

SIEMENS

Ingenuity for life

Industry Online Support

Home

Modbus/TCP mit den Anweisungen MB_CLIENT und MB_SERVER

S7-1500 CPU und S7-1200 CPU

<https://support.industry.siemens.com/cs/ww/de/view/102020340>

Siemens
Industry
Online
Support



Dieser Beitrag stammt aus dem Siemens Industry Online Support. Es gelten die dort genannten Nutzungsbedingungen (www.siemens.com/nutzungsbedingungen).

Security-hinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z.B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter <http://www.siemens.com/industrialsecurity>.

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter <http://www.siemens.com/industrialsecurity>.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 4 |
| 2 | Anwenderprogramm der S7-1500 CPU | 6 |
| 2.1 | S7-1500: Modbus TCP-Client | 7 |
| 2.1.1 | FB1 "ModbusClient" | 7 |
| 2.1.2 | Datenstruktur am Parameter "connectParamClient" | 8 |
| 2.1.3 | Parameter "dataBuffer" | 9 |
| 2.2 | S7-1500: Modbus TCP-Server | 10 |
| 2.2.1 | FB2 "ModbusServer" | 10 |
| 2.2.2 | Datenstruktur am Parameter "connectParamServer" | 11 |
| 2.2.3 | Parameter "dataBuffer" | 12 |
| 3 | Anwenderprogramm der S7-1200 CPU | 13 |
| 3.1 | S7-1200: Modbus TCP-Server | 14 |
| 3.1.1 | FB2 "ModbusServer" | 14 |
| 3.1.2 | Datenstruktur am Parameter "connectParamServer" | 15 |
| 3.1.3 | Parameter "dataBuffer" | 15 |
| 3.2 | S7-1200: Modbus TCP-Client | 16 |
| 3.2.1 | FB1 "ModbusClient" | 16 |
| 3.2.2 | Datenstruktur am Parameter "connectParamClient" | 17 |
| 3.2.3 | Parameter "dataBuffer" | 18 |
| 4 | Ein- und Ausgänge der FBs "ModbusClient" und "ModbusServer" | 19 |
| 4.1 | Ein- und Ausgänge des FB "ModbusClient" | 19 |
| 4.2 | Ein- und Ausgänge des FB "ModbusServer" | 21 |
| 4.3 | Parameter modbusMode und ModbusDataAddress | 23 |
| 5 | Aufbau der Struktur "TCON_IP_v4" | 25 |

1 Einleitung

Vorgeführt wird die Modbus-TCP-Kommunikation zwischen S7-1500 CPU und S7-1200 CPU.

Im Anwenderprogramm der S7-1200 CPU und S7-1500 CPU werden die Anweisungen "MB_CLIENT" und "MB_SERVER" aufgerufen und parametrisiert.

Die Anweisung "MB_CLIENT" kommuniziert als Modbus TCP-Client über die PROFINET-Schnittstelle der CPU. Zur Verwendung der Anweisung benötigen Sie kein zusätzliches Hardwaremodul. Über die Anweisung "MB_CLIENT" bauen Sie eine Verbindung zwischen den Client und Server auf, senden Anfragen und empfangen Antworten und steuern den Verbindungsabbau.

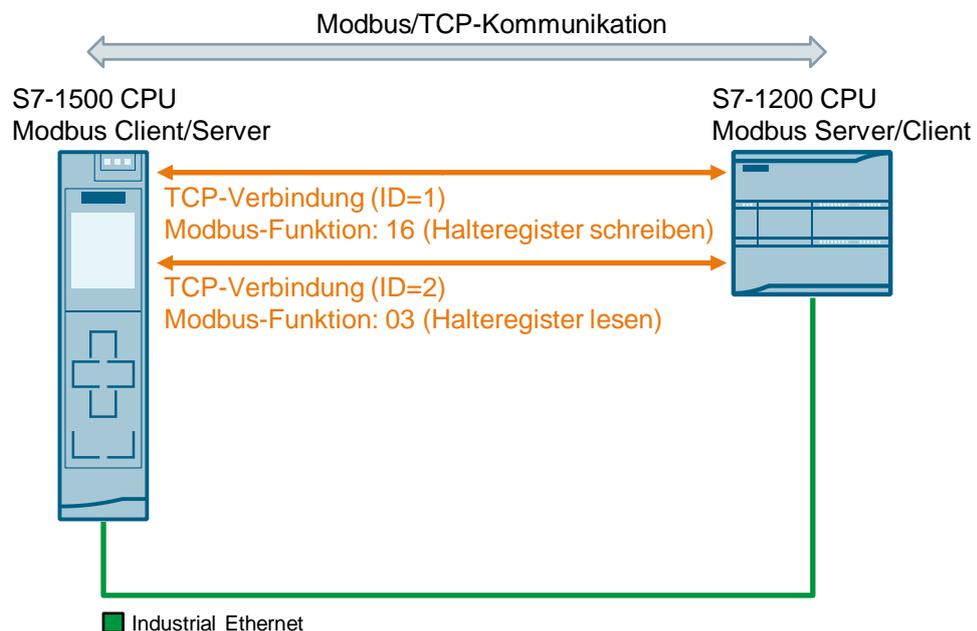
Die Anweisung "MB_SERVER" kommuniziert als Modbus TCP-Server über die PROFINET-Schnittstelle der CPU. Zur Verwendung der Anweisung benötigen Sie kein zusätzliches Hardwaremodul. Die Anweisung "MB_SERVER" verarbeitet Verbindungsanfragen eines Modbus TCP-Client, empfängt Anfragen von Modbus-Funktionen und sendet Antwort-Meldungen.

In diesem Beispiel werden zwei Modbus-Funktionen über zwei Modbus-TCP-Verbindungen demonstriert.

Die erste Verbindung baut die S7-1500 CPU als Modbus TCP-Client auf. Die S7-1200 CPU ist Modbus TCP-Server.

Die zweite Verbindung baut die S7-1200 CPU als Modbus TCP-Client auf. Die S7-1500 CPU ist Modbus TCP-Server.

Abbildung 1-1



Die Modbus-TCP-Verbindungen werden jeweils über ein Modbus-Anweisungspärchen (MB_CLIENT und MB_SERVER) aufgebaut.

Modbus-Funktion 16 (Halteregister schreiben)

[Tabelle 1-1](#) zeigt die Parametrierung der Modbus TCP-Verbindung und die Zuordnung der Anweisungspärchen für die Modbus-Funktion 16 (Halteregister schreiben).

Tabelle 1-1

| Parameter | S7-1500 | S7-1200 |
|---|---------------------------------------|---|
| Anweisung | MB_CLIENT | MB_SERVER |
| Modbus-Funktion | 16 (Halteregister schreiben) | |
| Verbindungsnummer (ID) | 1 | |
| Verbindungstyp | 0x0B (hex) = 11 (dez): TCP-Verbindung | |
| Verbindungsaufbau | aktiv | passiv |
| eigene IP-Adresse | 192.168.0.3 | 192.168.0.2 |
| IP-Adresse des entfernten Partner (remote IP-Adresse) | 192.168.0.2 | 192.168.0.3 |
| lokaler Port | 0: beliebiger Port | 502 |
| remoter Port | 502 | 0: Die Anweisung "MB_SERVER" soll Verbindungsanfragen von jedem entfernten Partner akzeptieren. |

Modbus-Funktion 3 (Halteregister lesen)

[Tabelle 1-2](#) zeigt die Parametrierung der Modbus TCP-Verbindung und die Zuordnung der Anweisungspärchen für die Modbus-Funktion 3 (Halteregister lesen).

Tabelle 1-2

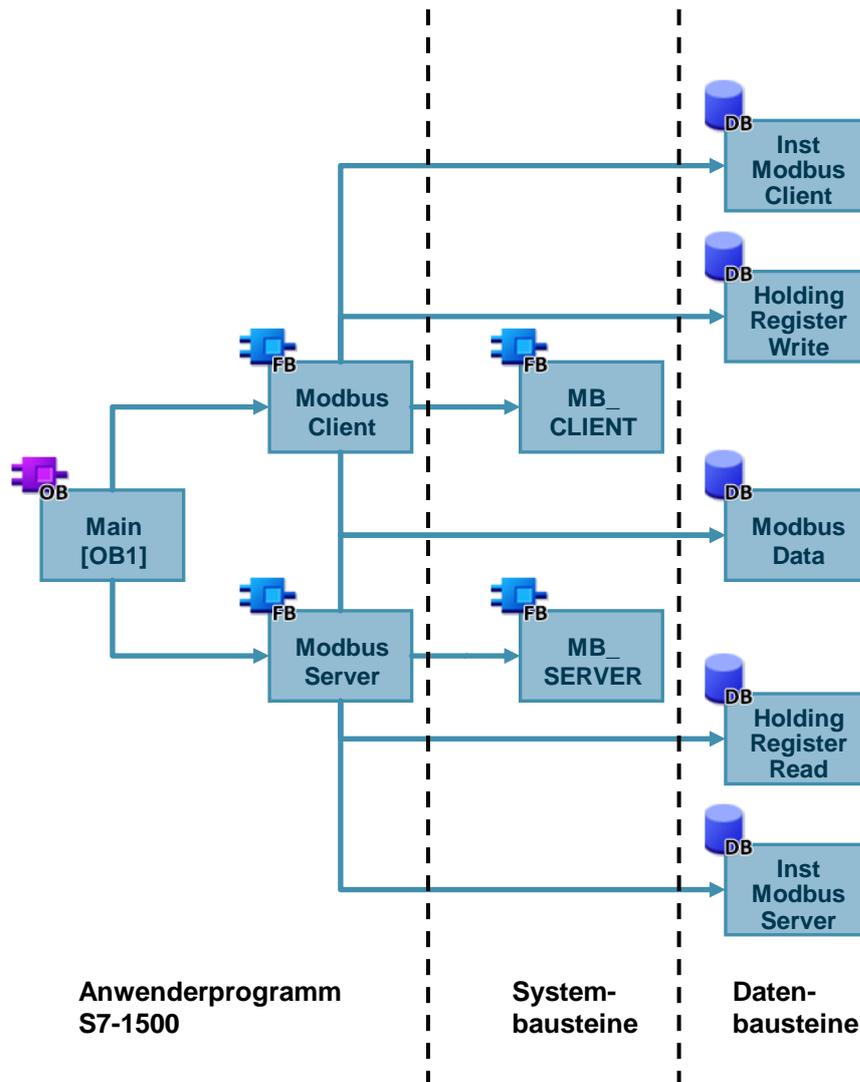
| Parameter | S7-1500 | S7-1200 |
|---|---|--------------------|
| Anweisung | MB_SERVER | MB_CLIENT |
| Modbus-Funktion | 3 (Halteregister lesen) | |
| Verbindungsnummer (ID) | 2 | |
| Verbindungstyp | 0x0B (hex) = 11 (dez): TCP-Verbindung | |
| Verbindungsaufbau | passiv | aktiv |
| eigene IP-Adresse | 192.168.0.3 | 192.168.0.2 |
| IP-Adresse des entfernten Partner (remote IP-Adresse) | 192.168.0.2 | 192.168.0.3 |
| lokaler Port | 503 | 0: beliebiger Port |
| remoter Port | 0: Die Anweisung "MB_SERVER" soll Verbindungsanfragen von jedem entfernten Partner akzeptieren. | 503 |

2 Anwenderprogramm der S7-1500 CPU

Übersicht

Im Anwenderprogramm der S7-1500 CPU wird die Anweisung "MB_CLIENT" und "MB_SERVER" für jede Modbus-TCP-Verbindung mit einer eindeutigen ID und einer eigenen Instanz aufgerufen. Der Aufruf der Anweisung "MB_CLIENT" und "MB_SERVER" erfolgt jeweils in einem eigenen Funktionsbaustein.

Abbildung 2-1



Als Modbus TCP-Client baut die S7-1500 CPU die Verbindung zum Modbus TCP-Server (S7-1200 CPU) auf und sendet die Anfrage zum Schreiben des Haltereisters.

Tabelle 2-1

| ID | Aufruf der Anweisung "MB_CLIENT" | Instanz-DB des FB "ModbusClient" | Beschreibung |
|----|----------------------------------|----------------------------------|---|
| 1 | FB1 "ModbusClient" | DB1 "InstModbusClient" | Modbus-Funktion 16 (Haltereister schreiben) |

Als Modbus TCP-Server verarbeitet die S7-1500 CPU die Verbindungsanfrage des Modbus TCP-Client (S7-1200 CPU) und empfängt die Anfrage zum Lesen des Haltereisters.

Tabelle 2-2

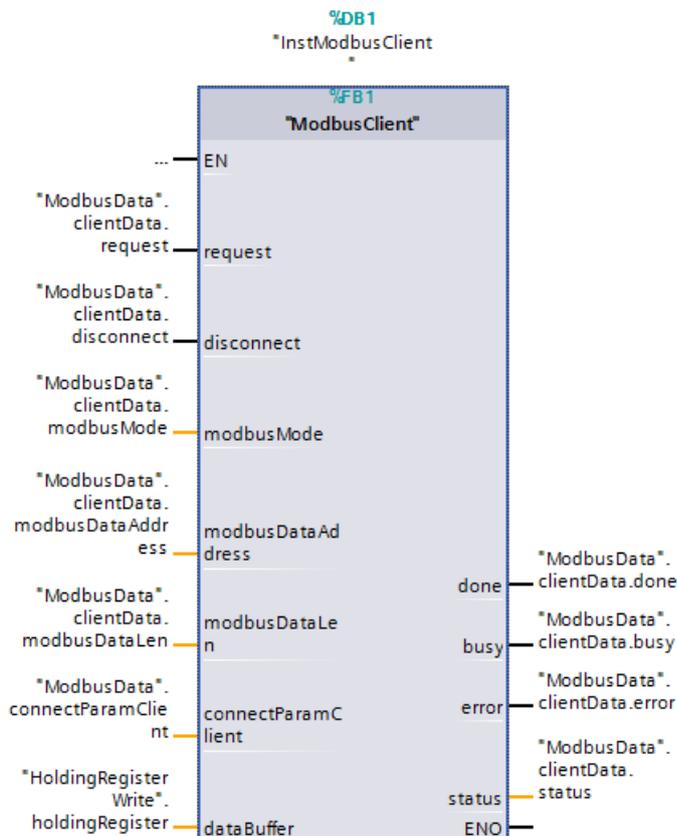
| ID | Aufruf der Anweisung "MB_SERVER" | Instanz-DB des FB "ModbusServer" | Beschreibung |
|----|----------------------------------|----------------------------------|--|
| 2 | FB2 "ModbusServer" | DB2 "InstModbusServer" | Modbus-Funktion 3 (Haltereister lesen) |

2.1 S7-1500: Modbus TCP-Client

2.1.1 FB1 "ModbusClient"

Der Funktionsbaustein FB1 "ModbusClient" wird zyklisch im OB1 aufgerufen.

Abbildung 2-2



Der FB1 "ModbusClient" ruft intern die Anweisung "MB_CLIENT" auf, um die Modbus-TCP-Verbindung mit der ID=1 aufzubauen und das Halteregeister in den Modbus TCP-Server zu schreiben.

Die Kommunikationsanfrage zum Schreiben des Halteregeisters wird über die Variable "ModbusData".clientData.request am Eingang "request" gesteuert.

In diesem Beispiel wird die Modbus-TCP-Verbindung mit der Verbindungsnummer=1 zum Port 502 des Modbus TCP-Server aufgebaut. Der Modbus-TCP-Server hat die IP-Adresse 192.168.0.2.

Es werden 10 Halteregeister auf der remoten Adresse 0 geschrieben. Dafür parametrieren Sie die Eingangsparameter "modbusMode", "modbusDataAddress" und "modbusDataLen" wie folgt:

- modbusMode = 116
- modbusDataAddress = 0
- modbusDataLen = 10

2.1.2 Datenstruktur am Parameter "connectParamClient"

Verwenden Sie für programmierte Verbindungen am Eingang "connectParamClient" folgende Struktur zur Verbindungsbeschreibung nach TCON_IP_v4.

- Achten Sie darauf, in der Struktur TCON_IP_v4 nur Verbindungen vom Typ TCP anzugeben.
- Die Verbindung darf die folgenden TCP-Port-Nummern nicht verwenden: 20, 21, 25, 80, 102, 123, 5001, 34962, 34963 und 34964.

Die folgende Abbildung zeigt den Aufbau der Struktur TCON_IP_v4 mit dem Namen "connectParamClient". Diese Struktur geben Sie am Parameter "connectParamClient" des FB "ModbusClient" an. Eine Beschreibung der Parameter der Struktur "TCON_IP_v4" finden Sie in Kapitel [5](#).

Abbildung 2-3

| | | |
|------------------------|---------------------|-----|
| ■ ▼ connectParamClient | TCON_IP_v4 | |
| ■ InterfaceId | HW_ANY | 64 |
| ■ ID | CONN_OUC | 1 |
| ■ ConnectionType | Byte | 11 |
| ■ ActiveEstablished | Bool | 1 |
| ■ ▼ RemoteAddress | IP_V4 | |
| ■ ▼ ADDR | Array[1..4] of Byte | |
| ■ ADDR[1] | Byte | 192 |
| ■ ADDR[2] | Byte | 168 |
| ■ ADDR[3] | Byte | 0 |
| ■ ADDR[4] | Byte | 2 |
| ■ RemotePort | UInt | 502 |
| ■ LocalPort | UInt | 0 |

2.1.3 Parameter "dataBuffer"

Am Parameter "dataBuffer" geben Sie den Datenbereich an, in dem die Daten gespeichert sind, die zum Modbus TCP-Server gesendet werden. Die Daten, die in das Haltereister des Modbus TCP-Server geschrieben werden, sind im Datenbaustein DB3 "HoldingRegisterWrite" gespeichert.

Tabelle 2-3

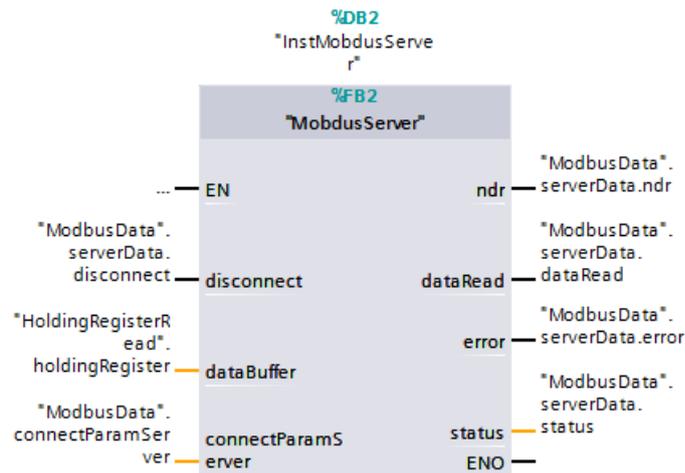
| Variablenname | Datentyp | Hinweis |
|-----------------|----------------------------|--|
| holdingRegister | Array [0 .. 65535] of Word | Entspricht dem Gesamtadressbereich des Haltereisters (0 bis 65535) |

2.2 S7-1500: Modbus TCP-Server

2.2.1 FB2 "ModbusServer"

Der FB2 "ModbusServer" wird zyklisch im OB1 aufgerufen.

Abbildung 2-4



Der Funktionsbaustein FB2 "ModbusServer" ruft intern die Anweisung "MB_SERVER" auf, um die Verbindungsanfrage zum Lesen des Haltereisters zu verarbeiten. Die Verbindungsanfrage erfolgt über die Modbus-TCP-Verbindung mit der ID=2 und den Port 503.

2.2.2 Datenstruktur am Parameter "connectParamServer"

Verwenden Sie für programmierte Verbindungen am Eingang "connectParamServer" folgende Struktur zur Verbindungsbeschreibung nach TCON_IP_v4.

- Achten Sie darauf, in der Struktur TCON_IP_v4 nur Verbindungen vom Typ TCP anzugeben.
- Die Verbindung darf die folgenden TCP-Port-Nummern nicht verwenden: 20, 21, 25, 80, 102, 123, 5001, 34962, 34963 und 34964.

Die folgende Abbildung zeigt den Aufbau der Struktur TCON_IP_v4 mit dem Namen "connectParamServer". Diese Struktur geben Sie am Parameter "connectParamServer" des FB "ModbusServer" an. Eine Beschreibung der Parameter der Struktur "TCON_IP_v4" finden Sie in Kapitel [5](#).

Abbildung 2-5

| | | |
|----------------------|---------------------|-----|
| ▼ connectParamServer | TCON_IP_v4 | |
| ■ InterfaceId | HW_ANY | 64 |
| ■ ID | CONN_OUC | 2 |
| ■ ConnectionType | Byte | 11 |
| ■ ActiveEstablished | Bool | 0 |
| ■ ▼ RemoteAddress | IP_V4 | |
| ■ ▼ ADDR | Array[1..4] of B... | |
| ■ ADDR[1] | Byte | 192 |
| ■ ADDR[2] | Byte | 168 |
| ■ ADDR[3] | Byte | 0 |
| ■ ADDR[4] | Byte | 2 |
| ■ RemotePort | UInt | 0 |
| ■ LocalPort | UInt | 503 |

2.2.3 Parameter "dataBuffer"

Der Parameter "dataBuffer" ist ein Zeiger auf einen Datenpuffer zur Ablage der Daten, die von oder auf den Modbus-Server gelesen oder geschrieben wurden. Als Speicherbereich können Sie einen globalen Datenbaustein oder einen Merker verwenden.

Die Daten, die gelesen werden, sind im Datenbaustein DB4 "HoldingRegisterRead" gespeichert.

Tabelle 2-4

| Variablenname | Datentyp | Hinweis |
|-----------------|----------------------------|--|
| holdingRegister | Array [0 .. 65535] of Word | Entspricht dem Gesamtadressbereich des Haltereisters (0 bis 65535) |

Die folgende Tabelle zeigt, wie die Modbus-Adressen auf das Haltereister für die Modbus-Funktion 3 (Word lesen) abgebildet werden.

Tabelle 2-5

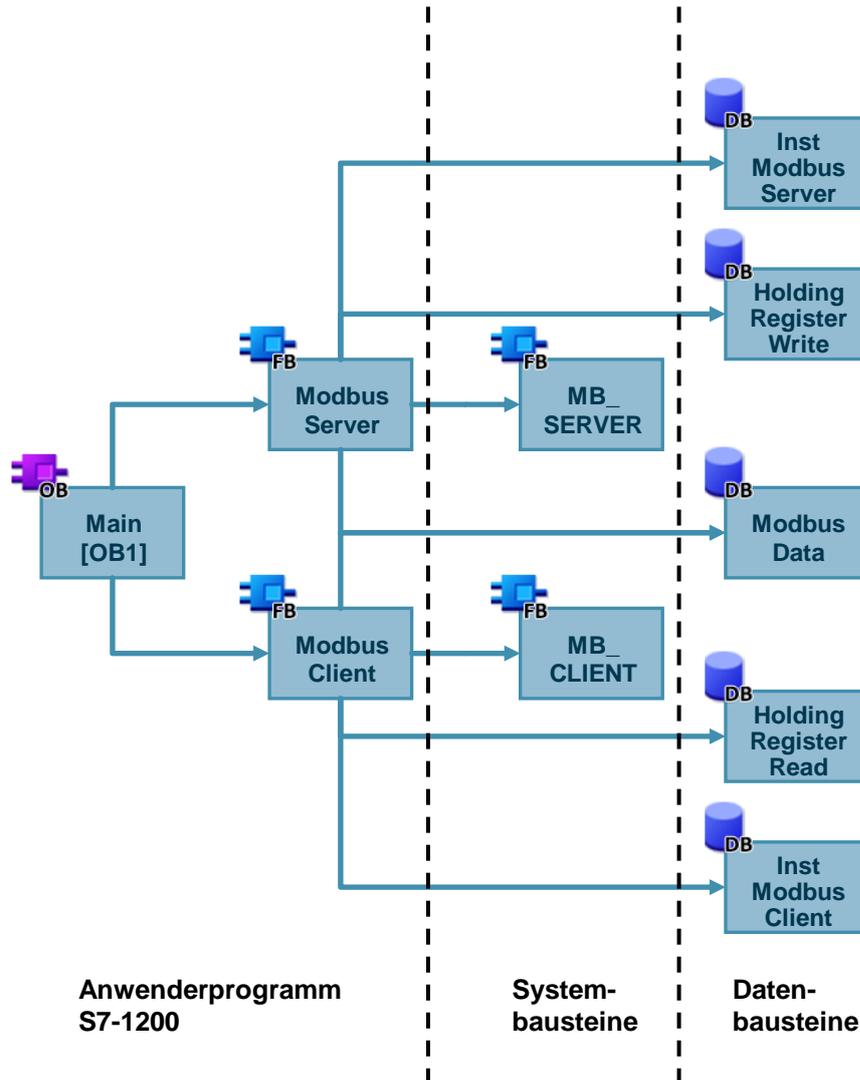
| Modbus-Adresse | dataBuffer |
|----------------|--|
| 0 | "holdingRegisterRead".holdingRegister[0] |
| 1 | "holdingRegisterRead".holdingRegister[1] |
| 2 | "holdingRegisterRead".holdingRegister[2] |
| 3 | "holdingRegisterRead".holdingRegister[3] |
| 4 | "holdingRegisterRead".holdingRegister[4] |
| 5 | "holdingRegisterRead".holdingRegister[5] |
| 6 | "holdingRegisterRead".holdingRegister[6] |
| 7 | "holdingRegisterRead".holdingRegister[7] |
| 8 | "holdingRegisterRead".holdingRegister[8] |
| 9 | "holdingRegisterRead".holdingRegister[9] |

3 Anwenderprogramm der S7-1200 CPU

Übersicht

Im Anwenderprogramm der S7-1200 CPU wird die Anweisung "MB_CLIENT" und "MB_SERVER" für jede Modbus-TCP-Verbindung mit einer eindeutigen ID und einem eigenen Instanz- Datenbaustein aufgerufen. Der Aufruf der Anweisung "MB_CLIENT" und "MB_SERVER" erfolgt jeweils in einer eigenen Funktion.

Abbildung 3-1



Als Modbus TCP-Server verarbeitet die S7-1200 CPU die Verbindungsanfrage des Modbus TCP-Client (S7-1500 CPU) und empfängt die Anfrage zum Schreiben des Haltereisters.

Tabelle 3-1

| ID | Aufruf der Anweisung "MB_SERVER" | Instanz-DB des FB "ModbusServer" | Beschreibung |
|----|----------------------------------|----------------------------------|---|
| 1 | FB2 "ModbusServer" | DB2 "InstModbusServer" | Modbus-Funktion 16 (Haltereister schreiben) |

Als Modbus TCP-Client baut die die S7-1200 CPU die Verbindung zum Modbus TCP-Server (S7-1500 CPU) auf und sendet die Anfrage zum Lesen des Haltereisters.

Tabelle 3-2

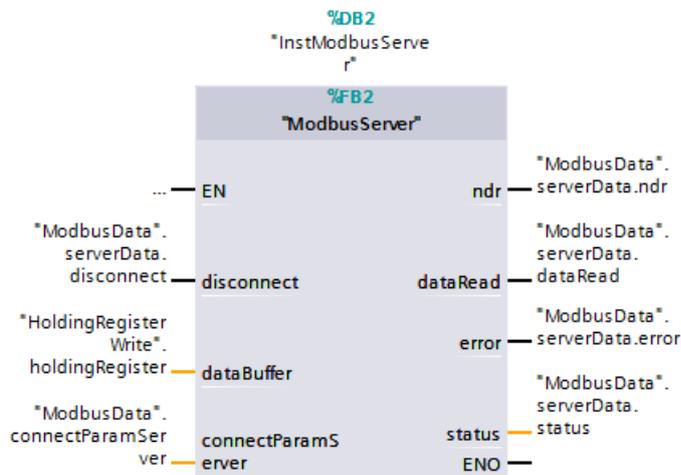
| ID | Aufruf der Anweisung "MB_CLIENT" | Instanz-DB des FB "ModbusClient" | Beschreibung |
|----|----------------------------------|----------------------------------|--|
| 2 | FB1 "ModbusClient" | DB1 "InstModbusClient" | Modbus-Funktion 3 (Haltereister lesen) |

3.1 S7-1200: Modbus TCP-Server

3.1.1 FB2 "ModbusServer"

Der FB2 "ModbusServer" wird zyklisch im OB1 aufgerufen.

Abbildung 3-2



Der Funktionsbaustein FB2 "ModbusServer" ruft intern die Anweisung "MB_SERVER" auf, um die Verbindungsanfrage zum Schreiben des Haltereisters zu verarbeiten. Die Verbindungsanfrage erfolgt über die Modbus-TCP-Verbindung mit der ID=1 und den Port 502.

3.1.2 Datenstruktur am Parameter "connectParamServer"

Verwenden Sie für programmierte Verbindungen am Eingang "connectParamServer" folgende Struktur zur Verbindungsbeschreibung nach TCON_IP_v4.

- Achten Sie darauf, in der Struktur TCON_IP_v4 nur Verbindungen vom Typ TCP anzugeben.
- Die Verbindung darf die folgenden TCP-Port-Nummern nicht verwenden: 20, 21, 25, 80, 102, 123, 5001, 34962, 34963 und 34964.

Die folgende Abbildung zeigt den Aufbau der Struktur TCON_IP_v4 mit dem Namen "connectParamServer". Diese Struktur geben Sie am Parameter "connectParamServer" des FB "ModbusServer" an. Eine Beschreibung der Parameter der Struktur "TCON_IP_v4" finden Sie in Kapitel [5](#).

Abbildung 3-3

| | | |
|--------------------|---------------------|-------|
| connectParamServer | TCON_IP_v4 | |
| InterfaceId | HW_ANY | 64 |
| ID | CONN_OUC | 16#1 |
| ConnectionType | Byte | 16#0B |
| ActiveEstablished | Bool | false |
| ▼ RemoteAddress | IP_V4 | |
| ■ ▼ ADDR | Array[1..4] of Byte | |
| ■ ADDR[1] | Byte | 192 |
| ■ ADDR[2] | Byte | 168 |
| ■ ADDR[3] | Byte | 0 |
| ■ ADDR[4] | Byte | 3 |
| RemotePort | UInt | 0 |
| LocalPort | UInt | 502 |

3.1.3 Parameter "dataBuffer"

Der Parameter "dataBuffer" ist ein Zeiger auf einen Datenpuffer zur Ablage der Daten, die von oder auf den Modbus-Server gelesen oder geschrieben wurden. Als Speicherbereich können Sie einen globalen Datenbaustein oder einen Merker verwenden.

Die Daten werden in den Datenbaustein DB3 "HoldingRegisterWrite" geschrieben und gespeichert.

Tabelle 3-3

| Variablenname | Datentyp | Hinweis |
|-----------------|---------------------------|---------|
| holdingRegister | Array [0 .. 4999] of Word | - |

Die folgende Tabelle zeigt, wie die Modbus-Adressen auf das Haltereister für die Modbus-Funktion 16 (Word schreiben) abgebildet werden.

Tabelle 3-4

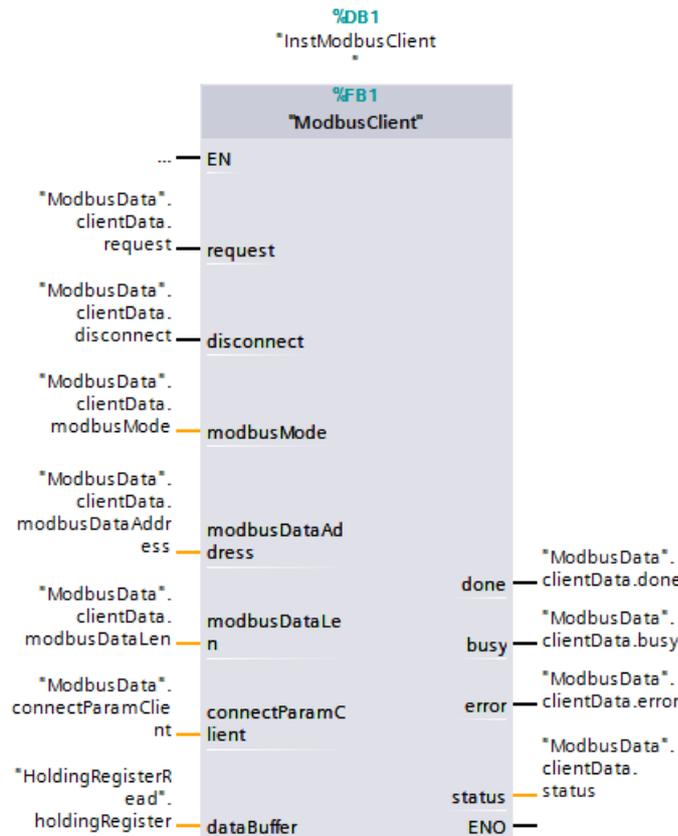
| Modbus-Adresse | dataBuffer |
|----------------|---|
| 0 | "holdingRegisterWrite".holdingRegister[0] |
| 1 | "holdingRegisterWrite".holdingRegister[1] |
| 2 | "holdingRegisterWrite".holdingRegister[2] |
| 3 | "holdingRegisterWrite".holdingRegister[3] |
| 4 | "holdingRegisterWrite".holdingRegister[4] |
| 5 | "holdingRegisterWrite".holdingRegister[5] |
| 6 | "holdingRegisterWrite".holdingRegister[6] |
| 7 | "holdingRegisterWrite".holdingRegister[7] |
| 8 | "holdingRegisterWrite".holdingRegister[8] |
| 9 | "holdingRegisterWrite".holdingRegister[9] |

3.2 S7-1200: Modbus TCP-Client

3.2.1 FB1 "ModbusClient"

Der FB1 "ModbusClient" wird zyklisch im OB1 aufgerufen.

Abbildung 3-4



Der FB1 "ModbusClient" ruft intern die Anweisung "MB_CLIENT" auf, um die Modbus-TCP-Verbindung mit der ID=2 aufzubauen und das Halteregeister vom Modbus TCP-Server zu lesen.

Die Kommunikationsanfrage zum Lesen des Halteregeisters wird über die Variable "ModbusData".clientData.request am Eingang "request" gesteuert.

In diesem Beispiel wird die Modbus-TCP-Verbindung mit der Verbindungsnummer=2 zum Port 505 des Modbus TCP-Server aufgebaut. Der Modbus-TCP-Server hat die IP-Adresse 192.168.0.3.

Es werden 10 Halteregeister auf der remoten Adresse 0 gelesen. Dafür parametrieren Sie die Eingangsparameter "modbusMode", "modbusDataAddress" und "modbusDataLen" wie folgt:

- modbusMode = 103
- modbusDataAddress = 0
- modbusDataLen = 10

3.2.2 Datenstruktur am Parameter "connectParamClient"

Verwenden Sie für programmierte Verbindungen am Eingang "connectParamClient" folgende Struktur zur Verbindungsbeschreibung nach TCON_IP_v4.

- Achten Sie darauf, in der Struktur TCON_IP_v4 nur Verbindungen vom Typ TCP anzugeben.
- Die Verbindung darf die folgenden TCP-Port-Nummern nicht verwenden: 20, 21, 25, 80, 102, 123, 5001, 34962, 34963 und 34964.

Die folgende Abbildung zeigt den Aufbau der Struktur TCON_IP_v4 mit dem Namen "connectParamClient". Diese Struktur geben Sie am Parameter "connectParamClient" des FB "ModbusClient" an. Eine Beschreibung der Parameter der Struktur "TCON_IP_v4" finden Sie in Kapitel [5](#).

Abbildung 3-5

| | | |
|--------------------|---------------------|-------|
| connectParamClient | TCON_IP_v4 | |
| InterfaceId | HW_ANY | 64 |
| ID | CONN_OUC | 16#2 |
| ConnectionType | Byte | 16#0B |
| ActiveEstablished | Bool | true |
| RemoteAddress | IP_V4 | |
| ADDR | Array[1..4] of Byte | |
| ADDR[1] | Byte | 192 |
| ADDR[2] | Byte | 168 |
| ADDR[3] | Byte | 0 |
| ADDR[4] | Byte | 3 |
| RemotePort | UInt | 503 |
| LocalPort | UInt | 0 |

3.2.3 Parameter "dataBuffer"

Am Parameter "dataBuffer" geben Sie den Datenbereich an, in dem die Daten gespeichert sind, die vom Modbus TCP-Server empfangen werden. Die Daten, die vom Haltereister des Modbus TCP-Server gelesen werden, werden im Datenbaustein DB4 "HoldingRegisterRead" gespeichert.

Tabelle 3-5

| Variablenname | Datentyp | Hinweis |
|-----------------|---------------------------|---------|
| holdingRegister | Array [0 .. 4999] of Word | - |

4 Ein- und Ausgänge der FBs "ModbusClient" und "ModbusServer"

4.1 Ein- und Ausgänge des FB "ModbusClient"

Eingänge

Die folgende Tabelle zeigt die Eingänge des Funktionsbausteins FB1 "ModbusClient".

Tabelle 4-1

| Eingang | Datentyp | Beschreibung |
|-------------------|----------|---|
| request | Bool | Modbus-Anfrage an den Modbus-Server, z. B. Halteregister schreiben, Halteregister lesen Der Parameter "request" ist Pegelgesteuert, d.h. solange der Eingang gesetzt ist, sendet die Anweisung Kommunikationsanfragen. |
| disconnect | Bool | Über den Parameter steuern Sie den Verbindungsaufbau- und abbau zum Modbus-Server <ul style="list-style-type: none"> 0: Kommunikationsverbindung zum konfigurierten Verbindungspartner aufbauen 1: Kommunikationsverbindung trennen. Während des Verbindungsabbaus wird keine andere Funktion ausgeführt. Nach dem erfolgreichen Abbau der Verbindung wird am Parameter "status" der Wert 0x0003 ausgegeben. Ist der Parameter "request" beim Verbindungsaufbau gesetzt, wird die Modbus-Anfrage sofort gesendet. |
| modbusMode | USint | Auswahl des Modbus der Modbus-Anfrage (Lesen, Schreiben oder Diagnose) Im Kapitel 4.3 finden Sie eine ausführliche Beschreibung des Parameters "modbusMode". |
| modbusDataAddress | UDint | Anfangsadresse der Daten, auf welche die Anweisung "MB_CLIENT" zugreift. Im Kapitel 4.3 finden Sie eine ausführliche Beschreibung des Parameters "modbusDataAddress". |
| modbusDataLen | UInt | Datenlänge: Anzahl der Bits oder Wörter für den Datenzugriff |
| dataBuffer | Variant | Zeiger auf einen Datenpuffer für die vom Modbus TCP-Server zu empfangenden oder zum Modbus TCP-Server zu sendenden Daten. In diesem Beispiel verweist der Zeiger auf einen globalen Datenbaustein (DB) mit optimiertem Bausteinzugriff. <ul style="list-style-type: none"> S7-1500: siehe Kapitel 2.1.3 S7-1200: siehe Kapitel 3.2.3 |

4 Ein- und Ausgänge der FBs "ModbusClient" und "ModbusServer"

| Eingang | Datentyp | Beschreibung |
|---------------------|----------|---|
| connect ParamClient | Variant | <p>Zeiger auf die Struktur der Verbindungsbeschreibung.</p> <p>Die folgenden Strukturen (Systemdatentypen) können verwendet werden:</p> <ul style="list-style-type: none"> • TCON_IP_v4: Enthält alle Adressierungsparameter, die für den Aufbau einer programmierten Verbindung benötigt werden. Bei Verwendung von TCON_IP_v4 wird die Verbindung beim Aufruf der Anweisung "MB_CLIENT" hergestellt. • TCON_Configured: Enthält die Adressierungsparameter einer konfigurierten Verbindung. Bei Verwendung von TCON_Configured wird eine vorhandene Verbindung genutzt, die nach dem Laden der Hardware-Konfiguration durch die CPU hergestellt wurde. <p>In diesem Beispiel wird die Struktur TCON_IP_v4 verwendet. Im Kapitel 5 ist der Aufbau der Struktur TCON_IP_v4 beschrieben.</p> <ul style="list-style-type: none"> • S7-1500: Parametrierung der Datenstruktur siehe Kapitel 2.1.2 • S7-1200 Parametrierung der Datenstruktur siehe Kapitel 3.2.2 |

Ausgänge

Die folgende Tabelle zeigt die Ausgänge des Funktionsbausteins FB1 "ModbusClient".

Tabelle 4-2

| Ausgang | Datentyp | Beschreibung |
|---------|----------|--|
| done | Bool | Das Bit am Ausgang "done" wird auf "1" gesetzt, sobald der letzte Auftrag ohne Fehler abgeschlossen ist. |
| busy | Bool | <ul style="list-style-type: none"> • 0: keine Modbus-Anfrage in Bearbeitung • 1: Modbus-Anfrage wird bearbeitet |
| error | Bool | <ul style="list-style-type: none"> • 0: kein Fehler • 1: Fehler aufgetreten. Die Fehlerursache wird am Ausgang "status" angezeigt. |
| status | Word | Detaillierte Statusinformation der Anweisung MB_CLIENT. |

4.2 Ein- und Ausgänge des FB "ModbusServer"

Eingänge

Die folgende Tabelle zeigt die Eingänge des Funktionsbausteins FB2 "ModbusServer"

Tabelle 4-3

| Eingang | Datentyp | Beschreibung |
|--------------------|----------|---|
| disconnect | Bool | <p>Die Anweisung "MB_SERVER" geht eine passive Verbindung mit einer Partner-Baugruppe ein. Der Server reagiert auf eine Verbindungsanfrage von der IP-Adresse, die in der Datenstruktur "TCON_IP_v4" am Eingang "connectParamServer" eingegeben ist.</p> <p>Über den Parameter können Sie steuern, wann eine Verbindungsanfrage akzeptiert wird.</p> <ul style="list-style-type: none"> • 0: Wenn keine Kommunikationsverbindung besteht, wird eine passive Verbindung aufgebaut. • 1: Initialisierung des Verbindungsstatus. Ist der Eingang gesetzt, werden keine anderen Vorgänge ausgeführt. Nach erfolgreichem Verbindungsabbau wird am Ausgang "status" der Wert 0x003 ausgegeben. |
| connectParamServer | Bool | <p>Zeiger auf die Struktur der Verbindungsbeschreibung.</p> <p>Die folgenden Strukturen (Systemdatentypen) können verwendet werden:</p> <ul style="list-style-type: none"> • TCON_IP_v4: Enthält alle Adressierungsparameter, die für den Aufbau einer programmierten Verbindung benötigt werden. Bei Verwendung von TCO_IP_v4 wird die Verbindung beim Aufruf der Anweisung "MB_SERVER" hergestellt. • TCON_Configured: Enthält die Adressierungsparameter einer konfigurierten Verbindung. Bei Verwendung von TCON_Configured wird die Verbindung beim Laden der Hardware-Konfiguration durch die CPU hergestellt. <p>In diesem Beispiel wird die Struktur TCON_IP_v4 verwendet. Im Kapitel 5 ist der Aufbau der Struktur TCON_IP_v4 beschrieben.</p> <ul style="list-style-type: none"> • S7-1500: Parametrierung der Datenstruktur siehe Kapitel 2.2.2 • S7-1200 Parametrierung der Datenstruktur siehe Kapitel 3.1.2 |

4 Ein- und Ausgänge der FBs "ModbusClient" und "ModbusServer"

| Eingang | Datentyp | Beschreibung |
|------------|----------|---|
| dataBuffer | Variante | <p>Zeiger auf das Modbus-Halteregister der Anweisung "MB_SERVER".</p> <p>Der Parameter "dataBuffer" muss immer auf einen Speicherbereich verweisen, der größer als 2 Byte ist.</p> <p>Das Halteregister beinhaltet die Werte, auf die ein Modbus TCP-Client über die Modbus-Funktionen 3 (lesen), 6 (schreiben) und 16 (Mehrfachschreiben) zugreifen darf.</p> <p>Verwenden Sie als Speicherbereich einen globalen Datenbaustein (DB) mit optimiertem Zugriff oder den Speicherbereich der Merker.</p> <p>Tabelle 3-4 zeigt, wie die Modbus-Adressen auf das Halteregister für die Modbus-Funktion 16 (Word schreiben) abgebildet werden.</p> <p>Tabelle 2-5 zeigt, wie die Modbus-Adressen auf das Halteregister für die Modbus-Funktion 3 (Word lesen) abgebildet werden.</p> |

Ausgänge

Die folgende Tabelle zeigt die Ausgänge des Funktionsbausteins FB2 "ModbusServer".

Tabelle 4-4

| Ausgang | Datentyp | Beschreibung |
|----------|----------|--|
| ndr | Bool | <p>"new data ready"</p> <ul style="list-style-type: none"> 0: keine neue Daten 1: neue Daten durch den Modbus TCP-Client geschrieben |
| dataRead | Bool | <p>"data read"</p> <ul style="list-style-type: none"> 0: keine Daten gelesen 1: Daten durch den Modbus TCP-Client gelesen |
| error | Bool | <p>Tritt während eines Aufrufs der Anweisung "MB_SERVER" ein Fehler auf, wird der Ausgang am Parameter "error" auf "1" gesetzt.</p> <p>Detailinformationen zur Fehlerursache wird am Ausgang "status" angezeigt.</p> |
| status | Word | <p>Detaillierte Statusinformation der Anweisung "MB_SERVER"</p> |

4.3 Parameter modbusMode und ModbusDataAddress

Für die Werte 0 und 1 von "modbusMode" legt die Kombination der Parameter "modbusModus", "modbusDataAddress" und "modbusDataLen" den Modbus-Funktionscode fest, der in der aktuellen Modbus-Nachricht verwendet wird:

- "modbusMode" enthält die Information, ob gelesen oder geschrieben werden soll.
 - 0: Lesen
 - 1 Schreiben
- "modbusDataAddress" enthält die Information was gelesen bzw. geschrieben werden soll, sowie eine Adressinformation aus der die Anweisung "MB_CLIENT" die remote Adresse berechnet.
- "modbusDataLen" enthält die Anzahl der zu lesenden / schreibenden Werte.

Für die Werte 101 bis 116 von "modbusMode" gilt:

- "modbusMode" legt den Modbus-Funktionscode fest.
- "modbusDataAddress" enthält die remote Adresse.
- "modbusDataLen" enthält die Anzahl der zu lesenden / schreibenden Werte.

Die folgende Tabelle zeigt den Zusammenhang zwischen den Eingängen des Funktionsbausteins "ModbusClient" und der Modbus-Funktion.

Tabelle 4-5

| modbusMode | modbusDataAddress | modbusDataLen | Modbus-Funktion | Funktion und Datentyp |
|------------|----------------------------------|--|-----------------|---|
| 116 | Anfangsadresse: • 0 bis 65535 | Datenlänge (WORD) pro Aufruf: • 1 bis 123 | 16 | 1 bis 123 Haltereister auf der remoten Adresse 0 bis 65535 schreiben. |
| 103 | Anfangsadresse: 0 bis 65535 | Datenlänge (WORD) pro Aufruf: 1 bis 125 | 3 | 1 bis 125 Haltereister auf der remoten Adresse 0 bis 65535 lesen. |

4 Ein- und Ausgänge der FBs "ModbusClient" und "ModbusServer"

| modbusMode | modbusDataAddress | modbusDataLen | Modbus-Funktion | Funktion und Datentyp |
|---------------------|--|---|-----------------|---|
| Alternative: | | | | |
| 1 | Anfangsadresse: <ul style="list-style-type: none"> • 40001 bis 49999 • 400001 bis 465535 | Datenlänge (WORD) pro Aufruf: <ul style="list-style-type: none"> • 2 bis 123 | 16 | 2 bis 123 Halteregeister auf der remoten Adresse 0 bis 9998 schreiben. 2 bis 123 Halteregeister auf der remoten Adresse 0 bis 65534 schreiben. |
| 0 | Anfangsadresse: <ul style="list-style-type: none"> • 40001 bis 49999 • 400001 bis 465535 | Datenlänge (WORD) pro Aufruf: 1 bis 125 | 3 | 1 bis 125 Halteregeister auf der remoten Adresse 0 bis 9998 lesen. 1 bis 125 Halteregeister auf der remoten Adresse 0 bis 65534 lesen. |

Hinweis

Weitere Informationen zur Adressierung der Speicherbereiche in der SIMATIC S7-1200/S7-1500 bei einem Modbus/TCP-Datenaustausch finden Sie unter folgendem Link:

<https://support.industry.siemens.com/cs/ww/de/view/100633819>

5 Aufbau der Struktur "TCON_IP_v4"

Folgende Tabelle beschreibt die Parameter der Struktur "TCON_IP_v4".

Tabelle 5-1

| Byte | Parameter | Datentyp | Beschreibung |
|-----------|-------------------|----------------------|---|
| 0 bis 1 | InterfaceID | HW_ANY | Hardware-Kennung der lokalen Schnittstelle (Wertebereich: 0 bis 65535) Die Hardware-Kennung finden Sie in der Gerätekonfiguration der CPU. Markieren Sie die PROFINET-Schnittstelle, um im Inspektorfenster die Eigenschaften der PROFINET-Schnittstelle anzuzeigen. Im Register "Allgemein" navigieren zu "HW-Kennung", um die Hardware-Kennung zu ermitteln. |
| 2 bis 3 | ID | CONN_OUC | Referenz auf diese Verbindung (Wertebereich: 1 bis 4095) Über den Parameter wird eine Verbindung innerhalb der CPU eindeutig identifiziert. Jede einzelne Instanz der Anweisung "MB_CLIENT" sowie "MB_SERVER" muss eine eindeutige ID verwenden. |
| 4 | ConnectionType | BYTE | Verbindungstyp Wählen Sie 11 (dezimal) für TCP. Andere Verbindungstypen sind nicht zulässig. |
| 5 | ActiveEstablished | BOOL | Kennung für die Art des Verbindungsaufbaus. True: aktiver Verbindungsaufbau False: passiver Verbindungsaufbau |
| 6 bis 9 | RemoteAddress | ARRAY [1..4] of BYTE | IP-Adresse des entfernten Verbindungspartners |
| 10 bis 11 | RemotePort | UINT | Port-Nummer des entfernten Verbindungspartners (Wertebereich: 1 bis 49151). <ul style="list-style-type: none"> • MB_CLIENT: Verwenden Sie die IP-Port-Nummer des Servers, zu dem der Client die Verbindung herstellt und über das TCP/IP-Protokoll kommuniziert • MB_SERVER: Verwenden Sie die IP-Port-Nummer des Clients, von dem die Verbindungsanfrage akzeptiert werden soll. Soll die Anweisung "MB_SERVER" von jedem entfernten Partner akzeptieren, verwenden Sie als Port-Nummer "0". |
| 12 bis 13 | LocalPort | UINT | Port-Nummer des lokalen Verbindungspartners (Wertebereich: 1 bis 49151) <ul style="list-style-type: none"> • MB_CLIENT: <ul style="list-style-type: none"> - Port-Nummern: 1 bis 49151 - Beliebiger Port: 0 • MB_SERVER: Die Nummer der IP-Ports legt fest, welcher IP-Port für Verbindungsanfragen des Modbus-Clients überwacht wird. |