



Beispieldokumentation Sample documentation

**Modbus TCP Client-Server-Demo:
SPS als Modbus-Daten-Relais**

**Modbus TCP Client-Server-Demo:
PLC as Modbus-Data-Relay**

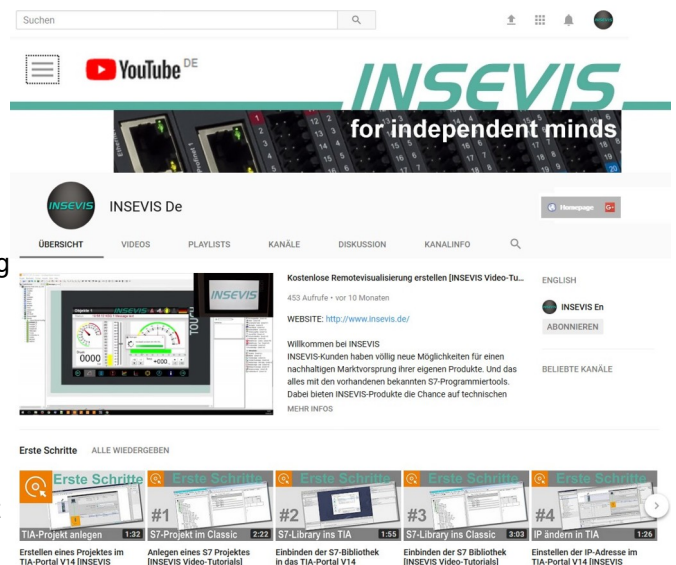
Hinweis zum besseren Verständnis durch Zusatzinformationen

Im deutschen INSEVIS-YouTube-Kanal INSEVIS DE stehen mehrere Playlists mit **Hantierungsvideos** für einzelne Details zur Verfügung.

Ebenfalls stehen **Handbücher** für die einzelnen Produktgruppen im Downloadbereich der Webseite insevis.de zur Verfügung

Bitte nutzen Sie diese Informationsquellen in Ergänzung zur vorliegenden Dokumentation. So können Sie sich noch leichter mit den INSEVIS-Funktionen vertraut machen.

Möchten Sie Erweiterungswünsche oder Fehler zu diesen Beispielen melden oder wollen Sie anderen eigene Beispielprogramme kostenlos zur Verfügung stellen? Gern werden Ihre Programme -auf Wunsch mit Benennung des Autors- allen INSEVIS- Kunden zur Verfügung gestellt.



Hinweis zu den verschiedenen Versionen der Beispielprogramme

Im Lieferumfang der Beispielprogramme können sich auch ältere Ausgabestände bzw. Versionen befinden. Diese wurden nicht aktualisiert und auf die neueste Siemens-Programmiersoftware angepasst, um einen Zugriff mit älteren Programmiersystemen weiterhin zu ermöglichen. Generell werden INSEVIS-Beispielprogramme immer mit dem aktuell neuesten Siemens-Programmierertools erstellt.

BEISPIELBESCHREIBUNG

Dieses Beispiel realisiert gleichzeitig einen Modbus-TCP-Client und -Server in einer INSEVIS-SPS. Hintergrund ist das Erfassen und vorverarbeiten von Daten mehrerer Modbus-Endgeräte (Server) und Weiterleiten an eine übergeordnete Leitebene.

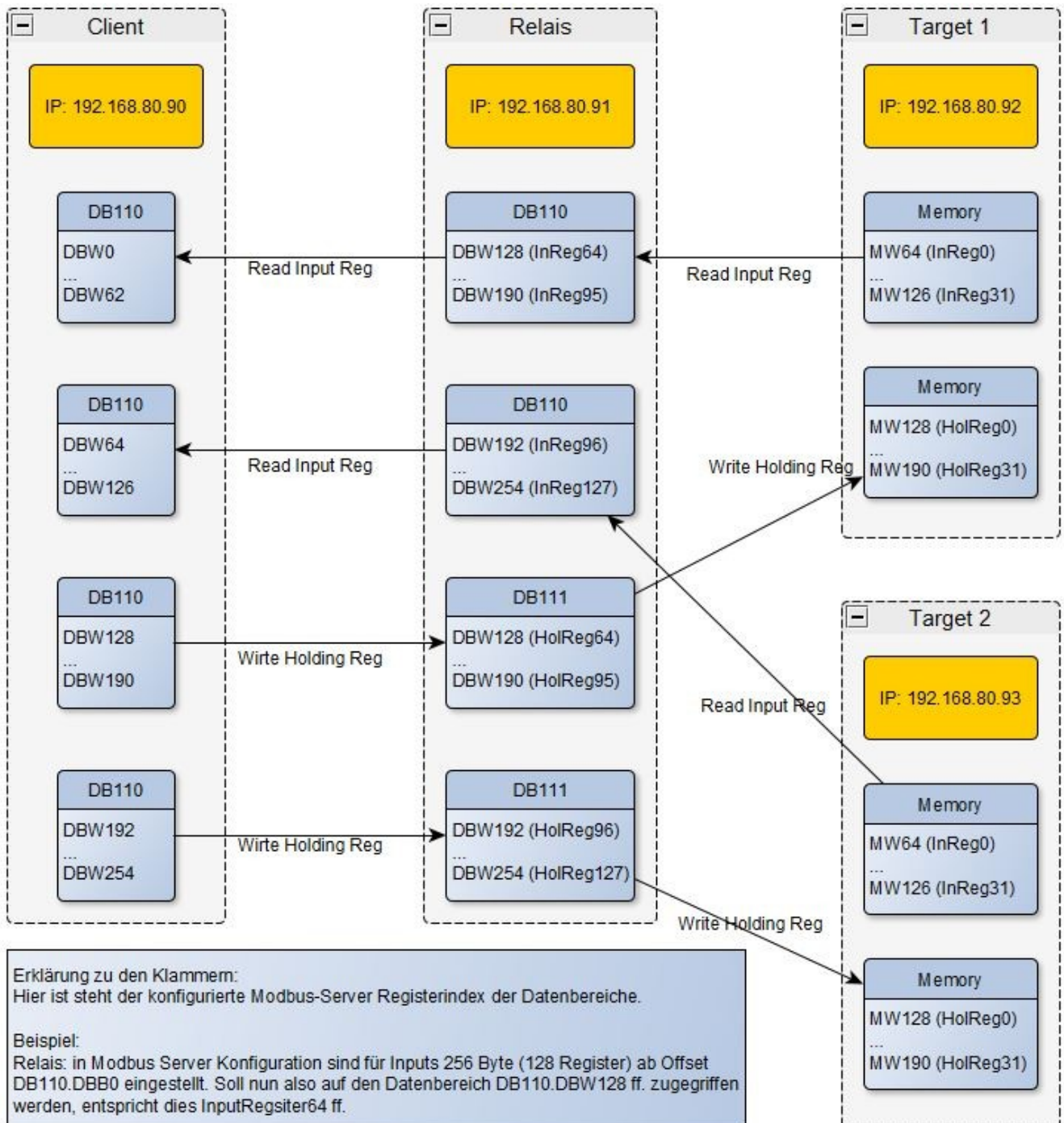
Das Projekt beinhaltet 4 SPSen: 2 Endgeräte (Target_1 und Target_2) als Server, eine Relaisstation (Server und Client zu den 2 Endgeräten) und ein Client zur Darstellung der Leitebene.

Vorgehensweise

- S7 Software in alle 4 SPSen laden
- Konfigurationsprojekte via ConfigStage in alle 4 SPSen laden und SPSen starten
- Beobachten/Steuern:
 1. erstes und letztes InputRegister Target: das erste und letzte InputRegister beider Targets werden zyklisch erhöht. Dies kann in allen SPSen beobachtet werden.
 2. sonstige InputRegister Target: werden sonstige InputRegister beider Targets verändert, ist auch diese Änderung in allen SPSen nachvollziehbar.
 3. HoldingRegister: werden HoldingRegister für Target 1 in der Client-SPS verändert, müssen diese Änderungen auch in der Relais-SPS und Target 1 nachvollziehbar sein. Werden HoldingRegister für Target 2 in der Client-SPS verändert, müssen diese Änderungen auch in der Relais-SPS und Target 2 nachvollziehbar sein.

S7 Programm

Datenstruktur:



Endgeräte (Target 1, Target 2) als Modbus Server:

Modbus-Input Register Index 0..31 => MW 64..126

Modbus-Holding Register Index 0..31 => MW 128..190

Funktion: OB1 inkrementiert 1. und letztes InputRegister (Index 0,31 bzw MW64, MW126) (Dummyapplikation)

Relaisgerät

als Modbus Client:

erfasst die Daten der beiden Endgeräte und bildet diese in einen einzigen Datenbereich ab:

Modbus Client liest 32 InputRegister (Worte) von Target 1 ab Index 0 nach DB110.DBW128

Modbus Client liest 32 InputRegister (Worte) von Target 2 ab Index 0 nach DB110.DBW192

Modbus Client schreibt 32 HoldingRegister (Worte) nach Target 1 ab Index 0 von DB111.DBW128

Modbus Client schreibt 32 HoldingRegister (Worte) nach Target 2 ab Index 0 von DB111.DBW192

als Modbus Server:

exportiert DB110 als Input Register und DB111 als Holding Register:

Modbus-Input Register DB110 DBW 0..254 (256 Bytes = 4x32 Worte)

Modbus-Holding Register DB111 DBW 0..254

Damit erscheinen „von außen“:

Modbus-Input Register Target 1 = Modbus-Input Register Relais Index 64.. 95

Modbus-Input Register Target 2 = Modbus-Input Register Relais Index 96..127

Modbus-Holding Register Target 1 = Modbus-Holding Register Relais Index 64.. 95

Modbus-Holding Register Target 2 = Modbus-Holding Register Relais Index 96..127

Client als Leitebene

- Zugriff auf Target1 über Index 64.. 95

- Zugriff auf Target2 über Index 96..127

abgebildet in DB110:

32 InputRegister (Worte) von Target 1 nach DB110.DBW 0.. 62

32 InputRegister (Worte) von Target 2 nach DB110.DBW 64..126

32 HoldingRegister (Worte) für Target 1 in DB110.DBW128..190

32 HoldingRegister (Worte) für Target 2 in DB110.DBW192..254

Visualisierung

nicht im Projekt enthalten

Konfiguration

Der Modbus-Server verarbeitet eingehende Verbindungen fest auf Port 502. Deshalb ist im Client der Zielport der TCP-Verbindung („Partner“) mit der IP des Servers und Port 502 zu konfigurieren. Die lokale Portnummer ist weitgehend frei aber darf nicht mit bestehenden Verbindungen kollidieren. (z.B. Port 502 für eigenen Server).

Target 1 und Target 2: ('Target_1.csproj', 'Target_2.csproj')

Für beide Targets ist je 1 Verbindung vom Typ Modbus TCP Server (Server für Relais-SPS) anzulegen. In den Projekten enthalten ist die Konfiguration des Modbus TCP Servers (für beide Targets gleich):

Relais: ('Relay_ClientServer.csproj')

Für die Relais-SPS sind 2 Verbindungen vom Typ TCP Send/Receive (Client zu Targets 1 und 2) anzulegen. Zusätzlich ist 1 Verbindung vom Typ Modbus TCP Server (Server für Client-SPS) anzulegen. Im den Projekt enthalten ist die Konfiguration des Modbus TCP Servers:

Client: ('Client.csproj')

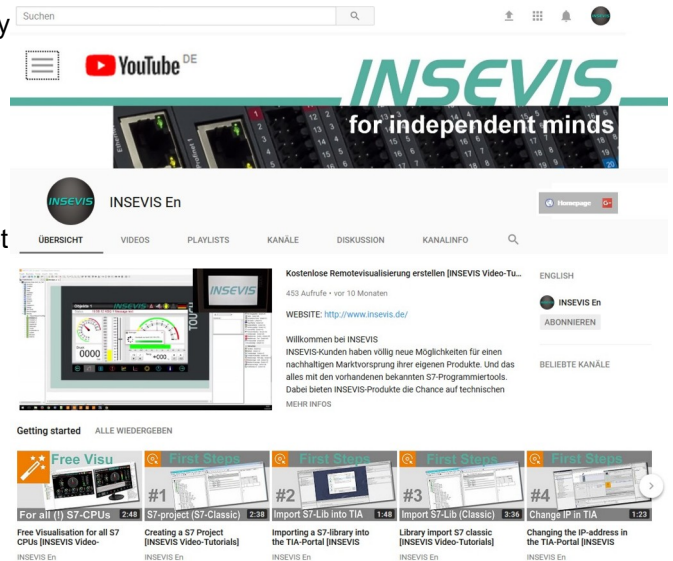
Für die Client-SPS ist 1 Verbindung vom Typ TCP Send/Receive (Client zu Relais-SPS) anzulegen.

Hint for better understanding by additional information

In the English YouTube-channel INSEVIS EN we supply different playlists with handling videos for single details. This will help you to get familiar with INSEVIS much faster.

Please download the referring manual from the download area of our English website insevis.com to get familiar with INSEVIS technology in detail.

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



Hint to different versions of the sample programs

There could be older versions in delivery scope of the sample programs too. These were not updated and converted to the newest programming tool versions to allow access by older programming tools too. INSEVIS sample programs will be created in the present newest Siemens-programming tool always.

SAMPLE DESCRIPTION

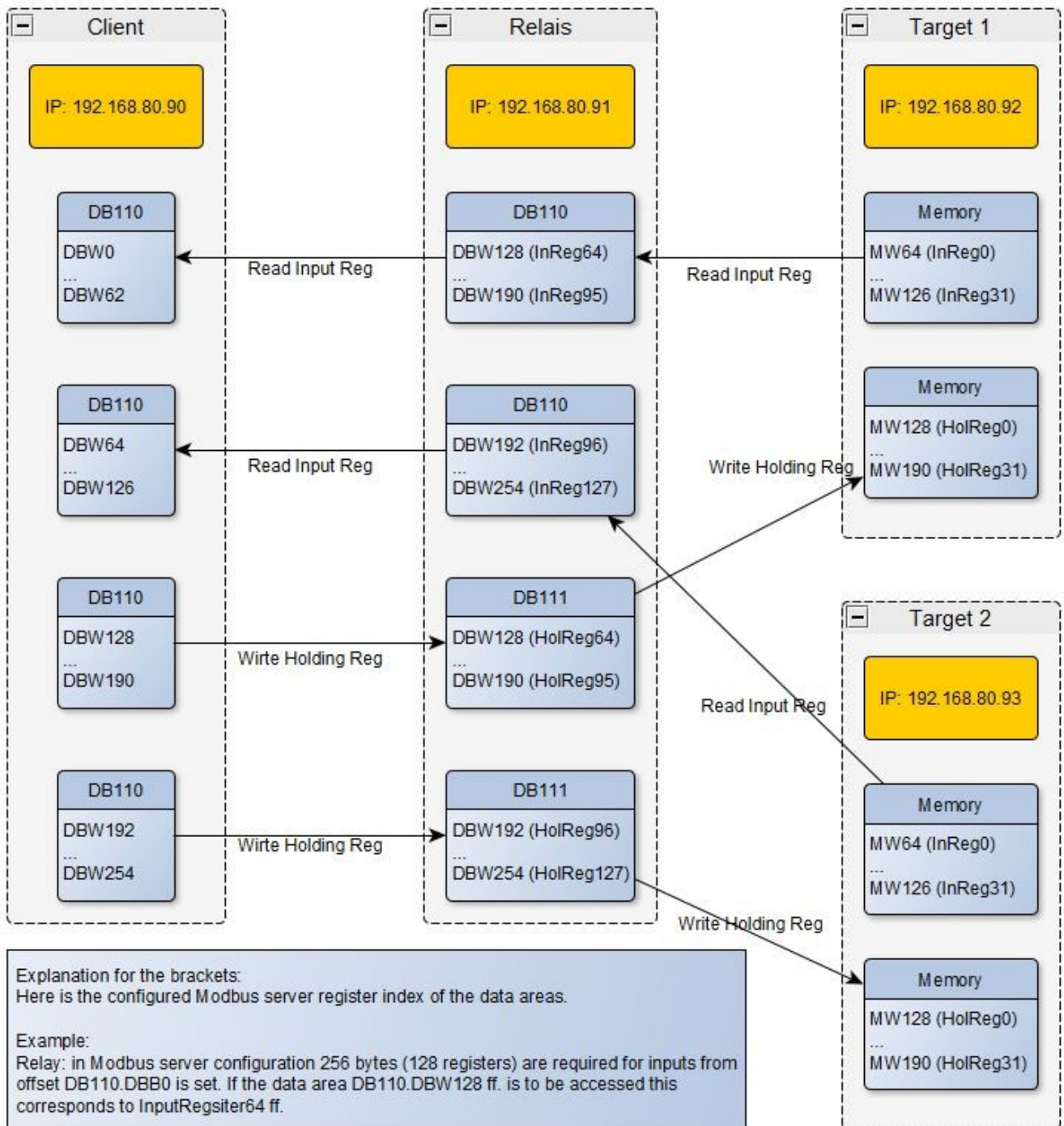
This example realizes simultaneously a Modbus TCP client and server in one INSEVIS-PLC. The target application may be collect and process data from several Modbus target devices and route this data to a higher level (control level).

Procedure

- transfer S7 software into all 4 plcs
- transfer configuration projects into all 4 plcs via ConfigStage and RUN plcs
- watch/control:
 1. first and last InputRegister of targets: the first and last InputRegister of both targets are incremented cyclically. This can be monitored in all plcs.
 2. other InputRegisters of targets: if the values of other InputRegisters of both targets are adapted, this can also be monitored in the other plcs.
 3. HoldingRegisters: if the values of HoldingRegisters for target 1 are adapted in the client-plc, this can also be monitored in the relay- and target 1 plc. If the values of HoldingRegisters for target 2 are adapted in the client-plc, this can also be monitored in the relay- and target 2 plc.

S7 program

Data structure:



Target devices (Target 1, Target 2) as Modbus server:

Modbus-input register Index 0..31 => MW 64..126

Modbus-holding register Index 0..31 => MW 128..190

Function: OB1 increments 1st und last Inputregister (Index 0,31 rsp. MW64, MW126) (Dummyapplication)

Relay device

as Modbus client:

captures data of both targets and maps it into one common data area:

Modbus client reads 32 InputRegister (words) from Target 1 starting at Index 0 into DB110.DBW128

Modbus client reads 32 InputRegister (words) from Target 2 starting at Index 0 into DB110.DBW192

Modbus client writes 32 HoldingRegister (words) to Target 1 starting at Index 0 out of DB111.DBW128

Modbus client writes 32 HoldingRegister (words) to Target 2 starting at Index 0 out of DB111.DBW192

as Modbus server:

exports DB110 as input register and DB111 as holding register:

Modbus-input register DB110 DBW 0..254 (256 bytes = 4x32 words)

Modbus-holding register DB111 DBW 0..254

Therefore data of target 1 and 2 appear in the relay server as followed :

Modbus-input register target 1 = Modbus-input register relay index 64.. 95

Modbus-input register target 2 = Modbus-input register relay index 96..127

Modbus-holding register target 1 = Modbus-holding register relay index 64.. 95

Modbus-holding register target 2 = Modbus-holding register relay index 96..127

Client device as control level:

- access to target 1 by index 64.. 95

- access to target 1 by index 96..127

mapped into DB110:

32 input register (words) from target 1 into DB110.DBW 0.. 62

32 inputRegister (words) from target 1 into DB110.DBW 64..126

32 holding register (words) into target 1 out of DB110.DBW128..190

32 holding register (words) into target 2 out of DB110.DBW192..254

Visualization

not included in project

Configuration

A Modbus server always listens to incoming connections on port 502. Hence the modbus client's partner of the TCP connection has to be configured with the IP address of the Modbus server with port 502. The local port is pretty unrestricted. But it must not conflict with existing communications (e.g. port 502 of the own Modbus server).

Target 1 and Target 2: ('Target_1.csproj', 'Target_2.csproj')

For each target create 1 connection of type Modbus TCP Server (server for relay-plc).

The projects contains the configuration of the Modbus TCP Server (equal for both targets):

Relay: ('Relay_ClientServer.csproj')

For the relay-plc create 2 connections of type TCP Send/Receive (client for targets 1 and 2).

Additional create 1 connection of type Modbus TCP Server (server for client-plc).

The project contains the configuration of the Modbus TCP Server:

Client: ('Client.csproj')

For the client-plc create 1 connection of type TCP Send/Receive (client for relay-plc).

INSEVIS Vertriebs GmbH

Am Weichselgarten 7
D - 91058 Erlangen

Fon: +49(0)9131-691-440
Fax: +49(0)9131-691-444
Web: www.insevis.de
E-Mail: info@insevis.de

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 die aus der Weitergabe der die Beispielinformationen beinhaltenden Software resultieren. Mit Nutzung dieser Dokumentation werden diese Nutzungsbedingungen anerkannt.

TERMS OF USE

The use of this sample programs is allowed only under acceptance of following conditions by the user:
The present software 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.
Use of this documentation constitutes acceptance of these terms of use.