-Code 03h* 100 Register ab Startadresse 2000 gelesen werden und im Merkerbereich ab MB200 abgelegt werden. Fehler sollen abgespeichert werden.

##### OB1

```

CALL FB 65 , DB65
  REQ      :=M100.0
  ID       :=W#16#1
  DONE     :=M100.1
  BUSY     :=
  ERROR    :=M100.2
  STATUS   :=MW102
  CONNECT:=P#DB255.DBX 0.0 BYTE 64

UN      M      100.2
SPB     ERR1
L       MW     102
T       MW     104
ERR1: NOP 0
U       M      100.1
R       M      100.0

CALL FB 70 , DB70
  REQ      :=M101.0
  ID       :=W#16#1
  MB_FUNCTION :=B#16#3
  MB_DATA_ADDR:=W#16#7D0
  MB_DATA_LEN :=100
  MB_DATA_PTR :=P#M 200.0 BYTE 200
  DONE     :=M101.1
  BUSY     :=
  ERROR    :=M101.2
  STATUS   :=MW106

UN      M      101.2
SPB     ERR2
L       MW     106
T       MW     108
ERR2: NOP 0
U       M      101.1
R       M      101.0

```

##### OB1 - Beschreibung

- 1.** ➤ Aufruf von FB 65 (TCON) zur Herstellung der Kommunikationsverbindung mit der Partnerstation.
- 2.** ➤ Aufruf des Modbus/TCP-Client Hantierungsbausteins mit den korrekten Parametern.
- 3.** ➤ Es ist keine Verbindung zur Partnerstation aufgebaut und MW102 liefert 7000h.

- 4.** ➔ M100.0 in der CPU auf TRUE setzen.
- ⇒ Wenn M100.0 automatisch zurück gesetzt wird, ist die Verbindung zur Partnerstation aufgebaut und MW108 liefert 7004h.
- 5.** ➔ M101.0 in der CPU auf TRUE setzen.
- ⇒ Der Modbus-Request wird versendet und auf eine Antwort gewartet.
- Wenn M101.0 automatisch zurück gesetzt wird, wurde der Auftrag fehlerfrei bearbeitet und die gelesenen Daten liegen ab Merkerbyte 200 in der CPU. MW108 liefert 7004h und signalisiert die Bereitschaft für einen neuen Auftrag.
- Wenn M101.0 nicht automatisch zurück gesetzt wird und MW108 einen Wert ungleich 0 liefert, ist ein Fehler aufgetreten. Die Fehlerursache kann über den Code in MW108 ausgelesen werden (z.B. MW108 = 8382h wenn die Startadresse 2000 im Server nicht vorhanden ist). MW108 liefert 7004h und signalisiert die Bereitschaft für einen neuen Auftrag.

## 4.1.2 FB 71 - TCP\_MB\_SERVER - Modbus/TCP-Server

### 4.1.2.1 Beschreibung

Dieser Funktionsbaustein ermöglicht den Betrieb einer Ethernet-Schnittstelle als Modbus/TCP-Server.

#### Aufrufparameter

Name	Deklara- tion	Typ	Beschreibung
ENABLE	IN	BOOL	Aktivierung/Deaktivierung Modbus-Server.
MB_DATA_PTR	IN	ANY	Modbus: Datenpuffer (nur Merkerbereich oder Datenbaustein vom Datentyp Byte zulässig) für den Zugriff mit <i>Funktions-Code 03h, 06h und 10h</i> .
ID	IN	WORD	ID von TCON.
NDR*	OUT	BOOL	Neue Daten wurden durch den Modbus-Client geschrieben.
DR*	OUT	BOOL	Daten wurden vom Modbus Client gelesen.
ERROR*	OUT	BOOL	Auftrag fertig mit Fehler - Parameter STATUS enthält die Fehlerinformation.
STATUS*	OUT	WORD	Erweiterte Status- und Fehlerinformationen.

\*) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung

#### Parameter im Instanz-DB

Name	Deklara- tion	Typ	Beschreibung
REQUEST_COUNT	STAT	WORD	Zähler für jedes empfangene Telegramm.
MESSAGE_COUNT	STAT	WORD	Zähler für jeden gültigen Modbus-Request.
XMT_RCV_COUNT	STAT	WORD	Zähler für jedes empfangene Telegramm, welches keinen gültigen Modbus-Request enthält.
EXCEPTION_COUNT	STAT	WORD	Zähler für jeden negativ quittierten Modbus-Request.

TCP &gt; FB 71 - TCP\_MB\_SERVER - Modbus/TCP-Server

Name	Deklara- tion	Typ	Beschreibung
SUCCESS_COUNT	STAT	WORD	Zähler für jeden positiv quittierten Modbus-Request.
FC1_ADDR_OUTPUT_START	STAT	WORD	Modbus <i>Funktions-Code 01h</i> Startregister für A0.0 Default: 0
FC1_ADDR_OUTPUT_END	STAT	WORD	Modbus <i>Funktions-Code 01h</i> Endregister für Ax.y Default: 19999
FC1_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 01h</i> Startregister für M0.0 Default: 20000
FC1_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 01h</i> Endregister für Mx.y Default: 39999
FC2_ADDR_INPUT_START	STAT	WORD	Modbus <i>Funktions-Code 02h</i> Startregister für E0.0 Default: 0
FC2_ADDR_INPUT_END	STAT	WORD	Modbus <i>Funktions-Code 02h</i> Endregister für Ex.y Default: 19999
FC2_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 02h</i> Startregister für M0.0 Default: 20000
FC2_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 02h</i> Endregister für Mx.y Default: 39999
FC4_ADDR_INPUT_START	STAT	WORD	Modbus <i>Funktions-Code 04h</i> Startregister für EW0 Default: 0
FC4_ADDR_INPUT_END	STAT	WORD	Modbus <i>Funktions-Code 04h</i> Endregister für EWx Default: 19999
FC4_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 04h</i> Startregister für MW0 Default: 20000
FC4_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 04h</i> Endregister für MWx Default: 39999
FC5_ADDR_OUTPUT_START	STAT	WORD	Modbus <i>Funktions-Code 05h</i> Startregister für A0.0 Default: 0
FC5_ADDR_OUTPUT_END	STAT	WORD	Modbus <i>Funktions-Code 05h</i> Endregister für Ax.y Default: 19999
FC5_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 05h</i> Startregister für M0.0 Default: 20000
FC5_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 05h</i> Endregister für Mx.y Default: 39999
FC15_ADDR_OUTPUT_START	STAT	WORD	Modbus <i>Funktions-Code 0Fh</i> Startregister für A0.0 Default: 0
FC15_ADDR_OUTPUT_END	STAT	WORD	Modbus <i>Funktions-Code 0Fh</i> Endregister für Ax.y Default: 19999

Name	Deklaration	Typ	Beschreibung
FC15_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 0Fh</i> Startregister für M0.0 Default: 20000
FC15_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 0Fh</i> Endregister für Mx.y Default: 39999

Hierbei ist folgendes zu beachten:

- Die *Aufrufparameter* sind beim Baustein-Aufruf anzugeben. Neben den *Aufrufparametern* finden Sie alle Parameter im Instanz-DB.
- Die Kommunikationsverbindung muss zuvor über FB 65 (TCON) initialisiert werden.
- FB 63 (TSEND) und FB 64 (TRCV) sind für die Verwendung des Bausteins erforderlich.
- Die INPUT/OUTPUT Modbus-Adressen eines *Funktions-Codes* müssen vor den MEMORY Modbus-Adressen liegen und somit immer kleiner sein.
- Innerhalb eines *Funktions-Code* darf keine Modbus-Adresse mehrfach definiert werden - auch die 0 nicht!
- Der Server kann nur einen Auftrag gleichzeitig bearbeiten. Neue Modbus-Anfragen während einer Auftragsbearbeitung werden ignoriert und nicht beantwortet.

### Status- und Fehleranzeige

Der Funktionsbaustein liefert über *STATUS* die folgenden Status- und Fehlerinformationen.

STATUS	NDR	DR	ERROR	Beschreibung
0000h	0 oder 1*		0	Anweisung fehlerfrei ausgeführt.
7000h	0	0	0	Keine Verbindung aufgebaut oder Kommunikationsfehler (TCON).
7005h	0	0	0	Daten werden gesendet.
7006h	0	0	0	Daten werden empfangen.
8210h	0	0	1	Die Hardware ist inkompatibel zur Baustein-Bibliothek Modbus RTU/TCP.
8380h	0	0	1	Empfangenes Modbus-Frame hat nicht das richtige Format oder es wurden zu wenige Bytes empfangen.
8381h	0	0	1	<i>Exception-Code 01h</i> , <i>Funktion-Code</i> wird nicht unterstützt.
8382h	0	0	1	<i>Exception-Code 03h</i> , Datenlänge oder Datenwert ungültig.
8383h	0	0	1	<i>Exception-Code 02h</i> , Ungültige Startadresse bzw. Adressbereich.
8384h	0	0	1	<i>Exception-Code 04h</i> , Bereichslängenfehler beim Zugriff auf Eingänge, Ausgänge oder Merker.
8387h	0	0	1	Verbindungs-ID (TCON) passt nicht zur Instanz oder Client liefert falsche Protokoll-ID.
8187h	0	0	1	MB_DATA_PTR ungültig.

\*) Fehlerfreier Modbus-Auftrag mit *Funktions-Code 05h, 06h, 0Fh* oder *10h* liefert NDR=1 und DR=0.

Fehlerfreier Modbus-Auftrag mit *Funktions-Code 01h, 02h, 03h, 04h* liefert DR=1 und NDR=0.

TCP &gt; FB 71 - TCP\_MB\_SERVER - Modbus/TCP-Server

## 4.1.2.2 Beispiel

## Aufgabenstellung

Die CPU stellt 100 Byte Daten im Merkerbereich ab MB200 für einen Modbus-Client über die Modbus-Register 0...49 zur Verfügung. Die Daten können vom Modbus-Client mit dem *Funktions-Code 03h* gelesen und mit *Funktions-Code 06h, 10h* geschrieben werden. Der Ausgang A1.0 in der CPU soll von einem Modbus-Client über den *Funktions-Code 05h* und die Startadresse 5008 angesteuert werden können. Fehler sollen abgespeichert werden.

## OB1

```

CALL FB 65 , DB65
    REQ      :=M100.0
    ID       :=W#16#1
    DONE     :=M100.1
    BUSY     :=
    ERROR    :=M100.2
    STATUS   :=MW102
    CONNECT:=P#DB255.DBX 0.0 BYTE 64

    UN      M      100.2
    SPB     ERR1
    L       MW     102
    T       MW     104
ERR1: NOP 0
    U       M      100.1
    R       M      100.0

    L       5000
    T       DB71.DBW 52

CALL FB 71 , DB71
    ENABLE   :=M101.0
    MB_DATA_PTR:=P#M 200.0 BYTE 100
    ID       :=W#16#1
    NDR      :=M101.1
    DR       :=M101.2
    ERROR    :=M101.3
    STATUS   :=MW106

    UN      M      101.3
    SPB     ERR2
    L       MW     106
    T       MW     108
ERR2: NOP 0

```

## OB1 - Beschreibung

1. ➤ Aufruf von FB 65 (TCON) zur Herstellung der Kommunikationsverbindung mit der Partnerstation.
2. ➤ Aufruf des Modbus/TCP-Server Hantierungsbausteins mit den korrekten Parametern.
3. ➤ Es ist keine Verbindung zur Partnerstation aufgebaut und MW102 liefert 7000h.
4. ➤ M100.0 in der CPU auf TRUE setzen.  
⇒ Wenn M100.0 automatisch zurück gesetzt wird, ist die Verbindung zur Partnerstation aufgebaut und MW108 liefert 7006h.
5. ➤ Das Modbus-Startregister für die über *Funktions-Code 05h* erreichbaren Ausgänge im Prozessabbild wird im Beispiel über den Parameter FC5\_ADDR\_OUTPUT\_START (Wort 52 im Instanz-Datenbaustein) geändert.
6. ➤ M101.0 in der CPU auf TRUE setzen.  
⇒ Der Modbus-Server arbeitet nun.
7. ➤ Der Client sendet einen Modbus-Request mit *Funktions-Code 03h*, Startadresse 10 und Quantity 30.



- ⇒ Der Server antwortet mit 60 Byte ab MB220. DR wird für einen CPU-Zyklus angesteuert und somit M101.2 auf "1" gesetzt.
- 8. Der Client sendet einen Modbus-Request mit *Funktions-Code 05h*, Startadresse 5008 und dem Wert FF00h.
  - ⇒ Der Server quittiert den Auftrag und schreibt den Ausgang A1.0 auf "1". NDR wird für einen CPU-Zyklus angesteuert und somit M101.1 auf "1" gesetzt.
- 9. Der Client sendet einen Modbus-Request mit *Funktions-Code 03h*, Startadresse 50 (nicht vorhanden!) und Quantity 1.
  - ⇒ Der Server antwortet mit einem *Exception-Code 02h* und steuert ERROR/STATUS für einen CPU-Zyklus an. MW108 liefert 8383h.

## 4.2 RTU

### 4.2.1 FB 72 - RTU\_MB\_MASTER - Modbus-RTU-Master

#### 4.2.1.1 Beschreibung

Dieser Funktionsbaustein ermöglicht den Betrieb der internen seriellen RS485 Schnittstelle einer VIPA SPEED7 CPU oder eines VIPA System SLIO CP 040 als Modbus-RTU-Master.

#### Aufrufparameter

Name	Deklaration	Typ	Beschreibung
REQ	IN	BOOL	Auftrag starten mit Flanke 0-1.
HARDWARE	IN	BYTE	1 = System SLIO CP 040 / 2 = VIPA SPEED7 CPU
LADDR	IN	INT	Logische Adresse vom System SLIO CP 040 (Parameter wird für VIPA SPEED7 CPU ignoriert).
MB_UNIT_ID	IN	BYTE	Modbus: Geräteidentifikation = Adresse vom Slave (0 ... 247).
MB_FUNCTION	IN	BYTE	Modbus: <i>Funktions-Code</i> .
MB_DATA_ADDR	IN	WORD	Modbus: Startadresse oder <i>Sub-Funktions-Code</i> .
MB_DATA_LEN	IN	INT	Modbus: Anzahl der Register/Bits.
MB_DATA_PTR	IN	ANY	Modbus: Datenpuffer (nur Merkerbereich oder Datenbaustein vom Datentyp Byte zulässig) für den Zugriff mit <i>Funktions-Code 03h, 06h und 10h</i> .
DONE*	OUT	BOOL	Auftrag fertig ohne Fehler.
BUSY	OUT	BOOL	Auftrag ist in Bearbeitung.
ERROR*	OUT	BOOL	Auftrag fertig mit Fehler - Parameter <i>STATUS</i> enthält die Fehlerinformation.
STATUS*	OUT	WORD	Erweiterte Status- und Fehlerinformationen.

\*) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung

## Parameter im Instanz-DB

Name	Deklaration	Typ	Beschreibung
INIT	STAT	BOOL	Eine Flanke 0-1 führt einen Synchron Reset am System SLIO CP 040 durch. Nach erfolgreichem Reset wird das Bit automatisch zurück gesetzt.

Hierbei ist folgendes zu beachten:

- Die *Aufrufparameter* sind beim Baustein-Aufruf anzugeben. Neben den *Aufrufparametern* finden Sie alle Parameter im Instanz-DB.
- Die verwendete Schnittstelle muss zuvor konfiguriert werden:
  - VIPA System SLIO CP 040: Projektierung als "Modbus Master RTU" mit 60 Byte IO-Size in der Hardwarekonfiguration.
  - Interne serielle RS485 Schnittstelle einer CPU von VIPA:  
Projektierung über SFC 216 (SER\_CFG) mit Protokoll "Modbus Master RTU".
- FB 60 SEND und FB 61 RECEIVE (oder FB 65 SEND\_RECV) sind für die Verwendung des Bausteins zwingend erforderlich, auch wenn die interne serielle RS485 Schnittstelle einer CPU von VIPA verwendet wird.
- Während einer Auftragsbearbeitung werden Änderungen an den Eingangsparametern nicht ausgewertet.
- Broadcast Request über MB\_UNIT\_ID = 0 werden nur für schreibende Funktionen akzeptiert.
- Unter einer der folgenden Bedingungen ist eine Auftragsbearbeitung abgeschlossen bzw. wird abgebrochen:
  - *DONE* = 1 bei Auftrag ohne Fehler
  - *ERROR* = 1 bei Auftrag mit Fehler
  - Ablauf vom Timeout (Parametrierung bei der Schnittstelle)
- Wird *REQ* zurückgesetzt bevor *DONE* oder *ERROR* gesetzt ist, wird *STATUS* 8200h geliefert. Hierbei wird der aktive Auftrag weiterhin bearbeitet.

## Status- und Fehleranzeige

Der Funktionsbaustein liefert über STATUS die folgenden Status- und Fehlerinformationen.

STATUS	DONE	BUSY	ERROR	Beschreibung
0000h	1	0	0	Anweisung fehlerfrei ausgeführt.
7000h	0	0	0	Keine Verbindung aufgebaut oder Kommunikationsfehler.
7004h	0	0	0	Verbindung hergestellt und überwacht. Keine Auftragsbearbeitung aktiv.
7005h	0	1	0	Daten werden gesendet.
7006h	0	1	0	Daten werden empfangen.
8210h	0	0	1	Die Hardware ist inkompatibel zur Baustein-Bibliothek Modbus RTU/TCP.
8381h	0	0	1	Server liefert <i>Exception-Code 01h</i> .
8382h	0	0	1	Server liefert <i>Exception-Code 03h</i> oder falsche Startadresse.
8383h	0	0	1	Server liefert <i>Exception-Code 02h</i> .
8384h	0	0	1	Server liefert <i>Exception-Code 04h</i> .
8386h	0	0	1	Server liefert falschen <i>Funktions-Code</i> .
8388h	0	0	1	Server liefert falschen Wert oder falsche Quantity.

STATUS	DONE	BUSY	ERROR	Beschreibung
80C8h	0	0	1	Keine Antwort des Servers im definierten Zeitraum (Timeout über Schnittstelle parametrierbar).
8188h	0	0	1	MB_FUNCTION ungültig.
8189h	0	0	1	MB_DATA_ADDR ungültig.
818Ah	0	0	1	MB_DATA_LEN ungültig.
818Bh	0	0	1	MB_DATA_PTR ungültig.
8201h	0	0	1	HARDWARE ungültig.
8202h	0	0	1	MB_UNIT_ID ungültig.
8200h	0	0	1	Eine andere Modbus-Anfrage wird zur Zeit über den Port verarbeitet.

#### 4.2.1.2 Beispiel

##### Aufgabenstellung

Von einem Modbus-RTU-Slave mit Adresse 99 sollen mit *Funktions-Code 03h*, 100 Register ab Startadresse 2000 gelesen und im Merkerbereich ab MB200 abgelegt werden. Fehler sollen abgespeichert werden. Der Modbus-RTU-Master wird über die interne serielle Schnittstelle einer CPU von VIPA realisiert.

##### OB100

```
CALL SFC 216
  Protocol :=B#16#5
  Parameter :=DB10
  Baudrate:=B#16#9
  CharLen:=B#16#3
  Parity:=B#16#2
  StopBits:=B#16#1
  FlowControl:=B#16#1
  RetVal:=MW100
```

##### OB100 - Beschreibung

- Aufruf von SFC 216 (SER\_CFG) zur Konfiguration der internen seriellen Schnittstelle der CPU von VIPA.
- Protokoll: "Modbus Master RTU", 9600 Baud, 8 Datenbit, 1 Stoppbit, gerade Parität, kein Flusskontrolle.
- Der DB10 enthält eine Variable vom Typ WORD mit einem Modbus-Timeout (Wert in ms).

##### OB1

```
CALL FB 72 , DB72
  REQ      :=M101.0
  HARDWARE :=B#16#2
  LADDR    :=
  MB_UNIT_ID :=B#16#63
  MB_FUNCTION :=B#16#3
  MB_DATA_ADDR:=W#16#7D0
  MB_DATA_LEN :=100
  MB_DATA_PTR :=P#M 200.0 BYTE 200
  DONE      :=M101.1
  BUSY      :=
  ERROR     :=M101.2
  STATUS    :=MW102

  UN      M      101.2
  SPB     ERR1
  L       MW     102
```

RTU &gt; FB 73 - RTU\_MB\_SLAVE - Modbus-RTU-Slave

```

      T      MW      104
ERR1: NOP    0
      U      M      101.1
      R      M      101.0

```

**OB1 - Beschreibung**

1. ➔ Aufruf des Modbus-RTU-Master Hantierungsbausteins mit den korrekten Parametern.
2. ➔ Wenn die Schnittstelle im OB 100 korrekt initialisiert wurde, ist der Master einsatzbereit und MW102 liefert 7004h zurück.
3. ➔ M101.0 in der CPU auf TRUE setzen.
  - ⇒ Der Modbus-Request wird versendet und auf eine Antwort gewartet.

Wenn M101.0 automatisch zurück gesetzt wird, wurde der Auftrag fehlerfrei bearbeitet und die gelesenen Daten liegen ab Merkerbyte 200 in der CPU. MW104 liefert 7004h und signalisiert die Bereitschaft für einen neuen Auftrag.

Wenn M101.0 nicht automatisch zurück gesetzt wird und MW104 einen Wert ungleich 0 liefert, ist ein Fehler aufgetreten. Die Fehlerursache kann über den Code in MW104 ausgelesen werden (z.B. MW104 = 8382h wenn die Startadresse 2000 im Server nicht vorhanden ist). MW102 liefert 7004h und signalisiert die Bereitschaft für einen neuen Auftrag.

**4.2.2 FB 73 - RTU\_MB\_SLAVE - Modbus-RTU-Slave****4.2.2.1 Beschreibung**

Dieser Funktionsbaustein ermöglicht den Betrieb der internen seriellen RS485 Schnittstelle einer VIPA SPEED7 CPU oder eines VIPA System SLIO CP 040 als Modbus-RTU-Slave.

**Aufrufparameter**

Name	Deklaration	Typ	Beschreibung
ENABLE	IN	BOOL	Aktivierung/Deaktivierung des Modbus-Server.
HARDWARE	IN	BYTE	1 = System SLIO CP 040 / 2 = VIPA SPEED7 CPU
LADDR	IN	INT	Logische Adresse vom System SLIO CP 040 (Parameter wird für VIPA SPEED7 CPU ignoriert).
MB_UNIT_ID	IN	BYTE	Modbus: Geräteidentifikation = eigene Adresse (1 ... 247).
MB_DATA_PTR	IN	ANY	Modbus: Datenpuffer (nur Merkerbereich oder Datenbaustein vom Datentyp Byte zulässig) für den Zugriff mit <i>Funktions-Code 03h, 06h und 10h</i> .
NDR*	OUT	BOOL	Neue Daten wurden durch den Modbus-Client geschrieben.
DR*	OUT	BOOL	Daten wurden vom Modbus-Client gelesen.
ERROR*	OUT	BOOL	Auftrag fertig mit Fehler - Parameter <i>STATUS</i> enthält die Fehlerinformation.
STATUS*	OUT	WORD	Erweiterte Status- und Fehlerinformationen.

\*) Parameter steht bis zum nächsten Aufruf des FBs zur Verfügung

## Parameter im Instanz-DB

Name	Deklaration	Typ	Beschreibung
INIT	STAT	BOOL	Eine Flanke 0-1 führt einen Synchron Reset am System SLIO CP 040 durch.
REQUEST_COUNT	STAT	WORD	Zähler für jedes empfangene Telegramm.
MESSAGE_COUNT	STAT	WORD	Zähler für jeden gültigen Modbus-Request.
BROADCAST_COUNT	STAT	WORD	Zähler für jeden gültigen Modbus-Broadcast-Request.
EXCEPTION_COUNT	STAT	WORD	Zähler für jeden negativ quittierten Modbus-Request.
SUCCESS_COUNT	STAT	WORD	Zähler für jeden positiv quittierten Modbus-Request.
BAD_CRC_COUNT	STAT	WORD	Zähler für jeden gültigen Modbus-Request mit CRC-Fehler.
FC1_ADDR_OUTPUT_START	STAT	WORD	Modbus <i>Funktions-Code 01h</i> Startregister für A0.0 Default: 0
FC1_ADDR_OUTPUT_END	STAT	WORD	Modbus <i>Funktions-Code 01h</i> Endregister für Ax.y Default: 19999
FC1_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 01h</i> Startregister für M0.0 Default: 20000
FC1_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 01h</i> Endregister für Mx.y Default: 39999
FC2_ADDR_INPUT_START	STAT	WORD	Modbus <i>Funktions-Code 02h</i> Startregister für E0.0 Default: 0
FC2_ADDR_INPUT_END	STAT	WORD	Modbus <i>Funktions-Code 02h</i> Endregister für Ex.y Default: 19999
FC2_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 02h</i> Startregister für M0.0 Default: 20000
FC2_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 02h</i> Endregister für Mx.y Default: 39999
FC4_ADDR_INPUT_START	STAT	WORD	Modbus <i>Funktions-Code 04h</i> Startregister für EW0 Default: 0
FC4_ADDR_INPUT_END	STAT	WORD	Modbus <i>Funktions-Code 04h</i> Endregister für EWx Default: 19999
FC4_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 04h</i> Startregister für MW0 Default: 20000
FC4_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 04h</i> Endregister für MWx Default: 39999
FC5_ADDR_OUTPUT_START	STAT	WORD	Modbus <i>Funktions-Code 05h</i> Startregister für A0.0 Default: 0
FC5_ADDR_OUTPUT_END	STAT	WORD	Modbus <i>Funktions-Code 05h</i> Endregister für Ax.y Default: 19999

RTU &gt; FB 73 - RTU\_MB\_SLAVE - Modbus-RTU-Slave

Name	Deklaration	Typ	Beschreibung
FC5_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 05h</i> Startregister für M0.0 Default: 20000
FC5_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 05h</i> Endregister für Mx.y Default: 39999
FC15_ADDR_OUTPUT_START	STAT	WORD	Modbus <i>Funktions-Code 0Fh</i> Startregister für A0.0 Default: 0
FC15_ADDR_OUTPUT_END	STAT	WORD	Modbus <i>Funktions-Code 0Fh</i> Endregister für Ax.y Default: 19999
FC15_ADDR_MEMORY_START	STAT	WORD	Modbus <i>Funktions-Code 0Fh</i> Startregister für M0.0 Default: 20000
FC15_ADDR_MEMORY_END	STAT	WORD	Modbus <i>Funktions-Code 0Fh</i> Endregister für Mx.y Default: 39999

Hierbei ist folgendes zu beachten:

- Die *Aufrufparameter* sind beim Baustein-Aufruf anzugeben. Neben den *Aufrufparametern* finden Sie alle Parameter im Instanz-DB.
- Die verwendete Schnittstelle muss zuvor konfiguriert werden:
  - VIPA System SLIO CP 040: Projektierung als ASCII-Modul mit 60 Byte IO-Size in der Hardwarekonfiguration.
  - Interne serielle RS485 Schnittstelle einer CPU von VIPA:  
Projektierung über SFC 216 (SER\_CFG) mit Protokoll "ASCII".
- FB 60 SEND und FB 61 RECEIVE (oder FB 65 SEND\_RECV) sind für die Verwendung des Baustein zwingend erforderlich, auch wenn die interne serielle RS485 Schnittstelle einer CPU von VIPA verwendet wird.
- Broadcast Request über MB\_UNIT\_ID = 0 werden nur für schreibende Funktionen akzeptiert.
- Die INPUT/OUTPUT Modbus-Adressen eines *Funktions-Codes* müssen vor den MEMORY Modbus-Adressen liegen und somit immer kleiner sein.
- Innerhalb eines *Funktions-Codes* darf keine Modbus-Adresse mehrfach definiert werden, auch die 0 nicht!
- Der Slave kann nur einen Auftrag gleichzeitig bearbeiten. Neue Modbus-Anfragen während einer Auftragsbearbeitung werden ignoriert und nicht beantwortet.

### Status- und Fehleranzeige

Der Funktionsbaustein liefert über STATUS die folgenden Status- und Fehlerinformationen.

STATUS	NDR	DR	ERROR	Beschreibung
0000h	0 oder 1*		0	Anweisung fehlerfrei ausgeführt.
7000h	0	0	0	Keine Verbindung aufgebaut oder Kommunikationsfehler.
7005h	0	0	0	Daten werden gesendet.
7006h	0	0	0	Daten werden empfangen.
8210h	0	0	1	Die Hardware ist inkompatibel zur Baustein-Bibliothek Modbus RTU/TCP.
8380h	0	0	1	CRC-Fehler

STATUS	NDR	DR	ERROR	Beschreibung
8381h	0	0	1	<i>Exception-Code 01h</i> , Funktions-Code wird nicht unterstützt.
8382h	0	0	1	<i>Exception-Code 03h</i> , Datenlänge oder Datenwert ungültig.
8383h	0	0	1	<i>Exception-Code 02h</i> , Ungültige Startadresse bzw. Adressbereich.
8384h	0	0	1	<i>Exception-Code 04h</i> , Bereichslängenfehler beim Zugriff auf Eingänge, Ausgänge oder Merker
8187h	0	0	1	MB_DATA_PTR ungültig.
8201h	0	0	1	HARDWARE ungültig.
8202h	0	0	1	MB_UNIT_ID ungültig.
8203h	0	0	1	Fragmentiertes Empfangstelegramm ungültig (SFC 218).

\*) Fehlerfreier Modbus-Auftrag mit Funktions-Code 05h, 06h, 0Fh oder 10h liefert NDR=1 und DR=0.

Fehlerfreier Modbus-Auftrag mit Funktions-Code 01h, 02h, 03h, 04h liefert DR=1 und NDR=0.

#### 4.2.2.2 Beispiel

##### Aufgabenstellung

Die CPU stellt 100 Byte Daten im Merkerbereich ab MB200 für einen Modbus-Master über die Modbus-Register 0 ... 49 zur Verfügung. Die Daten können vom Modbus-Master über Funktions-Code 03h gelesen und über Funktions-Code 06h, 10h geschrieben werden. Der Ausgang A1.0 in der CPU soll von einem Modbus-Master über den Funktions-Code 05h und die Startadresse 5008 angesteuert werden können. Fehler sollen abgespeichert werden. Der Modbus-RTU-Slave mit der Adresse 99 wird über die interne serielle Schnittstelle einer CPU von VIPA realisiert.

##### OB100

```
CALL SFC 216
Protocol :=B#16#1
Parameter :=DB10
Baudrate:=B#16#9
CharLen:=B#16#3
Parity:=B#16#2
StopBits:=B#16#1
FlowControl:=B#16#1
RetVal:=MW100
```

##### OB100 - Beschreibung

- Aufruf von SFC 216 (SER\_CFG) zur Konfiguration der internen seriellen Schnittstelle der CPU von VIPA.
- Protokoll: "ASCII", 9600 Baud, 8 Datenbit, 1 Stoppbit, gerade Parität, kein Flusskontrolle.
- Der DB10 enthält eine Variable vom Typ WORD und muss als "Dummy" übergeben werden.

##### OB1

```
L 5000
T DB73.DBW 58

CALL FB 73 , DB73
ENABLE :=M101.0
HARDWARE :=B#16#2
LADDR :=
MB_UNIT_ID :=B#16#63
MB_DATA_PTR:=P#M 200.0 BYTE 100
```



```

NDR      :=M101.1
DR       :=M101.2
ERROR    :=M101.3
STATUS   :=MW102

```

```

UN      M      101.3
SPB     ERR1
L       MW     102
T       MW     104
ERR1: NOP      0

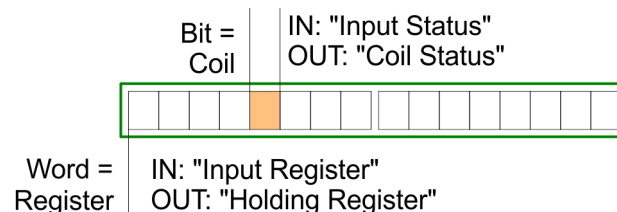
```

**OB1 - Beschreibung**

1. ➤ Aufruf des Modbus/TCP-Server Hantierungsbausteins mit den korrekten Parametern.
2. ➤ Wenn die Schnittstelle im OB100 korrekt initialisiert wurde, ist der Slave einsatzbereit und MW102 wird zu 7006h geliefert.
3. ➤ Das Modbus-Startregister für die über *Funktions-Code 05h* erreichbaren Ausgänge im Prozessabbild wird im Beispiel über den Parameter FC5\_ADDR\_OUTPUT\_START (Wort 58 im Instanz-Datenbaustein) geändert.
4. ➤ M101.0 in der CPU auf TRUE setzen.  
⇒ Der Modbus-Slave arbeitet nun.
5. ➤ Der Master sendet einen Modbus-Request mit *Funktions-Code 03h*, Startadresse 10 und Quantity 30.  
⇒ Der Slave antwortet mit 60Byte ab MB200. DR wird für einen CPU-Zyklus angesteuert und somit M101.2 auf "1" gesetzt.
6. ➤ Der Master sendet einen Modbus-Request mit *Funktions-Code 05h*, Startadresse 5008 und dem Wert FF00h.  
⇒ Der Slave quittiert den Auftrag und schreibt den Ausgang A1.0 auf "1". NDR wird für einen CPU-Zyklus angesteuert und somit M101.1 auf "1" gesetzt.
7. ➤ Der Master sendet einen Modbus-Request mit *Funktions-Code 03h*, Startadresse 50 (nicht vorhanden!) und Quantity 1.  
⇒ Der Slave antwortet mit einem *Exception Code 02h* und steuert ERROR/STATUS für einen CPU-Zyklus an. MW104 liefert 8383h.

**4.3 FKT Modbus-Funktionscodes - "FKT Codes"****Namenskonventionen**

Für Modbus gibt es Namenskonventionen, die hier kurz aufgeführt sind:



- Modbus unterscheidet zwischen Bit- und Wortzugriff; Bits = "Coils" und Worte = "Register".
- Bit-Eingänge werden als "Input-Status" bezeichnet und Bit-Ausgänge als "Coil-Status".
- Wort-Eingänge werden als "Input-Register" und Wort-Ausgänge als "Holding-Register" bezeichnet.

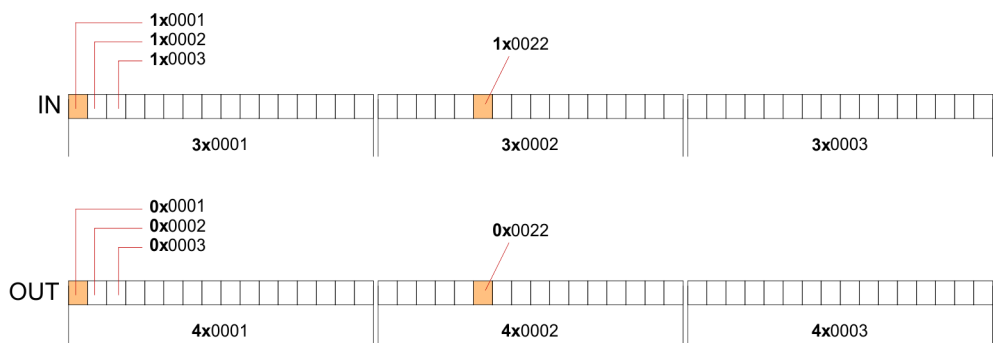
**Bereichsdefinitionen**

Üblicherweise erfolgt unter Modbus der Zugriff mittels der Bereiche 0x, 1x, 3x und 4x. Mit 0x und 1x haben Sie Zugriff auf digitale Bit-Bereiche und mit 3x und 4x auf analoge Wort-Bereiche.



Da aber bei den CPs von VIPA keine Unterscheidung zwischen Digital- und Analogdaten stattfindet, gilt folgende Zuordnung:

- 0x - Bit-Bereich für Ausgabe-Daten des Masters  
Zugriff über Funktions-Code 01h, 05h, 0Fh
- 1x - Bit-Bereich für Eingabe-Daten des Masters  
Zugriff über Funktions-Code 02h
- 3x - Wort-Bereich für Eingabe-Daten des Masters  
Zugriff über Funktions-Code 04h
- 4x - Wort-Bereich für Ausgabe-Daten des Masters  
Zugriff über Funktions-Code 03h, 06h, 10h, 16h



Übersicht

Mit folgenden Funktionscodes können Sie von einem Modbus-Master auf einen Slave zugreifen. Die Beschreibung erfolgt immer aus Sicht des Masters:

Code	Befehl	Beschreibung
01h	Read n Bits	n Bit lesen von Master-Ausgabe-Bereich 0x
02h	Read n Bits	n Bit lesen von Master-Eingabe-Bereich 1x
03h	Read n Words	n Worte lesen von Master-Ausgabe-Bereich 4x
04h	Read n Words	n Worte lesen von Master-Eingabe-Bereich 3x
05h	Write 1 Bit	1 Bit schreiben in Master-Ausgabe-Bereich 0x
06h	Write 1 Word	1 Wort schreiben in Master-Ausgabe-Bereich 4x
0Fh	Write n Bits	n Bit schreiben in Master-Ausgabe-Bereich 0x
10h	Write n Words	n Worte schreiben in Master-Ausgabe-Bereich 4x
16h	Mask 1 Word	1 Wort in Master-Ausgabe-Bereich 4x maskieren

Byte-Reihenfolge im Wort



Antwort des Kopplers

Liefert der Slave einen Fehler zurück, so wird der Funktionscode mit 80h "verodert" zurückgesendet. Ist kein Fehler aufgetreten, wird der Funktionscode zurückgeliefert.

Slave-Antwort:	Funktionscode OR 80h	→ Fehler & Fehlernummer
	Funktionscode	→ OK

Zusätzlich erhalten Sie im Fehlerfall in einem weiteren Byte eine Fehlernummer. Hier gibt es folgende Fehlernummern:

01h: Funktionsnummer wird nicht unterstützt

02h: Adressierung fehlerhaft

03h: Daten fehlerhaft

04h: System SLIO Bus ist nicht initialisiert

07h: Allgemeiner Fehler

### Read n Bits 01h, 02h

Code 01h: n Bit lesen von Master-Ausgabe-Bereich 0x.

Code 02h: n Bit lesen von Master-Eingabe-Bereich 1x.

### Kommandotelegramm

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits
x	x	0	0	0	6				
6Byte						1Byte	1Byte	1Wort	1Wort

### Antworttelegramm

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Anzahl gelesene Bytes	Daten 1. Byte	Daten 2. Byte	...
x	x	0	0	0							
6Byte						1Byte	1Byte	1Byte	1Byte	1Byte	
									max. 252Byte		

### Read n Words 03h, 04h

03h: n Worte lesen von Master-Ausgabe-Bereich 4x.

04h: n Worte lesen von Master-Eingabe-Bereich 3x.

### Kommandotelegramm

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse Wort	Anzahl der Worte
x	x	0	0	0	6				
6Byte						1Byte	1Byte	1Wort	1Wort

### Antworttelegramm

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Anzahl gelesene Bytes	Daten 1. Wort	Daten 2. Wort	...
x	x	0	0	0							
6Byte						1Byte	1Byte	1Byte	1Wort	1Wort	
									max. 126Worte		

**Write 1 Bit 05h**

Code 05h: 1 Bit schreiben in Master-Ausgabe-Bereich 0x.

Eine Zustandsänderung erfolgt unter "Zustand Bit" mit folgenden Werten:

"Zustand Bit" = 0000h → Bit = 0

"Zustand Bit" = FF00h → Bit = 1

**Kommandotelegramm**

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit
x	x	0	0	0	6				
6Byte						1Byte	1Byte	1Wort	1Wort

**Antworttelegramm**

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit
x	x	0	0	0	6				
6Byte						1Byte	1Byte	1Wort	1Wort

**Write 1 Word 06h**

Code 06h: 1 Wort schreiben in Master-Ausgabe-Bereich 4x.

**Kommandotelegramm**

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort
x	x	0	0	0	6				
6Byte						1Byte	1Byte	1Wort	1Wort

**Antworttelegramm**

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort
x	x	0	0	0	6				
6Byte						1Byte	1Byte	1Wort	1Wort

**Write n Bits 0Fh**

Code 0Fh: n Bit schreiben in Master-Ausgabe-Bereich 0x

Bitte beachten Sie, dass die Anzahl der Bits zusätzlich in Byte anzugeben sind.

**Kommandotelegramm**

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl Bits	Anzahl Bytes	Daten 1. Byte	Daten 2. Byte	...
x	x	0	0	0									
6Byte						1Byte	1Byte	1Wort	1Wort	1Byte	1Byte	1Byte	1Byte
											max. 248Byte		

**Antworttelegramm**

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse 1.Bit	Anzahl Bits
x	x	0	0	0	6				
6Byte						1Byte	1Byte	1Wort	1Wort

**Write n Words 10h**

Code 10h: n Worte schreiben in Master-Ausgabe-Bereich.

**Kommandotelegramm**

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl Worte	Anzahl Bytes	Daten 1. Wort	Daten 2. Wort	...
x	x	0	0	0									
6Byte						1Byte	1Byte	1Wort	1Wort	1Wort	1Wort	1Wort	1Wort
											max. 124Byte		

**Antworttelegramm**

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl Worte
x	x	0	0	0	6				
6Byte						1Byte	1Byte	1Wort	1Wort

**Mask 1 Word 16h**

Code 16h: Mit dieser Funktion können Sie ein Wort im Master-Ausgabe-Bereich 4x maskieren.

**Kommandotelegramm**

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse Wort	AND Mask	OR Mask
x	x	0	0	0	8					
6Byte						1Byte	1Byte	1Wort	1Wort	1Wort

Antworttelegramm

Modbus/TCP-Header						Slave-Adresse	Funktions-Code	Adresse Wort	AND Mask	OR Mask
x	x	0	0	0	8					
6Byte						1Byte	1Byte	1Wort	1Wort	1Wort