

## 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 CANopen NMT

##### Problem 1:

Es ist möglich, dass ein CANopen-Knoten nach gemeinsamen Einschalten mit der SPS zum Booten länger benötigt als die SPS. In der Konfigurationsphase (d.h. beim SDO-Download) oder beim NMT-Start ist der Knoten noch nicht bereit und kann von der SPS nicht konfiguriert und gestartet werden.

##### Problem 2:

Wenn die Applikation ermöglicht, dass CAN-Teilnehmer während des Betriebs zeitweise entfernt werden, müssen diese nach Wiederkehr manuell in OPERATIONAL geschaltet werden, um wieder an der Kommunikation teilzunehmen. Wenn die Konfiguration des Knotens nicht remanent im Knoten abgelegt ist, ist ebenfalls eine erneute Konfiguration über SDO-Transfers erforderlich.

Dies ist nach CANopen in der Verantwortung des Applikations-Programmierers.

##### Problem 3:

Es gibt CANopen-Knoten, welche NMT-Nachrichten nur in einem bestimmten Zeitabstand verarbeiten können. Wird dieser unterschritten, werden NMT-Kommandos "überhört". Da diese nicht quittiert werden, wird der falsche Status erst nach erfolgtem Nodeguard/Heartbeat erkannt.

Die Systemfunktion "NMT-Control" (in ConfigStage zu aktivieren) sendet die NMT-Nachrichten im Abstand von 1 ms. Wird eine längere Pause benötigt, ist das NMT-Control unter S7 zu programmieren.

##### Problem 4:

Es gibt CANopen-Knoten, welche nur synchrone PDO-Kommunikation unterstützen. Der Vorteil der synchronen PDOs ist, dass sowohl SPS-Zyklus und CAN-Knoten-Zyklus sowie auch die Kommunikation synchronisiert werden. Dafür ist unter S7 zyklisch ein SYNC-Telegramm zu senden und eine minimale Zykluszeit einzuhalten. Sinnvollerweise wird eine RxPDO-Event-Time unterhalb der SYNC-Zeit eingestellt.

#### Lösungen:

Im FB1 wird zyklisch über Aufruf von SFB114 der Status gelesen. Die globalen CAN-Status-Flags werden im Instanz-DB DB1.DBW2 abgelegt und können somit global geprüft werden.

- DB1.DBX2.0 CAN-Transmitbuffer overflow
- DB1.DBX2.1 CAN-Receivebuffer overflow
- DB1.DBX2.2 CAN-Anwender-Empfangspuffer overflow
- DB1.DBX2.3 CANopen NMT-Fehler (Node-Guarding, Heartbeat)
- DB1.DBX2.4 Ein CANopen Knoten nicht OPERATIONAL
- DB1.DBX2.5 CAN-Hardware Fehler
- DB1.DBX2.6 CANopen Knoteninitialisierung über SDO fehlerhaft

Über Bit 4 und Bit 6 kann die fehlerfreie Initialisierung geprüft werden, über Bit 3 und 4 könnte global „Handlungsbedarf“ erkannt werden.

Im Instanz-DB DB1 liegt ab Byte 4 eine Liste mit den Zuständen der Knoten 1..127.

**FB11** realisiert für jeden zu überwachenden Knoten eine kleine state-machine.

- Knoten wird ignoriert (= time out abwarten)  
nach Re-Start notwendig um Antwort über Guarding bzw Heartbeat-Telegramm abzuwarten und ggf. im Anlauf
- Knoten wird auf „PREOPERATIONAL“ geprüft  
=> Fall ja: Neuinitialisierung und Neustart
- Knoten befindet sich in Initialisierungsphase (über mehrere Zyklen)

a) im Anlauf wird ein Timeout initialisiert; wird dieser im Zyklus nicht beschrieben, gilt dieser dann global; ansonsten kann hier für den Anlauf eine separate Wartezeit eingestellt werden

b) über SFB107 wird die gesamte Konfiguration (die mit der ConfigStage in Systemdaten hinterlegt wurde) nochmals geschrieben

c) Senden des NMT-Kommandos „goto OPERATIONAL“ startet den Knoten;

Hier ist eine Wartezeit > Guardingtime notwendig, da die Statusrückmeldung erst nach erfolgreichem Nodeguarding erfolgt.

**FB2:** Option für Anwendung synchroner PDOs:

Am Zyklusende wird im FB2 über einen Zeitstempel und eine Warteschleife eine minimale Zykluszeit realisiert. Danach wird das CANopen-SYNC-Telegramm gesendet.

**FB3:** Weitere DEBUG-Informationen

Alle CAN-Telegramme die dem konfiguriertem CANopen-Protokoll nicht zugeordnet werden können, sind über SFB106 auslesbar. Diese werden in einem „one-shoot“ buffer aufgezeichnet. Zusätzlich werden EMERGENCY-Telegramme ausgefiltert und in einer Liste abgelegt, die in der Visualisierung dargestellt wird. Fehlerzustände von Knoten können über deren Fehlercodes identifiziert werden.

Hinweise:

Im aktuellen Betriebssystem des INSEVIS-CPU liefert der SFC64 die Systemzeit in einer Auflösung von 10ms.

## 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 CANopen NMT

#### Problem 1:

Sometimes CANopen nodes needs more time for bootup than the PLC. In this case configuration SDOs fails or the start-NMT message get lost. The PLC goes into run with CANopen nodes which are partial initiated or in PREOPERATIONAL.

#### Problem 2:

Sometimes applications needs temporary removable CANopen nodes. After recurrence these nodes must be manually set into OPERATIONAL. Caution: If these nodes need configuration, the according SDOs must be send manually too. According CiA CANopen this is in the responsibility of the application programmer.

#### Problem 3:

Some CANopen nodes needs a time distance between NMT commands. Otherwise commands are ignored. Cause NMT commands unreceipted, the error will still be recognized after next node guarding/heartbeat. The system function "NMT Control" (to be enabled in ConfigStage) sends NMT commands with a wait of 1ms. If a longer time is needed, the NMT command must be send with S7 manually.

#### Problem 4:

Some CANopen nodes supports synchronous PDO traffic only. This procedure synchronizes PLC-cycle, CANopen-node's cycle and PDO communication. To use this, send a CANopen SYNC-message every cycle and ensure a minimal cycle time. It makes sense to configure the RxPDO's Event time shorter than the used SYNC time.

#### Solution:

A cyclical call of SFB114 in **FB1** returns the CAN status. The global CAN status flags are written into DB1.DBW2:

- DB1.DBX2.0 CAN-Transmitbuffer overflow
- DB1.DBX2.1 CAN-Receivebuffer overflow
- DB1.DBX2.2 CAN-Layer2-receivebuffer overflow
- DB1.DBX2.3 CANopen NMT-Error (Node-Guarding, Heartbeat)
- DB1.DBX2.4 any CANopen Node not OPERATIONAL
- DB1.DBX2.5 CAN-Hardware errir
- DB1.DBX2.6 CANopen node initialisation by SDO failed

It is advised to check bit 4 and bit 6 to ensure an errorfree initialisation., bit 3 and 4 can be used as global state

The instance DB contains a list of all node's (1..127) state starting at byte DBB 4.

A small state machine in **FB11** controls

- node will be ignored (wait for timeout)  
in case of a restart to wait for valid state response or  
for a delayed start-up after power on
- node will be checked for „PREOPERATIONAL“  
repeat initialisation and re-start via NMT-command "goto OPERATIONAL"
- running re-initialisation takes several cycles

a) while startup a timeout must be initialized; if not changed cyclically, this value works globally; in other way setup a separate wait time until supervising for startup here

b) SFB107 writes again the whole configuration (described in ConfigStage and stored in system data)

c) Send the NMT-command „goto OPERATIONAL“ to restart the Node;

After command „goto Operational“ a wait time > Guardingtime is necessary until a valid NMT response will be received

**FB2:** Option for synchronous PDO-traffic:

At the end (!) of the cyclic program in FB2 a timestamp and a waitloop realize a minimum cycle time. After than the CANopen Synch telegramm is sent.

**FB3:** More DEBUG-Information

All CAN-messages not according configured CANopen-protocol are stored in a dedicated „L2-buffer“ and are readable with SFB106. FB3 stores all these messages in a „one-shoot“ buffer. Additionally EMERGENCY-messages are separated and stored into a list, which is displayed in the small demo visualisation project.

Reason of fails can be localized easier by their error code

Hint:

The SFC64 of the current operating system returns system time in a resolution of 10ms.

**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.