

SIMATIC Modbus/TCP

Kommunikation über die integrierte PN-Schnittstelle der CPU

Programmierhandbuch · 11/2017

SIMATIC

Answers for industry.

SIEMENS

SIMATIC Modbus/TCP Kommunikation über die integrierte PN-Schnittstelle der CPU

Programmierhandbuch

Vorwort

Produktbeschreibung

1

Getting Started

2

Inbetriebnahme

3

Parametrieren der Modbus-
Kommunikation

4

Lizenzierung

5

Funktionsbaustein
MODBUSPN

6

Additional Blocks

7

Diagnose

8

Applikationsbeispiele

9

Literatur

A

Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

GEFAHR

bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

WARNUNG

bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

VORSICHT

bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG

bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

WARNUNG

Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Vorwort

Zweck des Handbuchs

Die Informationen dieses Handbuchs ermöglichen es Ihnen, eine Kopplung zwischen einer CPU mit integrierter PN-Schnittstelle und einem Gerät, welches das Protokoll Modbus/TCP unterstützt, aufzubauen und in Betrieb zu nehmen.

Inhalte des Handbuchs

Im vorliegenden Handbuch sind die Funktion des Modbus Funktionsbausteins und dessen Parametrierung beschrieben.

Das Handbuch beinhaltet folgende Themen:

- Produktbeschreibung
- Getting Started
- Inbetriebnahme
- Parametrieren der Modbus-Kommunikation
- Lizenzierung
- Funktionsbaustein MODBUSPN
- Additional Blocks
- Diagnose
- Applikationsbeispiel

Gültigkeitsbereich des Handbuchs

Das vorliegende Handbuch ist gültig für folgende Software:

Produkt	Identifizierungsnummer	ab Version
MODBUS/TCP PN CPU	6AV6676-6MB20-3AX0 6AV6676-6MB20-3AD0	3.0
FB 900 "MODBUSPN"		4.0
FB 901 "MOD_CLI"		2.0
FB 903 "MOD_SERV"		2.0

Hinweis

Das vorliegende Handbuch enthält die Beschreibung der FBs, wie sie zum Zeitpunkt der Herausgabe des Handbuchs gültig ist.

Weiterführende Informationsquellen

Alle weiteren Informationen bezüglich der PN-CPUs und der IM 151-8 PN/DP CPU (Montage, Inbetriebnahme etc.) entnehmen Sie bitte den Handbüchern:

SIEMENS
SIMATIC S7-300
CPU 31xC und CPU 31x: Aufbauen
Betriebsanleitung
A5E00105491-07

SIEMENS
SIMATIC S7-300
CPU 31xC und CPU 31x, Technische Daten
Gerätehandbuch
A5E00105474-07

SIEMENS
SIMATIC S7-400
Automatisierungssystem S7-400: Aufbauen
Betriebsanleitung
A5E00850740-01

SIEMENS
SIMATIC S7-400
Automatisierungssystem S7-400 CPU-Daten
Gerätehandbuch
A5E00850745-06

SIEMENS
SIMATIC
Dezentrale Peripherie ET 200S
Interfacemodul IM151-8 PN/DP CPU
Betriebsanleitung
A5E02049033-01

SIEMENS
Produkt Information on
CPU314C-2 PN/DP, 6ES7314-6EH04-0AB0
CPU315-2 PN/DP, 6ES7315-2EH13-0AB0
CPU315F-2 PN/DP, 6ES7315-2FH13-0AB0
CPU317-2 PN/DP, 6ES7317-2EK13-0AB0
CPU317F-2 PN/DP, 6ES7317-2FK13-0AB0
CPU317-2 DP, 6ES7317-2AJ10-0AB0
CPU317F-2 DP, 6ES7317-6FF03-0AB0
CPU319-3 PN/DP, 6ES7318-3EL00-0AB0
CPU319F-3 PN/DP, 6ES7318-3FL00-0AB0
A5E01103134-03

Weitere Informationen bezüglich STEP 7 entnehmen Sie bitte den folgenden Handbüchern:

SIEMENS SIMATIC Software
Basissoftware für S7 und M7
STEP7 Benutzerhandbuch
C79000-G7000-C502-..

SIEMENS SIMATIC Software
Systemsoftware für S7-300/400
System- und Standardfunktionen
Referenzhandbuch
C79000-G7000-C503-02

Rückfragen

Bei Fragen zur Nutzung des in diesem Handbuch beschriebenen FBs, die Sie hier nicht beantwortet finden, wenden Sie sich bitte an Ihren Siemens-Ansprechpartner, von dem Sie diesen Funktionsbaustein erhalten haben.

Konventionen

In der vorliegenden Dokumentation wird im Folgenden die Bezeichnung PN-CPU verwendet. Die Ausführungen sind für die **PN-CPUs** der Reihe **314C, 315, 317, 319, 412, 414 und 416** sowie für die **IM 151-8 PN/DP CPU** gültig.

Einsatzbereich

Die in diesem Handbuch beschriebenen Funktionsbausteine stellen eine Verbindung zwischen einer PN-CPU und Modbusgeräten anderer Hersteller dar.

Inhaltsverzeichnis

Vorwort	5
1 Produktbeschreibung	11
1.1 Einsatzmöglichkeiten	11
1.2 Hard- und Softwarevoraussetzungen	12
2 Getting Started.....	13
3 Inbetriebnahme.....	15
3.1 Installieren der Bibliothek auf dem STEP 7-PG/PC.....	15
3.2 CPU - IP-Adresse zuweisen	16
3.3 Einfügen der Funktionsbausteine in das Programm.....	17
3.4 Mehrere Verbindungen auf Port 502	18
4 Parametrieren der Modbus-Kommunikation.....	19
4.1 Überblick	19
4.2 Parametrieren mit dem Wizard	20
4.3 Manuelle Parametrierung	21
5 Lizenzierung	31
6 Funktionsbaustein MODBUSPN.....	35
6.1 Funktionsweise des FB.....	35
6.2 Parameter des Funktionsbausteins MODBUSPN	38
6.3 Beispiel für die Adressabbildung	46
6.4 Vom FB verwendete Daten und Standardfunktionen	49
6.5 Umbenennen / Umverdrahten von Funktionen und Funktionsbausteinen	51
7 Additional Blocks.....	53
7.1 Unterstützung in CFC	53
7.2 Auftragsliste für zyklischen Telegrammverkehr	55
8 Diagnose	57
8.1 Diagnose über die Anzeigeelemente der CPU	58
8.2 Diagnosemeldungen des FB MODBUSPN.....	59
8.3 Diagnosemeldungen der eingebundenen Bausteine.....	65
8.4 Diagnosemeldungen des SFC24.....	66

9	Applikationsbeispiele	67
9.1	Beispielprojekt in AWL	68
9.2	Beispielprojekt in CFC	69
A	Literatur	71

Produktbeschreibung

1.1 Einsatzmöglichkeiten

Einordnen in die Systemumgebung

Der vorliegende Funktionsbaustein stellt ein Software-Produkt für CPUs mit integrierter PN-Schnittstelle der SIMATIC S7-300, S7-400 und IM 151-8 PN/DP CPU dar.

Funktion der FBs

Mit diesen Funktionsbausteinen wird eine Kommunikation zwischen einer S7-CPU mit integrierter PN-Schnittstelle und einem Gerät, welches das Protokoll Modbus/TCP unterstützt, ermöglicht.

Die Datenübertragung wird nach dem Client-Server-Prinzip abgewickelt.

Die SIMATIC S7 kann bei der Übertragung sowohl als Client als auch als Server betrieben werden.

Verwendung der Portnummer 502

Das Protokoll Modbus/TCP läuft üblicherweise über den Port 502. Diese Portnummer ist nur für PN-CPUs mit entsprechender Firmwareversion möglich. Die Information bezüglich der Freigabe der Portnummern finden Sie hier (<http://support.automation.siemens.com/WW/view/de/34010717>).

Bestimmte CPU-Typen können über den lokalen Port 502 Verbindungen zu mehreren Clients parallel halten und bedienen. Im Kapitel "Mehrere Verbindungen auf Port 502" sind die technischen Details zu diesem Thema erläutert.

1.2 Hard- und Softwarevoraussetzungen

Verwendbare Baugruppen für MODBUSPN

Die aktuellen Hardwarevoraussetzungen entnehmen Sie bitte hier (<http://support.automation.siemens.com/WW/view/de/104946406>).

Softwareausgabestände

Der Einsatz des FBs MODBUSPN ist ab **STEP7-Version 5.5** möglich.

Speicherbedarf

Der FB MODBUSPN benötigt ca. 8 kByte Arbeitsspeicher und ca. 9 kByte Ladespeicher.

Der FB MOD_CLI benötigt 9 kByte Arbeits- und Ladespeicher.

Der FB MOD_SERV benötigt 9 kByte Arbeits- und Ladespeicher.

Die exakten Längen der Bausteine können Sie über deren Eigenschaften im SIMATIC Manager ermitteln.

2

Getting Started

Vorgehensweise

1. Installation von "SIMATIC Modbus/TCP PN CPU" und Einfügen der Modbusbausteine in das Anwenderprojekt
=> Kapitel 3.1 (Seite 15)
2. Parametrierung des Parameter-DBs MODBUS_PARAM entsprechend den Anforderungen (IP-Adresse, Portnummer, Client/Server, Verbindungsauflaufbau bei Neustart, Modbusregister, DB-Bereiche etc.)
=> Kapitel 4 (Seite 19)
3. Aufruf und Parametrierung des Modbusbausteins MODBUSPN im notwendigen OB
=> Kapitel 6.1 (Seite 35) und 6.2 (Seite 38)
4. Laden des Anwenderprogramms in die CPU und Lizenzierung des Modbusbausteins für diese CPU
=> Kapitel 5 (Seite 31)

Inbetriebnahme

Allgemeines

Die im Folgenden verwendeten Angaben zu STEP 7 und zur Projektierung der Kommunikationsverbindungen beziehen sich auf die STEP7-Version 5.5.

Bei späteren Versionen können Abläufe, Namens- und Verzeichnisangaben geändert sein.

Voraussetzungen

STEP 7–Grundkenntnisse, AWL–Kenntnisse, SPS–Grundkenntnisse

3.1 Installieren der Bibliothek auf dem STEP 7-PG/PC

Lieferumfang

Die beiliegende CD enthält ein Setup, mit dem die Bibliothek "Modbus_PN_CPU", die Beispielprojekte und die Handbücher in Deutsch und Englisch in den entsprechenden STEP 7-Verzeichnissen installiert werden.

Zusätzlich befinden sich auf der CD die Handbücher im PDF-Format.

Voraussetzungen

Um die Installation durchführen zu können, muss vorher die Projektiersoftware STEP 7 V5.5 installiert worden sein.

Installation

Legen Sie die Modbus-CD in das CD-ROM-Laufwerk Ihres PGs/PCs ein. Wenn das Setupprogramm nicht automatisch startet, erfolgt die Installation folgendermaßen:

1. Wählen Sie im Windows Explorer das CD-ROM-Laufwerk, öffnen Sie das Verzeichnis Setup und starten Sie das Setupprogramm.
2. Befolgen Sie Schritt für Schritt die Anweisungen, die Ihnen das Installationsprogramm anzeigt.

Sie finden nun

- die Bibliothek in "\Program Files\Siemens\Step7\S7libs",
- die Beispielprojekte in "\Program Files\Siemens\Step7\Examples",
- das Handbuch in "\Program Files\Siemens\Step7\S7manual\S7Comm",
- das Software Registration Form in "\Program Files\Siemens\Step7\S7LIBS\Modbus_PN_CPU".

Das Handbuch kann auch über den Shortcut unter "\Program Files\Siemens\Dokumentation" geöffnet werden.

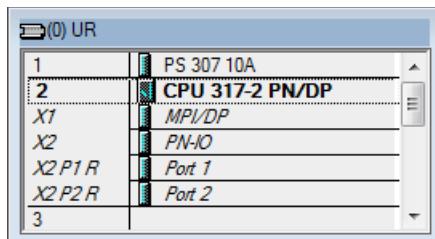
3.2 CPU - IP-Adresse zuweisen

In diesem Beispiel der Vergabe der IP-Adresse wird eine CPU 317-2PN/DP verwendet.

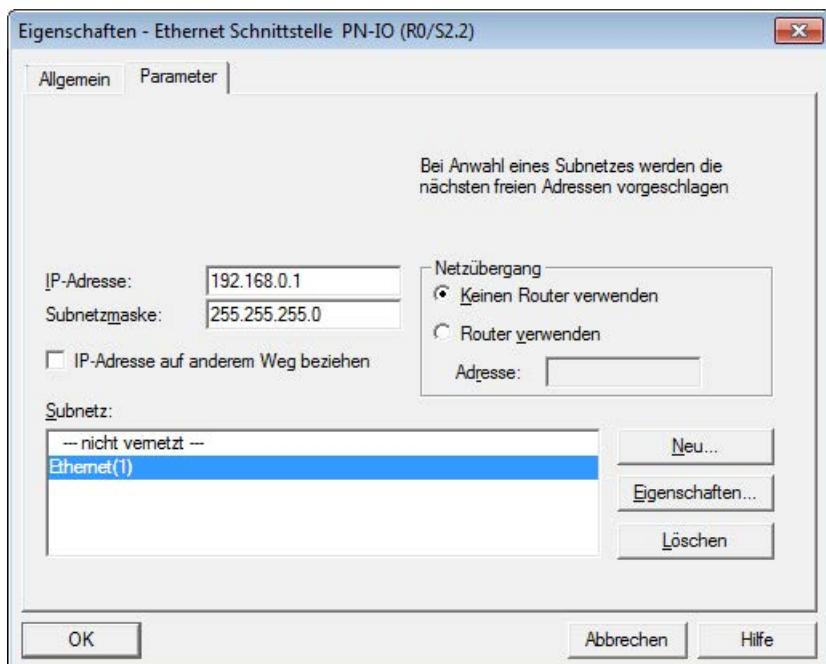
Vorgehensweise

Bevor Sie die Konfiguration vornehmen können, müssen Sie mit STEP 7 ein S7-Projekt angelegt haben.

1. Öffnen Sie HWKonfig und fügen Sie auf Steckplatz 2 eine CPU 317-2 PN/DP ein.



2. Durch Doppelklick auf die Zeile X2 öffnet sich der Eigenschaftsdialog der PN-IO-Schnittstelle.



3. Geben Sie die IP-Adresse und die Subnetzmaske ein. Wenn Sie eine Verbindung über einen Router aufbauen, müssen Sie zusätzlich noch die Adresse des Routers eingeben.
4. Klicken Sie auf die Schaltfläche "Neu..." und vergeben Sie einen Namen für ein neues Industrial Ethernet Subnetz. Klicken Sie dann auf die Schaltfläche "OK".

Ergebnis: Sie haben ein neues Industrial Ethernet-Subnetz angelegt.

5. Klicken Sie auf die Schaltfläche "OK".

Ergebnis: Das Eigenschaftsfenster der PN-IO-Schnittstelle der CPU 317-2 PN/DP wird geschlossen.

3.3 Einfügen der Funktionsbausteine in das Programm

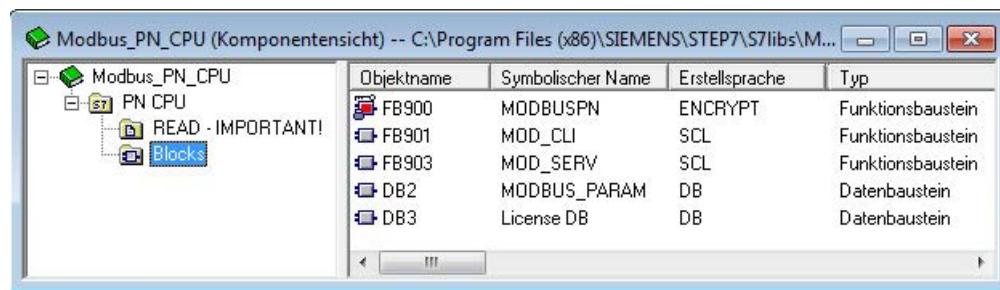
Inhalt der Modbus-Bibliothek

Folgende FBs werden für die Modbuskommunikation benötigt:

- MODBUSPN
- MOD_CLI
- MOD_SERV

Um diese in Ihr Projekt einzufügen zu können, müssen Sie die Bausteine aus der Bibliothek kopieren.

Des Weiteren befindet sich der Parameterdatenbaustein MODBUS_PARAM sowie der Lizenz-DB als Vorlage in der Bibliothek. Diese können Sie zur Arbeitserleichterung ebenfalls in Ihr Projekt kopieren.



Bausteine der Standardbibliothek

Folgende FBs werden für die Modbuskommunikation benötigt:

- TSEND (FB63)
- TRCV (FB64)
- TCON (FB65)
- TDISCON (FB66)

Diese Kommunikationsbausteine finden Sie in der Bibliothek "Standard Library - Communication Blocks" und müssen ebenfalls in Ihr Projekt eingefügt werden.

Hinweis

Beachten Sie, dass folgende Versionen der FBs Voraussetzung für den einwandfreien Betrieb des FB MODBUSPN sind:

- TSEND V2.1
- TRCV V2.2
- TCON V2.4
- TDISCON V2.1

3.4

Mehrere Verbindungen auf Port 502

Einige CPUs können TCP-Verbindungen multiplexen. Dabei können sich mehrere MODBUS Clients auf den Port 502 der CPU verbinden (Multiport). Die CPU fungiert als MODBUS Server.

Hier (<http://support.automation.siemens.com/WW/view/de/34010717>) finden Sie Informationen, welche CPU mit welchem Firmwarestand die mehrfache Nutzung des Ports 502 ermöglicht.

Voraussetzung

Um diese Funktion nutzen zu können, müssen bei der Parametrierung folgende Einstellungen vorgenommen werden:

- CP ist Server
- Port 502 als lokaler Port
- Passiver Verbindungsauflbau

Freigegebene Anzahl der Verbindungen

Die Anzahl der Verbindungen, die eine CPU auf dem Port 502 annehmen kann, ist geräteabhängig und kann den technischen Daten der CPU entnommen werden.

Projektierung

Für jeden Client, der sich auf den Port 502 des Servers verbinden will, sind je 1 eindeutige Verbindung im Parameter-DB und je 1 Modbusbaustein-Instanz im Programm notwendig.

Parametrieren der Modbus-Kommunikation

4.1 Überblick

Allgemeines

Für die Kommunikation über die integrierte PN-Schnittstelle der CPU ist keine Verbindungsprojektierung in NetPro notwendig. Die Verbindungen werden mit Hilfe der Funktionsbausteine TCON (FB65) und TDISCON (FB66) auf- bzw. abgebaut.

Es können mehrere Verbindungen zu verschiedenen Kommunikationspartnern projektiert und gleichzeitig aufgebaut werden. Die Anzahl der gleichzeitig aufgebauten Verbindungen ist CPU-abhängig.

Verbindungs-Datenbaustein MODBUS_PARAM

Die für den Aufbau der Verbindungen und die Bearbeitung der Modbustelegramme notwendigen Daten werden in einer Struktur – dem Parameterdatenbaustein MODBUS_PARAM – definiert. Dabei werden zuerst die verbindungsspezifischen Daten abgelegt. Nach den verbindungsspezifischen Daten folgen die Modbusparameter.

In dem Parameterdatenbaustein können die Projektierungsdaten aller Verbindungen enthalten sein. Es ist ebenso möglich, für jede Verbindung einen separaten Parameterdatenbaustein anzulegen. Eine vorgefertigte Struktur ist als Beispiel in der Bibliothek "Modbus_PN_CPU" enthalten.

Adresse	Name
	STRUCT
+0.0	Verbindung 1: Verbindungsparameter
+64.0	Verbindung 1: Modbusparameter
	END_STRUCT
	STRUCT
+650.0	Verbindung 2: Verbindungsparameter
+714.0	Verbindung 2: Modbusparameter
	END_STRUCT
...	...
	STRUCT
650*i	Verbindung i+1: Verbindungsparameter
650*i+64	Verbindung i+1: Modbusparameter
	END_STRUCT

Bild 4-1 Param_DB

Verbindungsparameter

Im Verbindungsblock werden die verbindungsspezifischen Parameter, wie z.B. die lokal verwendete Hardwareschnittstelle und die IP-Adresse des Kommunikationspartners, definiert. Mit Hilfe dieser Parameter können die Funktionen TCON und TDISCON eine Verbindung auf- bzw. abbauen. Den genauen Aufbau finden Sie in Kapitel 4.2.

Die Datenstruktur des Verbindungsparameterblocks muss zwingend eingehalten werden, da sonst keine Verbindung aufgebaut werden kann.

Modbusparameter

In den Modbusparametern werden die für die Betriebsart und Adressreferenz notwendigen Daten abgelegt, wie z.B. die Modbusbereiche, die in den Datenbausteinen abgebildet werden und die Betriebsart der S7 als Modbus Server oder Modbus Client. Die Datenstruktur der Modbusparameter muss eingehalten werden, da sonst keine fehlerfreie Bearbeitung möglich ist.

Projektierungsmöglichkeiten

Es gibt zwei Möglichkeiten, die Projektierung für die Verbindungs- und Modbusparameter vorzunehmen. Zum einen ist die Eingabe über einen Wizard möglich, mit dem sehr komfortabel die Parametrierung vorgenommen werden kann. Zum anderen können die Parameter durch Editieren der Struktur im Parameterdatenbaustein eingestellt werden.

Diese beiden Möglichkeiten werden in den folgenden Kapiteln 4.2 und 4.3 beschrieben.

4.2 Parametrieren mit dem Wizard

Allgemeines

Mit dem "Modbus/TCP Wizard" ist eine komfortable Projektierung der Verbindungen und der Modbusparameter im Parameterdatenbaustein MODBUS_PARAM möglich. Dabei wird der komplette Datenbaustein (Verbindungsparameter und Modbusparameter) angelegt.

Es wird empfohlen für die Parametrierung des MODBUS_PARAM den Wizard einzusetzen

Den Wizard finden Sie hier (<http://support.automation.siemens.com/WW/view/de/60735352>).

4.3 Manuelle Parametrierung

Vorgehensweise

Kopieren Sie den DB2 aus der Bibliothek "Modbus_PN_CPU" und fügen Sie diesen in Ihr Projekt ein. Wird die Nummer bereits anderweitig verwendet, kann der DB umbenannt werden.

Die Parameter im Baustein MODBUS_PARAM dürfen während der Laufzeit nicht verändert werden. Nach einer Änderung der Parameter muss der Modbusbaustein über den Parameter "Init" neu initialisiert werden.

Aufbau und Anpassungen der Verbindungsparameter

Pro Verbindung wird ein Block benötigt.

Adresse	Name	Typ	Anfangswert
0.0		STRUCT	
+0.0	OUCW_1	STRUCT	
+0.0	block_length	WORD	W#16#40
+2.0	id	WORD	W#16#1
+4.0	connection_type	BYTE	B#16#1
+5.0	active_est	BOOL	FALSE
+6.0	local_device_id	BYTE	B#16#2
+7.0	local_tsap_id_len	BYTE	B#16#2
+8.0	rem_subnet_id_len	BYTE	B#16#0
+9.0	rem_staddr_len	BYTE	B#16#0
+10.0	rem_tsap_id_len	BYTE	B#16#0
+11.0	next_staddr_len	BYTE	B#16#0
+12.0	local_tsap_id	ARRAY[1..16]	B#16#D0, B#16#7, B#16#0, B#16#0
*1.0		BYTE	
+28.0	rem_subnet_id	ARRAY[1..6]	B#16#0, B#16#0, B#16#0, B#16#0,
*1.0		BYTE	
+34.0	rem_staddr	ARRAY[1..6]	B#16#0, B#16#0, B#16#0, B#16#0,
*1.0		BYTE	
+40.0	rem_tsap_id	ARRAY[1..16]	B#16#0, B#16#0, B#16#0, B#16#0,
*1.0		BYTE	
+56.0	next_staddr	ARRAY[1..6]	B#16#0, B#16#0, B#16#0, B#16#0,
*1.0		BYTE	
+62.0	spare	WORD	W#16#0

block_length

Dieser Parameter bezeichnet die Länge der Verbindungsparameter und darf nicht verändert werden.

Fester Wert: W#16#40

id

Für jede logische Verbindung wird eine Verbindungs-ID vergeben. Diese muss bei Verwendung der T-Kommunikation auf der gesamten CPU eindeutig sein. Die ID wird beim Aufruf des FB MODBUSPN angegeben und bei den internen Aufrufen der T-Bausteine (TCON, TSEND, TRCV und TDISCON) verwendet.

Wertebereich:

W#16#1 bis W#16#FFF

connection_type

Hier wird der Verbindungstyp für den Aufbau der Verbindung durch die Funktion TCON definiert. Der einzustellende Wert ist CPU-abhängig.

TCP (Kompatibilitätsmode): B#16#01 für CPU 315 bzw. 317 <= FW V2.3

TCP: B#16#11 für CPU 315 bzw. 317 > FW V2.4, IM 151-8 PN/DP CPU, CPU314C, CPU319, CPU412, CPU414, CPU416.

Je nach verwendeter Firmware können diese Angaben variieren.

Weitere Angaben finden Sie hier

(<http://support.automation.siemens.com/WW/view/de/24294554>).

active_est

Dieser Parameter bezeichnet die Art des Verbindungsaufbaus, aktiv oder passiv. Der Modbus Client übernimmt den aktiven und der Modbus Server den passiven Verbindungsaufbau.

Aktiver Verbindungsaufbau: TRUE

Passiver Verbindungsaufbau: FALSE

local_device_id

Die local_device_id definiert die IE-Schnittstelle der verwendeten PN-CPU. Je nach PN-CPU-Typ werden unterschiedliche Einstellungen benötigt.

IM 151-8 PN/DP CPU	B#16#1
CPU 314C, 315 bzw. 317	B#16#2
CPU 319	B#16#3
CPU 412, 414, 416	B#16#5

local_tsap_id_len

Es wird die Länge des Parameters local_tsap_id (= lokale Portnummer) angegeben.

Aktiver Verbindungsaufbau: 0

Passiver Verbindungsaufbau: 2

rem_subnet_id_len

Dieser Parameter wird derzeit nicht verwendet und muss mit B#16#0 belegt sein.

rem_staddr_len

Es wird die Länge des Parameters rem_staddr, also der IP-Adresse des Kommunikationspartners, angegeben. Soll über eine unspezifizierte Verbindung kommuniziert werden, wird keine IP-Adresse für den Partner angegeben.

Unspezifizierte Verbindung: B#16#0

Spezifizierte Verbindung: B#16#4

rem_tsap_id_len

Dieser Parameter bezeichnet die Länge des Parameters rem_tsap_id, der Portnummer des remoten Kommunikationspartners.

Aktiver Verbindungsaufbau: 2

Passiver Verbindungsaufbau: 0

next_staddr_len

Hier wird die Länge des Parameters next_staddr festgelegt.

Bei PN-Schnittstelle: B#16#0

local_tsap_id

Mit diesem Parameter wird die lokale Portnummer eingestellt. Die Art der Darstellung wird dabei abhängig des Parameters connection_type unterschieden. Der Wertebereich ist CPU-abhängig. Die Portnummer muss auf der CPU eindeutig sein.

Tabelle 4- 1 Bei connection_type B#16#01

local_tsap_id[1]	low byte der Port-Nr. in Hex-Darstellung
local_tsap_id[2]	high byte der Port-Nr. in Hex-Darstellung
local_tsap_id[3-16]	B#16#00

Tabelle 4- 2 Bei connection_type B#16#11

local_tsap_id[1]	high byte der Port-Nr. in Hex-Darstellung
local_tsap_id[2]	low byte der Port-Nr. in Hex-Darstellung
local_tsap_id[3-16]	B#16#00

rem_subnet_id

Dieser Parameter wird derzeit nicht verwendet und muss mit 0 belegt werden.

rem_staddr

In diesem Byte-Array wird die IP-Adresse des remoten Kommunikationspartners eingetragen. Im Falle einer unspezifizierten Verbindung wird keine IP-Adresse eingetragen. Die Art der Darstellung wird abhängig vom Parameter connection_type unterschieden.

Beispiel: IP-Adresse 192.168.0.1:

Tabelle 4- 3 Bei connection_type B#16#01

rem_staddr[1]	B#16#01 (1)
rem_staddr[2]	B#16#00 (0)
rem_staddr[3]	B#16#A8 (168)
rem_staddr[4]	B#16#C0 (192)
rem_staddr[5-6]	B#16#00 (reserviert)

Tabelle 4- 4 Bei connection_type B#16#11

rem_staddr[1]	B#16#C0 (192)
rem_staddr[2]	B#16#A8 (168)
rem_staddr[3]	B#16#00 (0)
rem_staddr[4]	B#16#01 (1)
rem_staddr[5-6]	B#16#00 (reserviert)

rem_tsap_id

Mit diesem Parameter wird die remote Portnummer eingestellt. Die Art der Darstellung wird dabei abhängig vom Parameter connection_type unterschieden. Der Wertebereich ist CPU-abhängig.

Tabelle 4- 5 Bei connection_type B#16#01

rem_tsap_id[1]	low byte der Port-Nr. in Hex-Darstellung
rem_tsap_id[2]	high byte der Port-Nr. in Hex-Darstellung
rem_tsap_id[3-16]	B#16#00

Tabelle 4- 6 Bei connection_type B#16#11

rem_tsap_id[1]	high byte der Port-Nr. in Hex-Darstellung
rem_tsap_id[2]	low byte der Port-Nr. in Hex-Darstellung
rem_tsap_id[3-16]	B#16#00

next_staddr

Dieser Parameter bezeichnet die Rack- und Steckplatznummer des verwendeten CPs. Bei Verwendung der integrierten PN-Schnittstelle der CPU muss dieser Parameter auf 0 gesetzt werden.

next_staddr[1-6]	B#16#00
------------------	---------

spare

Dieser Parameter wird nicht verwendet und muss mit 0 vorbelegt werden.

Anpassungen der Modbusparameter

Mit den Modbusparametern im Baustein MODBUS_PARAM werden die Betriebsart der Modbuskommunikation und die Adressabbildung von Modbusadressen auf SIMATIC-Adressen festgelegt.

+64.0	server_client	BOOL	TRUE
+64.1	single_write	BOOL	FALSE
+64.2	connect_at_startup	BOOL	FALSE
+65.0	reserved	BYTE	B#16#0
+66.0	data_type_1	BYTE	B#16#3
+68.0	db_1	WORD	W#16#B
+70.0	start_1	WORD	W#16#1
+72.0	end_1	WORD	W#16#1F4
+74.0	data_type_2	BYTE	B#16#3
+76.0	db_2	WORD	W#16#C
+78.0	start_2	WORD	W#16#2D0
+80.0	end_2	WORD	W#16#384
+82.0	data_type_3	BYTE	B#16#4
+84.0	db_3	WORD	W#16#D
+86.0	start_3	WORD	W#16#2D0
+88.0	end_3	WORD	W#16#3E8
+90.0	data_type_4	BYTE	B#16#0
+92.0	db_4	WORD	W#16#4
+94.0	start_4	WORD	W#16#0
+96.0	end_4	WORD	W#16#64
+98.0	data_type_5	BYTE	B#16#1
+100.0	db_5	WORD	W#16#E
+102.0	start_5	WORD	W#16#280
+104.0	end_5	WORD	W#16#4E2
+106.0	data_type_6	BYTE	B#16#2
+108.0	db_6	WORD	W#16#F
+110.0	start_6	WORD	W#16#6A4
+112.0	end_6	WORD	W#16#8FC
+114.0	data_type_7	BYTE	B#16#1
+116.0	db_7	WORD	W#16#10
+118.0	start_7	WORD	W#16#6A4
+120.0	end_7	WORD	W#16#8FC
+122.0	data_type_8	BYTE	B#16#0
+124.0	db_8	WORD	W#16#8
+126.0	start_8	WORD	W#16#0
+128.0	end_8	WORD	W#16#64
+130.0	internal_send_buffer	ARRAY[1..260]	B#16#0
*1.0		BYTE	
+390.0	internal_recv_buffer	ARRAY[1..260]	B#16#0
*1.0		BYTE	
=650.0		END_STRUCT	
=650.0		END_STRUCT	

server_client

- TRUE: S7 ist Server
FALSE: S7 ist Client

single_write

In der Betriebsart "S7 ist Client" wird mit dem Parameter `single_write = TRUE` bei schreibenden Aufträgen mit Länge 1 die Funktionscodes 5 und 6 verwendet. Ist `single_write = FALSE`, werden bei allen schreibenden Aufträgen die Funktionscodes 15 und 16 verwendet.

connect_at_startup

Hiermit wird der Zeitpunkt des Verbindungsaufbaus festgelegt. Ist `connect_at_startup` auf `TRUE` gesetzt, wird der Verbindungsaufbau – unabhängig von `ENQ_ENR` – direkt nach dem CPU-Neustart ausgeführt. In diesem Fall darf erst ein Datenauftrag abgesetzt werden, wenn die Verbindung korrekt aufgebaut werden konnte (`CONN_ESTABLISHED = TRUE`) oder ein entsprechender Fehler an `ERROR` und `STATUS` angezeigt wird. Spätestens nach Ablauf von `CONN_TIMEOUT` werden die Status-Ausgänge aktualisiert.

- FALSE: Verbindungsaufbau bei gesetztem `ENQ_ENR`
TRUE: Verbindungsaufbau direkt nach Neustart

8 Datenbereiche

Es werden 8 Datenbereiche für die Abbildung der MODBUS Adressen im S7-Speicher angeboten. Es muss mindestens der erste Datenbereich definiert werden, die anderen 7 Datenbereiche sind optional. Aus den Datenbereichen wird abhängig vom Auftragstyp gelesen oder in sie geschrieben.

Mit einem Auftrag kann immer nur aus einem DB gelesen / in einen DB geschrieben werden. Zugriffe auf Register oder Bitwerte, die in mehreren DBs liegen, auch wenn die Nummern lückenlos hintereinander liegen, sind auf zwei Aufträge aufzuteilen. Dies ist bei der Projektierung zu berücksichtigen.

Es ist möglich in einen Datenbaustein mehr Modbusbereiche (Register oder Bitwerte) abzubilden als mit einem Telegramm bearbeitet werden können.

data_type_x

Mit dem Parameter `data_type_x` wird angegeben, welche MODBUS Datentypen in diesem Datenbaustein abgebildet werden. Wird in `data_type_x` der Wert 0 eingetragen, wird der entsprechende Datenbereich nicht verwendet.

Kennung	Datentyp	Datenbreite
0	Bereich nicht verwendet	
1	Coils	Bit
2	Inputs	Bit
3	Holding Register	Word
4	Input Register	Word

db_x

Der Parameter `db_x` legt den Datenbaustein fest, in dem die nachfolgend definierten MODBUS Register oder Bitwerte abgebildet werden. Die DB-Nummer 0 ist nicht erlaubt, da diese für das System reserviert ist.

Tabelle 4- 7 db_x

DB-Nummer	1 bis 65535 (W#16#0001 bis W#16#FFFF)
-----------	---------------------------------------

Der Datenbaustein muss 2 Byte länger angelegt werden als für die parametrierten Daten notwendig ist. Die beiden letzten Bytes werden für interne Zwecke verwendet.

start_x, end_x

Mit `start_x` wird die erste Modbusadresse, die im Datenwort 0 des DB abgebildet wird, angegeben. Der Parameter `end_x` definiert die Adresse der letzten Modbusadresse.

Bei Registerzugriffen berechnet sich die Datenwortnummer im S7 DB, in die die letzte Modbusadresse eingetragen wird, nach folgender Formel:

$$\text{DBW Nummer} = (\text{end}_x - \text{start}_x) * 2$$

Bei Bitzugriffen berechnet sich die Datenbytenummer im S7 DB, in die die letzte Modbusadresse eingetragen wird, nach folgender Formel:

$$\text{DBB Nummer} = (\text{end}_x - \text{start}_x + 7) / 8$$

Die definierten Datenbereiche dürfen sich nicht überlappen. Der Parameter `end_x` darf nicht kleiner als `start_x` sein. Im Fehlerfall wird der Anlauf des FBs mit Fehler beendet. Sind beide Werte gleich, wird 1 Modbusadresse (1 Register oder 1 Bitwert) zugeordnet.

In Kapitel 6.3 ist ein Beispiel für die Abbildung der MODBUS Adressen auf S7-Speicherbereiche dargestellt.

Tabelle 4- 8 start_x, end_x

MODBUS Adresse	0 bis 65535 (W#16#0000 bis W#16#FFFF)
----------------	---------------------------------------

internal_send_buffer

Dieses Array wird intern im FB für die Sendedaten verwendet. Zugriffe oder Änderungen in diesem Bereich sind nicht zulässig.

internal_recv_buffer

Dieses Array wird intern im FB für die Empfangsdaten verwendet. Zugriffe oder Änderungen in diesem Bereich sind nicht zulässig.

Lizenzierung

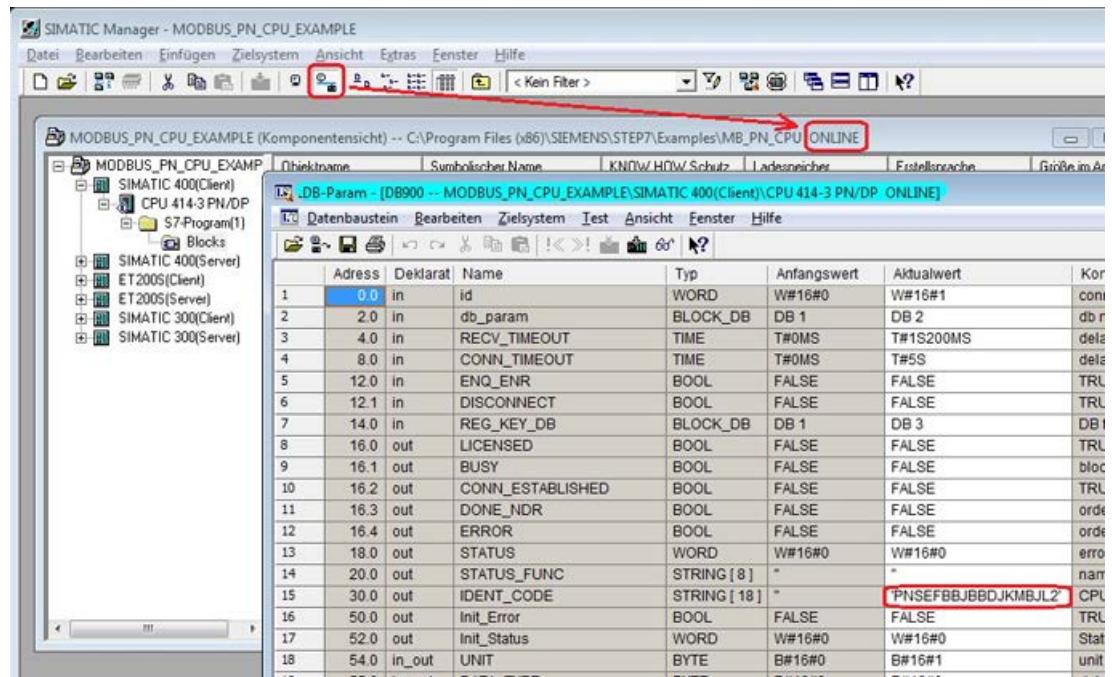
Allgemeines

Der Baustein MODBUSPN muss auf jeder CPU einzeln lizenziert werden. Die Lizenzierung erfolgt in 2 Schritten: dem Auslesen des IDENT_CODEs und der Eingabe des Freischaltcodes REG_KEY. Dabei muss der OB121 in der CPU vorhanden sein.

Auslesen des IDENT_CODES

Für das Auslesen des IDENT_CODE gehen Sie wie folgt vor:

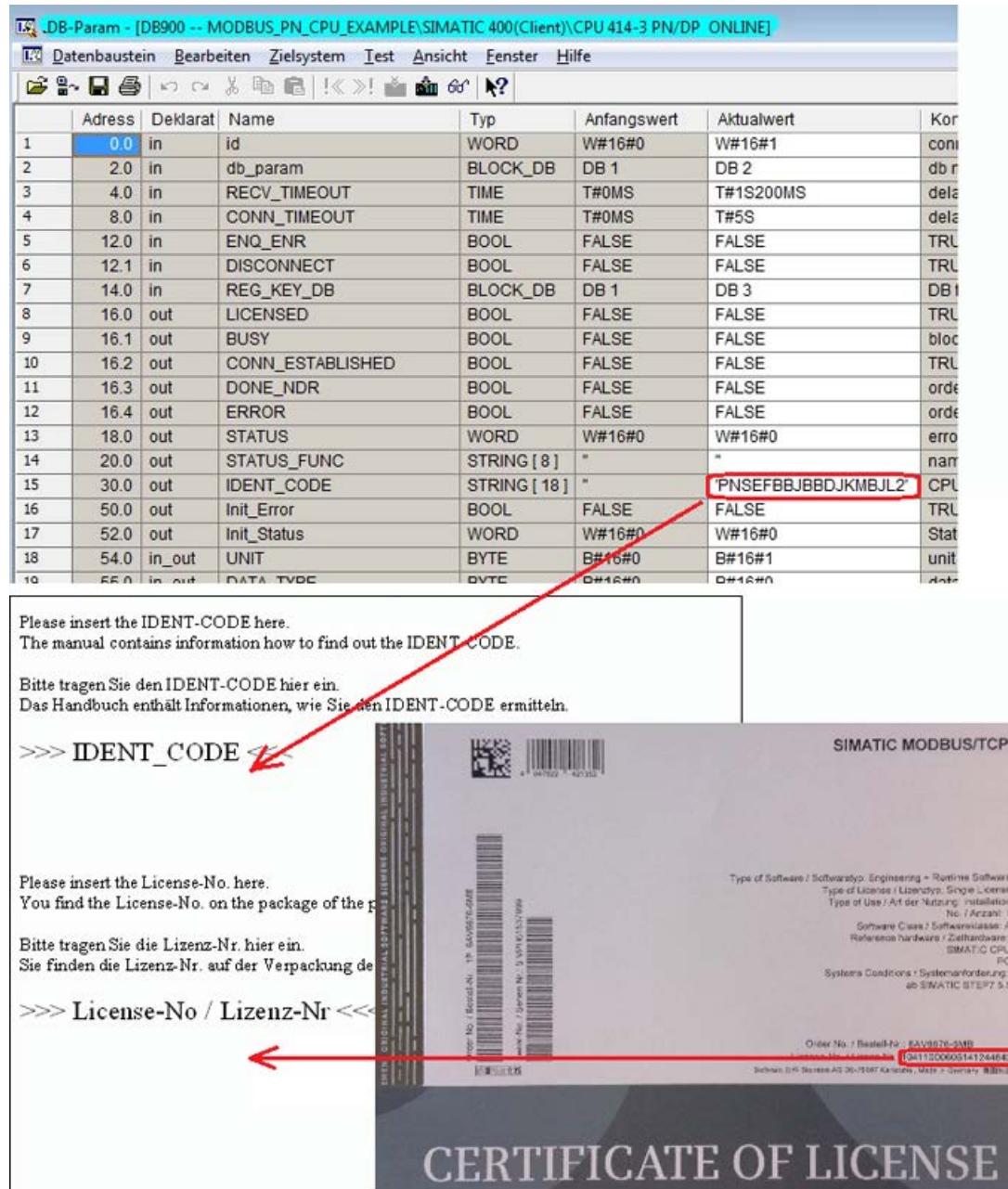
1. Parametrieren Sie den Baustein MODBUSPN Ihren Anforderungen entsprechend in einem zyklischen OB (OB1 oder Weckalarm-OB).
2. Laden Sie das Programm in die CPU und setzen Sie diese in RUN.
3. Öffnen Sie im SIMATIC Manager das Projekt im Online-Modus. Öffnen Sie in diesem Online-Projekt den Instanz-DB des Modbusbausteins.



3. Am Ausgang IDENT_CODE wird eine 18stellige Zeichenfolge angezeigt.

Kopieren Sie diesen String per Copy/Paste aus dem DB und fügen ihn in das Formular SOFTWARE REGISTRATION FORM ein. Dieses Formular wird bei der Installation im Bibliotheks-Pfad ..\Program Files\Siemens\Step7\S7LIBS\Modbus_PN_CPU abgelegt und liegt zusätzlich auch auf der Installations-CD.

Tragen Sie die Lizenz-Nr. von der Produktverpackung in das Formular ein.



4. Senden Sie das Formular als Support Request (<https://support.industry.siemens.com/my/ww/de/requests/#createRequest>) an den Customer Support. Sie erhalten daraufhin den Freischaltcode für ihre CPU.
5. Hinweis für die Verwendung in CFC: Der CFC-Editor kann online nur eine bestimmte Anzahl von Zeichen anzeigen. Der vollständige IDENT_CODE wird im Tooltipp des Ausgangsparameters bzw. im IDB anzeigt.

Eingabe des Freischaltcodes REG_KEY

Die Angabe des Freischaltcodes REG_KEY muss an jedem Modbus-Bausteinaufruf erfolgen.

Der REG_KEY muss in einem Global-DB gespeichert werden, über den alle Modbus-Bausteine den notwendigen Freischaltcode erhalten.

Für die Eingabe des Freischaltcodes REG_KEY gehen Sie wie folgt vor:

1. Kopieren Sie den vorgefertigten Lizenzierungsbaustein DB3 aus der Bibliothek "Modbus_PN_CPU" in Ihr Projekt. Wird die DB-Nummer bereits im Projekt verwendet, kann der Lizenz-DB auch umbenannt werden.
2. Öffnen Sie den Lizenz-DB und kopieren Sie den übermittelten 17-stelligen Freischaltcode per Copy/Paste in die Spalte "Anfangswert".

Die Eingabe von mehreren Keys als Liste ist möglich.

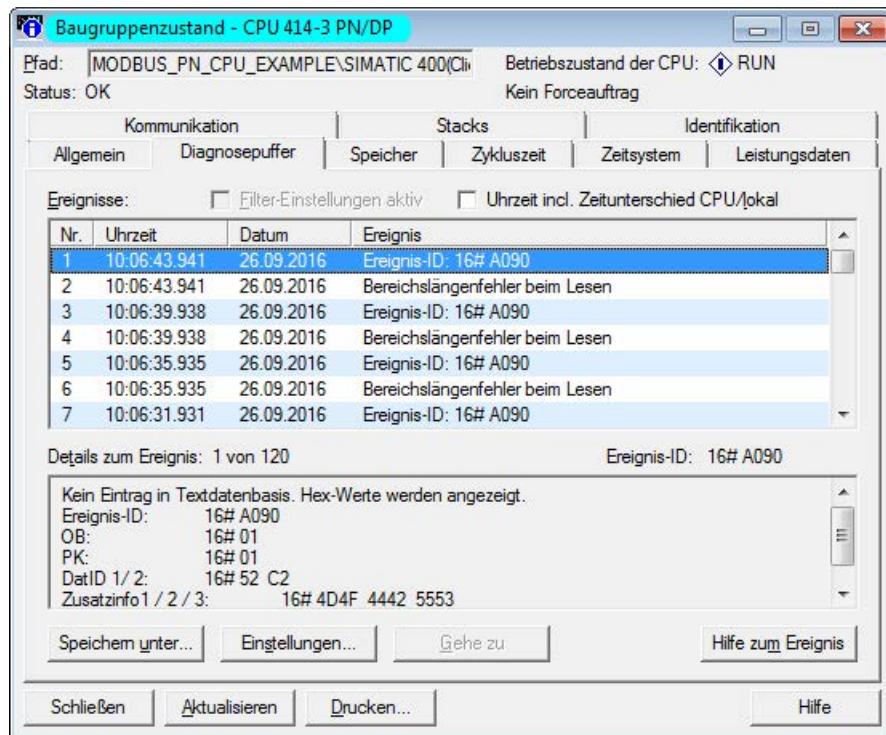
Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	REG_KEY	STRING[17]	'insert REG_KEY'	Registration Key
=20.0		END_STRUCT		

3. Damit der Freischaltcode nach dem Neuladen der CPU nicht erneut eingegeben werden muss, muss er im Datenbaustein fest eingetragen werden. Wechseln Sie dazu über den Menüpunkt "Ansicht" -> "Datensicht" in die Datensicht des DBs. Über den Menübefehl "Bearbeiten" -> "Datenbaustein initialisieren" werden dann alle Werte der Spalte "Anfangswert" in die Spalte "Aktualwert" übernommen.
4. Geben Sie im zyklischen OB am Parameter REG_KEY des Modbus-Bausteins die Datenbausteinnummer des Lizenz-DBs an.
5. Laden Sie die geänderten Bausteine in die CPU. Die Eingabe des Freischaltcodes kann zur Laufzeit erfolgen, ein Wechsel von STOP -> RUN ist nicht erforderlich.

Der Baustein ist nun für diese CPU lizenziert.

Fehlende oder fehlerhafte Lizenziierung

Ist kein oder ein falscher Freischaltcode eingetragen, blinkt die SF-LED (bei S7-300 und IM151-8) bzw. INTF-LED (bei S7-400) der CPU und es wird zyklisch ein Eintrag in den Diagnosepuffer bezüglich der fehlenden Lizenz vorgenommen. Die Fehlernummer für eine fehlende Lizenz ist W#16#A090.



! WARNUNG

Unbeabsichtigter STOP-Zustand

Falls der OB121 in der Steuerung fehlt, wird die CPU in den STOP-Zustand gesetzt.

Bei einem fehlenden oder falschen Freischaltcode wird die Modbus-Kommunikation bearbeitet, allerdings wird am Ausgang STATUS stets W#16#A090 "Keine gültige Lizenz vorhanden" angezeigt.

Funktionsbaustein MODBUSPN

6.1 Funktionsweise des FB

Allgemeines

Der Funktionsbaustein MODBUSPN erlaubt es eine Kommunikation zwischen einer CPU mit integrierter PN-Schnittstelle und einem Partner, der das Modbus/TCP Protokoll unterstützt, aufzubauen.

Es werden die Funktionscodes 1, 2, 3, 4, 5, 6, 15 und 16 unterstützt. Der FB kann je nach Parametrierung sowohl als Client als auch als Server betrieben werden.

Der Baustein MODBUSPN ruft intern die Bausteine MOD_CLI (FB901) und MOD_SERV (FB903) auf. Der Baustein MOD_CLI beinhaltet die Funktionalität des Modbus Clients, der Baustein MOD_SERV realisiert die Funktionalität des Modbus Servers.

Aufgaben des FB

Der Funktionsbaustein erfüllt folgende Aufgaben:

- Verbindungs- und Datenhandling unter Verwendung der T-Bausteine aus der Standard-Bibliothek
- MODBUS spezifischen Telegrammheader beim Senden generieren
- Prüfung des MODBUS spezifischen Telegrammheaders beim Empfang
- Prüfung, ob die angesprochenen Datenbereiche vorhanden sind
- Exception-Telegramme generieren, wenn ein Fehler aufgetreten ist (nur bei S7 ist Server)

Exception Code	Bedeutung
1	Der gesendete Funktionscode wird nicht unterstützt.
2	Es erfolgte ein Zugriff auf eine nicht vorhandene bzw. nicht zulässige Adresse.
3	Es wurde eine ungültige Länge für diesen Funktionscode angegeben.

- Datentransfer von/in den parametrierten DB
- Zeitliche Überwachung des Verbindungsaufbaus und –abbaus sowie des Empfangs von Daten
- Transaction Identifier TI führen
- Lizenzprüfung

Online-Hilfe

Für den Funktionsbaustein MODBUSPN steht im SIMATIC-Manager eine Baustein-Online-Hilfe zur Verfügung. Wenn der Baustein markiert und die Taste "F1" gedrückt wird, wird die Online-Hilfe mit den wichtigsten Informationen zum Baustein geöffnet.

Aufruf des FBs

Der Funktionsbaustein MODBUSPN muss für einen korrekten Programmablauf in einem zyklischen OB (OB1 oder einem zeitgesteuerten OB, z.B. OB35) eingebaut werden.

Die anderen in der Bibliothek enthaltenen FBs MOD_CLI und MOD_SERV werden unterlagert aufgerufen und dürfen nicht zusätzlich in einem OB aufgerufen werden.

Der gleichzeitige Aufruf des MODBUSPN im OB1 und in einem zeitgesteuerten OB (z.B. OB35) ist nicht zulässig.

Der OB121 muss in der CPU vorhanden sein. Nähere Informationen dazu erhalten Sie im Kapitel "Lizenzierung".

Initialisierung des FBs

Der Funktionsbaustein MODBUSPN wird mit einer positiven Flanke am Eingang "Init" initialisiert.

- Die Initialisierungsparameter müssen entsprechend der Anlagenkonfiguration belegt sein.
- Die Initialisierungsparameter werden in den Instanz-DB übernommen.
- Die Laufzeitparameter werden bei der Initialisierung nicht ausgewertet.
- Die Daten aus MODBUS_PARAM werden auf Plausibilität überprüft.

Wird am Parameter "Init" eine positive Flanke erkannt, werden die oben genannten Aktionen durchgeführt. Konnte die Prüfung ohne Fehler beendet werden, wird "Init" zurückgesetzt, "Init_Error" und "Init_Status" zeigen 0 an.

Traten während der Prüfung Fehler auf, wird dies an den Ausgängen "Init_Error" und "Init_Status" angezeigt. Solange ein Init-Fehler ansteht, ist keine Modbus/TCP-Kommunikation über diesen Baustein möglich. Der Init-Fehler muss erst korrigiert werden.

Zyklischer Betrieb des FBs

Im zyklischen Betrieb wird der MODBUSPN z.B. im OB35 aufgerufen.

- Anhand der Laufzeitparameter werden die Funktionen des Bausteins aktiviert.
- Während ein Auftrag läuft, werden Änderungen an den Laufzeitparametern nicht ausgewertet.
- Die Initialisierungsparameter werden nicht ausgewertet, solange keine Initialisierung durchgeführt wird.

Programmierfehler OB121

Ist der Modbus-Baustein für diese CPU noch nicht lizenziert, wird OB121 aufgerufen.

 WARNUNG
Unbeabsichtigter STOP-Zustand
Falls der OB121 in der Steuerung fehlt, wird die CPU in den STOP-Zustand gesetzt.

Verbindungsbearbeitung

Den aktiven Verbindungsaufbau führt der Modbus Client aus. Die Daten hierfür werden aus den Verbindungsparametern im DB MODBUS_PARAM ausgelesen.

Über einen Parameter im Verbindungsparameterblock (active_est) wird festgelegt, ob die PN-CPU als aktiver oder als passiver Kommunikationspartner fungieren soll.

Zur Laufzeit wird bei beiden Verbindungstypen, aktiv und passiv, mit der Funktion TCON ein Kommunikationskanal zum Koppelpartner geöffnet.

Der Zeitpunkt des Verbindungsaufbaus wird mit einem Parameter im DB MODBUS_PARAM festgelegt (connect_at_startup).

Der Verbindungsabbau wird mit dem Parameter DISCONNECT am FB MODBUSPN geregelt.

Auftragsanstoß, S7 ist Client

Durch einen positiven Flankenwechsel am Triggereingang ENQ_ENR wird ein Auftrag initiiert. Abhängig von den Eingangsparametern UNIT, DATA_TYPE, START_ADDRESS, LENGTH und WRITE_READ wird ein MODBUS Anforderungstelegramm generiert und zur Partnerstation über die TCP/IP Verbindung gesendet. Der Baustein wartet die parametrierte Zeit RECV_TIMEOUT auf eine Antwort vom Server. Kommt es zu einer Zeitüberschreitung (keine Antwort vom Server) wird der aktivierte Auftrag mit Fehler beendet. Ein neuer Auftrag kann initiiert werden.

Wird ein Antworttelegramm empfangen, wird eine Plausibilitätsprüfung durchgeführt. Verläuft diese positiv, werden die erforderlichen Aktionen durchgeführt und der Auftrag wird ohne Fehler beendet, der Ausgang DONE_NDR wird gesetzt. Wurden bei der Prüfung Fehler erkannt, wird der Auftrag mit Fehler beendet, das Bit ERROR wird gesetzt und eine Fehlernummer an STATUS angezeigt.

Aktivierung des FBs, S7 ist Server

Durch einen positiven Pegel am Triggereingang ENQ_ENR ist der FB zum Empfang eines Anforderungstelegramms vom Client bereit. Der Server verhält sich dabei passiv und wartet auf ein Telegramm vom Client. Das empfangene Telegramm wird überprüft. Verläuft die Prüfung positiv, wird das Anforderungstelegramm beantwortet. Der beendete Telegrammverkehr wird dem Anwender mitgeteilt indem das Bit DONE_NDR gesetzt wird. Zu diesem Zeitpunkt wird die ausgeführte Funktion an den Ausgängen UNIT, DATA_TYPE, START_ADDRESS, LENGTH und WRITE_READ angezeigt.

Ein fehlerhaftes Anforderungstelegramm bewirkt eine Fehlermeldung. Das Bit ERROR wird gesetzt, im STATUS wird die Fehlernummer angezeigt und die Anforderung des Client wird nicht bearbeitet. Je nach Fehler wird ein Exception-Telegramm zum Client geschickt.

6.2 Parameter des Funktionsbausteins MODBUSPN

Parameter	Dekl.	Typ	Beschreibung	Wertebereich	Init
id	IN	WORD	Verbindungs-ID gemäß der Projektierung "id" im Parameter-DB	1 bis 4095 W#16#1 bis W#16#FFF	ja
db_param	IN	BLOCK_DB	Parameter-DB, enthält alle Verbindungs- und Modbusdaten für diese Modbusbaustein-Instanz	CPU-abhängig	ja
REG_KEY_DB	IN	BLOCK_DB	Datenbaustein mit dem Registrierungsschlüssel für die Lizenzierung	CPU-abhängig	nein
RECV_TIMEOUT	IN	TIME	Überwachungszeit für Datenempfang, mind. 20ms	T#20ms bis T#+24d20h31m23s	nein
CONN_TIMEOUT	IN	TIME	Überwachungszeit für den Verbindungsauflauf und -abbau, mind. 100ms	T#100ms bis T#+24d20h31m23s	nein
DISCONNECT	IN	BOOL	S7 ist Client: TRUE: nach Empfang des Antworttelegramms wird die Verbindung abgebaut S7 ist Server: TRUE: bei ENQ_ENR = FALSE wird die Verbindung abgebaut	TRUE/FALSE	nein
ENQ_ENR	IN	BOOL	S7 ist Client: Auftragsanstoß bei positiver Flanke S7 ist Server: Empfangsbereit bei positivem Pegel	TRUE/FALSE	nein
LICENSED	OUT	BOOL	Lizenzzustand des Bausteins: Baustein ist lizenziert Baustein ist nicht lizenziert	TRUE FALSE	nein
CONN_ESTABLISHED	OUT	BOOL	TRUE: Verbindung ist aufgebaut FALSE: Verbindung ist abgebaut	TRUE/FALSE	nein
BUSY	OUT	BOOL	Bearbeitungszustand T-Funktionen (TCON, TDISCON, TSEND oder TRCV) in Bearbeitung nicht in Bearbeitung	TRUE FALSE	nein
DONE_NDR	OUT	BOOL	S7 ist Client: TRUE: aktiver Auftrag wurde fehlerfrei beendet S7 ist Server: TRUE: Anforderung vom Client wurde ausgeführt und beantwortet	TRUE/FALSE	nein
ERROR	OUT	BOOL	TRUE: Es ist ein Fehler aufgetreten.	TRUE/FALSE	nein

Parameter	Dekl.	Typ	Beschreibung	Wertebereich	Init
STATUS	OUT	WORD	Fehlernummer oder Statusinformation	0 bis FFFF	nein
STATUS_FUNC	OUT	STRING[8]	Name der Funktion, die den Fehler an STATUS verursacht hat	Character	nein
IDENT_CODE	OUT	STRING [18]	Identifikation für die Lizenzierung. Fordern Sie mit diesem Identifikationsstring die Lizenz an.	Character	nein
Init_Error	OUT	BOOL	TRUE: Bei der Initialisierung ist ein Fehler aufgetreten.	TRUE/FALSE	nein
Init_Status	OUT	WORD	Status der Initialisierung	0 bis FFFF	nein
UNIT	IN/OUT	BYTE	Unit Identifier (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion)	0 bis 255 B#16#0 bis B#16#FF	nein
DATA_TYPE	IN/OUT	BYTE	zu bearbeitender Datentyp (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion)		nein
			Coils	1	
			Inputs	2	
			Holding Register	3	
			Input Register	4	
START_ADDRESS	IN/OUT	WORD	MODBUS Startadresse (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion)	0 bis 65535 W#16#0000 bis W#16#FFFF	nein
LENGTH	IN/OUT	WORD	Anzahl der zu bearbeitenden Werte (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion)		nein
			Coils		
			Lesende Funktion	1 bis 2000	
			Schreibende Funktion	1 bis 1968	
			Inputs		
			Lesende Funktion	1 bis 2000	
			Holding Register		
			Lesende Funktion	1 bis 125	
			Schreibende Funktion	1 bis 123	
			Input Register		
			Lesende Funktion	1 bis 125	
WRITE_READ	IN/OUT	BOOL	Schreibzugriff Lesezugriff (INPUT bei CLIENT Funktion, OUTPUT bei SERVER Funktion)	TRUE FALSE	nein
Init	IN/OUT	BOOL	Initialisierung bei positiver Flanke	TRUE/FALSE	nein

Allgemeines

Die Parameter des FB MODBUSPN gliedern sich in zwei Gruppen:

- Initialisierungsparameter (klein geschrieben)
- Laufzeitparameter (groß geschrieben)

Die Initialisierungsparameter werden nur bei einer positiven Flanke am Parameter "Init" ausgewertet und in den Instanz-DB übernommen. Die Initialisierungsparameter sind in der obigen Tabelle in der Spalte "Init" mit "ja" gekennzeichnet.

Eine Änderung der Initialisierungsparameter während des laufenden Betriebs hat keine Auswirkung. Nach einer Änderung dieser Parameter z.B. im Testbetrieb muss der Instanz-DB (I-DB) durch eine positive Flanke am Parameter "Init" neu initialisiert werden.

Laufzeitparameter können im zyklischen Betrieb verändert werden. Es ist nicht sinnvoll die Eingangsparameter zu ändern während ein Auftrag läuft. Mit den Vorbereitungen für den nächsten Auftrag und den damit verbundenen Änderungen der Parameter sollte gewartet werden, bis der vorherige Auftrag mit DONE_NDR oder ERROR beendet wurde.

In der Betriebsart "S7 ist Server" dürfen die Ausgangsparameter nur bei gesetztem DONE_NDR ausgewertet werden.

Die Ausgangsparameter sind dynamische Anzeigen und stehen somit nur 1 CPU-Zyklus an. Sie müssen für eine eventuelle Weiterverarbeitung oder eine Anzeige in der Variablenliste in andere Speicherbereiche kopiert werden.

Wertebereiche

Bei den Wertebereichen für die verschiedenen Parameter sind ggf. auch CPU-spezifische Einschränkungen zu beachten.

id

Für jede Verbindung von der PN-CPU zu einem Kommunikationspartner wird eine Verbindungs-ID benötigt. Bei mehreren Kommunikationspartnern ist für jede logische Verbindung eine andere Verbindungs-ID zu verwenden. Diese Verbindungs-ID wird im Verbindungsparameterblock projektiert, welcher im Parameterdatenbaustein MODBUS_PARAM enthalten ist. Die Verbindungs-ID beschreibt eindeutig die Verbindung von der CPU zum Koppelpartner und kann Werte von 1 bis 4095 annehmen.

Die Verbindungs-ID aus dem Verbindungsparameterblock ist hier einzutragen und muss CPU-weit eindeutig sein.

db_param

Der Parameter db_param bezeichnet die Nummer des Datenbausteins MODBUS_PARAM. In diesem Parameterdatenbaustein sind die verbindungs- und modbusspezifischen Parameter hinterlegt, die für die Kommunikation zwischen der PN-CPU und dem Koppelpartner notwendig sind.

Der Wertebereich für diesen Parameter ist CPU-abhängig. Die DB-Nummer 0 ist nicht zulässig, da diese für das System reserviert ist. Die Eingabe der DB-Nummer erfolgt im Klartext in der Form "DBxy".

Sollen mehrere Verbindungen realisiert werden, kann der Parameterdatenbaustein die dazu notwendigen Parameter aller Verbindungen in sequentieller Abfolge enthalten. Es ist ebenso möglich, für jede Verbindung einen separaten Parameterdatenbaustein zu erstellen.

REG_KEY_DB

Der Baustein muss auf jeder CPU lizenziert werden. Mit der korrekten Eingabe des Freischaltcodes wird der Baustein lizenziert und die Modbus-Kommunikation kann ohne Einschränkungen genutzt werden. Es wird die Datenbausteinnummer angegeben, der den Freischaltcode enthält. Es ist möglich, mehrere Freischaltcodes untereinander in den DB einzutragen. Der Modbusbaustein durchsucht den DB nach dem passenden Freischaltcode. Weitere Informationen entnehmen Sie bitte dem Kapitel "Lizenzierung".

RECV_TIMEOUT

Die Überwachungszeit RECV_TIMEOUT überwacht den Empfang des Antworttelegramms vom Koppelpartner. Der Minimalwert beträgt 20ms.

Wenn in der Betriebsart "S7 ist Client" die RECV_TIMEOUT auf < 20ms gesetzt wird, erscheint eine entsprechende Fehlermeldung und der aktivierte Auftrag wird abgewiesen. Bei Ablauf der Überwachungszeit wird der aktivierte Auftrag mit Fehler beendet.

Wenn in der Betriebsart "S7 ist Server" die RECV_TIMEOUT auf < 20 ms gesetzt wird, wird der Defaultwert von 1,2 s verwendet. Bei Überschreitung der Überwachungszeit wird ein Fehler gemeldet. Die RECV_TIMEOUT überwacht die Laufzeit des TCP-Streams. Die Pause zwischen einzelnen Requests vom Client wird dabei nicht berücksichtigt.

CONN_TIMEOUT

Mit der Zeit CONN_TIMEOUT wird der Verbindungsaufbau bzw. -abbau überwacht. Der minimale Wert ist 100ms.

Konnte innerhalb der parametrierten Überwachungszeit die Verbindung nicht erfolgreich auf- bzw. abgebaut werden, erscheint eine entsprechende Fehlermeldung am Ausgang STATUS.

In der Betriebsart "S7 ist Client" wird bei connect_at_startup = TRUE eine zu gering parametrierte CONN_TIMEOUT auf 5 s gesetzt. Im zyklischen Betrieb wird bei einer zu kleinen CONN_TIMEOUT eine Fehlermeldung ausgegeben und der aktivierte Auftrag abgewiesen.

Wenn in der Betriebsart "S7 ist Server" die CONN_TIMEOUT auf < 100 ms gesetzt wurde, wird der Defaultwert von 5 s verwendet.

DISCONNECT

In der Betriebsart "S7 ist Client" wird mit DISCONNECT = TRUE wird festgelegt, dass die Verbindung nach dem Datentransfer abgebaut werden soll. In der Betriebsart "S7 ist Server" wird mit DISCONNECT = TRUE die Verbindung abgebaut, wenn der Parameter ENQ_ENR auf FALSE gesetzt wird.

Dieser Parameter ist ein Laufzeitparameter und kann entsprechend der Anforderung beliebig gesetzt bzw. zurückgesetzt werden.

ENQ_ENR

Betriebsart "S7 ist Client": Mit einer positiven Flanke wird der Datentransfer initiiert. Mit den Werten der Eingangsparameter UNIT, DATA_TYPE, START_ADDRESS, LENGTH und WRITE_READ wird das Anforderungstelegramm generiert. Ein neuer Auftrag kann nur gesendet werden, wenn der vorherige mit DONE_NDR oder ERROR abgeschlossen wurde.

Wenn die Verbindung nicht aufgebaut ist (CONN_ESTABLISHED = FALSE), wird zuerst die Verbindung aufgebaut und dann der Datentransfer ausgeführt.

Betriebsart "S7 ist Server": Mit einem positiven Pegel am Eingang wird der FB aktiviert. Es können Telegramme vom Client empfangen werden. Falls die Verbindung bei gesetztem ENQ_ENR nicht aufgebaut ist (CONN_ESTABLISHED = FALSE), wird der Verbindungsaufbau aktiviert.

Wechselt während des laufenden Betriebs ENQ_ENR von TRUE auf FALSE wird abhängig von der Einstellung am Parameter DISCONNECT ggf. die Verbindung abgebaut.

Bei einem nicht gesetzten Eingang ENQ_ENR und einer bestehenden Verbindung werden die empfangenen Daten verworfen.

LICENSED

Ist dieser Ausgang auf TRUE gesetzt, ist der MODBUS-Baustein auf dieser CPU lizenziert. Hat der Ausgang den Zustand FALSE, wurde kein oder ein fehlerhafter Lizenzstring eingetragen. Weitere Informationen entnehmen Sie bitte dem Kapitel "Lizenzierung".

CONN_ESTABLISHED

Mit CONN_ESTABLISHED = TRUE wird angezeigt, dass eine Verbindung zum Koppelpartner besteht und Daten übertragen werden können.

Ist CONN_ESTABLISHED = FALSE besteht keine Verbindung zum Koppelpartner.

BUSY

Wenn dieser Ausgang gesetzt ist, ist eine der T-Funktionen TCON, TDISCON, TSEND oder TRCV in Bearbeitung.

DONE_NDR

In der Betriebsart "S7 ist Client" wurde der aktivierte Auftrag fehlerfrei beendet. Bei einer lesenden Funktion wurden die Antwortdaten vom Server bereits im DB eingetragen, bei einer schreibenden Funktion wurde vom Server die Antwort auf das Anforderungstelegramm erhalten.

In der Betriebsart "S7 ist Server" zeigt der Ausgang einen fehlerfrei beendeten Telegrammverkehr mit dem Client an. In den Parametern UNIT, DATA_TYPE, START_ADDRESS, LENGTH und WRITE_READ werden die Auftragsparameter des Client angezeigt. Diese Ausgänge sind nur gültig so lange DONE_NDR gesetzt ist.

ERROR

Wenn dieser Ausgang gesetzt ist, wurde ein Fehler erkannt.

In der Betriebsart "S7 ist Client" wurde der aktivierte Auftrag mit Fehler abgeschlossen. Die zugehörige Fehlernummer wird am Ausgang STATUS angezeigt.

In der Betriebsart "S7 ist Server" wurde bei einem Anforderungstelegramm des Client oder beim Senden des Antworttelegramms ein Fehler erkannt. Die zugehörige Fehlernummer wird am Ausgang STATUS angezeigt.

STATUS

Der Ausgang STATUS zeigt bei gesetztem ERROR die Fehlernummer, bei nicht gesetztem ERROR Statusinformationen an.

Die Fehlernummern und Statusinformationen sind im Kapitel "Diagnose" beschrieben.

STATUS_FUNC

An diesem Parameter wird als Character-String der Name der Funktion angezeigt, die den aufgetretenen Fehler verursacht hat.

IDENT_CODE

Nach dem Anlauf der CPU wird an diesem Parameter eine 18stellige Identifikationskennung angezeigt, mit dem der REG_KEY (Freischaltcode) für die Modbus-Kommunikation beantragt wird.

Weitere Informationen entnehmen Sie bitte dem Kapitel "Lizenzierung".

Init_Error

Trat bei der Initialisierung ein Fehler auf, wird dies mit Init_Error = TRUE angezeigt.

Init_Status

Der Ausgang Init_Status zeigt bei gesetztem Init_Error die Fehlernummer an. Die Fehlernummern sind im Kapitel "Diagnose" beschrieben.

UNIT

Der Parameter UNIT, Unit Identifier, bezeichnet die eindeutige Zuordnung des Koppelpartners. Er ist vor allem notwendig, wenn sich hinter einem Konverter mehrere serielle Teilnehmer befinden, die mit unterschiedlichen UNIT Nummern angesprochen werden.

In der Funktion "S7 ist Client" ist der Parameter UNIT ein Eingangsparameter. Dieser Eingang ist den Anforderungen entsprechend zu setzen. Der FB übernimmt diesen Wert in das Anforderungstelegramm und überprüft ihn beim Empfang der Antwort. Zu beachten ist, dass einige Koppelpartner eine bestimmte UNIT-Nummer erwarten.

In der Funktion "S7 ist Server" ist der Parameter UNIT ein Ausgangsparameter. Der FB übernimmt den Wert aus dem Anforderungstelegramm in die Antwort. Der Ausgang wird bei beendetem Auftrag mit dem empfangenen Wert gesetzt.

DATA_TYPE

Der Parameter DATA_TYPE zeigt an, welcher MODBUS-Datentyp mit dem aktuellen Telegramm bearbeitet wird. Es sind folgende Werte zulässig:

Coils	B#16#1
Inputs	B#16#2
Holding Register	B#16#3
Input Register	B#16#4

In der Betriebsart "S7 ist Client" handelt es sich um einen Eingangsparameter, in der Betriebsart "S7 ist Server" um einen Ausgangsparameter. Die unterschiedlichen Datentypen haben einen direkten Zusammenhang mit den verwendeten Funktionscodes.

Datentyp	DATA_TYPE	Funktion	Länge	single_write	Funktionscode
Coils	1	lesen	beliebig	irrelevant	1
Coils	1	schreiben	1	TRUE	5
Coils	1	schreiben	1	FALSE	15
Coils	1	schreiben	>1	irrelevant	15
Inputs	2	lesen	beliebig	irrelevant	2
Holding Register	3	lesen	beliebig	irrelevant	3
Holding Register	3	schreiben	1	TRUE	6
Holding Register	3	schreiben	1	FALSE	16
Holding Register	3	schreiben	>1	irrelevant	16
Input Register	4	lesen	beliebig	irrelevant	4

START_ADDRESS

Der Parameter START_ADDRESS bestimmt die erste MODBUS-Adresse, die geschrieben bzw. gelesen wird.

In der Betriebsart "S7 ist Client" handelt es sich um einen Eingangsparameter, in der Betriebsart "S7 ist Server" um einen Ausgangsparameter.

LENGTH

Der Parameter LENGTH bestimmt die Anzahl der MODBUS-Werte, die geschrieben bzw. gelesen wird.

Bei lesenden Funktionen sind pro Telegramm für Holding und Input Register maximal 125 Register möglich. Für Coils und Inputs sind maximal 2000 Bits möglich. Bei schreibenden Funktionen beträgt bei Holding Register die maximale Anzahl 123 Register und bei Coils 1968 Bits.

Die mit einem Anforderungstelegramm bearbeiteten Register bzw. Bitwerte müssen innerhalb eines DBs liegen.

In der Betriebsart "S7 ist Client" handelt es sich um einen Eingangsparameter, in der Betriebsart "S7 ist Server" um einen Ausgangsparameter.

WRITE_READ

Dieser Parameter definiert ob eine lesende oder schreibende Funktion ausgeführt werden soll. Hat der Ein-/Ausgang den Wert FALSE, handelt es sich um eine lesende Funktion. Der Wert TRUE definiert eine schreibende Funktion.

Es können nur Holding Register und Coils beschrieben werden. Input Register und Inputs lassen sich lediglich lesen. In der Betriebsart "S7 ist Client" handelt es sich um einen Eingangsparameter, in der Betriebsart "S7 ist Server" um einen Ausgangsparameter.

Init

Mit einer positiven Flanke am Parameter Init wird der Modbusbaustein initialisiert. Die Initialisierung kann nur durchgeführt werden, wenn aktuell kein Auftrag läuft. Dies muss mit ENQ_ENR = FALSE und BUSY = FALSE programmtechnisch sichergestellt werden.

Hinweis

Bei einer Initialisierung werden die projektierten Verbindungen ab- und wieder aufgebaut. Bei Änderung des id-Parametes, müssen die Verbindungen vor der Initialisierung mit DISCONNECT = TRUE manuell abgebaut werden.

6.3 Beispiel für die Adressabbildung

Interpretation der MODBUS-Adressen

Das MODBUS Datenmodell basiert auf einer Reihe von Speicherbereichen die unterschiedliche Charakteristiken haben. Die Unterscheidung dieser Speicherbereiche erfolgt bei einigen Systemen, z.B. MODICON PLCs über die Register- bzw. Bitadresse. So wird z.B. das Holding Register mit Offset 0 als Register 40001 bezeichnet (Speichertyp 4xxxx, Reference 0001).

Es führt immer wieder zur Verwirrung, weil in manchen Handbüchern die Registeradresse des Application Layers und in anderen die tatsächlich im Protokoll übertragene Register-/Bitadresse beschrieben und gemeint ist.

Der FB MODBUS verwendet bei seinen Parametern start_x, end_x und START_ADDRESS die tatsächlich übertragene Modbusadresse. Es können also mit jedem Funktionscode Register-/Bitadressen von 0000 hex bis FFFF hex übertragen werden.

Beispiel: Parametrierung der Datenbereiche

Datenbereich 1	data_type_1	B#16#3	Holding Register
	db_1	W#16#B	DB 11
	start_1	W#16#0	Anfangsadresse: 0
	end_1	W#16#1F3	Endadresse: 499
Datenbereich 2	data_type_2	B#16#3	Holding Register
	db_2	W#16#C	DB 12
	start_2	W#16#2D0	Anfangsadresse: 720
	end_2	W#16#384	Endadresse: 900
Datenbereich 3	data_type_3	B#16#4	Input Register
	db_3	B#16#D	DB 13
	start_3	W#16#2D0	Anfangsadresse: 720
	end_3	W#16#3E8	Endadresse: 1000
Datenbereich 4	data_type_4	B#16#0	nicht verwendet
	db_4	0	0
	start_4	0	0
	end_4	0	0
Datenbereich 5	data_type_5	B#16#1	Coils
	db_5	W#16#E	DB 14
	start_5	W#16#280	Anfangsadresse: 640
	end_5	W#16#4E2	Endadresse: 1250
Datenbereich 6	data_type_6	B#16#2	Inputs
	db_6	W#16#F	DB 15
	start_6	W#16#6A4	Anfangsadresse: 1700
	end_6	W#16#8FC	Endadresse: 2300

Datenbereich 7	data_type_7	B#16#1	Coils
	db_7	W#16#10	DB 16
	start_7	W#16#6A4	Anfangsadresse: 1700
	end_7	W#16#8FC	Endadresse: 2300
Datenbereich 8	data_type_8	B#16#0	Nicht verwendet
	db_8	0	0
	start_8	0	0
	end_8	0	0

Für dieses Beispiel hat:

- DB11 eine Größe von 1002 Byte, es werden insgesamt 500 Register abgebildet (Register 0 – Register 499) + 2 reservierte Byte
- DB12 eine Größe von 364 Byte, es werden insgesamt 181 Register abgebildet (Register 720 – Register 900) + 2 reservierte Byte
- DB13 eine Größe von 564 Byte, es werden insgesamt 281 Input Register abgebildet (Register 720 - Register 1000) + 2 reservierte Byte
- DB14 eine Größe von 80 Byte, es werden insgesamt 611 Coils (Bits) abgebildet (Coil 640 – Coil 1250) + 2 reservierte Byte
- DB15 eine Größe von 78 Byte, es werden insgesamt 601 Inputs (Bits) abgebildet (Input 1700 – Input 2300) + 2 reservierte Byte
- DB16 eine Größe von 78 Byte, es werden insgesamt 601 Coils (Bits) abgebildet (Coil 1700 – Coil 2300) + 2 reservierte Byte

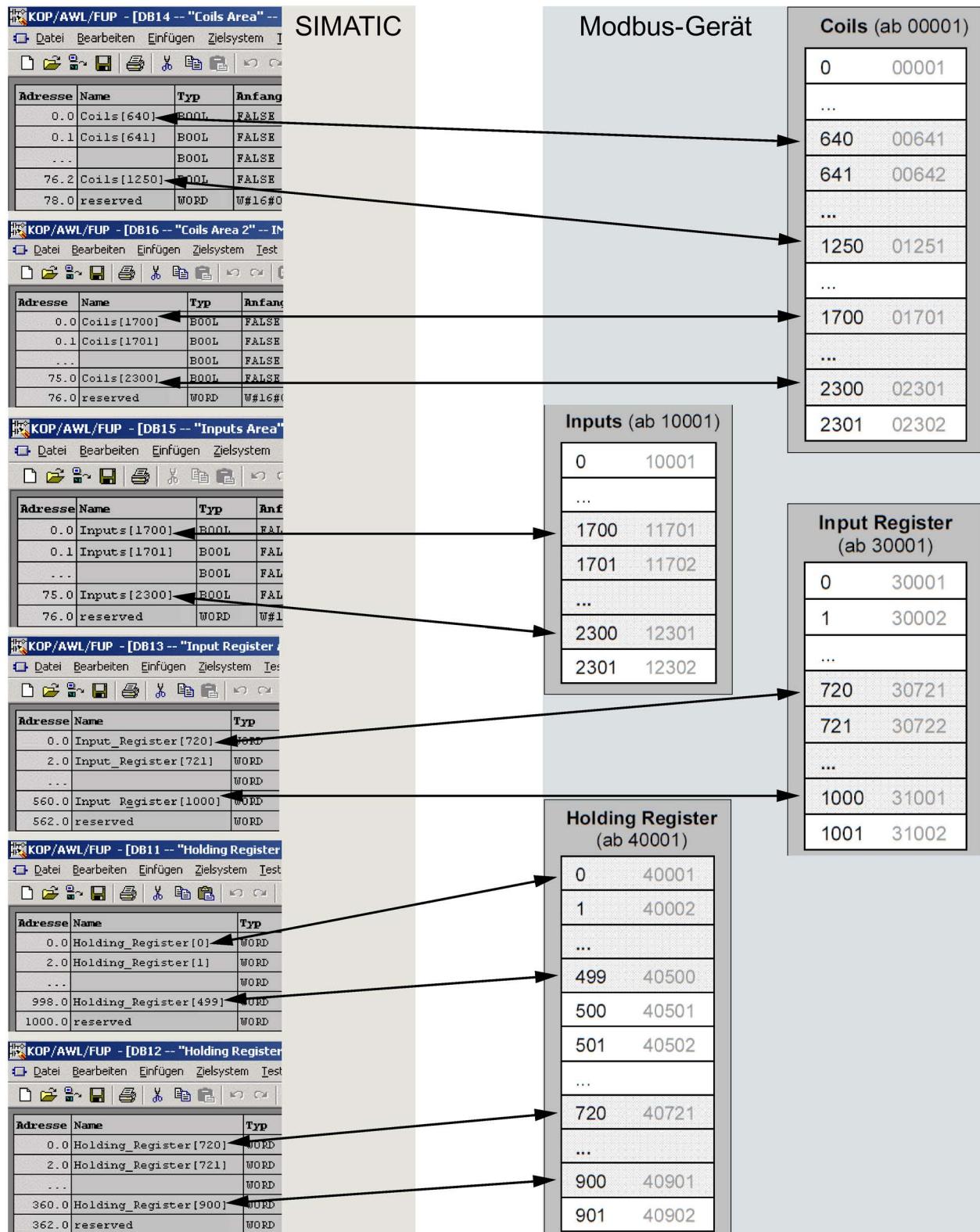
Adressabbildung

Im folgenden Bild sehen Sie die Gegenüberstellung der SIMATIC-Speicherbereiche mit der register- und bitorientierten Speicheraufteilung der Modbus-Geräte. Dabei wird auf obige Parametrierung Bezug genommen.

Im Modbus-Gerät: Die Modbus-Adressen, die schwarz dargestellt sind, betreffen den Data Link Layer, die grau dargestellten den Application Layer.

In der SIMATIC: Die in der 1. Spalte dargestellten SIMATIC-Adressen sind der Offset im DB. In den eckigen Klammern sind die Modbus Registernummern eingetragen.

6.3 Beispiel für die Adressabbildung



6.4 Vom FB verwendete Daten und Standardfunktionen

Instanz-DB

Der Funktionsbaustein MODBUSPN speichert seine Daten in einem Instanz-DB. Dieser Instanz-DB wird beim ersten Aufruf des FB durch STEP 7 generiert.

Der Instanz-Datenbaustein enthält Parameterwerte vom Typ Input, Output, Input/Output sowie statische Variablen, die er für seinen Ablauf benötigt. Diese Variablen sind remanent und behalten zwischen den FB-Aufrufen ihre Gültigkeit. Über die Variablen wird der interne Ablauf des FBs gesteuert.

Speicherbedarf des Instanz-DBs:

Instanz-DB	Arbeitsspeicher	Ladespeicher
MODBUSPN	ca. 1,5 kByte	ca. 3 kByte

Lokale Variablen

In Summe werden maximal 156 Byte Lokaldaten für einen FB MODBUSPN-Aufruf benötigt.

Parameter-DB

Die verbindungs- und modbusspezifischen Parameter werden in dem Parameter-DB MODBUS_PARAM gespeichert.

Zeiten

Der Funktionsbaustein verwendet keine Timer

Merker

Der Funktionsbaustein verwendet keine Merker.

Standard-FBs zur Verbindungsbearbeitung

Die im FB MODBUSPN aufgerufenen Bausteine MOD_CLI und MOD_SERV verwenden die Bausteine TCON und TDISCON aus der Standard-Bibliothek für den Auf- und Abbau der Verbindungen zwischen der CPU und dem Kommunikationspartner.

Standard-FBs zur Datenübertragung

Die im FB MODBUSPN aufgerufenen Bausteine MOD_CLI und MOD_SERV verwenden die Bausteine TSEND und TRCV aus der Standard-Bibliothek für die Datenübertragung zwischen der CPU und dem Kommunikationspartner.

MODBUSPN: SFCs für sonstige Funktionen

Der FB MODBUSPN verwendet folgende SFCs aus der Standardbibliothek:

- SFC20 "BLKMOV"
- SFC24 "TEST_DB"
- SFC51 "RDSYSST"
- SFC52 "WR_USMSG"

MOD_CLI/MOD_SERV:

SFCs für sonstige Funktionen

Die FBs MOD_CLI/MOD_SERV verwenden folgende SFCs und SFBs aus der Standardbibliothek:

- SFC20 "BLKMOV"
- SFC24 "TEST_DB"
- SFB4 "TON"

Weitere Informationen

Der Parameter TI (Transaction Identifier) wird bei dem Baustein MODBUSPN im Client-Modus intern geführt und bei jedem neuen Auftrag um 1 inkrementiert.

Die Zeit, in welcher ein Verbindungsabbruch erkannt werden kann, kann mit dem Parameter Keep Alive-Time beeinflusst werden. Diesen Parameter finden Sie in den Eigenschaften der CPUs in der HW-Konfig.

6.5 Umbenennen / Umverdrahten von Funktionen und Funktionsbausteinen

Veranlassung

Falls in Ihrem Projekt die Nummern der Standardfunktionen bereits verwendet werden oder der Nummernbereich für andere Applikationen reserviert ist (z.B. in CFC), können Sie die intern aufgerufenen Funktionsbausteine FB63, FB64, FB65 und FB66 umverdrahten.

Der Baustein MODBUSPN ist BlockPrivacy-geschützt. Dadurch können die intern aufgerufenen Bausteine FB901 MOD_CLI und FB903 MOD_SERV nicht umverdrahtet werden.

Die Systemfunktionen SFC20, SFC24, SFC51 und SFC52 sowie der Systemfunktionsbaustein SFB4 können ebenfalls nicht umverdrahtet werden.

Umverdrahten

Gehen Sie zum Verdrahten für die FBs folgendermaßen vor:

1. Holen Sie mittels "Extras > Referenzdaten > Anzeigen" Informationen über die verwendeten Operanden ein.
2. Stellen Sie den Operandenvorrang in den Objekteigenschaften des Bausteinordners auf "Absolutwert".
3. Rufen Sie im SIMATIC Manager die Funktion "Extras > Umverdrahten" auf, um die Operanden in freie Bereiche umzuverdrahten.
4. Um die Symbolik in Diagnosetools weiter nutzen zu können, ziehen Sie die Änderungen in der Symboltabelle nach.

Über "Extras > Referenzdaten > Anzeigen" können Sie die Änderungen überprüfen.

Additional Blocks

7.1 Unterstützung in CFC

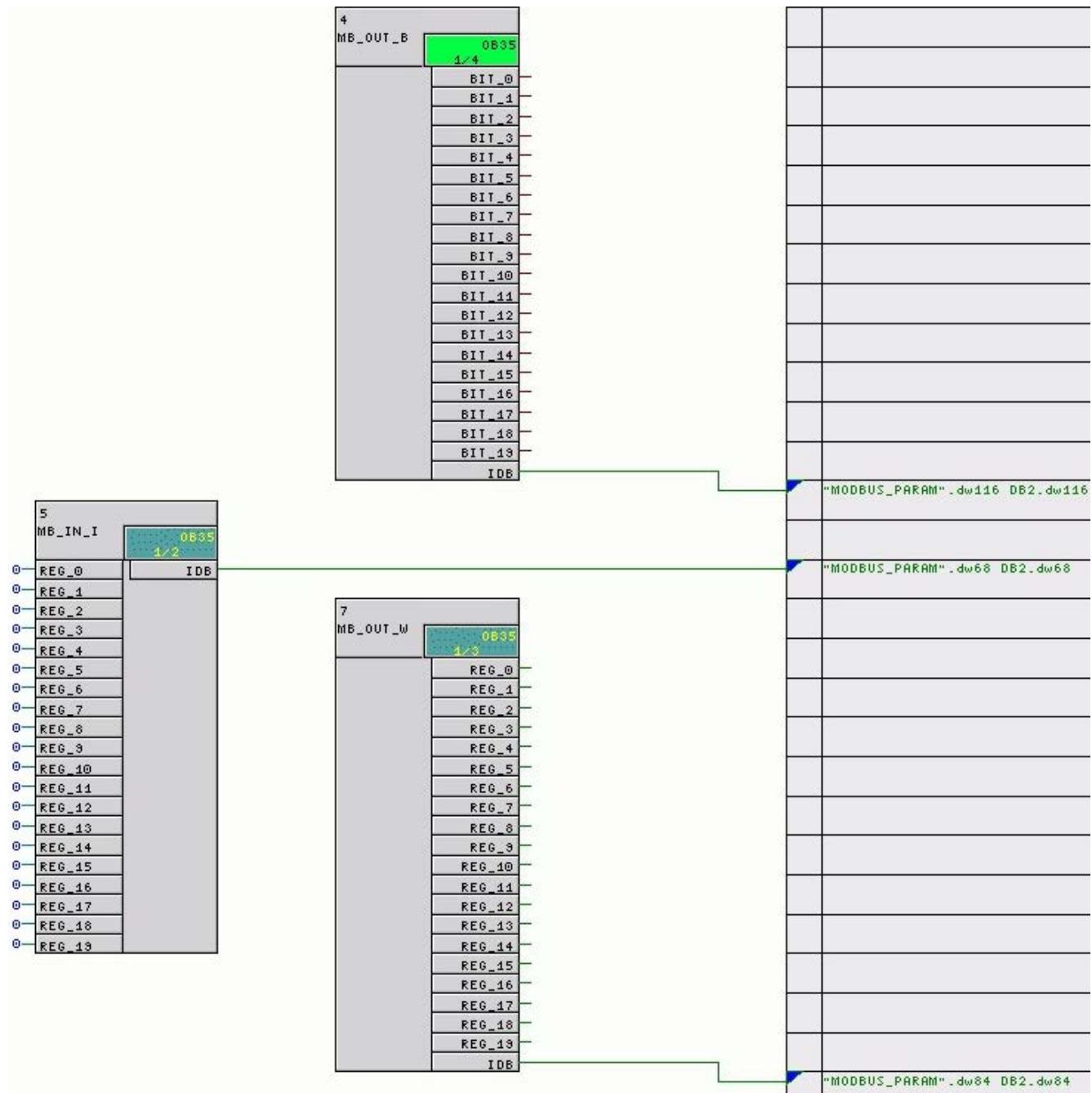
Allgemein

Um die Projektierung in CFC zu unterstützen, gibt es die Möglichkeit, die Modbuswerte nicht über Global-DBs sondern über "DataCollector-FBs" zu projektieren. Dabei werden die Sende- und Empfangspuffer für die Werte per Drag & Drop in den CFC-Plan gezogen.

Verwendung - Beispiel

Die DataCollector-FBs werden im CFC-Plan platziert. Der Ausgang "IDB" wird mit den DB-Parametern db_1 bis db_8 im Parameter-Datenbaustein verbunden.

Im weiteren Verlauf können die Modbuswerte direkt von den Channel-Bausteinen an den DataCollector-FB verschaltet werden.



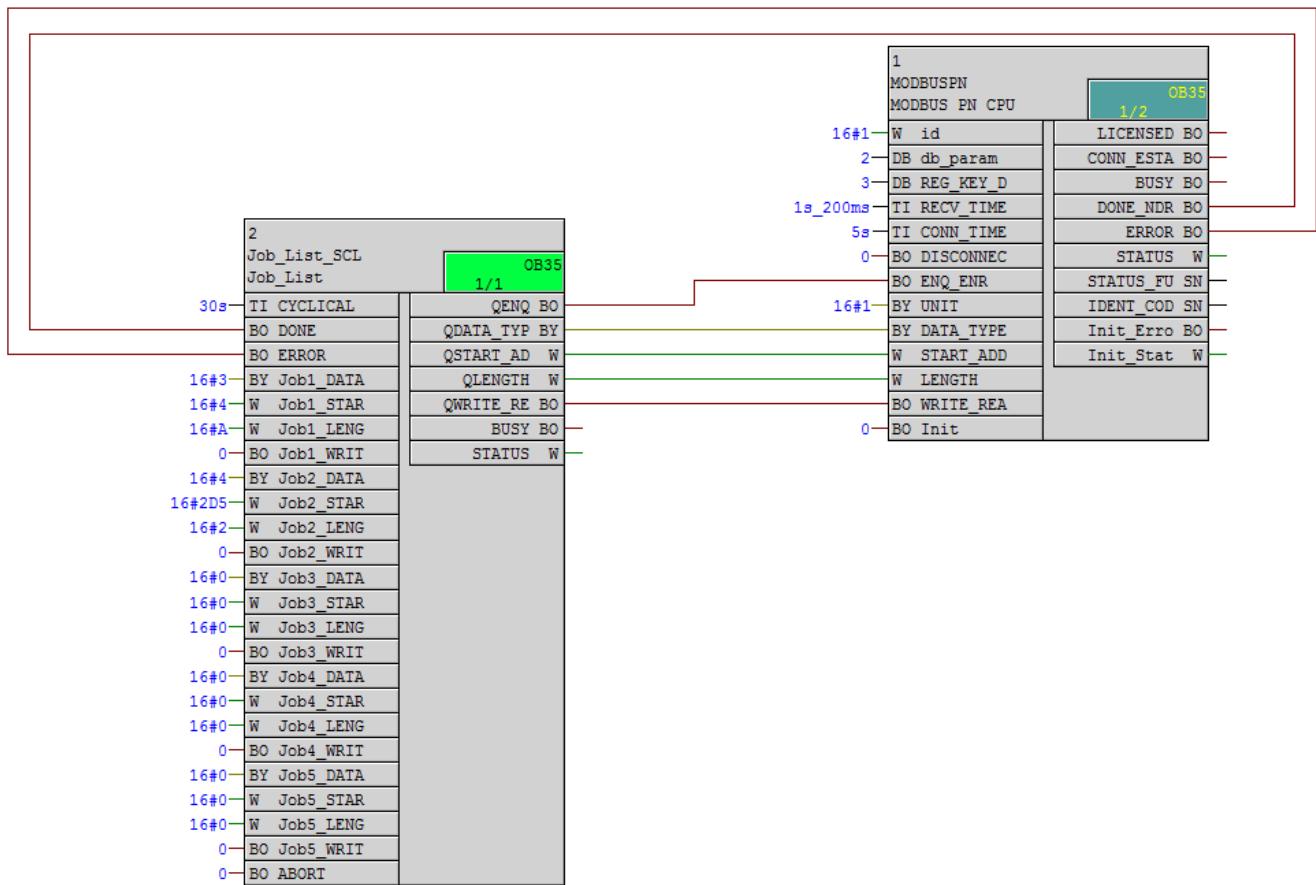
Die zusätzlichen Bausteine sowie eine detaillierte Beschreibung finden Sie hier (<http://support.automation.siemens.com/WW/view/de/62830463>).

7.2 Auftragsliste für zyklischen Telegrammverkehr

Allgemein

Mit dem Baustein Job_List ist eine Liste von Aufträgen parametrierbar, die zyklisch abgearbeitet wird.

Verwendungsbeispiel



Den zusätzlichen Baustein sowie eine detaillierte Beschreibung finden Sie hier (<http://support.automation.siemens.com/WW/view/de/62830463>).

Diagnose

Diagnosefunktionen

Die Diagnosefunktionen der PN-CPU erlauben Ihnen ein schnelles Lokalisieren aufgetretener Fehler. Folgende Diagnosemöglichkeiten stehen zur Verfügung:

- Diagnose über die Anzeigeelemente der CPUs
- Diagnose über die STATUS-Ausgänge des Modbusbausteins

Anzeigeelemente (LED)

Die Anzeigeelemente informieren Sie über den Betriebszustand bzw. über mögliche Fehlerzustände der CPUs. Die Anzeigeelemente geben Ihnen einen ersten Überblick über aufgetretene interne bzw. externe Fehler sowie schnittstellenspezifische Fehler.

STATUS-Ausgänge des MODBUSPN

Für eine Fehlerdiagnose besitzt der MODBUSPN-Funktionsbaustein STATUS-Ausgänge. Durch Lesen der STATUS-Ausgänge erhalten Sie allgemeine Aussagen zu Fehlern, die bei der Kommunikation aufgetreten sind. Die STATUS-Parameter können sie im Anwenderprogramm auswerten.

8.1 Diagnose über die Anzeigeelemente der CPU

Anzeigefunktionen

Über die Anzeigeelemente der CPU erhalten Sie Auskunft über den Baugruppenzustand. Zu unterscheiden sind folgende Anzeigefunktionen:

Tabelle 8- 1 Sammelstörungsanzeigen

PN-CPU 300 und IM 151-8/DP CPU	PN-CPU 400
SF Sammelfehler Blinkt diese LED, dann ist der Modbus-Baustein noch nicht lizenziert. Weitere Informationen entnehmen Sie dem Kapitel "Lizenzierung".	INTF Sammelfehler Blinkt diese LED, dann ist der Modbus-Baustein noch nicht lizenziert. Weitere Informationen entnehmen Sie dem Kapitel "Lizenzierung".

Tabelle 8- 2 Sonderanzeigen

PN-CPU 300 und IM 151-8 PN/DP	PN-CPU 400
RX/TX Ein Telegramm wird über die Schnittstelle übertragen.	

Eine detaillierte Beschreibung der Anzeigeelemente finden Sie im jeweiligen Gerätehandbuch der CPUs.

8.2 Diagnosemeldungen des FB MODBUSPN

Meldungen an den STATUS-Ausgängen des FBs

An den Status-Ausgängen des FBs MODBUSPN werden die Fehlermeldungen angezeigt.

- Der Ausgang STATUS gibt Fehlermeldungen und Statusinformationen während der Bausteinbearbeitung aus.
- Der Ausgang STATUS_FUNC gibt den Namen der Funktion aus, die den Fehler verursacht hat.
- Der Ausgang Init_Status gibt Fehlermeldungen und Statusinformationen während der Initialisierung aus.

Nachfolgend finden Sie eine Aufstellung der FB-spezifischen Fehlermeldungen.

Fehlermeldungen der aufgerufenen SFCs und FBs

Die Modbus-FBs benutzen die Standardbausteine SFC20, SFC24, SFC51 und SFC52. Die Fehlermeldungen dieser Bausteine werden unverändert an STATUS weitergegeben.

MOD_CLI bzw. MOD_SERV benutzen die Standardbausteine SFB4, FB63, FB64, FB65 und FB66. Die Fehlermeldungen dieser Bausteine werden ebenfalls unverändert an STATUS weitergegeben.

Im Diagnosepuffer oder der Online-Hilfe zu den SFCs/FCs aus dem SIMATIC Manager finden Sie weitere Hinweise zu diesen Fehlermeldungen.

Tabelle 8- 3 Fehlermeldungen des FBs MODBUSPN an STATUS und Init_Status

Status (Hex)	Ereignistext	Abhilfe
A001	Der Parameter-DB MODBUS_PARAM ist zu kurz oder zu lang.	Korrigieren Sie den DB MODBUS_PARAM.
A002	Der Parameter end_x ist kleiner als start_x.	Korrigieren Sie die Angaben im DB MODBUS_PARAM.
A003	<p>Ein DB, auf den MODBUS-Adressen abgebildet werden sollen, ist zu kurz. Minimallänge in Byte: - bei Registern: $(end_x - start_x + 1) * 2 + 2$ - bei Bitwerten: $(end_x - start_x) / 8 + 1 + 2$</p> <p>weitere mögliche Ursachen:</p> <ul style="list-style-type: none"> S7 ist Client: falsche Aufrufparameter S7 ist Server: falscher Adressbereich im Anforderungstelegramm des Clients. Die S7 antwortet mit einem Exceptiontelegramm 	Verlängern Sie den DB. <ul style="list-style-type: none"> S7 ist Client: Korrigieren Sie die Auftragsparameter START_ADDRESS oder LENGTH. S7 ist Server: Ändern Sie die Anforderung des Client.

Status (Hex)	Ereignistext	Abhilfe								
A004	Nur S7 ist Client: Es wurde eine unzulässige Kombination von DATA_TYPE und WRITE_READ angegeben.	Korrigieren Sie die Aufrufparameter. Es können nur die Datentypen 1 und 3 geschrieben werden.								
A005	S7 ist Client: Es wurde ein unzulässiger Wert am Parameter LENGTH angegeben S7 ist Server: Die Register-/Bitanzahl im Anforderungstelegramm ist unzulässig. Die S7 antwortet mit einem Exceptiontelegramm. Wertebereiche: <table border="1"><tr><td>Coils/Inputs lesen</td><td>1 bis 2000</td></tr><tr><td>Coils schreiben</td><td>1 bis 1968</td></tr><tr><td>Register lesen</td><td>1 bis 125</td></tr><tr><td>Holding Register schreiben</td><td>1 bis 123</td></tr></table>	Coils/Inputs lesen	1 bis 2000	Coils schreiben	1 bis 1968	Register lesen	1 bis 125	Holding Register schreiben	1 bis 123	S7 ist Client: Korrigieren Sie den Parameter LENGTH. S7 ist Server: Verändern Sie im Anforderungstelegramm des Client die Anzahl.
Coils/Inputs lesen	1 bis 2000									
Coils schreiben	1 bis 1968									
Register lesen	1 bis 125									
Holding Register schreiben	1 bis 123									
A006	Der über DATA_TYPE, START_ADDRESS und LENGTH angegebene Bereich existiert nicht in data_type_1 bis data_type_8. S7 ist Server: Die S7 antwortet mit einem Exceptiontelegramm.	S7 ist Client: Korrigieren Sie die Kombination DATA_TYPE, START_ADDRESS und LENGTH. S7 ist Server: Ändern Sie die Anforderung des Client oder korrigieren Sie die Parametrierung im Parameter-DB.								
A007	S7 ist Client: Es wurde eine ungültige Überwachungszeit an RECV_TIMEOUT oder CONN_TIMEOUT parametriert. Für RECV_TIMEOUT muss ein Wert \geq 20 ms, für CONN_TIMEOUT \geq 100 ms eingetragen werden.	Korrigieren Sie die Parametrierung.								
A009	S7 ist Client: Der empfangene Transaction Identifier TI ist ungleich dem gesendeten. Die Verbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.								
A00A	S7 ist Client: Die empfangene UNIT ist ungleich der gesendeten.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.								
A00B	S7 ist Client: Der empfangene Funktionscode ist ungleich dem gesendeten. S7 ist Server: Es wurde ein ungültiger Funktionscode empfangen. Die S7 antwortet mit einem Exceptiontelegramm.	S7 ist Client: Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners. S7 ist Server: Ändern Sie die Anforderung des Client. Der Modbus-FB bearbeitet die Funktionscodes 1, 2, 3, 4, 5, 6, 15 und 16.								
A00C	Der empfangene Bytecount passt nicht zur Registerzahl. Die Verbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.								
A00D	S7 ist Client: Die Register-/Bitadresse oder Register-/Bitanzahl im Antworttelegramm ist ungleich der im Anforderungstelegramm.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.								

Status (Hex)	Ereignistext	Abhilfe
A00E	Die Längenangabe im modbusspezifischen Telegrammheader passt nicht zu den Angaben der Register-/Bitanzahl oder des Bytecount im Telegramm. Der FB verwirft die Daten. Die Verbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners
A00F	Es wurde ein Protocol Identifier ungleich 0 empfangen. Die Verbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners
A010	Bei den Parametern db_1 bis db_8 wurde eine DB-Nummer doppelt vergeben.	Korrigieren Sie die Parametrierung im DB MODBUS_PARAM.
A011	Am Eingangsparameter DATA_TYPE wurde ein unzulässiger Wert angegeben (zulässige Werte sind 1- 4).	Korrigieren Sie die Aufrufparameter.
A012	Die parametrierten Bereiche data_type_1 und data_type_2 überlappen.	Korrigieren Sie die Parametrierung.
A013	Die parametrierten Bereiche data_type_1 und data_type_3 überlappen.	Die Datenbereiche dürfen keinen gemeinsamen Registeradressbereich besitzen.
A014	Die parametrierten Bereiche data_type_1 und data_type_4 überlappen.	
A015	Die parametrierten Bereiche data_type_1 und data_type_5 überlappen.	
A016	Die parametrierten Bereiche data_type_1 und data_type_6 überlappen.	
A017	Die parametrierten Bereiche data_type_1 und data_type_7 überlappen.	
A018	Die parametrierten Bereiche data_type_1 und data_type_8 überlappen.	
A019	Einer der Parameter db_x wurde auf 0 gesetzt, obwohl der zugehörige data_type_x mit > 0 parametriert ist. DB0 darf nicht verwendet werden, weil dieser für das System reserviert ist.	Korrigieren Sie die Parametrierung an db_x auf > 0.
A01A	Falsche Länge im Header: Es sind 3 bis 253 Byte zulässig. Die Verbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A01B	S7 ist Server und Funktionscode 5: Es wurde ein ungültiger Zustand für Coil empfangen. Die S7 antwortet mit einem Exceptiontelegramm.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A023	Die parametrierten Bereiche data_type_2 und data_type_3 überlappen.	Korrigieren Sie die Parametrierung.
A024	Die parametrierten Bereiche data_type_2 und data_type_4 überlappen.	Die Datenbereiche dürfen keinen gemeinsamen Registeradressbereich besitzen.
A025	Die parametrierten Bereiche data_type_2 und data_type_5 überlappen.	
A026	Die parametrierten Bereiche data_type_2 und data_type_6 überlappen.	
A027	Die parametrierten Bereiche data_type_2 und data_type_7 überlappen.	

Status (Hex)	Ereignistext	Abhilfe
A028	Die parametrierten Bereiche data_type_2 und data_type_8 überlappen.	
A034	Die parametrierten Bereiche data_type_3 und data_type_4 überlappen.	
A035	Die parametrierten Bereiche data_type_3 und data_type_5 überlappen.	
A036	Die parametrierten Bereiche data_type_3 und data_type_6 überlappen.	
A037	Die parametrierten Bereiche data_type_3 und data_type_7 überlappen.	
A038	Die parametrierten Bereiche data_type_3 und data_type_8 überlappen.	
A045	Die parametrierten Bereiche data_type_4 und data_type_5 überlappen.	
A046	Die parametrierten Bereiche data_type_4 und data_type_6 überlappen.	
A047	Die parametrierten Bereiche data_type_4 und data_type_7 überlappen.	
A048	Die parametrierten Bereiche data_type_4 und data_type_8 überlappen.	
A056	Die parametrierten Bereiche data_type_5 und data_type_6 überlappen.	
A057	Die parametrierten Bereiche data_type_5 und data_type_7 überlappen.	
A058	Die parametrierten Bereiche data_type_5 und data_type_8 überlappen.	
A067	Die parametrierten Bereiche data_type_6 und data_type_7 überlappen.	
A068	Die parametrierten Bereiche data_type_6 und data_type_8 überlappen.	
A078	Die parametrierten Bereiche data_type_7 und data_type_8 überlappen.	
A079	Die am Parameter id angegebene Verbindungs-ID ist im Parameter-DB MODBUS_PARAM nicht enthalten.	Korrigieren Sie die Parametrierung am Eingang id.
A07A	An Parameter id des Bausteins wurde ein unzulässiger Wert (Wertebereich von 1 bis 4095) vergeben.	
A07B	Die angegebene id ist im Parameter-DB doppelt enthalten.	
A07C	Im Parameter-DB wurde ein unzulässiger Wert am Parameter data_type_x angegeben (zulässige Werte sind 0 bis 4).	Korrigieren Sie die Parametrierung im DB MODBUS_PARAM.
A07D	Im Parameter-DB enthält der Parameter data_type_1 keinen Eintrag. Der Parameterbereich "_1" ist der Initialbereich und muss parametriert werden.	
A07E	An db_x wurde die Nummer des Parameter-DBs MODBUS_PARAM oder die Nummer des Instanz-DBs vom Baustein MODBUSPN angegeben.	

Status (Hex)	Ereignistext	Abhilfe
A07F	Der an db_param angegebene DB ist kein Modbus-Parameter-DB. Es wurde die Längeninformation im DBW0 geändert oder ein falscher DB angegeben.	Korrigieren Sie die Parametrierung am Eingang db_param.
A080	Der Modbusbaustein wurde noch nicht initialisiert.	Nach dem Transferieren des IDBs in die CPU muss der Modbusbaustein mit Init = TRUE initialisiert werden.
A081	Nur bei S7 ist Client und Funktionscode 5: Die Daten des Antworttelegramms sind nicht das Echo der Anforderung.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A082	Nur bei S7 ist Client und Funktionscode 6: Der empfangene Registerwert ist ungleich dem gesendeten.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A083	S7 ist Client: Es ist ein Auftrag angestoßen worden, während der vorherige Auftrag noch läuft. Der Auftrag wird nicht ausgeführt. Dies ist eine Statusinformation. Das Bit ERROR ist nicht gesetzt. Es wurde versucht, den Baustein zu initialisieren während noch ein Auftrag läuft bzw. während ENQ_ENR gesetzt war.	Client: Stoßen Sie erst dann einen neuen Auftrag an, wenn der vorherige Auftrag mit DONE_NDR = TRUE oder ERROR = TRUE beendet wurde. Warten Sie mit der Initialisierung bis kein Auftrag mehr läuft. Setzen Sie ENQ_ENR = FALSE.
A084	Es konnte keine Identifizierungskennung IDENT_CODE für die Lizenzierung ermittelt werden.	Wenden Sie sich an den Produkt Support.
A085	Es trat ein Fehler bei der Lizenzermittlung auf. Der Fehler wird beim 1. Auftreten mit ERROR = TRUE angezeigt. Im Folgenden wird es als Statusmeldung mit ERROR = FALSE angezeigt.	Prüfen Sie, dass keine unerlaubten Schreib-Zugriffe auf den Lizenz-DB vorhanden sind. Die Struktur des REG_KEYs darf nicht verändert werden. Wenden Sie sich gegebenenfalls an den Produkt Support.
A086	Es wurde versucht, in einen schreibgeschützten Datenbaustein zu schreiben.	Entfernen Sie den Schreibschutz des Datenbausteins oder verwenden Sie einen anderen DB.
A090	Der Modbus-Baustein ist für diese CPU noch nicht lizenziert. Dies ist eine Statusinformation. Das Bit ERROR ist nicht gesetzt. Die Modbus-Kommunikation läuft auch ohne Lizenz.	Lesen Sie den Identifikationsstring IDENT_CODE für diese CPU aus und fordern Sie den Registrierungsschlüssel an (siehe Kapitel 5 "Lizenzierung").
A091	Nur bei S7 ist Client: Als Antwort wurde ein Exception Telegramm mit Exception Code 1 empfangen.	Der Koppelpartner unterstützt die angeforderte Funktion nicht.
A092	Nur bei S7 ist Client: Als Antwort wurde ein Exception Telegramm mit Exception Code 2 empfangen. Es erfolgte ein Zugriff auf eine nicht vorhandene/nicht zulässige Adresse beim Koppelpartner.	Korrigieren Sie LENGTH bzw. START_ADDRESS beim FB-Aufruf.

Status (Hex)	Ereignistext	Abhilfe
A093	Nur bei S7 ist Client: Als Antwort wurde ein Exception Telegramm mit Exception Code 3 empfangen.	Der Koppelpartner kann das empfangene Telegramm nicht verarbeiten (z.B. er unterstützt die angeforderte Länge nicht).
A094	Nur bei S7 ist Client: Als Antwort wurde ein Exception Telegramm mit Exception Code 4 empfangen.	Der Koppelpartner befindet sich in einem Zustand in dem er das empfangene Telegramm nicht bearbeiten kann.
A095	Nur bei S7 ist Client: Als Antwort wurde ein Exception Telegramm mit einem unbekannten Exception Code empfangen.	Kontrollieren Sie die Fehlermeldungen des Koppelpartners und überprüfen Sie ggf. die Daten mit einer Telegrammaufzeichnung.
A100	Für einen Auftrag ist die Überwachungszeit CONN_TIMEOUT oder RECV_TIMEOUT abgelaufen. Bei Ablauf der RECV_TIMEOUT wird die Verbindung abgebaut.	Prüfen Sie die Parametrierung der Verbindung.
A101	Die interne Überwachungszeit der Funktion TDISCON ist abgelaufen.	Kontaktieren Sie den Produkt Support.

8.3 Diagnosemeldungen der eingebundenen Bausteine

Diagnosemeldung

Tabelle 8- 4 Fehlermeldungen der eingebundenen FBs/SFCs am Ausgang STATUS

STATUS (Hex)	Ereignistext	Abhilfe
7xxx	Bitte entnehmen Sie die detaillierte Information der Online-Hilfe des SIMATIC Manager.	Siehe Online-Hilfe (SIMATIC Manager -> Baustein markieren -> Taste F1)
8xxx	Bitte entnehmen Sie die detaillierte Information der Online-Hilfe des SIMATIC Manager.	Siehe Online-Hilfe (SIMATIC Manager -> Baustein markieren -> Taste F1)

8.4 Diagnosemeldungen des SFC24

Diagnosemeldung

Tabelle 8- 5 Fehlermeldungen des SFC24 am Ausgang STATUS

STATUS (Hex)	Ereignistext	Abhilfe
80A1	DB Nummer = 0 oder zu groß für die CPU.	Wählen Sie eine zulässige DB Nummer.
80B1	Der DB existiert nicht auf der CPU.	Alle Datenbausteine, die an db_x angegeben werden, müssen angelegt und auf die CPU übertragen werden.
80B2	DB UNLINKED	DB nicht als UNLINKED generieren.

Applikationsbeispiele

Allgemein

Mit der Installation werden 2 Beispielprojekte unter \Program Files\Siemens\Step7\Examples abgelegt:

- 1 Beispielprojekt in AWL "MB_PN_CPU" und
- 1 Beispielprojekt in CFC "MB_PN_CPU_CFC".

In den Beispielprojekten sind für alle Funktionsvarianten Simatic-Stationen angelegt:

- SIMATIC-Station ist Client oder Server
- SIMATIC-Station ist S7-300, S7-400 oder IM 151-8 PN/DP CPU

Hinweis

Das S7-Programm soll als Informationsquelle dienen und kann nicht als verbindlicher Lösungsvorschlag kundenspezifischer Anlagenkonfigurationen betrachtet werden.

Programmbeispiel

Die Programmbeispiele bestehen aus den Bausteinen:

- Anlauf-Baustein OB100 mit Setzen des Init-Bits
- Programmierfehler-OB121
- Zyklischer Betrieb OB1 bzw. OB35 mit Aufruf des MODBUSPN
- Globale Datenbausteine zum Auftragsanstoß (z.B. mit Hilfe einer Variablenliste) und für die Lizenzierung
- Datenbausteine für Register- und Bitwerte

9.1 Beispielprojekt in AWL

Übersicht

MODBUS_PN_CPU_EXAMPLE (Komponentensicht) -- C:\Program Files (x86)\SIEMENS\STEP7\Examples\MB_PN_CPU				
	Objektname	Symbolischer Name	Erstellsprache	Typ
MODBUS_PN_CPU_EXAMPLE	System data	---	---	SDB
SIMATIC 400(Client)	OB1	CYCL_EXC	AWL	Organisationsbaustein
CPU 414-3 PN/DP	OB100	COMPLETE RESTART	AWL	Organisationsbaustein
S7-Program(1)	OB121	PROG_ERR	AWL	Organisationsbaustein
Sources	FB63	TSEND	AWL	Funktionsbaustein
Blocks	FB64	TRCV	AWL	Funktionsbaustein
SIMATIC 400(Server)	FB65	TCON	AWL	Funktionsbaustein
ET200S(Client)	FB66	TDISCON	AWL	Funktionsbaustein
ET200S(Server)	FB900	MODBUSPN	ENCRYPT	Funktionsbaustein
SIMATIC 300(Client)	FB901	MOD_CLI	SCL	Funktionsbaustein
SIMATIC 300(Server)	FB903	MOD_SERV	SCL	Funktionsbaustein
	DB1	CONTROL_DAT	DB	Datenbaustein
	DB2	MODBUS_PARAM	DB	Datenbaustein
	DB3	LICENSE_DB	DB	Datenbaustein
	DB11	Holding Register Area	DB	Datenbaustein
	DB12	Holding Register Area2	DB	Datenbaustein
	DB13	Input Register Area	DB	Datenbaustein
	DB14	Coils Area	DB	Datenbaustein
	DB15	Inputs Area	DB	Datenbaustein
	DB16	Coils Area 2	DB	Datenbaustein
	DB900	IDB_MODBUS	DB	Instanzdatenbaustein
	Client_Job	Client_Job		Variablenliste

Verwendete Bausteine

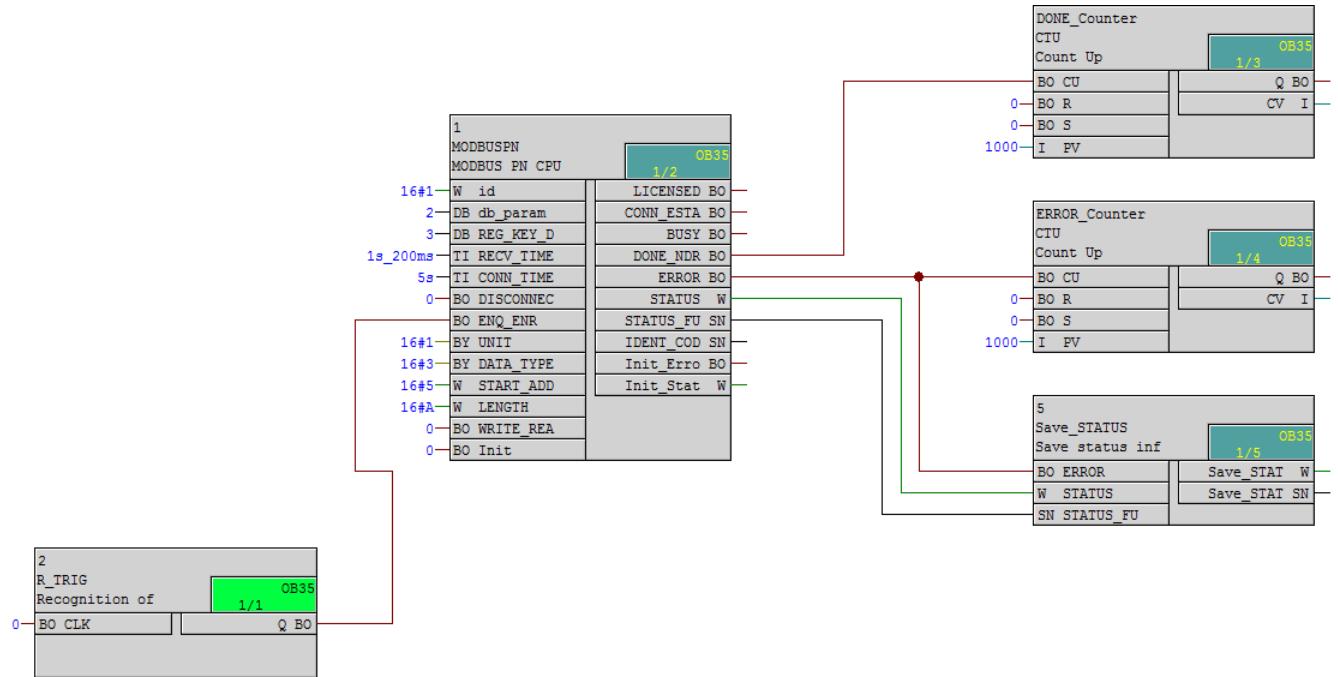
Diese Bausteinnummern werden im mitgelieferten Beispielprojekt verwendet.

Baustein	Symbol	Kommentar
OB 1	CYCL_EXC	Zyklische Programmbearbeitung
OB 100	COMPLETE RESTART	Anlauf-OB
OB 121	PROG_ERR	Programmierfehler-OB
FB 900	MODBUSPN	Anwenderbaustein FB MODBUSPN
FB 901	MOD_CLI	intern aufgerufener FB MOD_CLI
FB 903	MOD_SERV	intern aufgerufener FB MOD_SERV
DB 1	CONTROL_DAT	Arbeits-DB CONTROL DAT für MODBUSPN
DB 2	MODBUS_PARAM	Parameter-DB für MODBUSPN
DB 3	LICENSE_DB	Lizenz-DB für MODBUSPN
DB 11	Holding Register Area	Werte-DB für Bereich 1
DB 12	Holding Register Area 2	Werte-DB für Bereich 2
DB 13	Input Register Area	Werte-DB für Bereich 3
DB 14	Coils Area	Werte-DB für Bereich 5
DB 15	Inputs Area	Werte-DB für Bereich 6
DB 16	Coils Area 2	Werte-DB für Bereich 7
DB 900	IDB_MODBUS	Instanz-DB für MODBUSPN

9.2 Beispielprojekt in CFC

Übersicht

Das Beispielprojekt wurde mit CFC V8.0 Update 1 erstellt.



Verwendete Bausteine

Diese Bausteinnummern werden im mitgelieferten Beispielprojekt verwendet.

Baustein	Symbol	Kommentar
OB 35	CYCL_EXC	Zyklische Programmbearbeitung
OB 100	COMPLETE RESTART	Anlauf-OB für die Initialisierung
OB 121	PROG_ERR	Programmierfehler-OB
FB 99	Save_STATUS	Speicher-DB für Fehler
FB 900	MODBUSPN	Anwenderbaustein MODBUSPN
FB 901	MOD_CLI	intern aufgerufener FB MOD_CLI
FB 903	MOD_SERV	intern aufgerufener FB MOD_SERV
DB 2	MODBUS_PARAM	Parameter-DB für FB MODBUSPN
DB 3	LICENSE_DB	Lizenz-DB für MODBUSPN
DB 11	Holding Register Area	Werte-DB für Bereich 1
DB 12	Holding Register Area 2	Werte-DB für Bereich 2
DB 13	Input Register Area	Werte-DB für Bereich 3
DB 14	Coils Area	Werte-DB für Bereich 5
DB 15	Inputs Area	Werte-DB für Bereich 6
DB 16	Coils Area 2	Werte-DB für Bereich 7

Literatur

A

The MODBUS Organization

MODBUS APPLICATION PROTOCOL SPECIFICATION
V1.1b3, April, 2012

Modbus Homepage (<http://www.modbus.org>)

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
DEUTSCHLAND

Änderungen vorbehalten
A5E39465005-AB
© Siemens AG 2017