

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

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, Modbus-Kommando, Register-Index, Anzahl und Nutzdatenpointer übergeben. Daraus werden in FB2 die Telegramme erstellt, gesendet, empfangen und die Nettdaten kopiert. Für Modbus-Header und Checksumme besteht ein eigener Puffer im Instanzdatenbaustein. Unterstützt werden die Modbus-Kommandos 3, 4, 6 und 10 (hex). Es erfolgt ebenfalls eine Timeout-Überwachung. Erfolgt keine Antwort innerhalb 1 Sekunde, wird der Empfang abgebrochen und ein Errorflag gesetzt. Wird ein CRC-Fehler erkannt oder wird ein RTU-Request vom Server zurückgewiesen, wird ebenfalls das Errorflag gesetzt, weitere Protokollfehler werden nicht beachtet.

FC2 realisiert als Beispiel eine Modbus-RTU-Client Anwendung. Zyklisch werden verschiedene Kommunikationen durchgeführt.

- Über Modbus- Kommando '4', "Read Multiple Input Registers" werden 2 Blöcke a 96 und 16 Worte gelesen.
- wenn das Write-Flag gesetzt wird, erfolgt über Modbus-Kommando '10 hex', "Write Multiple Holding Registers" ein einmaliger Schreibzugriff.
- Über Modbus- Kommando '3', "Read Multiple Holding Registers" werden die geschriebenen Werte zurücklesen

Drei Bits dienen der Kontrolle: ACT initialisiert und startet einen Kommunikationszyklus, DONE wird gesetzt, wenn die Kommunikation erfolgreich abgeschlossen ist. Je nach Anwendungsfall sollte bei einem Fehler die Kommunikation neu gestartet, fortgesetzt oder wiederholt werden. Eine Signalisierung erfolgt über das OK-Flag und den Error-Status-Merker.

Als Anwendung steuert ein Bit den Schreibvorgang.

Modbus RTU Server:

Als Gegenpart wird ein (oder 2) Server konfiguriert, der nur einen Datenblock mit Dummydaten beinhaltet. ConfigStage-Projekte können als Konfigurationabeispiel geladen werden.

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 10ms 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**
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 nad checksum calculation regarding Modbus specification.

The S7 code FB2 works as application interface: Parameters are Node address (UID), Modbus-command, register index, register count and load data pointer. FB2 builds the telegramm, 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 3, 4, 6 and 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. Further invalid protocoll data are not handled.

FC2 is an example of Modbus RTU client application. Cyclically it executes different communications:

- read 2 blocks of 96 and 16 words by command '4', "Read Multiple Input Register"
- writes by command '10 hex', "Write Multiple Holding Register" if requested by application flag
- reads back written registers by command '3', "Read Multiple Holding Registers"

3 bits are used for control: ACT initializes and activates the communication, DONE will be set after communication finished sucessfully. In case of communication error the communication cycle shouldl be application depended restarted, retried or continued. The state of communication is signaled to the user by a OK flag (set when communication cycle completed successful) and the stored last failed state number.

The application enables the write operation by a control flag.

Modbus RTU Server:

As communication partner a server is configurated just containing dummy data.

Use ConfigStage-projects as template.

Modbus RTU Observer:

For troubleshooting and in case no PC with a RS485-UART interface is available, a 4th 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

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