

## 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

##### Modbus-Interface

Das Betriebssystem stellt Sende- und Empfangsfunktionen zur Modbus-RTU-Kommunikation bereit. Diese beinhalten die Rahmensynchronisation nach Modbus-Spezifikation einschließlich Checksummenberechnung. Als Interface zur Applikation dient FB2: Als Parameter werden Geräteadresse (UID), Modbus-Funktionscode (Cmd), Register- bzw. Bit-Adresse (Index), Anzahl der Register bzw. Bits (LEN) und Nutzdatenpointer übergeben. Daraus werden in FB2 die Telegramme erstellt, gesendet, empfangen und die Nettodaten kopiert. Für Modbus-Header und Checksumme besteht ein eigener Puffer im Instanzdatenbaustein. Unterstützt werden die Modbus-Kommandos 1 – 6, 15 (0f hex) und 16 (10 hex). Es erfolgt ebenfalls eine Timeout-Überwachung. Erfolgt keine Antwort innerhalb 1 Sekunde, wird der Empfang abgebrochen und das Errorflag gesetzt. Werden CRC-Fehler erkannt oder wird ein RTU-Request vom Server zurückgewiesen, wird ebenfalls das Errorflag gesetzt, der zurückgelesene Funktionscode wird überprüft, weitere Protokollfehler werden nicht beachtet.

##### STWA4MH

Das Beispielprogramm realisiert das Lesen von 2 Wandlern mit je 2 Gruppen von Modbus-Registern, die wahlweise zu- und abschaltbar sind sowie die Konfiguration der Modbus-Kommunikationsparameter.

Da keine Schalter für Geräte-IDs vorhanden sind, starten alle Wandler für 60 s mit der zusätzlichen ID 247.

a) In dieser ersten Minute darf keine Kommunikation mit ID 247 erfolgen ODER

b) Es wird extern sichergestellt, dass nur EIN Wandler mit dem System verbunden ist.

Innerhalb dieser 60 s muss FB1 mit passenden Parametern zyklisch aufgerufen werden, bis das Requestflag sich selbst zurückgesetzt hat.

##### Implementierung

Im OB1 werden per festem Parameter die Modbus-Slaveadressen, Transformer-Ratios, Kommunikationsparameter (Baudrate etc) sowie die Registergruppen zum Lesen festgelegt.

Nach Bedarf anpassen:

SlaveAddr_...	[1..246]		
TransfRatio_...	[1..10]		
BaudID	0: 4800	1: 9600 (default)	2: 19200
	3: 57600	4: 115200	
Parity	0: no	1: even (default)	2: odd
Stoppbits	[1 (default), 2]		

Da der Instanzdatenbaustein die Zustandsmerker enthält und diese null-initialisiert sein sollten, ist dieser auf „non-retain“ gesetzt und eignet sich daher nicht, die Parameter zu speichern.  
(Alternativ könnten diese in OB100 gelöscht werden)

Im OB1 wird per Timer die Kommunikation verzögert.

a) um nicht unnötig Ressourcen zu belegen und b) um das debuggen der statemachine beobachtbar zu machen. Der Timeout kann angepasst werden.

### Inbetriebnahme

Im Werkszustand sind die Wandler auf 9600/8E1 konfiguriert (→ passendes ConfigStage-Projekt laden!).  
→ Die Baudrate ist aber remanent, d.h. die eingestellt Baudrate muss man sich merken, um später wieder "dranzukommen" !

- 1) S7-Programm und ConfigStage-Projekt laden (RS485 auf Modbus, 9600/8E1)
- 2) „VAT\_CFG\_Control“ öffnen und Konfigurationsdaten im DB1 einsehen:
  - Slave-Adressen und Transformer-Ratio
  - Baud / Parität / Stoppbits
- 3) EINEN Wandler anschließen und innerhalb von 60 Sekunden das Konfigurations-Requestflag für den entsprechenden Wandler setzen:
  - DB1.DBX 18.0 "STWA4\_Instance".Set\_Cfg\_A oder
  - DB1.DBX 18.1 "STWA4\_Instance".Set\_Cfg\_B
- 4) Bei erfolgreicher Konfiguration wird das Requestflag binnen weniger Kommunikationszyklen zurückgesetzt.

Beim Wechsel der Baudrate wird diese vom Knoten erst mit dem nächsten Powercycle übernommen.

### Fehlercodes

Die Rückgabewerte des FB2 sind in eine Fehlerquelle (ErrSrc) und einen StatusCode (ErrStatus) aufgeteilt und werden bis in OB1-Ebene hochgereicht. Die Fehlerflags teilen sich alle Funktionen. ErrSrc entspricht dem state der statemachine in FB2, in dem der Fehler aufgetreten ist. Davon abhängig sind die jeweiligen Fehlercodes:

ErrSrc	ErrStatus	Bedeutung
0	8001 <sub>hex</sub>	RS485 nicht in Betriebsart ModbusRTU
	sonst Rückgabewerte des SFB 61 bei Initialisierung	
1	8001 <sub>hex</sub>	UID > 247
	8002 <sub>hex</sub>	Ungültiges Modbus-CMD (function code)
	8003 <sub>hex</sub>	Ungültige Längenangabe (Register > 250, Bits/Coils > 2000)
	8004 <sub>hex</sub>	ungültiger Bereich Nutzdaten (≠ E, A, M, DB)
2	8000 <sub>hex</sub>	SFB 60 belegt (Sendepuffer overflow)
	Rückgabewert SFB20 oder SFB 60 (Senden)	
3	Rückgabewert SFB 61 bei Empfang	
	CAFE <sub>hex</sub>	Timeout
4	9001 <sub>hex</sub>	Bit 7 im Empfangswert des Funktionscodes gesetzt = Server weist Anfrage zurück (ungültige Parameter)
	9000 <sub>hex</sub>	sonstiger Fehler im Empfangswert des Funktionscodes
	Rückgabewert des SFC 20 beim Kopieren der Nutzdaten	

### Modbus RTU Observer:

Falls kein PC mit einer RS485-Schnittstelle verfügbar ist, kann im Problemfall eine weitere SPS als Protokollbeobachter eingesetzt werden. Alle empfangenen Bytes werden mit einem Zeitstempel in einen Ringpuffer geschrieben. (Das Zeitraster ist mit 10 ms zu grob, um einzelne Telegramme zu trennen, aber die Anfragen des Client können zeitlich beliebig verzögert werden. Die Hardwarekonfiguration über ConfigStage muss als plain ASCII erfolgen.

### 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

The operating system supports Modbus-RTU communication by low level send- and receive functions. These handles frame synchronisation and checksum calculation regarding Modbus specification.

The S7 code FB2 works as application interface: Parameters are Node address (UID), Modbus-function code (Cmd), register or bit address (Index), register or bit count (LEN) and user data pointer. FB2 builds the telegram, sends and receives and copies data into specified user area. Buffer for Modbus header and checksum is in FB2's instance datablock. Supported modbus commands are 1 – 6, 15 (0F hex) and 16 (10 hex). A timeout of 1 second cancels receive and signals by errorflag. In case of detected CRC-errors or denied RTU-requests the errorflag is set too. The received function code is compared to the sended code, further invalid protocoll data are not handled.

### STWA4MH

The example program implements the reading of 2 current transformers with 2 groups of Modbus registers each, which can be optionally used as well as the configuration of the Modbus communication parameters. Since there are no switches for device IDs, all converters starts for 60 s with the additional ID 247.

- a) No communication with ID 247 is allowed in this first minute OR
- b) It is ensured externally that only ONE converter is connected to the system.

Within these 60 s FB1 must be called cyclically with suitable parameters until the request flag has reset itself.

### Implementation

In OB1, the Modbus slave addresses, transformer ratios, communication parameters (baud rate, etc.) and the register groups for reading are defined by fixed parameters.

Adapt as required:

SlaveAddr_...	[1..246]		
TransfRatio_...	[1..10]		
BaudID	0: 4800	1: 9600 (default)	2:19200
	3:57600	4:115200	
Parity	0: no	1: even (default)	2: odd
Stopbits	[1 (default), 2]		

Since the instance data block contains the states of the statemachine and these should be zero initialized, it is set to "non-retain" and is therefore not suitable for saving the parameters.

(Alternatively these could be initialized in OB100)

In OB1 the communication is delayed by a timer.

- a) to avoid using unnecessary resources and b) to make the debugging of the statemachine observable.

The timeout can be adjusted.

### Bring up

In the factory setting, the converters are configured to 9600/8E1 (→ load suitable ConfigStage project!).  
 → However, the baud rate is remanent, i.e. the set baud rate must be remembered, to "get to it" again later!

- 1) Load S7 program and ConfigStage project (RS485 to Modbus, 9600/8E1)
- 2) Open „VAT\_CFG\_Control“ an supervise data in DB1:
  - slave addresses and ratio values
  - Baud / Parität / Stopbits
- 3) Connect ONE transformer and within 60 seconds set the configuration request flag for the corresponding converter in DB1.
 

DB1.DBX	18.0	"STWA4_Instance".Set_Cfg_A
DB1.DBX	18.1	"STWA4_Instance".Set_Cfg_B

If the baud rate is changed, it will be taken over by the node with the next power cycle.

### Errorcodes

Return values of FB2 are divided into error source (ErrSrc) and a status code (ErrStatus).  
 Error source accords with the last state of the state machine in FB2, as the error occurred. Related to the error source the status code contains information about the cause of error:

ErrSrc	ErrStatus	description
0	8001 <sub>hex</sub>	RS485 not configured as ModbusRTU
		else Status return of SFB 61 while initialization
1	8001 <sub>hex</sub>	UID > 127
	8002 <sub>hex</sub>	Invalid CMD (function code)
	8003 <sub>hex</sub>	Invalid LEN (register > 250, bits/coils > 2000)
	8004 <sub>hex</sub>	Invalid area of payload data (≠ I, O, M, DB)
2	8000 <sub>hex</sub>	SFB 60 busy (send buffer overflow)
		else Status return SFB20 or SFB 60
3		Status return SFB 61 while receive
	CAFE <sub>hex</sub>	Timeout
4	9001 <sub>hex</sub>	Bit 7 of returned function code set = Server denies request (invalide parameter)
	9000 <sub>hex</sub>	Returned function code invalid
		Status return of SFC 20 copying user data

### Modbus RTU Observer:

For troubleshooting and in case no PC with a RS485-UART interface is available, another PLC can be used to observe running communication. All received bytes are stored with a timestamp into a ringbuffer. The time resolution of 10ms is too coarse to detect single frames but the requests by the client are apart enough  
 Notice that the hardware must be be configured by ConfigStage as plain ASCII

### 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.