



## Beispieldokumentation Sample documentation

Überschrift / Thema  
deutsch

Überschrift / Thema  
englisch

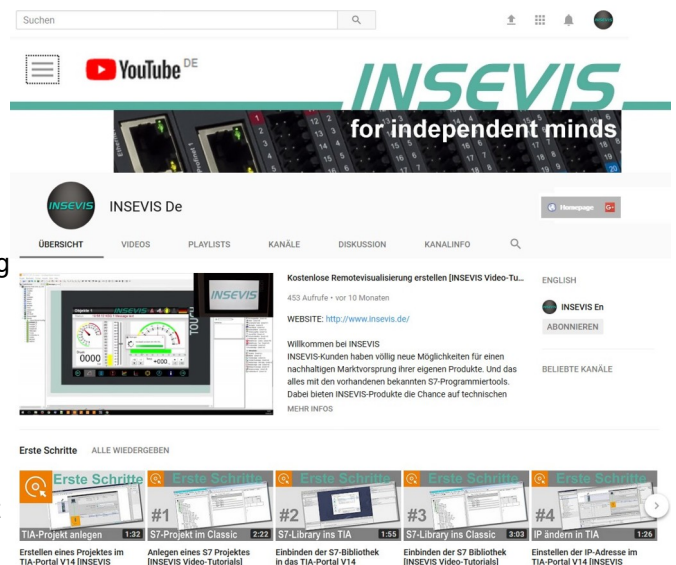
## Hinweis zum besseren Verständnis durch Zusatzinformationen

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](http://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

### Vorgehensweise:

#### BEISPIELBESCHREIBUNG CANopen Quickstart

#### Vorgehensweise bei der Inbetriebnahme eines CAN-Gerätes - Checkliste

##### Vorbereitung von SPS und CAN-Gerät:

- Verdrahtung:
  - CAN\_H ↔ CAN\_H
  - CAN\_L ↔ CAN\_L
  - CAN\_GND ↔ SPS-0V
- Knotennummer einstellen am CAN-Gerät
- Baudrate einstellen am CAN-Gerät
- CAN-Abschlusswiderstand in SPS per Schalter aktivieren  
(2 oder mehr Abschlusswiderstände beeinflussen die Signalqualität, ohne Abschlusswiderstand geht **gar nichts**)

##### Vorbereitung in der ConfigStage:

- unter CAN (Ressourcen → linke Spalte):  
Baudrate einstellen (muss mit CAN-Gerät übereinstimmen)
- unter Extra - Importiere CAN-Gerät → EDS-File des CAN-Gerätes einlesen (einmalig)  
(EDS-Datei beim Hersteller des CAN-Gerätes besorgen!)
- Automatisches Öffnen des Beschreibungsfeldes → Ausfüllen,  
danach erscheint dieses CAN-Gerät unter CAN (Katalogspalte → rechte Seite)

##### Einstellungen in der ConfigStage:

- Neu angelegtes CAN-Gerät (CAN-Node) per drag-n-drop auf die CPU ziehen, (auf die CAN-Steckerpins oder auf die Status-Felder)

- im Eigenschaftsfenster des CANopen-Gerätes nachfolgende Einstellungen vornehmen
  - Knotennummer eintragen (muss bekannt sein und mit CANopen Gerät übereinstimmen)
  - Button TxPDO / RxPDO: = Nutzdaten festlegen (= PDO-Mapping)

**Fall 1 Im EDS vordefiniertes Mapping**

Die PDO-Konfigurationsdaten sind schon ausgefüllt. Hier kann man aber muss nichts mehr einstellen.

Sind die Einträge im EDS als „read only“ gekennzeichnet, erscheinen die Werte ausgegraut und sind nicht veränderbar.

**Fall 2 EDS ohne vorgegebenes Mapping** (ohne Default-Werte)

Über "Zeige CANObjekte" werden alle verfügbare Variablen angezeigt,

die gewünschten Variablen per drag n drop auf die Mapping-Einträge ziehen

→ *Handbuch ConfigStage: „Dezentrale Fremdperipherie per EDS konfigurieren“*

**Fall 3: vordefiniertes Mapping ohne EDS** (Beispiel DS401\_example.csproj)

Einstellung der Variablentypen (Byte/Word) zur Längen- (und Byte-Swap-)-Definition, passend zum im Gerät im Gerät implementierten PDO-Mapping

→ *Handbuch ConfigStage: „Dezentrale Fremdperipherie manuell konfigurieren“*

**Fall 4: Anwendungsspezifisches Mapping ohne EDS** (Beispiel DS402\_minimum.csproj.)

Einstellung der Variablen mit Typ (Byte/Word), Index und Subindex lt. Handbuch des CAN-Gerätes

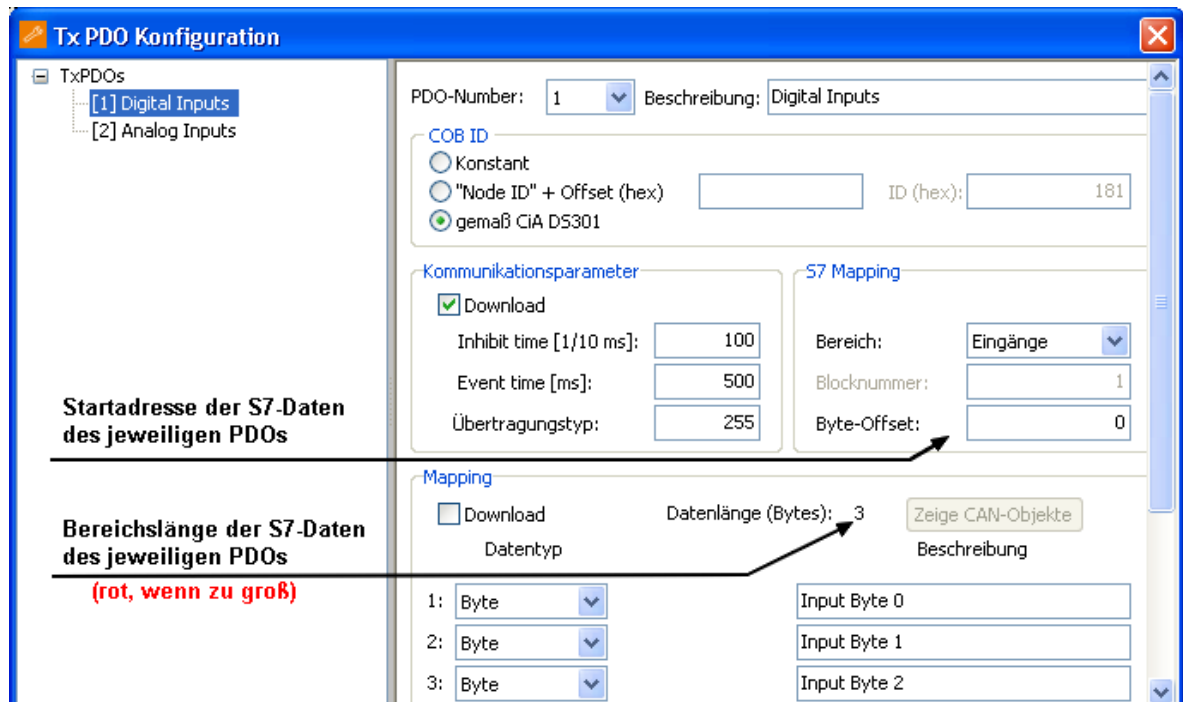
Checkbox „download“ aktivieren

→ *Handbuch ConfigStage: „Dezentrale Fremdperipherie manuell konfigurieren“*

- optional: konfiguriertes CAN-Gerät für später abspeichern,
- restliche SPS-Funktionen konfigurieren (CPU, Schnittstellen, Peripherie)
- Konfiguration ins die SPS downloaden.

**Hinweise:**

- Jede PDO enthält bis zu 8 Bytes Nutzdaten, die (fast) beliebig zusammengestellt werden können)
- Für jede PDO wird hier ein S7-Operand festgelegt, in dessen Bereich (Start-Adresse + Anzahl Bytes je nach PDO-Mapping) die CANopen-PDO-Nutzdaten abgebildet werden
- Es ist wichtig, dass die Länge der PDOs mit dem tatsächlich vorhandenen Peripherieausbau übereinstimmt. (Sonst werden u.U. die Daten wegen Längenfehler verworfen.)
- Ist das konfigurierte Mapping nicht beim Hochlauf des CANopen-Gerätes vorhanden (steht in Beschreibung CAN-Gerät), muss die Checkbox „Download“ aktiviert werden.



**Inbetriebnahme / Fehlersuche unter Step7:**

- SFB114 aus der Insevis-S7-Bibliothek im OB1-Zyklus aufrufen um eventuelle Fehler zu finden.
- zuerst die Status-Bits des SFB114 online prüfen:  
TXOFL und HW\_ERROR müssen 0 sein,  
sonst liegt ein Verdrahtungsproblem oder eine falsche Baudrate vor
- danach im Array NODESTATE das Statusbyte des Knotens prüfen:  
'5' bedeutet Operational, d.h. die PDO-Daten kommunizieren (CAN-Gerät ist im Run),
- **Erst jetzt** hat es Sinn, mit den konfigurierten S7-Operanden zu arbeiten

**Beispiel: Inbetriebnahme eines CANopen-Gerätes nach Profil DS401 „Generic I/O Modules“**

Zuerst ist das PDO-Mapping in der ConfigStage zu konfigurieren und zu laden.

**Beispielkonfiguration: „DS401\_example.csproj“ öffnen**

Das Beispiel enthält 3 Bytes digitale Eingänge (TxPDO1) und Ausgänge (RxPDO1) sowie 2 analoge Eingänge (TxPDO2) und Ausgangswörter.

Diese Prozessdaten werden automatisch in das Ein/Ausgangsabbild abgebildet und können über die Variablen-tabelle verifiziert werden.

Die Aufteilung der o.g. Prozessdaten z.B. auf verschiedene I/O-Module ist hier irrelevant.

Es ist zur CAN-Kommunikation praktisch kein S7-Programm notwendig.

Nur zur Überwachung und Fehlersuche wird SFB114 implementiert.

**Beispiel: Inbetriebnahme eines CANopen-Gerätes nach Profil DS402 „Drives and Motion Control“**

Zuerst ist das PDO-Mapping in der ConfigStage zu konfigurieren und zu laden. Das Beispiel enthält nur Control- und Statuswort und muss für eine Anwendung erweitert werden.

Die Prozessdaten (Ein/Ausgänge bzw interne Variablen des Antriebsreglers) werden automatisch in das Ein/Ausgangsabbild abgebildet und können über die Variablen-tabelle verifiziert werden.

Änderungen im Controlwort (nach Handbuch des Antriebsreglers(!)) sollten entsprechende Änderungen im Statuswort erzeugen.

Jetzt können Variablen des Antriebsreglers (z.B. Sollposition) in das PDO-Mapping aufgenommen und eine Anwendung programmiert werden.

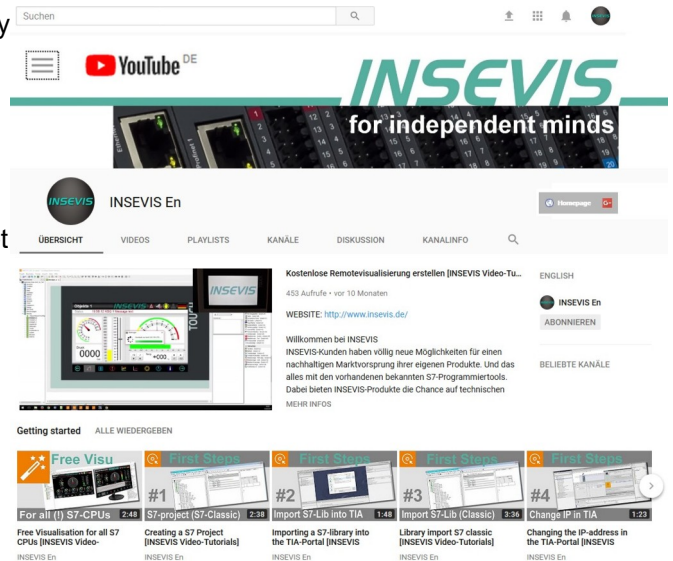
Es ist zur CAN-Kommunikation praktisch kein S7-Programm notwendig.  
Nur zur Überwachung und Fehlersuche wird SFB114 implementiert.

#### 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.

## 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](http://www.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

### Procedure:

### SAMPLE DESCRIPTION CANopen Quickstart

#### Procedure to bring up a CAN-device - Checklist

##### Preparation of PLC and CAN-device:

- Wiring:
  - CAN\_H ↔ CAN\_H
  - CAN\_L ↔ CAN\_L
  - CAN\_GND ↔ SPS-0V
- Set up Node-ID-no. at the CAN-device
- Set up Baud rate at the CAN-device
- activate CAN-terminating resistor at the PLC by switch (2 or more terminating resistor affect signal quality, without terminating resistor **nothing goes**)

##### Preparation in der ConfigStage:

- Assign baud rate at CAN (Resources column → left side) (must match with the CAN-device)
- Import EDS-file of the CAN-device at „Extra - Import CAN-Device“ (once only) (get EDS-file from the vendor of the CAN-device!)
- Property field opens automatically → Fill it out, than your CAN-device appears at CAN (Catalog column → right side)

##### Settings in the ConfigStage:

- Move the new configured CAN-device (CAN-Node) by drag-n-drop to the CPU (to the CAN-connector pins or to the state fields)
- Assign in the property window of the CANopen-device following settings
  - Set up Node-ID-no. at the CAN-device (must be known and match with the CAN-device)
  - Button TxPDO / RxPDO: = assign user data (= PDO-mapping)



**Case 1: predefined mapping in the EDS**

Mapping-values are available yet, nothing to do (but possible).

In case values in EDS-file are marked „read only“, data appear grayed-out and are not changeable.

**Case 2: EDS without defined mapping** (no default-values)

Button "Show CAN objects" will display all available variables,

Move the desired variables to the mapping inputs by drag'n drop

→ *Manual ConfigStage: „Configure decentral external peripherie by EDS-file“*

**Case 3: predefined mapping without EDS** (Sample DS401\_example.csproj)

Assigning of the variables types (Byte/Word) to Length- (and Byte-Swap-)-definition

→ *Manual ConfigStage: „Configure decentral external periphery manually“*

**Case 4: application-defined mapping without EDS** (Sample DS402\_minimum.csproj.)

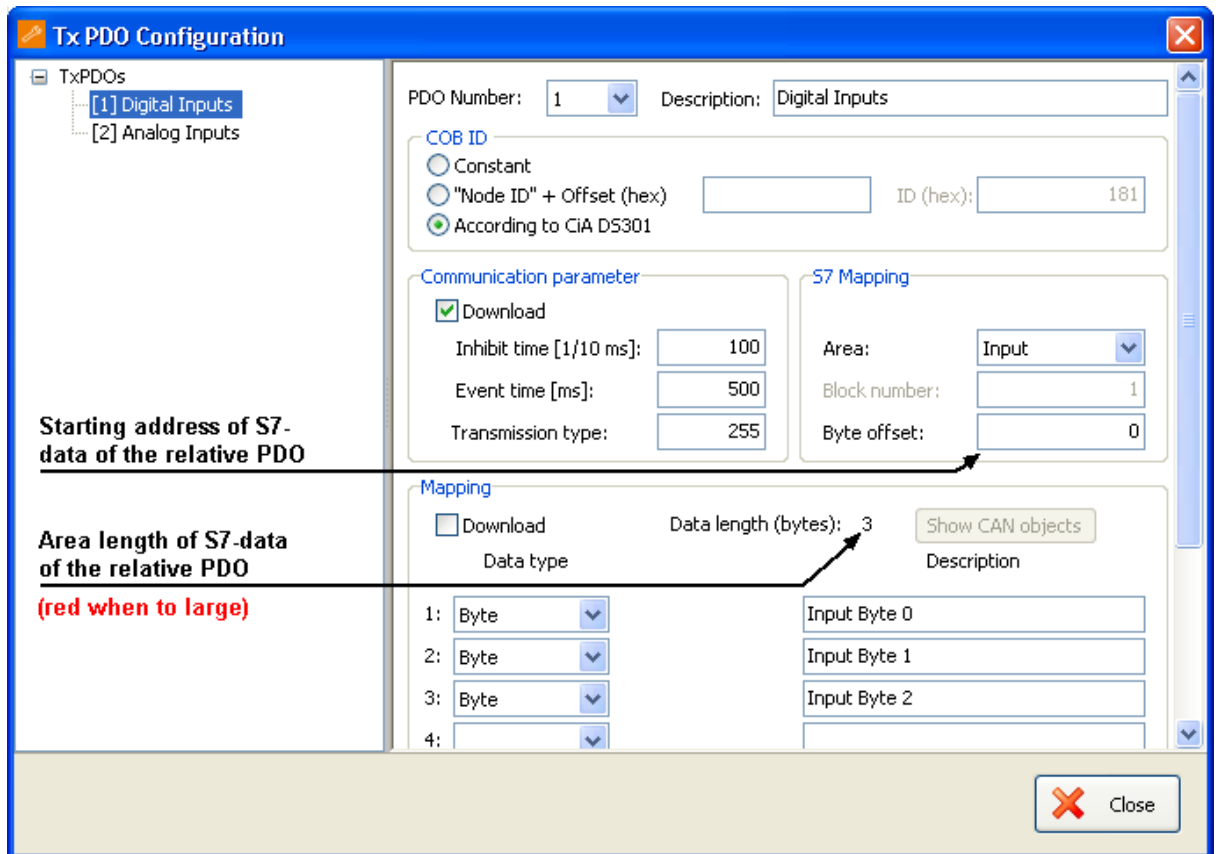
Assigning of the variables by type (Byte/Word), index and subindex regarding application needs and device' user manual,

*Don't forget to enable mapping „download“*

- Save the configured CAN-device,
- Configure the rest of the PLC-functions (CPU, interfaces, periphery)
- Download the configuration into the PLC

**Hints:**

- Each PDO contains up to 8 bytes user data, what can be composed (nearly) any way)
- For each PDO will be assigned an S7-operand, in those area (Start-address + number of bytes according to PDO-mapping) the CANopen-PDO-user data will be mapped.
- It is important, that the length of the PDOs match with the real existing periphery (Otherwise the data could be ignored because of length mismatch)
- If the configured mapping is not available at the booting time of the CAN-device (as described in the CAN-devices manual), the Checkbox „Download“ must be activated.



**Startup / debugging at Step7:**

- Call SFB114 from Insevis-S7-Bibliothek in the OB1-cycle to locate errors.
  - first check the state bits of the SFB114 online:
    - TXOFL and HW\_ERROR must be 0,
    - otherwise a wiring problem or a wrong baud rate is probably appeared
  - then check the state byte of the node in the array NODESTATE:
    - '5' means Operational, (it means the PDO-data communicate / CAN-device is in Run-condition),
- **Not before yet** it makes sense to work with the configured S7-operands

**Example: „Bring-up“ of CANopen-Devices according Profile DS401 „Generic I/O Modules“**

1<sup>st</sup> do the PDO-Mapping in the software tool „ConfigStage“. The example configuration contains 3 bytes digital I/O and 2 words analog I/O.

All process data are copied into process image and can be handled for testing by variable tables.

The way of dividing these process data e.g. to different I/O modules is unimportant here.

There is not really any S7 programm necessary to establish a communication. SFB114 is implemented just for monitoring and error detection.

**Example: „Bring-up“ of CANopen-Devices according Profile DS402 „Drives and Motion Control“**

1<sup>st</sup> do the PDO-Mapping in the software tool „ConfigStage“ and download it to PLC. The example configuration



contains only controlword and statusword, to be expanded to a „real“ application.

All process data are copied automatically into the process image and can be handled for testing by variable tables.

Manual changes of bits in the controlword (according drive manual(!)) should cause responses in the statusword.

Now drive variables e.g. target-position should be added to the PDO configuration to setup an application.

There is not really any S7 programm necessary to run the CAN communication. SFB114 is implemented just for monitoring and error detection.

#### 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.

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](http://www.insevis.de)  
E-Mail: [info@insevis.de](mailto: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.