

## Beispieldokumentation

### Deutsche Beschreibung

#### NUTZUNGSBEDINGUNGEN

Die Verwendung der Beispielprogramme erfolgt ausschließlich unter Anerkennung folgender Bedingungen durch den Benutzer:

INSEVIS bietet kostenlose Beispielprogramme für die optimale Nutzung der S7-Programmierung und zur Zeitersparnis bei der Programmerstellung. Für direkte, indirekte oder Folgeschäden des Gebrauchs dieser Software schließt INSEVIS jegliche Gewährleistung genauso aus, wie die Haftung für alle Schäden, die aus der Weitergabe der die Beispielinformationen beinhaltenden Software resultieren.

#### BEISPIELBESCHREIBUNG

##### Inhalt

Dieses Beispiel wendet das allgemeine Modbus-TCP-Beispiel an für ein Interface der Fa. Weigel.

##### Das Modbus-TCP-Interface des WEZ

Das Modbus-TCP Interface WEZ bildet alle Prozessdaten in Registern ab, die sowohl mit function code 3 als auch 4 gelesen werden können. Die Statusbits werden als coils abgebildet und werden über function code 1 gelesen. Ein Steuerregister kann über function code 10 geschrieben werden.

##### S7-Programm

FB1 dient als Modbus-TCP client Treiberbaustein und nutzt die Systembausteine zum Senden und Empfangen über TCP/IP und bleibt **unverändert**.

Als Parameter werden VerbindungsID-Nummer, Knotennummer (UID), Modbus- Kommando (function code 1, 2, 3, 4, 6, 15 oder 16) und Nutzdatenpointer übergeben.

```
CALL  "ModbusTCP_Client" , "tcp"      // FB1 mit DB1 als Instanz-DB
R      :=FALSE                       // Reset-Eingang, einmalig nach Hochlauf setzen
ConnectID:=1                          // VerbindungsID-Nummer aus ConfigStage
UID     :=                             // obsolete Knotennummer aus ModbusRTU
CMD     :=B#16#2                       // Modbus Kommando (funktion code 1,2,3,4,6,15,16)
Index   :=0                            // Register bzw. Bit-Adresse (0...65535)
LEN     :=256                          // Anzahl zu übertragender Register (1..125) bzw. Bits (1..2000)
DATA    :="Daten_TCP".Inputs          // ANY-Pointer auf Nutzdatenbereich
DONE    :="tcp1_done"                  // TRUE wenn erfolgreich
ERROR   :="tcp1_error"                 // TRUE bei Fehler
ErrSrc  :="tcp1_ErrorSrc"              // Fehlercode s. Tabelle
ErrStatus:="tcp1_ErrCode"
```

Der Aufruf muss wiederholt werden, bis DONE oder ERROR zurückgegeben wird.

Die Längenangabe im Pointer DATA wird zum Kopieren der Nutzdaten benutzt und muss zu den Datenpaketen passen (Beim Senden wird sonst ggf. der Sendepuffer unvollständig gefüllt oder andere Werte überschrieben. Beim Empfang werden ggf. andere Nutzdaten überschrieben)

Alle lokalen Variablen des Kommunikationstreiberbausteins FB1 sowie Sende- und Empfangsdaten liegen in dem zugehörigen Instanzdatenbaustein.

FC1 stellt die Modbus-Kundenapplikation dar und **wurde** wie folgt **angepasst**:

Da die relevanten Prozessdaten verstreut liegen, erfolgen mehrere Kommunikationszyklen auf Datengruppen. Im NW3 werden 6 Register mit den Spannungswerten L1-L3 gelesen, im NW4 6 Register mit den Stromwerten L1-L3 und in NW5 die Leistungsfaktoren. NW6 liest die Frequenz.

In NW7 werden die Statusflags über function code 1 gelesen.

#### RÜCKMELDUNGEN

Möchten Sie Erweiterungswünsche oder Fehler zu diesen Beispielen melden oder wollen Sie anderen eigene Beispielprogramme kostenlos zur Verfügung stellen? **Bitte informieren Sie uns unter [info@insevis.de](mailto:info@insevis.de)**  
Gern werden Ihre Programme -auf Wunsch mit Benennung des Autors- allen INSEVIS- Kunden zur Verfügung gestellt.

## English description

### TERMS OF USE

The use of this sample programs is allowed only under acceptance of following conditions by the user:

The present software which is for guidance only aims at providing customers with sampling information regarding their S7-programs in order to save time. As a result, INSEVIS shall not be held liable for any direct, indirect or consequential damages respect to any claims arising from the content of such software and/or the use made by customers of this sampling information contained herein in connection with their own programs.

### SAMPLE DESCRIPTION

#### Abstract

This example transfers the generic Modbus TCP client demo to an interface "WEZ" from Weigel.

#### The Modbus-TCP-Interface of WEZ

The Modbus-TCP Interface WEZ maps all process data into register. They are to read via function code 3 as code 4. Some statusbits are as coils to read via function code 1. A control register could be written by function code 10.

#### S7-program

FB1 works as Modbus-TCP client driver and handels the systemcalls for send und receive via TCP/IP and is **no to change**.

As parameter it uses connection ID-number, node-number (UID), Modbus-command (function code 1, 2, 3, 4, 6, 15 oder 16) und payload data pointer.

```
CALL  "ModbusTCP_Client" , "tcp"      // FB1 + DB1 as Instance-DB
R      :=FALSE                       // Reset-input, to set once while startup
ConnectID:=1                          // connection ID-number from ConfigStage
UID     :=                             // obsolete node number from ModbusRTU
CMD     :=B#16#2                       // Modbus command (funktion code 1,2,3,4,6,15,16)
Index   :=0                            // Register rsp. Bit-address (0...65535)
LEN     :=256                          // number of register (1..125) or bits (1..2000) to transfer
DATA    :="Daten_TCP".Inputs          // ANY-Pointer payload dataa area
DONE    :="tcp1_done"                 // TRUE if well done
ERROR   :="tcp1_error"                // TRUE in case of trouble
ErrSrc  :="tcp1_ErrorSrc"             // errorcode s. table
ErrStatus:="tcp1_ErrCode"
```

The call of FB1 must repeated until DONE or ERROR are returned.

The length information of Pointer DATA will be used to copy data and must match to the used data packets. (Otherwise sending the buffer could be filled incomplete or other data are overwritten.)

All local data are kept in the instance data block

FC1 represents the Modbus customes applikation and **was adapted as follows:**

Due to relevant process data are scattered, several communication cycles handles groups of data.

NW3 reads 6 register with voltage L1-L3, NW4 reads 6 register with currents L1-L3, NW5 reads the power factors and NW6 the freuquency.

NW7 reads statusflags via function code 1.

### FEEDBACK

Do you want to inform us about necessary increments or errors or do you want to provide us with your sample programs to offer it for free to all customers? **Please inform us at [info@insevis.de](mailto:info@insevis.de)**

Gladly we would provide your program -if you wish with the authors name- to all other customers of INSEVIS.