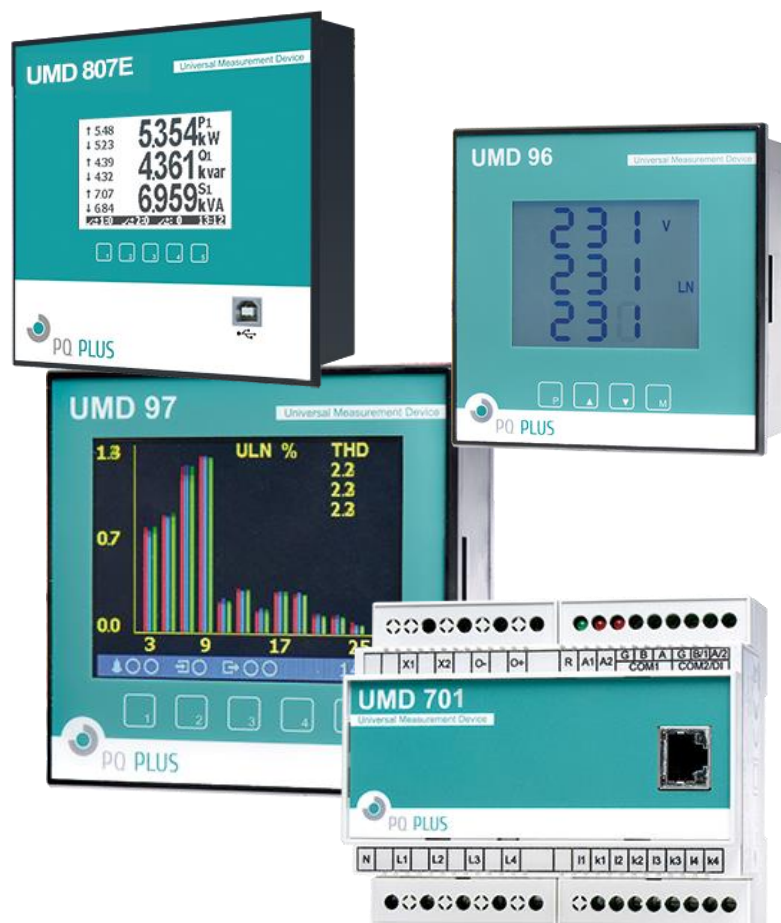


Softwarebeschreibung

Modbus Kopplung Simatic S7 PN

mit Simatic S7 300/400 über Profinet (OnBoard Ethernet)

V2.0 - 1/2016



Inhaltsverzeichnis

1. Haftung und Gewährleistung.....	3
2. Allgemein	4
2.1 Beschreibung.....	4
2.2 Lieferumfang.....	4
2.3 Einstellung am UMD der Fa. PQ PLUS GmbH	4
3. FB 192 Parameter	5
3.1 Aufruf im Programm.....	5
3.2 Datenablage im RX_DB.....	8
3.3 Beispiel eines Programmaufrufes.....	9
3.4 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. Allgemein

2.1 Beschreibung

Der FB 192 wird verwendet, um die Daten eines UMD mittels Ethernet-Kopplung auszulesen. Die Register können einzeln oder blockweise gelesen werden.

Die Verwendung ist mit allen CPU's mit Onboard Profinet – Schnittstelle oder mittels Profinet-CP möglich.

Die Datenübergabe erfolgt in einem frei wählbaren Datenbaustein.

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

Einbindung des FB 192 ins Programm

- FB 192 wird mit einem zugeordneten Instanz-DB aufgerufen. Die Nummer des DB's ist frei wählbar.
- FB 192 kann auch umbenannt werden.
- FB 192 ist Multi-Instanz fähig.
- Der Aufruf erfolgt im OB1.
- Es werden folgende umbenannte Siemens Standard-Bausteine verwendet (im Projektumfang enthalten):
FB 1063 – SendData (FB63)
FB 1064 – ReceiveData (FB64)
FB 1065 – Connect (FB65)
FB 1066 – Disconnect (FB66)

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.

Objektname	Änderungsdatum	Kommentar
DB41	17.05.2014 15:10:39	RX Data
DB42	17.05.2014 15:10:51	TX Data
DB192	25.03.2015 16:35:38	InstanzDB
FB192	26.01.2016 21:57:13	NET_Modbus_UMD /////...
FB1063	19.09.2006 08:34:56	Send Data
FB1064	10.01.2007 15:15:49	Receive Data
FB1065	22.09.2008 15:00:13	Connect
FB1066	19.09.2006 08:34:57	Disconnect
OB1	26.01.2016 22:00:51	Zyklus //////////////////////////////////...
SFC1	02.11.1994 11:13:22	
SFC24	13.12.1995 17:11:46	
Systemdaten	17.05.2014 14:57:38	---
UDT65	17.06.2013 17:26:04	Connection Parameters for ...
VAT_Fb192	08.07.2014 17:21:36	
VAT_LAN	18.11.2013 21:51:39	

Lieferumfang des STEP7 – Projektes

2.3 Einstellung am UMD der Fa. PQ PLUS GmbH

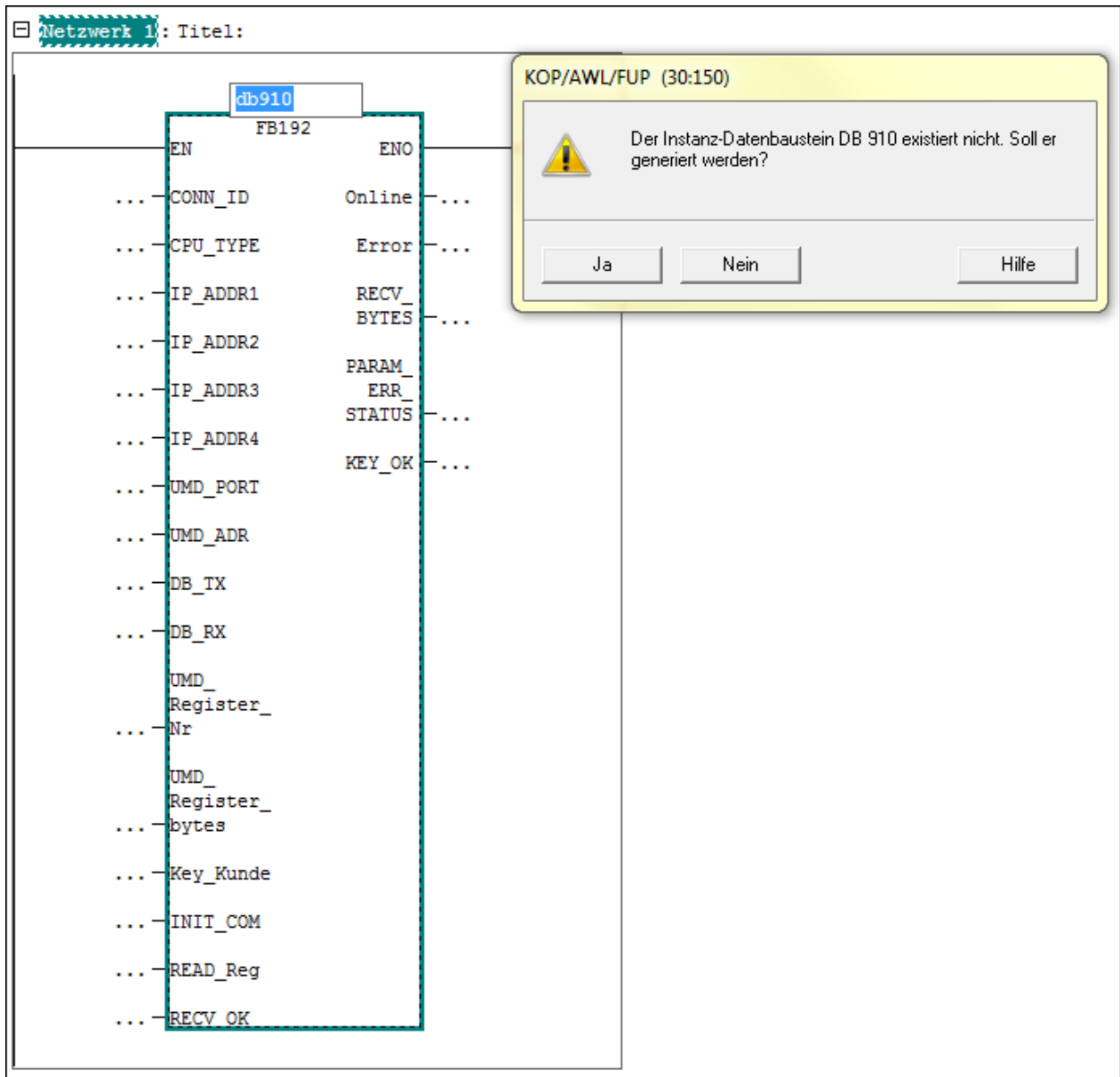
Für eine fehlerfreie Kommunikation muss am UMD die in diesem STEP7 Projekt gewählte Ethernet-Adresse eingestellt werden (man beachte auch die Subnet – Maske).

In der Simatic Hardwarekonfiguration muss keine Verbindung für das UMD angelegt werden.

3. FB 192 Parameter

3.1 Aufruf im Programm


Man öffne einen Baustein für die zu programmierende Verbindung und integriere den beigeestellten FB 192.



The screenshot shows a software interface for configuring a function block (FB192). The block is titled "db910" and "FB192". The interface lists various parameters and their values:

Parameter	Value
EN	ENO
CONN_ID	Online
CPU_TYPE	Error
IP_ADDR1	RECV_BYTES
IP_ADDR2	PARAM_ERR_STATUS
IP_ADDR3	KEY_OK
IP_ADDR4	
UMD_PORT	
UMD_ADR	
DB_TX	
DB_RX	
UMD_Register_Nr	
UMD_Register_bytes	
Key_Kunde	
INIT_COM	
READ_Reg	
RECV_OK	

The dialog box "KOP/AWL/FUP (30:150)" contains the following text:

 Der Instanz-Datenbaustein DB 910 existiert nicht. Soll er generiert werden?

Buttons: Ja, Nein, Hilfe

Beim Anlegen wird nach dem gewünschten Instanz DB gefragt.

Hinweis:

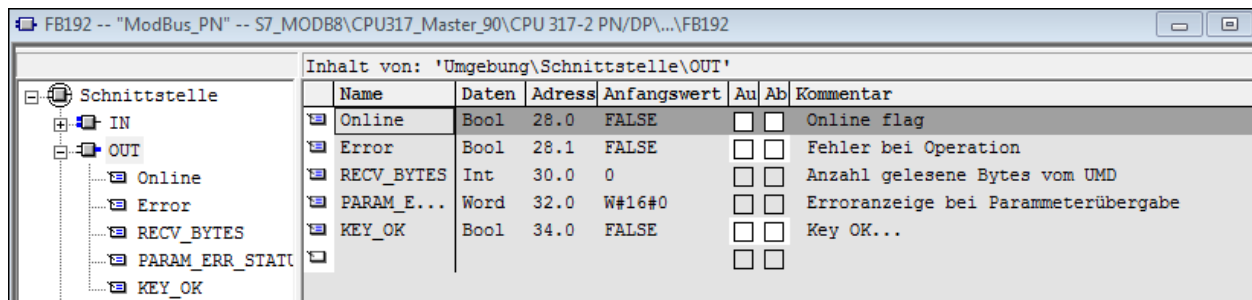
Die Bausteinnummer FB192 ist jederzeit änderbar (..speichern als ..).

In der folgenden Tabelle werden die Input Parameter zusammengestellt:

Name	Datentyp	Address	An	Au	Ab	Kommentar
CONN_ID	Int	0.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Connection ID (je Aufruf eigene ID) [1..1000]
CPU_TYPE	Int	2.0	0	<input type="checkbox"/>	<input type="checkbox"/>	0=CP443 2=315/317 3=319 5=413/414/416
IP_ADDR1	Int	4.0	0	<input type="checkbox"/>	<input type="checkbox"/>	UMD IP1
IP_ADDR2	Int	6.0	0	<input type="checkbox"/>	<input type="checkbox"/>	UMD IP2
IP_ADDR3	Int	8.0	0	<input type="checkbox"/>	<input type="checkbox"/>	UMD IP3
IP_ADDR4	Int	10.0	0	<input type="checkbox"/>	<input type="checkbox"/>	UMD IP4
UMD_PORT	Int	12.0	0	<input type="checkbox"/>	<input type="checkbox"/>	PortNr. UMD (Standard 502)
UMD_ADR	Int	14.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Geräteadresse UMD (1...x)
DB_TX	Int	16.0	0	<input type="checkbox"/>	<input type="checkbox"/>	DB Nr zum Senden
DB_RX	Int	18.0	0	<input type="checkbox"/>	<input type="checkbox"/>	DB Nr zum Empfangen...
UMD_Register_Nr	Int	20.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Registernummer zum Auslesen
UMD_Register_bytes	Int	22.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Anzahl Bytes zum Lesen
Key_Kunde	DInt	24.0	L.	<input type="checkbox"/>	<input type="checkbox"/>	Key des Kunden...

Input	Datentyp	Beschreibung
CONN_ID	INT	Nummer der Verbindung [1..1000]
CPU_TYPE	INT	Beschreibt die verwendete CPU / CP Type: 0-CP 443 2-CPU 315/317 3-CPU 319 4-CPU 4xx
IP_ADDR1	INT	IP-Adresse Teil 1 (z.B. 192)
IP_ADDR2	INT	IP-Adresse Teil 2 (z.B. 168)
IP_ADDR3	INT	IP-Adresse Teil 3 (z.B. 104)
IP_ADDR4	INT	IP-Adresse Teil 4 (z.B. 47)
UMD_PORT	INT	Nummer des UMD Ports des Gerätes (Default: 502)
UMD_ADR	INT	Nummer der Modbusadresse des UMD Gerätes (z.B. 1)
DB_RX	INT	Nummer eines Datenbausteines zum Senden von Daten über den Modbus (Länge: mind. 150 Worte) Bereich: 1 – 16.000. Dieser DB wird intern verwendet und sollte vom Anwender nicht geändert werden.
DB_TX	INT	Nummer eines Datenbausteines zum Empfangen von Daten über den Modbus (Länge: mind. 12 Bytes). Bereich: 1 – 16.000 Nach Übergabe der Parameter und dem Start mit <i>READ_Reg</i> = 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).
UMD_Register	INT	Angabe des ersten zu lesenden Registers (Nummern siehe UMD Anleitung PQ Plus) (0..32767)
UMD_Register_bytes	INT	Anzahl der zu lesenden Bytes Bereich: 2 – 160 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:

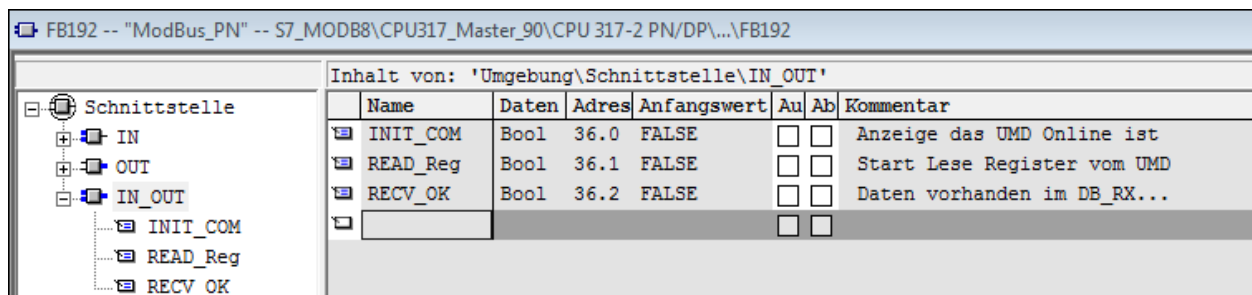


Name	Daten	Adress	Anfangswert	Au	Ab	Kommentar
Online	Bool	28.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	Online flag
Error	Bool	28.1	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	Fehler bei Operation
RECV_BYTES	Int	30.0	0	<input type="checkbox"/>	<input type="checkbox"/>	Anzahl gelesene Bytes vom UMD
PARAM_E...	Word	32.0	W#16#0	<input type="checkbox"/>	<input type="checkbox"/>	Erroranzeige bei Parameterübergabe
KEY_OK	Bool	34.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	Key OK...

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 <i>PARAM_ERR_STATUS</i>
RECV_BYTES	INT	Anzahl der empfangenen Bytes im <i>DB_RX</i>
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

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:



Name	Daten	Adres	Anfangswert	Au	Ab	Kommentar
INIT_COM	Bool	36.0	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	Anzeige das UMD Online ist
READ_Reg	Bool	36.1	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	Start Lese Register vom UMD
RECV_OK	Bool	36.2	FALSE	<input type="checkbox"/>	<input type="checkbox"/>	Daten vorhanden im DB_RX...

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 werden im <i>DB_RX</i> ab Datenwort 0 eingetragen. 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.

3.2 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
//RX-Daten		
//Telegrammkopf Antwort UMG		
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
//Netto Daten		
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 UMD97.

Operand	Statuswert	Steuerwert
//RX-Daten		
//Telegrammkopf Antwort UMG		
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
//Netto Daten		
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.

3.3 Beispiel eines Programmaufrufes

Im folgenden Beispiel wird der Treiber im FC2 aufgerufen.

```

Netzwerk 1: UMD 1

## NET COMM Modbus #####

CALL FB 192 , DB192
CONN_ID      :=1          // Connection ID (1.1000)
CPU_TYPE     :=2          // 2=315/317 | 3=319 | 5=41x
IP_ADDR1     :=192        // IP0 Adresse UMD
IP_ADDR2     :=168        // IP1
IP_ADDR3     :=47         // IP2
IP_ADDR4     :=111        // IP3
UMD_PORT     :=502        // Standard UMD (FIX)
UMD_ADR      :=1          // SGeräteadresse UMD
DB_TX        :=42         // DB-Nr. Sendepuffer
DB_RX        :=41         // DB-Nr. Empfangspuffer
UMD_Register_Nr :=19000    //MW 1010, // 1. zu lesendes Register (z.B. 19000)
UMD_Register_bytes:=100    //MW 1012, // Anzahl der Registerbytes (2/4...100)
Key_Kunde    :=L#26012016 // Kundenschlüssel (individuell)
Online       :=M10.0      // true=Online Meldung
Error        :=M10.1      // true=Fehler aufgetrete
RECV_BYTES   :=MW1002     // Anzahl Bytes empfangen (Netto Daten)
PARAM_ERR_STATUS :=MW1004 // Error Register
KEY_OK       :=M11.2      // true=Kundenschlüssel OK
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:

CPU_TYPE: 2 steht für CPU315/317

Die IP-Adresse des UMD wird mit **192.168.47.111** eingestellt.

Der Port des UMD wird auf **502** eingestellt.

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 eigegeben 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

	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

	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

3.4 INDEX

B

- Beispiel eines Programmaufrufes · 11
- Beispiel Lese Istfrequenz** · 12
- Beispiel Phasenspannung** · 12
- Beschreibung · 4

C

- Connection ID** · 7
- CPU Typ** · 7

E

- Einbindung ins Anwenderprogramm · 6

F

- FB192 Parameter · 6

H

- Haftung · 5

I

- In_Out parameter** · 9
- Input parameter** · 7
- IP Adresse** · 7

L

- Lieferumfang · 4

O

- Output parameter** · 8

P

- Port Nummer** · 7
- Programmaufruf · 6

U

- UMG_ModbusAdresse** · 7