

# CANopen Library Extensions

## Overview

Besides the standardization of communication objects CANopen specifies application objects used in various device profiles. These device profiles guarantee a defined device behavior and provide thereby interchangeability of CANopen devices.

For employment of these profiles *port* provides extension modules for its *CANopen Library*. These modules make it possible to use the desired device profiles easily.

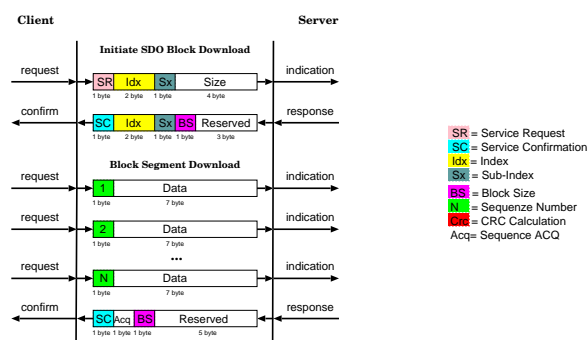
Currently *port* supports the following device profiles:

- CiA 301 SDO Block Transfer
- CiA 301A Multiplexed PDO's
- CiA 302 Flying Master
- CiA 302 CANopen Redundancy Support
- CiA 302 SDO Manager/  
SDO Requesting Devices
- CiA 303-3 LED Modul CANopen Indicator
- CiA 304 Safety-Relevant Communication
- CiA 305 LSS Layer Setting Services
- CiA 401 Generic I/O Modules
- CiA 402 Drives Support

All modules are available in ANSI-C source code and can be used with all versions of the *CANopen Library*.

## Description

### CiA 301 SDO Block Transfer



SDO transfers are based on the client-server model with a handshake after each transfer. For a larger block of data this will take a large amount of time. Therefore a new SDO mode has been defined in CiA 301 V4.x. It is called SDO block transfer. Using the block transfer a sequence of blocks can be transmitted without a large overhead. Each block is a sequence of up to 127 segments (e.g. CAN telegrams)

containing only a sequence number and the data.

### CiA 301A Multiplexed PDO's

If the application has a lot of data with the same properties a special PDO type can be used. It is called a Multiplexed PDO (MPDO). MPDOs transmit with every telegram the index and the sub-index of the given data. Therefore the maximum data length can be only 4 bytes. The transmitted index and sub-index can be the index and the sub-index of the producers object dictionary (MPDO Source Addressing Mode) or the index and the sub-index of the consumers object dictionary (MPDO Destination Addressing Mode).

### CiA 302 Flying Master



The package *CiA 302 Flying Master* is an extension to the *CANopen Library* library of *port* and provides functions for implementing the flying master functionality for CANopen devices. Flying master means that a second device in a network can take over the master role when the master dropped out.

The provided functions fulfill all demands of the standard:

- determination of master device in a network
- detection of the active master
- priority driven master negotiation
- check for multiple master

The application is informed about changes of the master state with indication functions.

**CiA 302 Redundancy Communication**



For employment of communication systems in maritime applications a redundant bus system is required. Because only the communication has to be redundant all services provided by CANopen can be used by this package. With function calls the redundant communication is handled automatically by the library. Information about other events in the different bus systems like drop out of one bus line, restoration of a bus line, missing communication and others are provided to the application with indication functions. Redundancy needs at least two CAN controllers. *port* offers implementations for stand alone CAN controllers as well as with microcontrollers with more than one CAN integrated.

**CiA 302 SDO Manager/SDO Requesting Devices**

SDO connections exist exclusively between Client and Server. For the operation of configuration and analysis software as well as HMI which are only temporarily present in a network static SDO connections are not possible. For these cases the use of dynamic SDO connections is foreseen. With the extension module *CiA 302 SDO Manager/SDO Requesting Devices* the *CANopen Library* of *port* can be expanded to use this functionality.

The module contains functions for implementing either an SDO manager or an SDO requesting device. All functions described in the standard are available:

- manage dynamic SDO connections (request, release, configure, monitor)
- manage the COB IDs of SDO
- manage the SDO connection table

The application is informed about events by means of indication functions.

Furthermore functions to request or release SDO connections are enclosed:

- register as an SDO requesting device
- request/release SDO connections
- request all default connections

The package contains source code, documentation and example programs for getting into it quickly.

**CiA 303-3 LED Module**



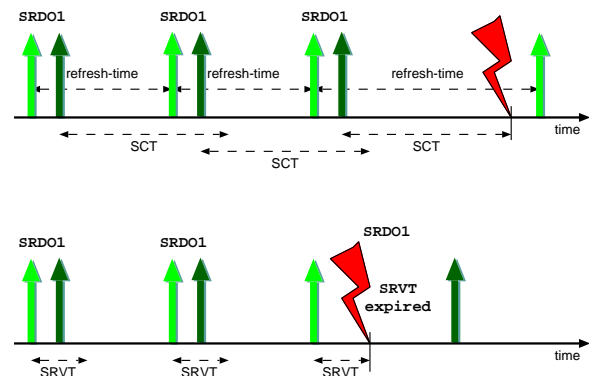
For a consistent state or error indication respective error diagnosis of CANopen devices the module *LED* is available. It is included in the default delivery of the *CANopen Library*.

According to the specification of CiA 303-3 a NMT-LED and an Error-LED can be used as two single colored LED or as one bicolored LED. The state machine for controlling the LED is realized within the module. The desired configuration can set with the configuration tool.

On state change of either NMT or error state the application is informed by an indication function which will then control the LED, i.e. switch on or off the appropriate LED.

The LED module is alltimes delivered by the library.

**CiA 304 Safety-Relevant Communication**



In order to transmit safety relevant data in a CAN network further measures have to be taken. This is done with this extension package. Communication takes place with so called SRDO i.e. Safety Relevant Data Object. These objects have communication and mapping parameter like PDO. The further measures consist of:

- sending data as plain and inverted data with different CAN identifier
- monitoring of sending data cyclically and in correct order
- protecting communication and mapping parameter with a checksum
- additional activation flag in the object directory

All monitoring functions can be realized with functions of the *CANopen Library*, but considering safety aspects monitoring functions should be implemented

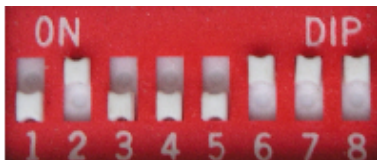
by the application developer.

Consistency of the data in the object directory is achieved through further safety measures.

- modification of communication and mapping parameter is only allowed in the state PRE-OPERATIONAL
- on modification all data lose their validity
- mapping data is also stored inverted
- a checksum calculated over all data of an SRDO is checked when changing into state OPERATIONAL

With this all planned safety measures of the standard are realizable.

### CiA 305 Layer Setting Services



The extension package *LSS* (Layer Setting Services) for the *CANopen Library* contains all functions to operate CANopen devices as LSS Master or LSS Slaves. With it unconfigured devices in a network can be identified by their unique manufacturer, product, serial and revision number and configured. After identification bit rate and node id can be configured for each device.

Configuration with LSS can only be implemented on a CANopen master. With function calls existing slaves can be identified or addressed. In indication functions the responses of slave devices can be processed by the master.

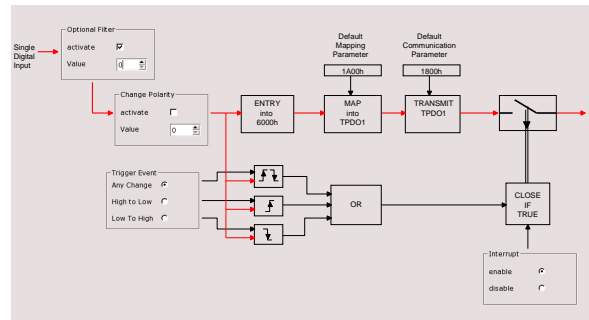
Responses of the LSS slaves to LSS requests of the master are generated automatically by the *CANopen Library*. Only when new parameters like bit rate or node id are set the application will be informed with indication functions.

Delivery comprises source code, documentation and example code to get started quickly.

This extension module for CANopen LSS functionality is included in the default delivery of the CANopen Library.

### CiA 401 generic I/O modules

The device profile for Generic I/O-Module is an extension to the *CANopen Library* of



*port* and provides functions for realizing I/O devices that conform to the device profile CiA 401. The implemented functions realize the functionality as specified in the standard, i.e. logic operations, polarity switching, filter mechanisms, limit monitoring and others. The developer has to provide functions to access the hardware, ie. reading inputs, writing outputs.

Delivery comprises source code and documentation.

### CiA 402 Drives Support

Drives Support is available from *port* in different ways. First of all using the *CANopen Design Tool* and its CiA 402 profile data base. Second, from a lot of implementations we did, we have some collected and assembled supporting functions in C combined in a CiA 402 framework. And third, our experience. Implementing drives and the CiA 402 profile is one of the most complicated programming tasks.

For the test of an drives implementation according to CiA 402 the *CANopen Device Monitor* with its drive specific plug-in is suggested. Look at the drive state machine or the different movement modes in a graphical way.

The CANopen Drive Control Library Extension provides a framework and functions to control single or multi-axis drives compliant to CiA 402. It supports the Profile Position Mode and is built on top of *port's* CANopen Library but can be used as well as with *port's* CANopen Master/Slave DLL.

The basic drive control functions can be used with the CANopen Slave Library, but full functionality requires the CANopen Master Library. In this case the CANopen Drive Control Library Extension is able to monitor the node using Node Guarding or Heartbeat.

## Ordering Information

0564/50	CiA 401 Source Code Generic I/O Modules
0564/51	CiA 401 Source Code and Profile Database for Generic I/O Modules
0564/52	CiA 302 Source Code Flying Master
0564/55	CiA 304 Source Code Safety-Relevant Communication
0564/57	CiA 307 Source Code CANopen Redundancy
0564/58	CiA 304 Source Code and Profile Database for Safety-Relevant Communication
0564/60	CiA 302 Source Code dyn. SDO Manager/ SDO Requesting Devices
0564/62	CiA 301A Source Code Multiplexed PDO's
0564/63	CiA 301 Source Code SDO Block Transfer
0564/66	CiA 402 Source Code (Template for Drives functionality)
0564/67	CiA 402 Source Code (Template for Drives functionality) and Profile Database

### Engineering Services

*port* is providing engineering services and trainings for our business activities:

- CAN and CAN-based protocols: CANopen, J1939
- Industrial Ethernet Protocols: POWERLINK, EtherNet/IP, EtherCAT PROFINET
- Implementation of devices according to CANopen device profiles
- VHDL based solutions for industrial applications
- application specific implementations or enhancements
- embedded LINUX projects

### Notice

Brands and product names are trademarks or registered trademarks of their respective companies. The product will be continuously improved. *port* therefore reserves the right to change technical properties at any time without appointment.



*port* GmbH  
Regensburger Straße 7b  
D-06132 Halle/Saale  
+49 345 777 55 0  
service@port.de