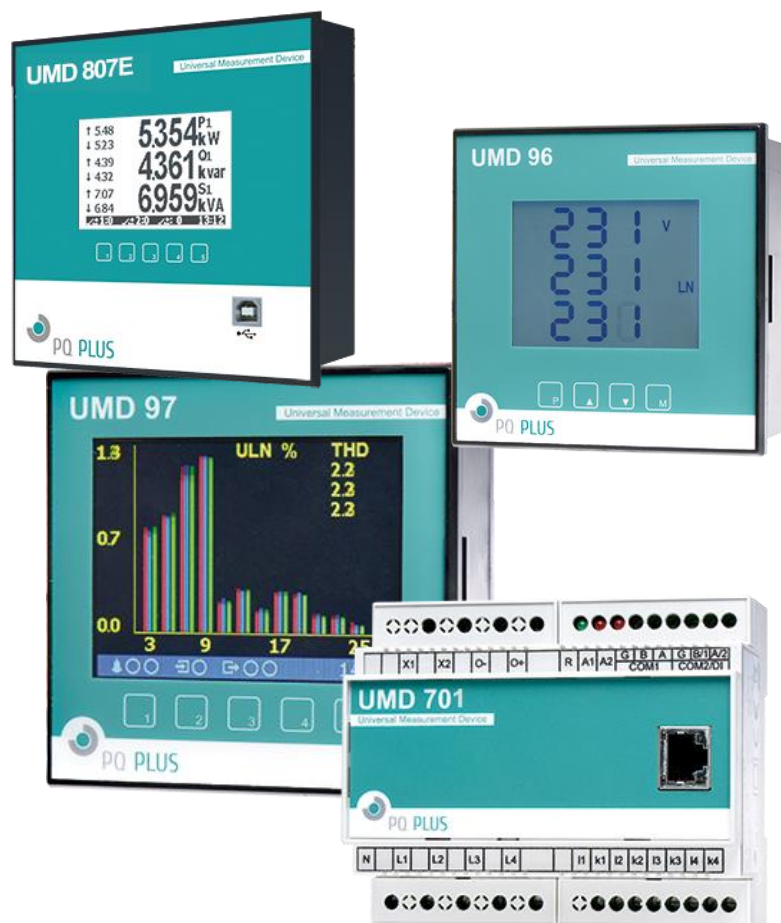


Softwarebeschreibung

# Modbus Kopplung Simatic S7 (STEP 7)

mit Simatic S7 300 über CP343-1

V2.0 - 1/2016



# Inhaltsverzeichnis

<b>1</b>	<b>Haftung und Gewährleistung</b>	<b>3</b>
<b>2</b>	<b>STEP7 V5.x</b>	<b>4</b>
2.1	Beschreibung	4
2.2	Lieferumfang	4
2.3	Einstellung am UMD der Fa. PQ PLUS GmbH	4
2.4	Verbindung über CP343-1	4
2.5	Verbindung im NetPro anlegen	5
2.6	FB 193 Parameter	6
2.6.1	Datenablage im RX_DB	8
2.7	Beispiel eines Programmaufrufes	9
2.8	INDEX	10

# 1 Haftung und Gewährleistung

---

Der Inhalt dieser Dokumentation ist nicht Teil einer früheren oder bestehenden Vereinbarung, Zusage oder eines Rechtsverhältnisses oder soll dieses abändern.

Die Fa. PQ Plus GmbH weist darauf hin, dass der Benutzer für den Einsatz der Software und deren programmtechnischen Einbindung selbst verantwortlich ist.

Die Verantwortung für die vorschriftsmäßige Errichtung der Gesamtanlage hat in diesem Fall der Anlagenbetreiber oder dessen Generalauftragnehmer.

## **Software**

Da Daten unter bestimmten Bedingungen in jedem elektronischen Speicher verloren gehen oder geändert werden, übernimmt die Fa. PQ Plus GmbH keine Haftung für Daten, die aufgrund missbräuchlicher Verwendung, Reparaturen bzw. Defekten an der Hardware oder aus irgendwelchen anderen Gründen verloren gehen oder anderweitig unbrauchbar werden.

## **HINWEIS**

Die Fa. PQ Plus GmbH übernimmt keine Haftung, direkt oder indirekt, für finanzielle Verluste oder Schadensansprüche Dritter, die aus der Nutzung dieser Software und seiner Funktionen entstehen.

Zur Nutzung der Software ist nur der jeweilige Vertragspartner berechtigt, eine Weitergabe an Dritte ist nicht gestattet.

Diese Bedienungsanleitung wurde mit der gebotenen Sorgfalt erarbeitet; es wird keinerlei Haftung für Fehler oder Auslassungen übernommen. Ebenso wird für Schäden die sich aus der Nutzung von Informationen dieser Bedienungsanleitung ergeben nicht gehaftet.

## 2 STEP7 V5.x

### 2.1 Beschreibung

Der FB 193 wird verwendet, um die Daten eines UMD mittels CP343-1 Ethernet-Kopplung auszulesen. Die Register können einzeln oder blockweise gelesen werden.

Die Verwendung ist mit allen CPU's mittels Profinet-CP343-1 möglich.

Die Datenübergabe erfolgt in einem frei wählbaren Datenbaustein.

Die Daten werden mittels TCP/IP Kommunikation vom angeschlossenen UMD gelesen.

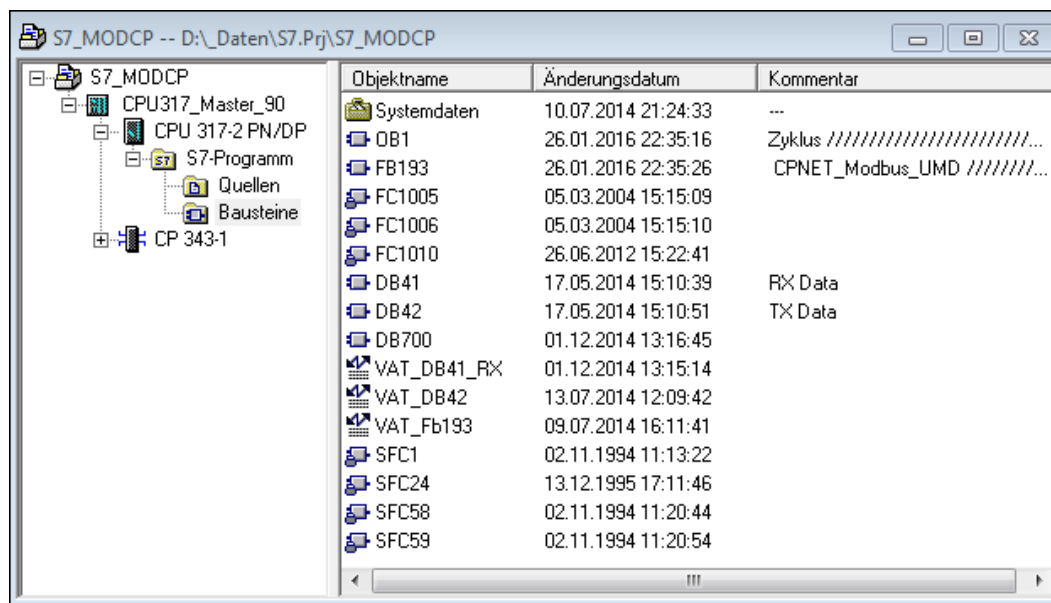
#### Einbindung des FB 193 ins Programm

- FB 193 wird mit einem zugeordneten Instanz-DB aufgerufen. Die Nummer des DB's ist frei wählbar.
- FB 193 kann auch umbenannt werden.
- FB 193 ist Multi-Instanz fähig.
- Der Aufruf erfolgt im OB1.
- Es werden folgende umbenannte Siemens Standard-Bausteine verwendet (im Projektumfang enthalten):  
**FC 1005** – AG\_SEND (FC5)  
**FC 1006** – AG\_RECV (FC6)  
**FC 10** – AG\_CNTRL (FC10)  
**SFC 1** – Read\_CLK  
**SFC 24** – Test\_DB

Die Standard FB's können nicht umbenannt werden

### 2.2 Lieferumfang

Der Lieferumfang besteht aus einem Simatic STEP7 V5.5 Projekt, welches unter Step7 V5.4/5.5 oder Step7 Professional 2006/2010 eingesetzt werden kann.



Lieferumfang des STEP7 – Projektes

### 2.3 Einstellung am UMD der Fa. PQ PLUS GmbH

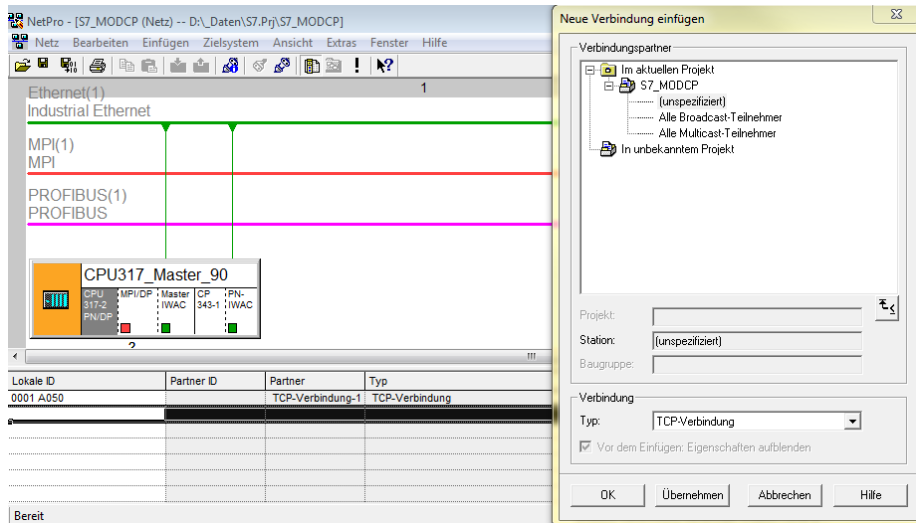
Für eine fehlerfreie Kommunikation muss am UMD die in diesem STEP7 Projekt gewählte Ethernet-Adresse eingestellt werden (man beachte auch die Subnet – Maske). Die UMD-Adresse muss in der NetPro Verbindung eingetragen werden.

### 2.4 Verbindung über CP343-1

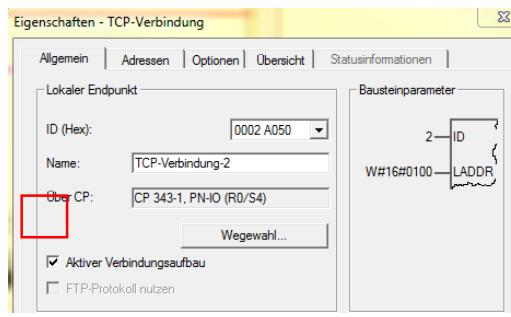
Es wurde der CP343-1 mit Firmwarestand V3.0 getestet. Es können hierbei maximal 16 offene Verbindungen angelegt und betrieben werden je CP.

## 2.5 Verbindung im NetPro anlegen

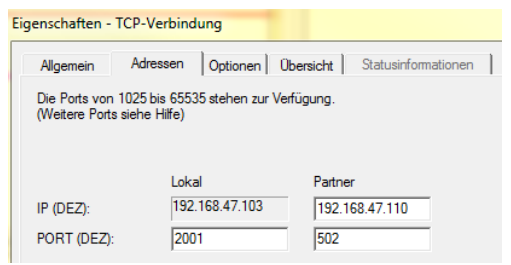
Im Simatic NetPro muss dann eine Verbindung für jeden UMD Strang angelegt werden.



Verbindung erstellen



Anlegen der TCP-Verbindungseigenschaften (aktiv)



	Lokal	Partner
IP (DEZ):	192.168.47.103	192.168.47.110
PORT (DEZ):	2001	502

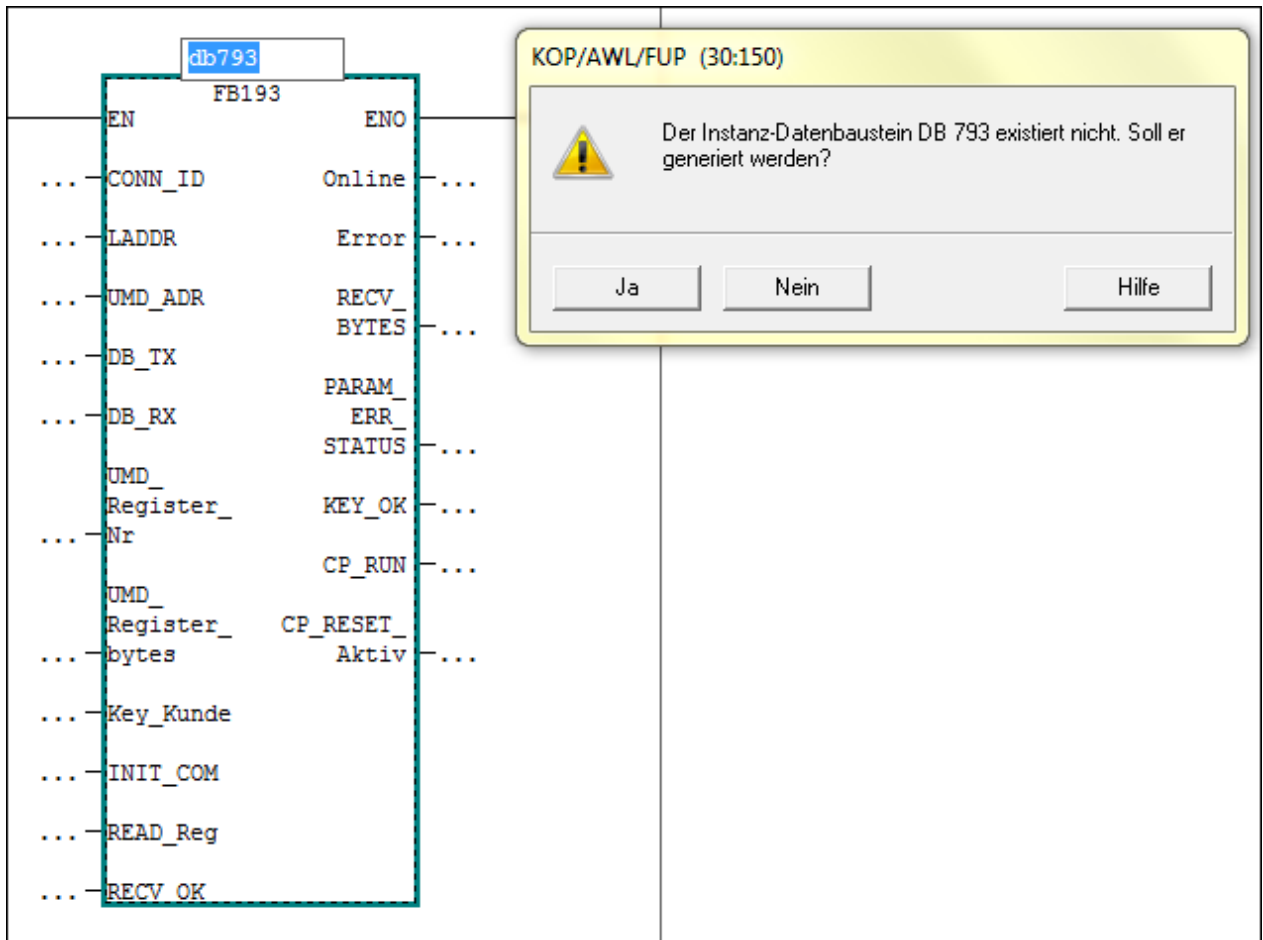
Eintragen der IP-Adresse und des Ports (UMD immer 502)

*Hinweis: Die für jede Verbindung angezeigte ID und die LADDR muss am Treiber des FB193 ebenfalls angegeben werden.*

## 2.6 FB 193 Parameter

Aufruf im Programm:

Man öffne einen Baustein für die zu programmierende Verbindung und integriere den beigegebenen FB 193.



Beim Anlegen wird nach dem gewünschten Instanz DB gefragt.

### Hinweis:

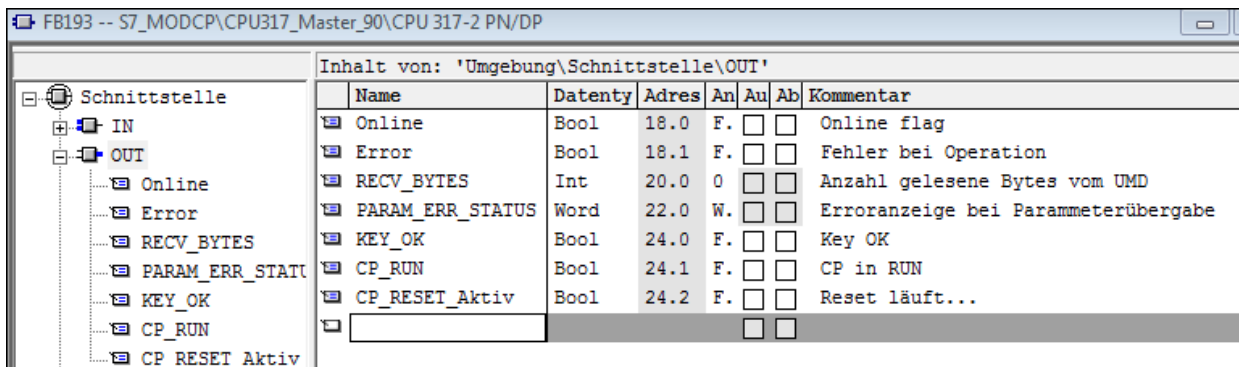
Die Bausteinnummer FB193 ist jederzeit änderbar (..speichern als ..).

In der folgenden Tabelle werden die Input Parameter zusammengestellt:

Schnittstelle		Inhalt von: 'Umgebung\Schnittstelle\IN'					
Name	Datent	Adress	Anfang	Au	Ab	Kommentar	
CONN_ID	Int	0.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Connection ID (HW-config ID) [1..16]	
LADDR	Word	2.0	W#16#0	<input type="checkbox"/>	<input type="checkbox"/>	CP-Adresse ((HW-config ID)	
UMD_ADR	Int	4.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Geräteadresse UMD (1...x)	
DB_TX	Int	6.0	0	<input type="checkbox"/>	<input type="checkbox"/>	DB Nr zum Senden	
DB_RX	Int	8.0	0	<input type="checkbox"/>	<input type="checkbox"/>	DB Nr zum Empfangen...	
UMD_Register_Nr	Int	10.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Registernummerzum Auslesen	
UMD_Register_bytes	Int	12.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Anzahl Bytes zum Lesen	
Key_Kunde	DInt	14.0	L#0	<input type="checkbox"/>	<input type="checkbox"/>	Key des Kunden...	

Input	Datentyp	Beschreibung
CONN_ID	INT	Nummer der Verbindung [1..16] gemäß NetPro
LADDR	WORD	CP-Adresse [z.B. W#16#0100 für Adresse 256dez] gemäß NetPro
UMD_ADR	INT	Nummer der Modbusadresse des UMD Gerätes (z.B. 1),
DB_TX	INT	Nummer eines Datenbausteines zum Senden von Daten über den Modbus (mind. Länge: 12 Bytes) Bereich: 1 – 16.000. Dieser DB wird intern beschrieben und sollte vom Anwender nicht geändert werden.
DB_RX	INT	Nummer eines Datenbausteines zum Empfangen von Daten über den Modbus (Länge: min 150 Worte). Bereich: <b>1 – 16.000</b> Nach Übergabe der Parameter und dem Start mit READ_Reg = TRUE werden bei fehlerfreier Ausführung die Daten in dem DB ab Datenwort DBW0 abgelegt. Je nach Format des/der gelesenen Register müssen die entsprechenden Formate im DB zur Auswertung gewählt werden (z.B. Simatic Gleitpunkt für Float Register). Der Header wird ebenfalls mit abgelegt, die Nettodaten stehen ab DBB9 im DB_RX
UMD_Register_Nr	INT	Angabe des ersten zu lesenden Registers (Nummern siehe UMD Anleitung PQ Plus GmbH) ( <b>0..32767</b> )
UMD_Register_bytes	INT	Anzahl der zu lesenden Bytes Bereich: <b>2 – 200</b> Es können somit auch Registerbereiche gelesen werden (z.B. 12 Byte entsprechen 3 Float-Register (für 3 Phasenströme o.ä.)).
Key_Kunde	DINT	Angabe des mitgelieferten Kundenschlüssels (Key)

In der folgenden Tabelle werden die OUTPUT Parameter zusammengestellt:



Inhalt von: 'Umgebung\Schnittstelle\OUTI'							
Name	Datentyp	Adres	An	Au	Ab	Kommentar	
Online	Bool	18.0	F.	<input type="checkbox"/>	<input type="checkbox"/>	Online flag	
Error	Bool	18.1	F.	<input type="checkbox"/>	<input type="checkbox"/>	Fehler bei Operation	
RECV_BYTES	Int	20.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Anzahl gelesene Bytes vom UMD	
PARAM_ERR_STATUS	Word	22.0	W.	<input type="checkbox"/>	<input type="checkbox"/>	Erroranzeige bei Parameterübergabe	
KEY_OK	Bool	24.0	F.	<input type="checkbox"/>	<input type="checkbox"/>	Key OK	
CP_RUN	Bool	24.1	F.	<input type="checkbox"/>	<input type="checkbox"/>	CP in RUN	
CP_RESET_Aktiv	Bool	24.2	F.	<input type="checkbox"/>	<input type="checkbox"/>	Reset läuft...	

Output	Datentyp	Beschreibung
Online	BOOL	Zeigt den Status Verbindung zum UMD an: TRUE: Verbindung ist aufgebaut FALSE: keine Verbindung ist aufgebaut
Error	BOOL	Signalisiert einen Fehler bei der Parameterübergabe des Anwenders. TRUE: Fehler in den Parametern, Detailinformation findet man unter PARAM_ERR_STATUS
RECV_BYTES	INT	Anzahl der empfangenen Bytes im DB_RX
PARAM_ERR_STATUS	WORD	0h: kein Fehler aufgetreten 8xxxh: Fehler bei Parameterübergabe (Details siehe Liste im STEP7 Projekt) Fehler können sein: DB nicht vorhanden, DB zu kurz, CPU type falsch, IP Adresse ungültig usw.
KEY_OK	BOOL	TRUE: Schlüsselnummer OK
CP_RUN	BOOL	TRUE: CP343 im Run Modus
CP_RESET_AKTIV	BOOL	TRUE: CP wird initialisiert

Vor einem Datenabruf muss das Flag „KEY\_OK“ = true sein, ebenso muss das Flag „Online“ = true sein.

In der folgenden Tabelle werden die IN OUT Parameter zusammengestellt:

Schnittstelle		Inhalt von: 'Umgebung\Schnittstelle\IN_OUT'					
Name	Datentyp	Adresse	An	Au	Ab	Kommentar	
INIT_COM	Bool	26.0	F.	<input type="checkbox"/>	<input type="checkbox"/>	Befehl zum Initialisieren des UMD (set vom Anwender)	
READ_Reg	Bool	26.1	F.	<input type="checkbox"/>	<input type="checkbox"/>	Start Lese Register vom UMD (set vom Anwender)	
RECV_OK	Bool	26.2	F.	<input type="checkbox"/>	<input type="checkbox"/>	Daten vorhanden im DB_RX (set von FB193)...	

Output	Datentyp	Beschreibung
INIT_COM	BOOL	Muss vom Anwender auf TRUE gesetzt werden, wenn die Verbindung neu initialisiert werden soll (z.B. nach Netz-Aus oder beim Auftreten der Offline-Meldung)
READ_Reg	BOOL	Nach dem Eintragen der gewünschten Registernummer(n) und der Anzahl der Bytes wird der Lesevorgang durch Setzen auf TRUE gestartet.  Nach Ausführung wird die Variable vom Treiber auf FALSE gesetzt.  Wenn das Lesen der Register erfolgreich war, wird das Flag <i>RECV_OK</i> auf TRUE gesetzt, die Anzahl der Bytes wird in <i>RECV_BYTES</i> eingetragen und die Daten mit Header werden im <i>DB_RX</i> ab Datenwort 0 eingetragen. <u>Die Nettodaten stehen ab Byte 9 im <i>DB_RX</i>.</u>  Wenn das Lesen der Register nicht erfolgreich war, wird das Flag <i>Error</i> auf TRUE gesetzt und die Fehlernummer wird in der Variable <i>PARAM_ERR_STATUS</i> eingetragen.
RECV_OK	BOOL	FALSE: kein Auftrag gelesen TRUE: Daten wurden empfangen. Das Bit sollte vom Anwender nach der Verarbeitung der Daten wieder auf FALSE gesetzt werden.

### 2.6.1 Datenablage im RX\_DB

Die empfangenen Daten werden im **DB\_RX** abgelegt (im Beispiel DB41) und es wird das Bit **RECV\_OK** gesetzt. Die Nettodaten werden ab Byte 9 im DB abgelegt (DBB9 und folgende).

Operand	Statuswert	Steuerwert
<b>//RX-Daten</b>		
<b>//Telegrammkopf Antwort UMG</b>		
DB41.DBB 0	0	//Transaktions Id High
DB41.DBB 1	1	//Transaktions Id Low
DB41.DBB 2	0	// NULL
DB41.DBB 3	0	// NULL
DB41.DBB 4	0	//Anzahl folgender Bytes Hi
DB41.DBB 5	7	//Anzahl folgender Bytes Lo
DB41.DBB 6	1	// Modbus Slave Adresse
DB41.DBB 7	3	//Funktionsnummer (3)
DB41.DBB 8	4	// Anzahl Datenbytes
<b>//Netto Daten</b>		
DB41.DBD 9	49.99464	//wert1
DB41.DBD 13	0.0	//Wert2
DB41.DBD 17	0.0	//Wert2
DB41.DBB 9	B#16#42	
DB41.DBB 10	B#16#47	
DB41.DBB 11	B#16#FA	
DB41.DBB 12	B#16#82	
DB41.DBB 13	B#16#00	

Beispiel: Empfang von 4 Bytes  
Abgefragt wurde das Register 4100 [1004hex] (Istfrequenz) des UMD 97.



Operand	Statuswert	Steuerwert
<b>//RX-Daten</b>		
<b>//Telegrammkopf Antwort UMG</b>		
DB41.DBB 0	0	//Transaktions Id High
DB41.DBB 1	1	//Transaktions Id Low
DB41.DBB 2	0	// NULL
DB41.DBB 3	0	// NULL
DB41.DBB 4	0	//Anzahl folgender Bytes Hi
DB41.DBB 5	15	//Anzahl folgender Bytes Lo
DB41.DBB 6	1	// Modbus Slave Adresse
DB41.DBB 7	3	//Funktionsnummer (3)
DB41.DBB 8	12	// Anzahl Datenbytes
<b>//Netto Daten</b>		
DB41.DBD 9	234.6838	//wert1
DB41.DBD 13	234.7173	//Wert2
DB41.DBD 17	234.6703	//Wert2
DB41.DBB 9	B#16#43	
DB41.DBB 10	B#16#6A	
DB41.DBB 11	B#16#AF	
DB41.DBB 12	B#16#0C	
DB41.DBB 13	B#16#43	
DB41.DBB 14	B#16#6A	
DB41.DBB 15	B#16#B7	
DB41.DBB 16	B#16#A4	
DB41.DBB 17	B#16#43	
DB41.DBB 18	B#16#6A	
DB41.DBB 19	B#16#AB	
DB41.DBB 20	B#16#9B	
DB41.DBB 21	B#16#00	

Beispiel: Empfang von 12 Bytes

Abgefragt wurden die Register 4352/4352/4356 [1100/1102/11104hex] (U1/U2/U3) des UMD97.

## 2.7 Beispiel eines Programmaufrufes

Im folgenden Beispiel wird der Treiber im OB1 aufgerufen.

```

CALL FB 193 , DB700
CONN_ID      :=1           // Connection ID (1..16)
LADDR       :=W#16#100    // Adresse des CP343
UMD_ADR     :=1           // Geräteadresse UMD (1..32)
DB_TX       :=42          // DB-Nr. Sendespuffer [1..16000]
DB_RX       :=41          // DB-Nr. Empfangspuffer [1..16000]
UMD_Register_Nr :=19000   // 1. zu lesendes Register (z.B. 19000)
UMD_Register_bytes:=16    // Anzahl der Registerbytes (2/4...200)
Key_Kunde   :=L#12345678 // Kundenschlüssel (individuell)
Online      :=M10.0      // true=Online Meldung
Error       :=M10.1      // true=Fehler aufgetreten
RECV_BYTES  :=MW1002     // Anzahl Bytes empfangen (Netto Daten)
PARAM_ERR_STATUS :=MW1004 // Error Register
KEY_OK      :=M11.2      // true=Kundenschlüssel OK
CP_RUN      :=M11.3      // true=CP in RUN modus
CP_RESET_Aktiv :=M10.1   // true=CP im Reset/ Init modus
INIT_COM    :=M10.2      // true=Init Schnittstelle (Reset intern)
READ_Reg    :=M11.0      // true=Start Anfordern Registerdaten (Reset intern)
RECV_OK     :=M11.1      // true=Daten wurden im RX-DB abgelegt! (Reset nach auswerten durch Anwender)
  
```

Zur Erklärung der Parameterversorgung (für UMD97):

Die Geräteadresse Modbus UMD wird auf **1** eingestellt.

Als Sendebaustein wird der DB **42** verwendet, als Empfangsbaustein wird der DB **41** verwendet.

Das/die zu lesende(n) Register wird im Beispiel mit **4100 [auf MW1010 geschrieben]** (Istfrequenz) vom Anwender übergeben.

Die Anzahl zu lesender Registerbytes werden im Beispiel mit **4 [auf MW1012 geschrieben]** vom Anwender übergeben.

Eine Übergabe über Datenworte ode o.ä. ist ebenfalls möglich.

Die Anzahl der empfangenen Netto Bytes werden vom Treiber im MW1002 eingetragen.

Für die Status anzeigen (BOOL) werden Merker verwendet.

### Key\_Kunde:

Hier muss die mitgelieferte Schlüsselzahl eingegeben werden, um eine Kommunikation zu ermöglichen. Nach 10 Versuchen mit falschen Keycode wird die Verbindung blockiert. Danach muss man den Instanz-DB neu laden, um wieder eine Kommunikation mit dem korrekten Key aufbauen zu können.

**Beispiel: Lese Register 4100 – Measured frequency**

VAT_Fb192 -- @S7_MODB1\CPU317_Master_90\CPU 317-2 PN/DP\S7-Programm ONLINE					
	Operand	Symbol	Anzeigeform	Statuswert	Steuerwert
1	M 10.0	"ONLINE"	BOOL	true	
2	M 10.1	"ERROR"	BOOL	false	
3	M 10.2	"INIT"	BOOL	false	
4	M 11.0	"REQU_DA	BOOL	false	true
5	M 11.1	"RX_DATA	BOOL	true	
6	M 11.2		BOOL	true	
7	Z 20		ZÄHLER	C#0	
8	// Nr. Register				
9	MW 1010		DEZ	19050	19050
10	//Nr. Bytes				
11	MW 1012		DEZ	4	4
12	MW 1002	"RX_Bytes	DEZ	13	
13	MW 1004	"ERR_Nr"	HEX	W#16#0000	
14	DB41.DBD 9		GLEITPU...	49.9798	// 1.Wert
15	DB41.DBD 13		GLEITPU...	0.0	// 2.Wert
16	DB41.DBD 17		GLEITPU...	0.0	// 3.Wert
17	DB41.DBD 21		GLEITPU...	0.0	// 4.Wert

**Beispiel: Lese Register 4352 - 4356 – Voltage U1-N / U2-N / U3-N**

VAT_Fb192 -- @S7_MODB1\CPU317_Master_90\CPU 317-2 PN/DP\S7-Programm ONLINE					
	Operand	Symbol	Anzeigeform	Statuswert	Steuerwert
1	M 10.0	"ONLINE"	BOOL	true	
2	M 10.1	"ERROR"	BOOL	false	
3	M 10.2	"INIT"	BOOL	false	
4	M 11.0	"REQU_DA	BOOL	false	true
5	M 11.1	"RX_DATA	BOOL	true	
6	M 11.2		BOOL	true	
7	Z 20		ZÄHLER	C#0	
8	// Nr. Register				
9	MW 1010		DEZ	19000	19000
10	//Nr. Bytes				
11	MW 1012		DEZ	12	12
12	MW 1002	"RX_Bytes	DEZ	21	
13	MW 1004	"ERR_Nr"	HEX	W#16#0000	
14	DB41.DBD 9		GLEITPU...	234.7674	// 1.Wert
15	DB41.DBD 13		GLEITPU...	234.7914	// 2.Wert
16	DB41.DBD 17		GLEITPU...	234.7497	// 3.Wert
17	DB41.DBD 21		GLEITPU...	0.0	// 4.Wert

## 2.8 INDEX

### *B*

Beispiel eines Programmaufrufes · 10  
**Beispiel Phasenspannung** · 11

**Input parameter** · 6  
**IP Adresse** · 6

### *C*

**Connection ID** · 6  
**CPU Typ** · 6

### *L*

Lieferumfang · 4

### *E*

Einbindung ins Anwenderprogramm · 5

### *O*

**Output parameter** · 7

### *F*

FB192 Parameter\_TIA · 6

### *P*

**Port Nummer** · 6  
 Programmaufruf · 5

### *H*

Haftung · 3

### *T*

TIA V12 · 4

### *I*

**In\_Out parameter** · 8

### *U*

**UMG\_Modbus**