

SIMATIC

MODBUS (TCP)

Programmier- und Bedienhandbuch

## 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.

 <b>GEFAHR</b>
bedeutet, dass Tod oder schwere Körperverletzung eintreten <b>wird</b> , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 <b>WARNUNG</b>
bedeutet, dass Tod oder schwere Körperverletzung eintreten <b>kann</b> , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 <b>VORSICHT</b>
bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

<b>ACHTUNG</b>
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:

 <b>WARNUNG</b>
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.

# Inhaltsverzeichnis

<b>1</b>	<b>MODBUSPN: Als Modbus/TCP-Client oder Modbus/TCP-Server kommunizieren.....</b>	<b>5</b>
1.1	Allgemeine Informationen.....	5
1.2	Inbetriebnahme.....	5
1.3	Parameter-Datenbaustein.....	7
1.4	Beschreibung MODBUSPN.....	13
1.5	Lizenzierung mit den Parametern IDENT_CODE und REG_KEY.....	17
1.6	Adressabbildung.....	21
1.7	Parameter ID und DB_PARAM.....	24
1.8	Parameter RECV_TIME und CONN_TIME.....	25
1.9	Parameter ENQ_ENR und DISCONNECT.....	25
1.10	Parameter DATA_TYPE, START_ADDRESS, LENGTH, TI, WRITE_READ und UNIT.....	26
1.11	Parameter ERROR, STATUS_MODBUS, STATUS_CONN und STATUS_FUNC.....	28
	<b>Index.....</b>	<b>35</b>



# MODBUSPN: Als Modbus/TCP-Client oder Modbus/TCP-Server kommunizieren

# 1

## 1.1 Allgemeine Informationen

### Allgemein

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

Mit dieser Anweisung wird eine Kommunikation zwischen einer SIMATIC-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.

### Schritt-für-Schritt Anleitung

1. Zuweisung der IP-Adresse der CPU
2. Aufruf der Anweisung MODBUSPN in den notwendigen OBs - siehe Inbetriebnahme (Seite 5)
3. Parametrierung des Parameter-DBs entsprechend den Anforderungen (ID, Portnummer, Client/Server, Verbindungsaufbau bei Neustart, Modbusregister, DB-Bereiche etc.) - siehe Parameter-Datenbaustein (Seite 7)
4. Parametrierung des Modbusbausteins für die Initialisierung und für die Laufzeit - siehe Funktionsweise der Anweisung (Seite 13)
5. Laden des Anwenderprogramms in die CPU und Lizenzierung des Modbusbausteins für diese CPU - siehe Lizenzierung (Seite 17)

## 1.2 Inbetriebnahme

### Voraussetzung und Grundlage

Der Einsatz der Anweisung MODBUSPN ist ab **STEP7 V13 (TIA Portal)** möglich.

Die Anweisung MODBUSPN wurde auf Grundlage der Modbus Application Protocol Specification V1.1b3, April 26, 2012 erstellt - siehe Modbus Homepage (<http://www.modbus.org>).

## Aufruf der Anweisung

Die Anweisung MODBUSPN muss für einen korrekten Programmablauf in 2 OBs eingebaut werden:

- im Anlauf-OB100 und
- in einem zyklischen OB (OB1 oder in einem zeitgesteuerten OB, z.B. OB35)

Dabei muss derselbe Instanz-Datenbaustein verwendet werden. Der gleichzeitige Aufruf der Anweisung 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 unter Lizenzierung (Seite 17).

## Einfügen des Modbusbausteins

Öffnen Sie den Baustein „COMPLETE RESTART [OB100]“. Falls dieser in den Programmbausteinen noch nicht vorhanden ist, fügen Sie ihn mit „Neuen Baustein hinzufügen > Organisationsbaustein > Startup > COMPLETE RESTART [OB 100]“ ein. Die Anweisung MODBUSPN ist in der Task Card, Palette und Ordner „Anweisungen > Kommunikation > Weitere“ enthalten. Öffnen Sie darin den Ordner „MODBUS TCP“ und ziehen Sie die Anweisung MODBUSPN in den OB100-Baustein.

Unter „Systembausteine > Programmressourcen“ werden nun neben dem MODBUSPN (FB70) die unterlagerten Anweisungen MOD\_CLI (FB72), MOD\_SERV (FB73) und TCP\_COMM (FB71) angezeigt. Diese dürfen nicht zusätzlich in einem OB aufgerufen werden. Weiterhin werden die ebenfalls intern aufgerufenen Kommunikationsanweisungen TSEND (FB63), TRCV (FB64), TCON (FB65) und TDISCON (FB66) angezeigt.

---

### Hinweis

**Beachten Sie, dass folgende Versionen Voraussetzung für den einwandfreien Betrieb der Anweisung MODBUSPN sind:**

TSEND	V3.0
TRCV	V3.0
TCON	V3.0
TDISCON	V2.1

---

Öffnen Sie den Baustein “Main [OB1]” oder einen zyklischen Baustein und ziehen Sie die Anweisung MODBUSPN [FB70] in den OB. Wählen Sie als Instanz-Datenbaustein den Datenbaustein MODBUSPN\_DB aus dem OB100-Aufruf aus. Es darf kein neuer Instanz-Datenbaustein angelegt werden.

## Mehrfache Client- und Server-Verbindungen

Eine S7-CPU kann mehrere TCP-Verbindungen unterstützen, wobei die maximale Anzahl der Verbindungen von der verwendeten CPU abhängt. Die gesamte Anzahl der Verbindungen einer CPU, inklusive der Modbus/TCP-Verbindungen, darf die maximale Anzahl der unterstützten Verbindungen nicht überschreiten.

## Verwendung der Portnummer 502

Das Protokoll Modbus/TCP läuft üblicherweise über den Port 502. Diese Portnummer ist nur für PN-CPU's mit entsprechender Firmwareversion möglich. Die Information bezüglich der Freigabe der Portnummern finden Sie hier: "Welche Ports sind für die Modbus/TCP-Kommunikation freigegeben und wie viele Modbus-Clients können mit einer SIMATIC S7-CPU als Modbus-Server kommunizieren? (<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 (Multiport). Bei der Parametrierung müssen folgende Einstellungen vorgenommen werden:

- CPU ist Server
- Port 502 als lokaler Port
- unspezifizierte TCP-Verbindung
- passiver Verbindungsaufbau

Die Anzahl der Verbindungen, die eine CPU auf dem Port 502 annehmen kann, ist geräteabhängig und kann den technischen Daten der CPU entnommen werden. Für jeden Client, der sich auf den Port 502 des Servers verbinden will, ist je 1 eindeutige Verbindung im Parameter-DB und je 1 Modbusbaustein-Instanz im OB100 bzw. zyklischen OB notwendig.

## 1.3 Parameter-Datenbaustein

### Parametrieren der Modbus-Kommunikation

Für die Kommunikation über die integrierte PN-Schnittstelle der CPU ist keine Verbindungsprojektierung im Netzwerkeditor notwendig. Die Verbindungen werden mit Hilfe der Anweisungen TCON und TDISCON auf- bzw. abgebaut.

#### Parameter-Datenbaustein

Die für den Aufbau der Verbindungen und die Bearbeitung der Modbustelegramme notwendigen Daten werden in dem PLC-Datentyp **MB\_PN\_PARAM** definiert. Dieser PLC-Datentyp enthält eine Struktur für die verbindungs-spezifischen Daten und eine Struktur für die Modbusparameter.

Für jede Verbindung zu einem Kommunikationspartner wird in einem Datenbaustein 1 Instanz des PLC-Datentyps benötigt, in der die Verbindungsparameter und die Modbusparameter definiert werden. Für jede weitere Verbindung kann der Datenbaustein erweitert oder ein neuer Datenbaustein angelegt werden.

Dieser Datenbaustein bzw. diese Datenbausteine sind nur für die Verbindungs- und Modbusparameter vorgesehen, es dürfen keine anderen Parameter darin gespeichert werden.

Jede Instanz der Anweisung "MODBUSPN" benötigt eine eindeutige Verbindung. Erstellen Sie entsprechend für jede Instanz der Anweisung eine eigene Struktur der Verbindungsbeschreibung.

1.3 Parameter-Datenbaustein

PARAM_DB								
Name	Datentyp	Offset	Startwert	Remanenz	Sichtbar i...	Einstellwert	Kommentar	
1	Static							
2	Connection_1	MB_PN_PARAM			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	Connection settings	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Einstellungen für die Verbindungsparameter
4	block_length	Word	...	W#16#0040	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Länge der Connection_settings: 64 Bytes (fest)
5	id	Word	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Referenz auf diese Verbindung (Wertebereich: ...)
6	connection_type	Byte	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		B#16#11: TCP/IP native; B#16#12: ISO on TCP; ...
7	active_est	Bool	...	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE: passiver Verbindungsaufbau; TRUE: akt...
8	local_device_id	Byte	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Erlaubte Werte: B#16#0, B#16#2, B#16#3, B...
9	local_tsap_id_len	Byte	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Verwendete Länge des Parameters local_tsap...
10	rem_subnet_id_len	Byte	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unbenutzt; muss B#16#00 sein
11	rem_staddr_len	Byte	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Bedeutung des Parameters rem_staddr: B#16...
12	rem_tsap_id_len	Byte	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Verwendete Länge des Parameters rem_tsap_...
13	next_staddr_len	Byte	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		B#16#1 bei local_device_id = 0; sonst B#16#0...
14	local_tsap_id	Array[1..16] of Byte	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Abhängig von dem Parameter connection_ty...
15	rem_subnet_id	Array[1..6] of Byte	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unbenutzt; muss B#16#00 sein
16	rem_staddr	Array[1..6] of Byte	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IP-Adresse des remoten Verbindungsendpunk...
17	rem_tsap_id	Array[1..16] of Byte	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Abhängig von dem Parameter connection_ty...
18	next_staddr	Array[1..6] of Byte	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Abhängig von dem Parameter local_device_id...
19	spare	Word	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unbenutzt; muss B#16#00 sein
20	Modbus settings	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Einstellungen für die Modbusparameter
21	server_client	Bool	...	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE: S7 ist Client; TRUE: S7 ist Server
22	single_write	Bool	...	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE: Verwendung der Funktionscodes 15 u...
23	connect_at_startup	Bool	...	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE: Verbindungsaufbau bei gesetztem EN...
24	reserved	Byte	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Unbenutzt; muss B#16#00 sein
25	data_areas	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Datenbereiche
26	data_area_1	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Datenbereich 1
27	data_type	Byte	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		1: Coils; 2: Inputs; 3: Holding Register; 4: Input...
28	db	Word	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		DB-Nummer für die Datenblage
29	start	Word	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Erste gespeicherte Register/Bit-Adresse im Da...
30	end	Word	...	16#0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Letzte gespeicherte Register/Bit-Adresse im Da...
31	data_area_2	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Datenbereich 2
32	data_area_3	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Datenbereich 3
33	data_area_4	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Datenbereich 4
34	data_area_5	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Datenbereich 5
35	data_area_6	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Datenbereich 6
36	data_area_7	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Datenbereich 7
37	data_area_8	Struct	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Datenbereich 8
38	internal_send_buffer	Array[1..260] of Byte	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Für internen Gebrauch
39	internal_recv_buffer	Array[1..260] of Byte	...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Für internen Gebrauch

Bild 1-1 Aufbau Param\_DB

**Verbindungsparameter in der Struktur "Connection settings"**

In der 1. Struktur "Connection settings" werden die verbindungs-spezifischen Parameter, wie z.B. die lokal verwendete Hardwareschnittstelle und die IP-Adresse des Kommunikationspartners, definiert. Mit Hilfe dieser Parameter können die Anweisungen TCON und TDISCON eine Verbindung auf- bzw. abbauen.

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

**Modbusparameter in der Struktur "Modbus settings"**

In der 2. Struktur - den "Modbus settings" - 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.

### Projektierung

Es gibt 2 Möglichkeiten die Projektierung für die Verbindungs- und Modbusparameter vorzunehmen.

1. Möglichkeit:

Erstellen Sie einen neuen Global-Datenbaustein und öffnen Sie diesen. Fügen Sie einen Parameter hinzu und wählen Sie für diesen Parameter den Datentyp **MB\_PN\_PARAM**. Falls dieser Datentyp in der Auswahlliste noch nicht angezeigt wird, tragen Sie ihn manuell ein.

Bei dieser Möglichkeit ist das Einfügen von mehreren Instanzen in 1 Datenbaustein möglich.

2. Möglichkeit:

Erstellen Sie mit „Neuen Baustein hinzufügen“ einen neuen Datenbaustein und wählen Sie als „Typ“ **MB\_PN\_PARAM** aus. Es öffnet sich der neue Datenbaustein mit der eingefügten Verbindungs- und Modbusstruktur.

Dieser Baustein ist schreibgeschützt. Es können keine weiteren Parameter hinzugefügt werden. Das Editieren der vorhandenen Parameter ist möglich.

### Änderung der Werte

Die Werte im Parameter-Datenbaustein dürfen während der Laufzeit nicht verändert werden. Nach einer Änderung der Parameter muss die CPU mit STOP -> RUN neu gestartet werden.

### Verbindungsparameter "Connection settings"

Parameter	Beschreibung
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 im gesamten Parameter-Datenbaustein eindeutig sein. Die ID wird beim Aufruf der Anweisung 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 Anweisung 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 und WinAC RTX Je nach verwendeter Firmware können diese Angaben variieren.
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

1.3 Parameter-Datenbaustein

Parameter	Beschreibung														
local_device_id	<p>Die local_device_id definiert die IE-Schnittstelle der verwendeten PN-CPU. Je nach PN-CPU-Typ werden unterschiedliche Einstellungen benötigt.</p> <table border="1"> <tr> <td>IM 151-8 PN/DP CPU, WinAC RTX, IF 1:</td> <td>B#16#1</td> </tr> <tr> <td>CPU 314C, 315 bzw. 317:</td> <td>B#16#2</td> </tr> <tr> <td>CPU 319:</td> <td>B#16#3</td> </tr> <tr> <td>CPU 412, 414 bzw. CPU 416</td> <td>B#16#5</td> </tr> <tr> <td>WinAC RTX, IF 2</td> <td>B#16#6</td> </tr> <tr> <td>WinAC RTX, IF 3</td> <td>B#16#B</td> </tr> <tr> <td>WinAC RTX, IF 4</td> <td>B#16#F</td> </tr> </table>	IM 151-8 PN/DP CPU, WinAC RTX, IF 1:	B#16#1	CPU 314C, 315 bzw. 317:	B#16#2	CPU 319:	B#16#3	CPU 412, 414 bzw. CPU 416	B#16#5	WinAC RTX, IF 2	B#16#6	WinAC RTX, IF 3	B#16#B	WinAC RTX, IF 4	B#16#F
IM 151-8 PN/DP CPU, WinAC RTX, IF 1:	B#16#1														
CPU 314C, 315 bzw. 317:	B#16#2														
CPU 319:	B#16#3														
CPU 412, 414 bzw. CPU 416	B#16#5														
WinAC RTX, IF 2	B#16#6														
WinAC RTX, IF 3	B#16#B														
WinAC RTX, IF 4	B#16#F														
local_tsap_id_len	<p>Es wird die Länge des Parameters local_tsap_id (= lokale Portnummer) angegeben.</p> <p>Aktiver Verbindungsaufbau: 0                      Passiver Verbindungsaufbau: 2</p>														
rem_subnet_id_len	<p>Dieser Parameter wird derzeit nicht verwendet und muss mit B#16#0 belegt sein.</p>														
rem_staddr_len	<p>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.</p> <p>Unspezifizierte Verbindung: B#16#0                      Spezifizierte Verbindung: B#16#4</p>														
rem_tsap_id_len	<p>Dieser Parameter bezeichnet die Länge des Parameters rem_tsap_id, der Portnummer des entfernten Kommunikationspartners.</p> <p>Aktiver Verbindungsaufbau: 2                      Passiver Verbindungsaufbau: 0</p>														
next_staddr_len	<p>Hier wird die Länge des Parameters next_staddr festgelegt.</p> <p>Bei PN-Schnittstelle: B#16#0</p>														
local_tsap_id	<p>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.</p> <table border="1"> <tr> <td>Bei connection_type B#16#01: local_tsap_id[1] local_tsap_id[2] local_tsap_id[3-16]</td> <td>low byte der Port-Nr. in Hex-Darstellung high byte der Port-Nr. in Hex-Darstellung B#16#00</td> </tr> <tr> <td>Bei connection_type B#16#11: local_tsap_id[1] local_tsap_id[2] local_tsap_id[3-16]</td> <td>high byte der Port-Nr. in Hex-Darstellung low byte der Port-Nr. in Hex-Darstellung B#16#00</td> </tr> </table>	Bei connection_type B#16#01: local_tsap_id[1] local_tsap_id[2] local_tsap_id[3-16]	low byte der Port-Nr. in Hex-Darstellung high byte der Port-Nr. in Hex-Darstellung B#16#00	Bei connection_type B#16#11: local_tsap_id[1] local_tsap_id[2] local_tsap_id[3-16]	high byte der Port-Nr. in Hex-Darstellung low byte der Port-Nr. in Hex-Darstellung B#16#00										
Bei connection_type B#16#01: local_tsap_id[1] local_tsap_id[2] local_tsap_id[3-16]	low byte der Port-Nr. in Hex-Darstellung high byte der Port-Nr. in Hex-Darstellung B#16#00														
Bei connection_type B#16#11: local_tsap_id[1] local_tsap_id[2] local_tsap_id[3-16]	high byte der Port-Nr. in Hex-Darstellung low byte der Port-Nr. in Hex-Darstellung B#16#00														
rem_subnet_id	<p>Dieser Parameter wird derzeit nicht verwendet und muss mit 0 belegt werden.</p>														

Parameter	Beschreibung				
rem_staddr	<p>In diesem Byte-Array wird die IP-Adresse des remoten Kommunikationspartners eingetragen. Im Falle einer un spezifizierten Verbindung wird keine IP-Adresse eingetragen. Die Art der Darstellung wird abhängig vom Parameter connection_type unterschieden. Beispiel: IP-Adresse 192.168.0.1:</p> <table border="0"> <tr> <td>Bei connection_type B#16#01: rem_staddr[1] = rem_staddr[2] = rem_staddr[3] = rem_staddr[4] = rem_staddr[5-6]=</td> <td>B#16#01 (1) B#16#00 (0) B#16#A8 (168) B#16#C0 (192) B#16#00 (reserviert)</td> </tr> <tr> <td>Bei connection_type B#16#11: rem_staddr[1] = rem_staddr[2] = rem_staddr[3] = rem_staddr[4] = rem_staddr[5-6]=</td> <td>B#16#C0 (192) B#16#A8 (168) B#16#00 (0) B#16#01 (1) B#16#00 (reserviert)</td> </tr> </table>	Bei connection_type B#16#01: rem_staddr[1] = rem_staddr[2] = rem_staddr[3] = rem_staddr[4] = rem_staddr[5-6]=	B#16#01 (1) B#16#00 (0) B#16#A8 (168) B#16#C0 (192) B#16#00 (reserviert)	Bei connection_type B#16#11: rem_staddr[1] = rem_staddr[2] = rem_staddr[3] = rem_staddr[4] = rem_staddr[5-6]=	B#16#C0 (192) B#16#A8 (168) B#16#00 (0) B#16#01 (1) B#16#00 (reserviert)
Bei connection_type B#16#01: rem_staddr[1] = rem_staddr[2] = rem_staddr[3] = rem_staddr[4] = rem_staddr[5-6]=	B#16#01 (1) B#16#00 (0) B#16#A8 (168) B#16#C0 (192) B#16#00 (reserviert)				
Bei connection_type B#16#11: rem_staddr[1] = rem_staddr[2] = rem_staddr[3] = rem_staddr[4] = rem_staddr[5-6]=	B#16#C0 (192) B#16#A8 (168) B#16#00 (0) B#16#01 (1) B#16#00 (reserviert)				
rem_tsap_id	<p>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.</p> <table border="0"> <tr> <td>Bei connection_type B#16#01: rem_tsap_id[1] rem_tsap_id[2] rem_tsap_id[3-16]</td> <td>low byte der Port-Nr. in Hex-Darstellung low byte der Port-Nr. in Hex-Darstellung B#16#00</td> </tr> <tr> <td>Bei connection_type B#16#11: rem_tsap_id[1] rem_tsap_id[2] rem_tsap_id[3-16]</td> <td>high byte der Port-Nr. in Hex-Darstellung low byte der Port-Nr. in Hex-Darstellung B#16#00</td> </tr> </table>	Bei connection_type B#16#01: rem_tsap_id[1] rem_tsap_id[2] rem_tsap_id[3-16]	low byte der Port-Nr. in Hex-Darstellung low byte der Port-Nr. in Hex-Darstellung B#16#00	Bei connection_type B#16#11: rem_tsap_id[1] rem_tsap_id[2] rem_tsap_id[3-16]	high byte der Port-Nr. in Hex-Darstellung low byte der Port-Nr. in Hex-Darstellung B#16#00
Bei connection_type B#16#01: rem_tsap_id[1] rem_tsap_id[2] rem_tsap_id[3-16]	low byte der Port-Nr. in Hex-Darstellung low byte der Port-Nr. in Hex-Darstellung B#16#00				
Bei connection_type B#16#11: rem_tsap_id[1] rem_tsap_id[2] rem_tsap_id[3-16]	high byte der Port-Nr. in Hex-Darstellung low byte der Port-Nr. in Hex-Darstellung B#16#00				
next_staddr	<p>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</p>				
spare	<p>Dieser Parameter wird nicht verwendet und muss mit 0 vorbelegt werden.</p>				

### Modbusparameter "Modbus settings"

Parameter	Beschreibung
server_client	<p>TRUE:        S7 ist Server FALSE:       S7 ist Client</p>
single_write	<p>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.</p>
connect_at_startup	<p>Hiermit wird der Zeitpunkt des Verbindungsaufbaus festgelegt. Ist connect_at_startup auf TRUE gesetzt, wird der Verbindungsaufbau 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_CONN angezeigt wird. FALSE:        Verbindungsaufbau bei gesetztem ENQ_ENR TRUE:         Verbindungsaufbau direkt nach Neustart</p>



## 1.4 Beschreibung MODBUSPN

### Beschreibung

Die Anweisung 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. Die Anweisung kann je nach Parametrierung sowohl als Client (S7 ist Client) als auch als Server (S7 ist Server) betrieben werden. Über die Anweisung MODBUSPN bauen Sie eine Verbindung zwischen den Kommunikationspartnern auf, führen die Datenübertragung durch und steuern den Verbindungsabbau.

Bei der Datenübertragung werden folgende Aktionen ausgeführt:

- Generierung des MODBUS-spezifischen Telegrammheader beim Senden
- Prüfung des MODBUS-spezifischen Telegrammheaders beim Empfang
- Prüfung ob die vom Client angesprochenen Datenbereiche vorhanden sind
- Generierung von Exception-Telegrammen wenn ein Fehler aufgetreten ist (nur bei S7 ist Server)
- Datentransfer von/in den parametrisierten Datenbaustein

Weiterhin erfolgt eine zeitliche Überwachung des Verbindungsaufbaus und -abbaus sowie des Datenempfangs.

Die Anweisung MODBUSPN V1.0 kann sowohl für die S7-300 als auch für die S7-400 verwendet werden. Die Verbindung erfolgt über die lokale Schnittstelle der CPU. Zur Verwendung der Anweisung benötigen Sie kein zusätzliches Hardwaremodul.

### Funktionsweise der Anweisung

#### Anlauf

Die Anweisung MODBUSPN wird im OB100 einmal aufgerufen.

- Die Initialisierungsparameter müssen entsprechend der Anlagenkonfiguration belegt sein.
- Die Initialisierungsparameter werden in den Instanz-DB übernommen.
- Die Laufzeitparameter werden im Anlauf nicht ausgewertet.
- Die Daten aus dem Parameter-Datenbaustein werden auf Plausibilität überprüft.

#### Zyklischer Betrieb

Im zyklischen Betrieb wird die Anweisung MODBUSPN im OB1 oder in einem Weckalarm-OB aufgerufen.

- Anhand der Laufzeitparameter werden die Funktionen des Bausteins aktiviert.
- Während ein Auftrag läuft, werden Änderungen an den Laufzeitparametern nicht ausgewertet.
- Initialisierungsparameter werden nicht ausgewertet.

### Neustart bei Inbetriebnahme

Bei der Inbetriebnahme kann ein wiederholter CPU-Neustart nach einer Änderung der Initialisierungsparameter sehr aufwendig sein. Über das manuelle Setzen des im statischen Bereich vorhandenen Parameters „Init\_Start“, kann der Neustart-Programmteil des Modbusbausteins durchlaufen werden. Bei manueller Initialisierung darf kein Auftrag laufen. Für eine korrekte Initialisierung müssen alle Initialisierungsparameter im zyklischen OB parametrisiert werden.

### Verbindungsbearbeitung

Den aktiven Verbindungsaufbau führt der Modbus-Client aus. Die Daten hierfür werden aus den Verbindungsparametern im Parameter-Datenbaustein 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 Anweisung TCON ein Kommunikationskanal zum Koppelpartner geöffnet.

Der Zeitpunkt des Verbindungsaufbaus wird mit dem Parameter connect\_at\_startup im Parameter-Datenbaustein festgelegt.

Der Verbindungsabbau wird mit dem Parameter DISCONNECT an der Anweisung MODBUSPN geregelt.

### Auftragsanstoß bei "S7 ist Client" bzw. Aktivierung der Anweisung bei "S7 ist Server"

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

#### S7 ist Client: Auftragsanstoß

Durch einen positiven Flankenwechsel am Triggereingang ENQ\_ENR wird ein Auftrag aktiviert. Abhängig von den Eingangsparametern UNIT, DATA\_TYPE, START\_ADDRESS, LENGTH, TI und WRITE\_READ wird ein MODBUS Anforderungstelegramm generiert und zur Partnerstation über die TCP/IP-Verbindung gesendet. Der Client wartet die parametrisierte Zeit RECV\_TIME 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.

Nach dem Empfang des Antworttelegramms 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 in STATUS\_MODBUS angezeigt.

#### S7 ist Server: Aktivierung der Anweisung

Durch einen positiven Pegel am Triggereingang ENQ\_ENR ist die Anweisung 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, TI und WRITE\_READ angezeigt.

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

## Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "MODBUSPN":

Parameter	Deklaration	Datentyp	Beschreibung	Wertebereich
ID (Seite 24)	Input	WORD	Verbindungs-ID, muss identisch sein mit dem zugehörigen Parameter id im Parameter-DB	1 bis 4095 W#16#1 bis W#16#FFF
DB_PARAM (Seite 24)	Input	BLOCK_DB	Nummer des Parameter-DBs	CPU abhängig
RECV_TIME (Seite 25)	Input	TIME	Überwachungszeit für den Empfang von Daten vom Koppelpartner Die minimal einstellbare Zeit ist 20 ms	T#20ms bis T# +24d20h31m2 3s647ms
CONN_TIME (Seite 25)	Input	TIME	Überwachungszeit für den Verbindungsaufbau bzw. -abbau Die minimal einstellbare Zeit ist 100 ms	T#100ms bis T# +24d20h31m2 3s647ms
KEEP_ALIVE	Input	TIME	nicht verwendet	
ENQ_ENR (Seite 25)	Input	BOOL	S7 ist Client: Auftragsanstoß bei positiver Flanke <ul style="list-style-type: none"> <li>• Änderungen an den Eingangsparametern wirken sich erst aus, wenn eine Antwort des Servers erfolgt ist oder eine Fehlermeldung ausgegeben wurde.</li> <li>• Wird während einer laufenden Modbus-Anfrage der Parameter ENQ_ENR erneut gesetzt, wird im Anschluss keine weitere Übertragung durchgeführt.</li> </ul> S7 ist Server: Empfangsbereit bei positivem Pegel	TRUE FALSE
DISCONNECT (Seite 25)	Input	BOOL	Über den Parameter steuern Sie den Verbindungsaufbau und -abbau zu dem Modbus-Server: 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
REG_KEY (Seite 17)	Input	STRING [17]	Registrierungsschlüssel (Freischaltcode) für die Lizenzierung	Character
LICENSED (Seite 17)	Output	BOOL	Lizenzzustand des Bausteins Baustein ist lizenziert Baustein ist nicht lizenziert	TRUE FALSE

1.4 Beschreibung MODBUSPN

Parameter	Deklaration	Datentyp	Beschreibung	Wertebereich
BUSY	Output	BOOL	Bearbeitungszustand der T-Funktionen (TCON, TDISCON, TSEND oder TRCV) eine T-Funktion ist in Bearbeitung eine T-Funktion ist nicht in Bearbeitung	TRUE FALSE
CONN_ESTABLISHED	Output	BOOL	Verbindung zum Koppelpartner ist aufgebaut Verbindung zum Koppelpartner ist abgebaut	TRUE FALSE
DONE_NDR	Output	BOOL	S7 ist Client: TRUE: aktivierter Auftrag wurde fehlerfrei beendet S7 ist Server: TRUE: Anforderung vom Client wurde ausgeführt und beantwortet	TRUE FALSE
ERROR	Output	BOOL	FALSE: Kein Fehler TRUE: Fehler aufgetreten. Die Fehlerursache wird über die Parameter STATUS_MODBUS und STATUS_CONN angezeigt.	FALSE TRUE
STATUS_MODBUS (Seite 28)	Output	WORD	Fehlernummer für Protokollfehler bei der Bearbeitung der Modbustelegramme	0 bis FFFF
STATUS_CONN (Seite 28)	Output	WORD	Fehlernummer für Verbindungsfehler bei der Bearbeitung der T-Funktionen (TCON, TSEND, TRCV, TDISON)	0 bis FFFF
STATUS_FUNC (Seite 28)	Output	STRING [8]	Name der Anweisung, die den Fehler an STATUS_MODBUS bzw. STATUS_CONN verursacht hat	Character
IDENT_CODE (Seite 17)	Output	STRING [18]	Identifikationsnummer für die Lizenzierung Mit dieser Kennung können Sie den Freischaltcode REG_KEY für Ihre Lizenz beantragen.	Character
UNIT (Seite 26)	InOut	BYTE	Unit Identifier (INPUT bei Client-Funktion, OUTPUT bei Server-Funktion)	0 bis 255 B#16#0 bis B#16#FF
DATA_TYPE (Seite 26)	InOut	BYTE	zu bearbeitender Datentyp: (INPUT bei Client-Funktion, OUTPUT bei Server-Funktion) Coils Inputs Holding Register Input Register	1 2 3 4
START_ADDRESS (Seite 26)	InOut	WORD	MODBUS Startadresse (INPUT bei Client-Funktion, OUTPUT bei Server-Funktion)	0 bis 65535 W#16#0000 bis W#16#FFFF

Parameter	Deklaration	Datentyp	Beschreibung	Wertebereich
LENGTH (Seite 26)	InOut	WORD	Anzahl der zu bearbeitenden Werte (INPUT bei Client-Funktion, OUTPUT bei Server-Funktion) Coils Lesende Funktion Schreibende Funktion Inputs Lesende Funktion Holding Register Lesende Funktion Schreibende Funktion Input Register Lesende Funktion	1 bis 2000 1 bis 1968  1 bis 2000  1 bis 125 1 bis 123  1 bis 125
TI (Seite 26)	InOut	WORD	Transaction Identifier (INPUT bei Client-Funktion, OUTPUT bei Server-Funktion)	0 bis 65535 W#16#0 to W#16#FFFF
WRITE_READ (Seite 26)	InOut	BOOL	Schreibzugriff oder Lesezugriff (INPUT bei Client-Funktion, OUTPUT bei Server-Funktion)	TRUE FALSE

## Siehe auch

Inbetriebnahme (Seite 5)

## 1.5 Lizenzierung mit den Parametern IDENT\_CODE und REG\_KEY

### Beschreibung

Die Anweisung MODBUSPN muss auf jeder CPU einzeln lizenziert werden. Die Lizenzierung erfolgt in 2 Schritten:

- dem Auslesen des IDENT\_CODE und
- der Eingabe des Freischaltcodes REG\_KEY.

Dabei muss der OB121 in der CPU vorhanden sein.

Für das Auslesen des IDENT\_CODE gehen Sie wie folgt vor:

1. Parametrieren Sie die Anweisung MODBUSPN Ihren Anforderungen entsprechend in einem zyklischen OB und im OB100. Laden Sie das Programm in die CPU und setzen Sie diese in RUN.
2. Öffnen Sie den Instanz-DB der Modbus-Anweisung und klicken Sie auf die Schaltfläche „Alle beobachten“ .

3. Am Ausgang IDENT\_CODE wird eine 18-stellige Zeichenfolge angezeigt.

MODBUSPN_DB					
	Name	Datentyp	Offset	Startwert	Beobachtungswert
1	▼ Input				
2	ID	Word	0.0	16#0	16#0001
3	DB_PARAM	Block_DB	2.0	DB 1	DB1
4	RECV_TIME	Time	4.0	T#0ms	T#500MS
5	CONN_TIME	Time	8.0	T#0ms	T#5S
6	KEEP_ALIVE	Time	12.0	T#0ms	T#0MS
7	ENQ_ENR	Bool	16.0	false	FALSE
8	DISCONNECT	Bool	16.1	false	FALSE
9	REG_KEY	String[17]	18.0	'	'
10	▼ Output				
11	LICENSED	Bool	38.0	false	FALSE
12	BUSY	Bool	38.1	false	FALSE
13	CONN_ESTABLISHED	Bool	38.2	false	FALSE
14	DONE_NDR	Bool	38.3	false	FALSE
15	ERROR	Bool	38.4	false	FALSE
16	STATUS_MODBUS	Word	40.0	16#0	16#A090
17	STATUS_CONN	Word	42.0	16#0	16#0000
18	STATUS_FUNC	String[8]	44.0	"	"
19	IDENT_CODE	String[18]	54.0	"	'MBDCALKIFABJKMBJL2'
20	▼ InOut				
21	UNIT	Byte	74.0	16#0	16#00
22	DATA_TYPE	Byte	75.0	16#0	16#00
23	START_ADDRESS	Word	76.0	16#0	16#0000
24	LENGTH	Word	78.0	16#0	16#0000
25	TI	Word	80.0	16#0	16#0000
26	WRITE_READ	Bool	82.0	false	FALSE

Bild 1-2 IDENT\_CODE im DB

- Kopieren Sie diesen String per Copy/Paste aus dem Datenbaustein und fügen ihn in das Formular SOFTWARE REGISTRATION FORM ein. Dieses Formular ist auf der Installations-CD enthalten.

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

MODBUSPN_DB					
	Name	Datentyp	Offset	Startwert	Beobachtungswert
1	Input				
2	ID	Word	0.0	16#0	16#0001
3	DB_PARAM	Block_DB	2.0	DB 1	DB1
4	RECV_TIME	Time	4.0	T#0ms	T#500MS
5	CONN_TIME	Time	8.0	T#0ms	T#5S
6	KEEP_ALIVE	Time	12.0	T#0ms	T#0MS
7	ENQ_ENR	Bool	16.0	false	FALSE
8	DISCONNECT	Bool	16.1	false	FALSE
9	REG_KEY	String[17]	18.0	'	'
10	Output				
11	LICENSED	Bool	38.0	false	FALSE
12	BUSY	Bool	38.1	false	FALSE
13	CONN_ESTABLISHED	Bool	38.2	false	FALSE
14	DONE_NDR	Bool	38.3	false	FALSE
15	ERROR	Bool	38.4	false	FALSE
16	STATUS_MODBUS	Word	40.0	16#0	16#A090
17	STATUS_CONN	Word	42.0	16#0	16#0000
18	STATUS_FUNC	String[8]	44.0	"	"
19	IDENT_CODE	String[18]	54.0	"	'MBCALKIFABJKMBJL2'
20	InOut				
21	UNIT	Byte	74.0	16#0	16#00
22	DATA_TYPE	Byte	75.0	16#0	16#00
23	START_ADDRESS	Word	76.0	16#0	16#0000
24	LENGTH	Word	78.0	16#0	16#0000
25	TI	Word	80.0	16#0	16#0000
26	WRITE_READ	Bool	82.0	false	FALSE

Please insert the IDENT-CODE here.  
The manual contains information how to find out the IDENT-CODE.

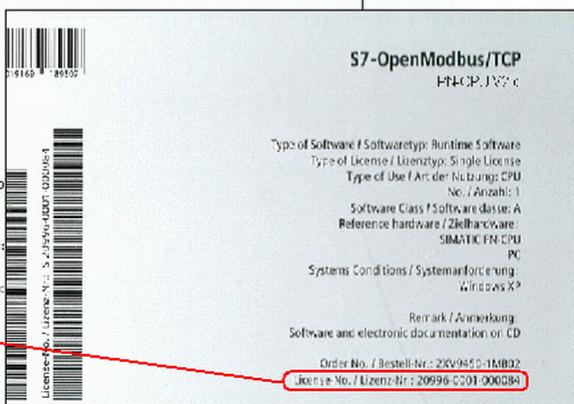
Bitte tragen Sie den IDENT-CODE hier ein.  
Das Handbuch enthält Informationen, wie Sie den IDENT-CODE ermitteln.

>>> IDENT\_CODE <<<

Please insert the License-No. here.  
You find the License-No. on the package of the product.

Bitte tragen Sie die Lizenz-Nr. hier ein.  
Sie finden die Lizenz-Nr. auf der Verpackung des Produktes.

>>> License-No / Lizenz-Nr <<<



**S7-OpenModbus/TCP**  
FW-CP, JX2.c

Type of Software / Softwaretyp: Runtime Software  
Type of License / Lizenztyp: Single License  
Type of Use / Art der Nutzung: CPU  
No. / Anzahl: 1  
Software Class / Softwareklasse: A  
Reference hardware / Zielhardware: SIMATIC PN-CPU  
PC  
Systems Conditions / Systemanforderung:  
Windows XP

Remark / Anmerkung:  
Software and electronic documentation on CD

Order No. / Bestell-Nr.: 2XV9450-1MB02  
License No. / Lizenz-Nr.: 20996-001-000084

Bild 1-3 IDENT\_CODE und Lizenzformular

- Senden Sie das Formular über einen Service Request an den Customer Support (<https://support.automation.siemens.com/WW/view/de/38718979>). Sie erhalten daraufhin den Freischaltcode für ihre CPU.

Die Angabe des Freischaltcodes REG\_KEY muss an jeder MODBUSPN-Anweisung erfolgen. Der REG\_KEY sollte in einem Global-Datenbaustein gespeichert werden, über den alle MODBUSPN-Anweisungen den notwendigen Freischaltcode erhalten.

1.5 Lizenzierung mit den Parametern IDENT\_CODE und REG\_KEY

Für die Eingabe des Freischaltcodes REG\_KEY gehen Sie wie folgt vor:

- Fügen Sie mit „Neuen Baustein hinzufügen...“ einen neuen Global-Datenbaustein mit einem eindeutigen symbolischen Namen, z.B. „Lizenz\_DB“, ein.
- Legen Sie in diesem Baustein einen Parameter REG\_KEY mit dem Datentyp STRING[17] an.

Lizenz_DB				
	Name	Datentyp	Offset	Startwert
1	Static			
2	REG_KEY	String[17]	0.0	

Bild 1-4 REG\_KEY in DB

- Kopieren Sie den übermittelten 17-stelligen Freischaltcode per Copy/Paste in die Spalte ‚Startwert‘.
- Geben Sie im zyklischen OB am Parameter REG\_KEY der MODBUSPN-Anweisung den Wert „Lizenz\_DB.REG\_KEY“ an.
- Laden Sie die geänderten Bausteine in die CPU. Die Eingabe des Freischaltcodes kann zur Laufzeit erfolgen, ein Wechsel von STOP -> RUN ist nicht erforderlich.

Die Modbus/TCP-Kommunikation über die Anweisung MODBUSPN ist nun für diese CPU lizenziert, das Ausgangsbit LICENSED ist TRUE.

**Fehlende oder fehlerhafte Lizenzierung**

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.

Diagnosepuffer

**Ereignisse**

CPU-Zeitstempel berücksichtigt lokale PG/PC-Zeit

Nr.	Datum und Uhrzeit	Ereignis
1	04.03.2014 13:43:37.728	Ereignis-ID: 16# A090
2	04.03.2014 13:43:37.728	Bereichslängenfehler beim Lesen
3	04.03.2014 13:43:33.706	Ereignis-ID: 16# A090
4	04.03.2014 13:43:33.706	Bereichslängenfehler beim Lesen
5	04.03.2014 13:43:29.685	Ereignis-ID: 16# A090
6	04.03.2014 13:43:29.685	Bereichslängenfehler beim Lesen
7	04.03.2014 13:43:25.658	Ereignis-ID: 16# A090
8	04.03.2014 13:43:25.658	Bereichslängenfehler beim Lesen

Anzeige einfrieren

**Details zum Ereignis**

Details zum Ereignis: 1 von 120 Ereignis-ID: 16# A090

Beschreibung:

Ereignis-ID:	16# A090
OB:	16# 01
Prioritätsklasse:	16# 01
DatID 1/2:	16# 50 C0
Zusatzinfo 1 / 2 / 3:	16# 2020 2020 2020

Zeitstempel: 04.03.2014 13:43:37.728

Kommend/Gehend: Gehendes Ereignis

Hilfe zum Ereignis Im Editor öffnen Speichern unter...

Bild 1-5 Diagnosepuffer mit A090

**! WARNUNG**

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

Bei einem fehlenden oder falschen Freischaltcode wird die Modbus/TCP-Kommunikation bearbeitet, allerdings wird am Ausgang STATUS\_MODBUS stets W#16#A090 „Keine gültige Lizenz vorhanden“ angezeigt. Das Ausgangsbit LICENSED ist FALSE.

Siehe auch

Beschreibung MODBUSPN (Seite 13)

## 1.6 Adressabbildung

### Interpretation der Modbus-Adressen

Das MODBUS-Datenmodell enthält folgende Bereiche:

- Coils
- Inputs

1.6 Adressabbildung

- Holding Register
- Input Register

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.

Die Anweisung MODBUSPN verwendet bei seinen Parametern start, end und START\_ADDRESS die **tatsächlich übertragene Modbusadresse**. Es können also mit jedem Funktionscode Register-/Bitadressen von 0000H bis FFFFH übertragen werden.

**Beispiel**

Die Modbusadressen können im Parameter-DB in dezimaler oder in hexadezimaler Form angegeben werden.

Parameter	Dezimale Schreibweise	Hexadezimale Schreibweise	Bedeutung
data_type	3	B#16#3	Holding Register
db	11	W#16#B	DB 11
start	0	W#16#0	Anfangsadresse: 0
end	499	W#16#1F3	Endadresse: 499
data_type	3	B#16#3	Holding Register
db	12	W#16#C	DB 12
start	720	W#16#2D0	Anfangsadresse: 720
end	900	W#16#384	Endadresse: 900

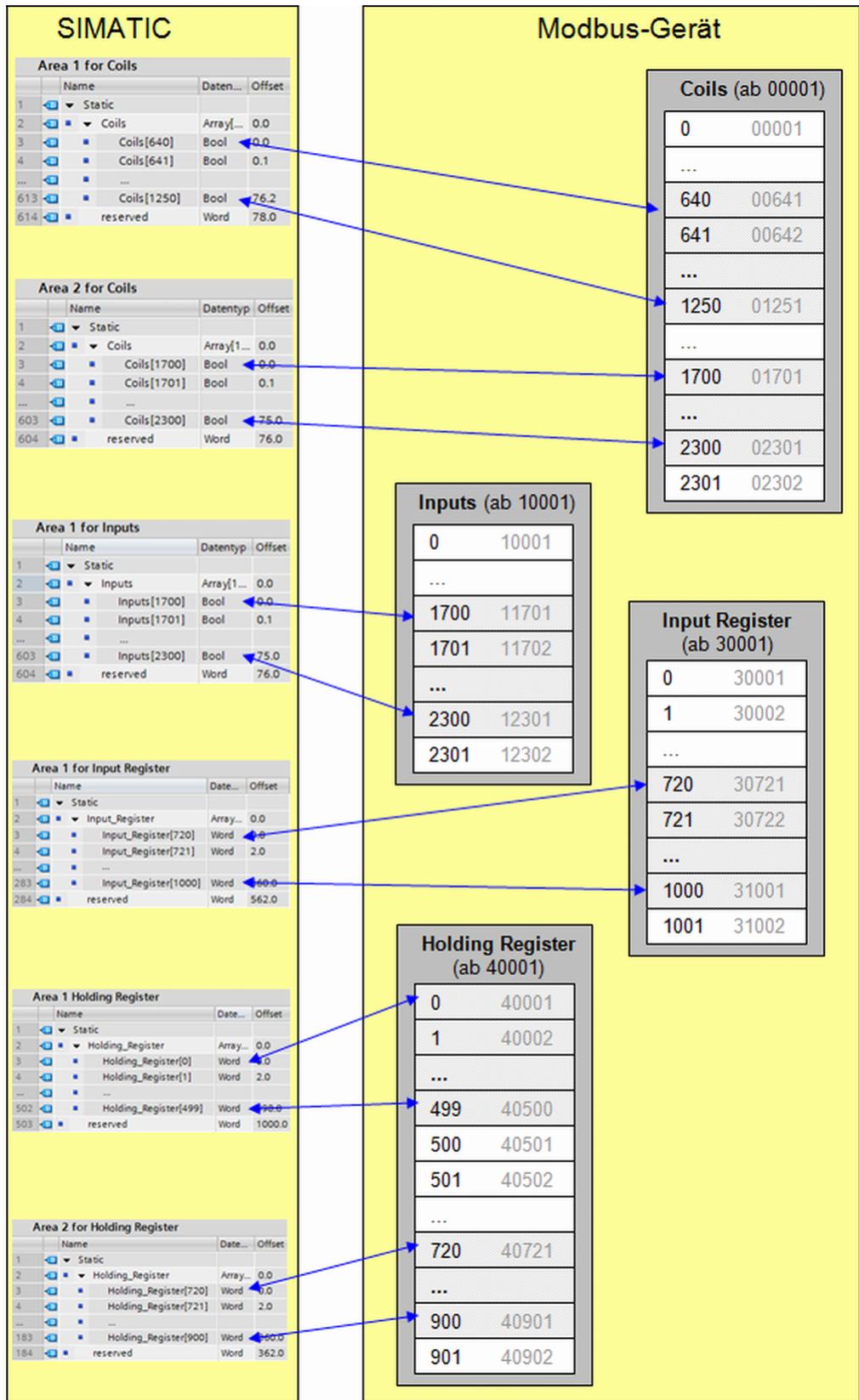
Im folgenden Bild sehen Sie die Gegenüberstellung der SIMATIC-Speicherbereiche mit der register- und bitorientierten Speicheraufteilung der Modbus-Geräte. Dabei wird auf obige Parametrierung Bezug genommen.

**Im Modbus-Gerät:**

Die Modbus-Adressen, die schwarz dargestellt sind, betreffen den Data Link Layer, die grau dargestellten den Applikation 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.



## 1.7 Parameter ID und DB\_PARAM

Bild 1-6 Adressabbildung

### Siehe auch

Beschreibung MODBUSPN (Seite 13)

## 1.7 Parameter ID und DB\_PARAM

### Beschreibung

#### 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 projiziert, welcher im Parameter-Datenbaustein 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 Parameter-Datenbausteins. In diesem Parameter-Datenbaustein sind die verbindungs- und modbuspezifischen 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 Parameter-Datenbaustein die dazu notwendigen Parameter aller Verbindungen in sequentieller Abfolge enthalten. Es ist ebenso möglich für jede Verbindung einen separaten Parameter-Datenbaustein zu erstellen.

### Siehe auch

Beschreibung MODBUSPN (Seite 13)

Parameter-Datenbaustein (Seite 7)

## 1.8 Parameter RECV\_TIME und CONN\_TIME

### Beschreibung

#### RECV\_TIME

Die Überwachungszeit RECV\_TIME überwacht den Empfang der Daten vom Koppelpartner. Bei Überschreitung der Überwachungszeit wird ein Fehler gemeldet und die Verbindung abgebaut.

Der Minimalwert beträgt 20 ms.

Wenn in der Betriebsart „**S7 ist Client**“ die RECV\_TIME auf < 20 ms gesetzt wird, erscheint eine entsprechende Fehlermeldung und der aktivierte Auftrag wird abgewiesen.

Wenn in der Betriebsart „**S7 ist Server**“ die RECV\_TIME auf < 20 ms gesetzt wird, wird der Defaultwert von 1,2 s verwendet. Die RECV\_TIME überwacht die Laufzeit des TCP-Streams. Die Pause zwischen einzelnen Request vom Client wird dabei nicht berücksichtigt.

#### CONN\_TIME

Die CONN\_TIME gibt die Zeit für die Überwachung des Verbindungsaufbaus bzw. -abbaus an. Konnte innerhalb der parametrisierten Überwachungszeit die Verbindung nicht erfolgreich auf- bzw. abgebaut werden, erscheint eine entsprechende Fehlermeldung am Ausgang STATUS\_CONN.

Der minimale Wert ist 100 ms.

In der Betriebsart „**S7 ist Client**“ wird bei connect\_at\_startup = TRUE eine zu gering parametrisierte CONN\_TIME auf den Defaultwert von 5 s gesetzt. Im zyklischen Betrieb wird bei einer zu kleinen CONN\_TIME eine Fehlermeldung ausgegeben und der aktivierte Auftrag abgewiesen.

Wenn in der Betriebsart „**S7 ist Server**“ die CONN\_TIME auf < 100 ms gesetzt wurde, wird ebenfalls der Defaultwert von 5 s verwendet.

### Siehe auch

Beschreibung MODBUSPN (Seite 13)

## 1.9 Parameter ENQ\_ENR und DISCONNECT

### Beschreibung

#### Betriebsart "S7 ist Client"

Mit einer positiven Flanke an ENQ\_ENR wird der Datentransfer initiiert. Mit den Werten der Eingangsparameter UNIT, DATA\_TYPE, START\_ADDRESS, LENGTH, TI und WRITE\_READ wird das Anforderungstelegramm generiert. Ein neues Telegramm kann nur gesendet werden, wenn das vorherige Telegramm mit DONE\_NDR oder ERROR abgeschlossen wurde.

Wenn die Verbindung nicht aufgebaut ist (CONN\_ESTABLISHED = FALSE), wird zuerst die Verbindung aufgebaut und anschließend der Datentransfer ausgeführt.

Ist der Parameter DISCONNECT = TRUE gesetzt, wird die Verbindung nach dem Datentransfer abgebaut.

**Betriebsart "S7 ist Server"**

Mit einem positiven Pegel am Eingang ENQ\_ENR wird die Anweisung aktiviert. Es werden Requests vom Client ausgewertet und beantwortet. 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 bei gesetztem Parameter DISCONNECT = TRUE die Verbindung abgebaut.

Bei einer bestehenden Verbindung und ENQ\_ENR = FALSE werden empfangene Requests verworfen.

**Siehe auch**

Beschreibung MODBUSPN (Seite 13)

**1.10 Parameter DATA\_TYPE, START\_ADDRESS, LENGTH, TI, WRITE\_READ und UNIT**

**Beschreibung**

In der Betriebsart „S7 ist Client“ handelt es sich um Eingangs-, in der Betriebsart „S7 ist Server“ um Ausgangsparameter.

**DATA\_TYPE**

Der Parameter DATA\_TYPE zeigt an, welcher Modbus-Datentyp mit dem aktuellen Telegramm bearbeitet wird. Es sind folgende Werte zulässig:

Modbus-Datentyp	DATA_TYPE
Coils	B#16#1
Inputs	B#16#2
Holding Register	B#16#3
Input Register	B#16#4

Die unterschiedlichen Datentypen haben einen direkten Zusammenhang mit den verwendeten Funktionscodes.

Modbus-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

Modbus-Datentyp	DATA_TYPE	Funktion	Länge	single_write	Funktionscode
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.

### 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 Datenbausteins liegen.

### TI

Der Parameter TI, Transaction Identifier, wird gemäß MODBUS Spezifikation vom Server aus dem Anforderungstelegramm in das Antworttelegramm umkopiert.

In der Betriebsart „S7 ist Client“ handelt es sich um einen Eingangsparameter. Die Anweisung übernimmt diesen Wert in das Anforderungstelegramm und überprüft ihn beim Empfang der Antwort.

In der Betriebsart „S7 ist Server“ handelt es sich um einen Ausgangsparameter. Die Anweisung übernimmt den Wert aus dem Anforderungstelegramm in die Antwort.

Der Transaction Identifier dient zur Erkennung von Telegrammen bzw. der eindeutigen Zuordnung von den Anfragen zu den Antworten. Die Anweisung MODBUSPN kann diese Zuordnung nur vornehmen, wenn der TI auch bei jedem Telegramm verändert wird. Nur dann ist eine zuverlässige Funktion der Anweisung gewährleistet.

Deshalb wird empfohlen, den TI bei jedem Request um 1 zu erhöhen.

### 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.

### 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.

### 1.11 Parameter ERROR, STATUS\_MODBUS, STATUS\_CONN und STATUS\_FUNC

In der Funktion „**S7 ist Client**“ ist der Parameter UNIT ein Eingangsparameter. Dieser Eingang ist entsprechend den Anforderungen zu setzen. Die Anweisung übernimmt diesen Wert in das Anforderungstelegramm und überprüft ihn beim Empfang der Antwort.

In der Funktion „**S7 ist Server**“ ist der Parameter UNIT ein Ausgangsparameter. Die Anweisung übernimmt den Wert aus dem Anforderungstelegramm in das Antworttelegramm und zeigt ihn nach einem beendeten Auftrag an.

#### 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, TI und WRITE\_READ werden die Auftragsparameter des Client angezeigt. Diese Ausgänge sind nur gültig so lange DONE\_NDR gesetzt ist.

#### Siehe auch

Beschreibung MODBUSPN (Seite 13)

Adressabbildung (Seite 21)

## 1.11 Parameter ERROR, STATUS\_MODBUS, STATUS\_CONN und STATUS\_FUNC

### Fehlerauswertung

Zur Fehlerdiagnose besitzt die Anweisung MODBUSPN 3 Statusausgänge: STATUS\_MODBUS, STATUS\_CONN und STATUS\_FUNC.

#### 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 in den Ausgängen STATUS\_MODBUS oder STATUS\_CONN 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 in den Ausgängen STATUS\_MODBUS oder STATUS\_CONN angezeigt.

#### STATUS\_MODBUS

An STATUS\_MODBUS werden die Fehlermeldungen und Statusinformationen bezüglich der modbusspezifischen Telegrammbearbeitung angezeigt. Die Anweisung MODBUSPN verwendet verschiedene Systembausteine. Die Fehlermeldungen dieser Bausteine werden unverändert an STATUS\_MODBUS weitergegeben.

### STATUS\_CONN

An STATUS\_CONN werden die Fehlermeldungen bezüglich der Verbindungsbearbeitung angezeigt. Zusätzlich werden in STATUS\_CONN auch Meldungen wie „Auftrag in Bearbeitung“ angezeigt. In diesem Fall ist das ERROR-Bit nicht gesetzt. Weiterhin werden intern die Anweisungen TCON, TSEND, TRCV und TDISCON aufgerufen. Die Fehlermeldungen dieser Bausteine werden unverändert an STATUS\_CONN weitergegeben.

### STATUS\_FUNC

An STATUS\_FUNC wird der Name der Funktion angezeigt, die den Fehler verursacht hat.

Nachfolgend finden Sie eine Aufstellung der Anweisungs-spezifischen Fehlermeldungen.

#### Parameter STATUS\_MODBUS bei STATUS\_FUNC = 'MODBUSPN'

STATUS* (W#16#)	Beschreibung	Abhilfe
A001	Der Parameter-Datenbaustein ist zu kurz.	Korrigieren Sie die Länge des Parameter-DBs.
A002	Der Parameter end ist kleiner als start.	Korrigieren Sie die Angaben im Parameter-DB.
A003	Ein DB, auf den MODBUS-Adressen abgebildet werden sollen, ist zu kurz. Minimallänge: <ul style="list-style-type: none"> <li>• bei Registern: <math>(end - start + 1) * 2 + 2</math></li> <li>• bei Bitwerten: <math>(end - start) / 8 + 1 + 2</math></li> </ul> weitere mögliche Ursachen: <ul style="list-style-type: none"> <li>• S7 ist Client: falsche Aufrufparameter</li> <li>• S7 ist Server: falscher Adressbereich im Anforderungstelegramm des Clients. Die S7 antwortet mit einem Exception-Telegramm.</li> </ul>	Verlängern Sie den DB. S7 ist Client: Korrigieren Sie die Aufrufparameter START_ADDRESS oder LENGTH. S7 ist Server: Ändern Sie die Anforderung des Client.
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 Exception-Telegramm. Wertebereiche: 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.
A006	Der über DATA_TYPE, START_ADDRESS und LENGTH angegebene Bereich existiert nicht in data_type von data_area_1 bis data_area_8. S7 ist Server: Die S7 antwortet mit einem Exception-Telegramm.	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.

1.11 Parameter ERROR, STATUS\_MODBUS, STATUS\_CONN und STATUS\_FUNC

STATUS* (W#16#)	Beschreibung	Abhilfe
A007	S7 ist Client: Es wurde eine ungültige Überwachungszeit an RECV_TIME oder CONN_TIME parametriert. Für RECV_TIME muss ein Wert >= 20 ms, für CONN_TIME >= 100 ms eingetragen werden.	Korrigieren Sie die Parametrierung.
A009	S7 ist Client: Der empfangene Transaction Identifier TI ist ungleich dem gesendeten. Die Kommunikationsverbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A00A	S7 ist Client: Der empfangene Unit Identifier 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 Exception-Telegramm.	S7 ist Client: Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.  S7 ist Server: Ändern Sie die Anforderung des Client. Die Anweisung MOD-BUSPN bearbeitet die Funktionscodes 1, 2, 3, 4, 5, 6, 15 und 16.
A00C	Der empfangene Bytecount passt nicht zur Registeranzahl. Die Kommunikationsverbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A00D	Nur bei S7 ist Client: Die Register-/Bitadresse bzw. die Register-/Bitanzahl im Antworttelegramm ist ungleich der im Anforderungstelegramm.	
A00E	Die Längenangabe im modbusspezifischen Telegrammheader passt nicht zu den Angaben der Register-/Bitanzahl oder des Bytecount im Telegramm. Die Anweisung verwirft alle Daten. Die Kommunikationsverbindung wird abgebaut.	
A00F	Es wurde ein Protocol Identifier ungleich 0 empfangen. Die Kommunikationsverbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.
A010	Bei dem Parameter db von data_area_1 bis data_area_8 wurde eine DB-Nummer doppelt vergeben.	Korrigieren Sie die Parametrierung im Parameter-DB.
A011	Am Eingangsparameter DATA_TYPE wurde ein unzulässiger Wert angegeben (zulässige Werte sind 1, 2, 3 und 4).	Korrigieren Sie die Aufrufparameter.
A012	Die parametrierten Bereiche data_area_1 und data_area_2 überlappen.	Korrigieren Sie die Parametrierung.  Die Datenbereiche dürfen keinen gemeinsamen Registeradressbereich besitzen.
A013	Die parametrierten Bereiche data_area_1 und data_area_3 überlappen.	
A014	Die parametrierten Bereiche data_area_1 und data_area_4 überlappen.	
A015	Die parametrierten Bereiche data_area_1 und data_area_5 überlappen.	
A016	Die parametrierten Bereiche data_area_1 und data_area_6 überlappen.	
A017	Die parametrierten Bereiche data_area_1 und data_area_7 überlappen.	
A018	Die parametrierten Bereiche data_area_1 und data_area_8 überlappen.	
A019	Einer der Parameter db wurde auf 0 gesetzt, obwohl der zugehörige data_type mit > 0 parametriert ist.	Korrigieren Sie die Parametrierung am Parameter db auf > 0.
A01A	Falsche Länge im Header: Es sind 1 bis 253 Byte zulässig. Die Kommunikationsverbindung wird abgebaut.	Überprüfen Sie mit Hilfe einer Telegrammaufzeichnung die Daten des Koppelpartners.

1.11 Parameter ERROR, STATUS\_MODBUS, STATUS\_CONN und STATUS\_FUNC

STATUS* (W#16#)	Beschreibung	Abhilfe
A01B	S7 ist Server und Funktionscode 5: Es wurde ein ungültiger Zustand für Coil empfangen. Die S7 antwortet mit einem Exception-Telegramm.	Überprüfen Sie mit Hilfe einer Telegammaufzeichnung die Daten des Koppelpartners.
A023	Die parametrisierten Bereiche data_area_2 und data_area_3 überlappen.	Korrigieren Sie die Parametrierung im Parameter-DB.  Die Datenbereiche dürfen keinen gemeinsamen Registeradressbereich besitzen.
A024	Die parametrisierten Bereiche data_area_2 und data_area_4 überlappen.	
A025	Die parametrisierten Bereiche data_area_2 und data_area_5 überlappen.	
A026	Die parametrisierten Bereiche data_area_2 und data_area_6 überlappen.	
A027	Die parametrisierten Bereiche data_area_2 und data_area_7 überlappen.	
A028	Die parametrisierten Bereiche data_area_2 und data_area_8 überlappen.	
A034	Die parametrisierten Bereiche data_area_3 und data_area_4 überlappen.	
A035	Die parametrisierten Bereiche data_area_3 und data_area_5 überlappen.	
A036	Die parametrisierten Bereiche data_area_3 und data_area_6 überlappen.	
A037	Die parametrisierten Bereiche data_area_3 und data_area_7 überlappen.	
A038	Die parametrisierten Bereiche data_area_3 und data_area_8 überlappen.	
A045	Die parametrisierten Bereiche data_area_4 und data_area_5 überlappen.	
A046	Die parametrisierten Bereiche data_area_4 und data_area_6 überlappen.	
A047	Die parametrisierten Bereiche data_area_4 und data_area_7 überlappen.	
A048	Die parametrisierten Bereiche data_area_4 und data_area_8 überlappen.	
A056	Die parametrisierten Bereiche data_area_5 und data_area_6 überlappen.	
A057	Die parametrisierten Bereiche data_area_5 und data_area_7 überlappen.	
A058	Die parametrisierten Bereiche data_area_5 und data_area_8 überlappen.	
A067	Die parametrisierten Bereiche data_area_6 und data_area_7 überlappen.	
A068	Die parametrisierten Bereiche data_area_6 und data_area_8 überlappen.	
A078	Die parametrisierten Bereiche data_area_7 und data_area_8 überlappen.	
A079	Die am Parameter ID angegebene Verbindungs-ID ist im Parameter-DB nicht enthalten.	Korrigieren Sie die Parametrierung am Eingang id.
A07A	Es wurde ein unzulässiger Wert am Parameter ID angegeben (Wertebereich von 1 bis 4095).	
A07B	Die angegebene ID ist im Parameter-DB doppelt enthalten.	Korrigieren Sie die Parametrierung im Parameter-DB.
A07C	Im Parameter-DB wurde ein unzulässiger Wert am Parameter data_type angegeben (zulässige Werte sind 0 bis 4).	
A07D	Im Parameter-DB enthält der Parameter data_type des data_area_1 keinen Eintrag. Der Parameterbereich „1“ ist der Initialbereich und muss parametrisiert werden.	
A07E	An db wurde die Nummer des Parameter-DBs oder die Nummer des Instanz-DBs der Anweisung MODBUSPN angegeben.	
A07F	Der an DB_PARAM angegebene DB ist kein Modbus-Parameter-DB. Es wurde die Längenangabe im DBW0 geändert oder ein falscher DB angegeben.	Korrigieren Sie die Parametrierung am Eingang DB_PARAM.
A080	Diese Fehlermeldung tritt auf, wenn für den Aufruf der Anweisung MODBUSPN im OB1 bzw. Weckalarm-OB und OB100 unterschiedliche Instanz-DBs verwendet werden.	Verwenden Sie denselben IDB im OB100 und im zyklischen OB.

1.11 Parameter ERROR, STATUS\_MODBUS, STATUS\_CONN und STATUS\_FUNC

STATUS* (W#16#)	Beschreibung	Abhilfe
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.	Stoßen Sie erst dann einen neuen Auftrag an, wenn der vorherige Auftrag mit DONE_NDR = TRUE oder ERROR = TRUE beendet wurde.
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.	Prüfen Sie auf unerlaubte Schreibzugriffe auf den Lizenz-DB. Wenden Sie sich ggf. an den Produkt Support.
A086	Es wurde versucht in einen schreibgeschützten Datenbaustein zu schreiben.	Entfernen Sie den Schreibschutz oder verwenden Sie einen anderen DB.
A090	Der 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 IDENT_CODE für diese CPU aus und fordern Sie damit den Freischaltungscode an. Siehe „Lizenzierung (Seite 17)“.
A091	Als Antwort wurde ein Exception-Telegramm mit Exception Code 1 empfangen (nur bei S7 ist Client).	Der Koppelpartner unterstützt die geforderte Funktion nicht.
A092	Als Antwort wurde ein Exception-Telegramm mit Exception Code 2 empfangen (nur bei S7 ist Client). Es erfolgte ein Zugriff auf eine nicht vorhandene/ nicht zulässige Adresse beim Koppelpartner.	Korrigieren Sie LENGTH bzw. START_ADDRESS beim Aufruf der Anweisung.
A093	Als Antwort wurde ein Exception-Telegramm mit Exception Code 3 empfangen (nur bei S7 ist Client).	Der Koppelpartner kann das empfangene Telegramm nicht verarbeiten (z.B. er unterstützt die angeforderte Länge nicht).
A094	Als Antwort wurde ein Exception-Telegramm mit Exception Code 4 empfangen (nur bei S7 ist Client).	Der Koppelpartner ist in einem Zustand in dem er das empfangene Telegramm nicht bearbeiten kann.
A095	Als Antwort wurde ein Exception-Telegramm mit einem unbekanntem Exception Code empfangen (nur bei S7 ist Client)	Kontrollieren Sie die Fehlermeldungen des Koppelpartners und überprüfen Sie ggf. die Daten mit einer Telegrammaufzeichnung.

Parameter STATUS\_MODBUS bei STATUS\_FUNC = 'RD\_SINFO', 'BLKMOV', 'TEST\_DB', 'RDSYSST' oder 'WR\_USMSG'

STATUS* (W#16#)	Beschreibung	Abhilfe
7xxx	Bitte entnehmen Sie die detaillierte Information der Online-Hilfe.	Siehe Online-Hilfe (TIA Portal -> Baustein markieren -> Taste F1)
80B1	STATUS_FUNC = 'TEST_DB': Der DB existiert nicht auf der CPU.	Alle Datenbausteine, die an db angegeben werden, müssen angelegt und auf die CPU übertragen werden.
80B2	STATUS_FUNC = 'TEST_DB': DB UNLINKED	DB nicht als UNLINKED generieren.
8xxx	Bitte entnehmen Sie die detaillierte Information der Online-Hilfe.	Siehe Online-Hilfe (TIA Portal -> Baustein markieren -> Taste F1)

Parameter STATUS\_CONN bei STATUS\_FUNC = 'MODBUSPN'

STATUS* (W#16#)	Beschreibung	Abhilfe
A100	Für einen Auftrag ist die Überwachungszeit CONN_TIME oder RECV_TIME abgelaufen. Bei Ablauf der RECV_TIME wird die Verbindung abgebaut.	Prüfen Sie die Parametrierung der Verbindung.
A101	Die interne Überwachungszeit der Funktion TDISCON ist abgelaufen.	Kontaktieren Sie den Produkt Support.

Parameter STATUS\_CONN bei STATUS\_FUNC = 'TCON', 'TSEND', 'TRCV' oder 'TDISCON'

STATUS* (W#16#)	Beschreibung	Abhilfe
7xxx	Siehe Online-Hilfe.	Siehe Online-Hilfe (TIA Portal -> Baustein markieren -> Taste F1)
8xxx	Siehe Online-Hilfe.	Siehe Online-Hilfe (TIA Portal -> Baustein markieren -> Taste F1)

Siehe auch

Beschreibung MODBUSPN (Seite 13)



# Index

## M

MODBUSPN, 5

