

# CANopen Library

## Overview

The *CANopen Library* provides all required services for a CANopen compliant communication according to the communication profile CiA 301 V 4.2. It facilitates easy and fast development of master and slave devices and is able to serve one or multiple CAN-Controller in one device.

The *CANopen Library* is available in different expansion stages:

- Slave
- Master/Slave

The functional range can be extended by additional modules (Redundancy, Flying Master, CANopen safety, ...) (see *CANopen-Library Extension modules*).

Access to the hardware is carried out via a defined driver interface, which is available for many CPU- and CAN controller with and without operating system (see *CANopen Driver Package*).

## Application

The *CANopen Library* is based on ground of the communication profile CiA 301 V4.2 of the CiA e.V. "CANopen Application Layer and Communication Profile" and EN50325-4, respectively, and provides all services specified therein. It is completely written in ANSI-C and can be compiled with every ANSI-C compliant compiler.

Depending on the required scope of functionality the *CANopen Library* is available in different expansion stages. For development of small sensors and actuators with limited CANopen services the

With the *Slave-Version* of the *CANopen Library* all services are provided for development of full-featured slave devices.

The functionality of the network management master as well as the comfortable node monitoring functionality is provided by the *Master/Slave-Version*, which of course contains the services of the *Slave-Version*.

Additional services of other communication profiles (CiA 3xx) are provided by means of extension modules.

All versions of the *CANopen Library* are compatible to each other and are constantly tested with the current CANopen Conformance Test for compliance with the standard.

	Slave	Master/ Slave	Extra Package
SDO Server	128	128	
SDO Client	128	128	
SDO Segmented Transfer	✓	✓	
SDO Block Transfer			✓
Dyn. SDO Slave			✓
SDO Manager			✓
Program Download	✓	✓	
PDO Consumer	512	512	
PDO Producer	512	512	
Dynamic Mapping	✓	✓	
Bitwise Mapping	✓	✓	
MPDO Source Mode			✓
MPDO Dest. Mode			✓
Nodeguarding Master		✓	
Nodeguarding Slave	✓	✓	
Lifeguarding	✓	✓	
Heartbeat Consumer	128	128	
Heartbeat Producer	✓	✓	
EMCY Consumer	128	128	
EMCY Producer	✓	✓	
Time Consumer	✓	✓	
Time Producer		✓	
SYNC Consumer	✓	✓	
SYNC Producer		✓	
NMT Slave	✓	✓	
NMT Master		✓	
NMT Flying Master			✓
Bootup Procedure		✓	
Configuration Manager		✓	
LED Indicator	✓	✓	
Safety Communication			✓
Redundancy Support			✓
LSS Services	✓	✓	
CiA 401 Framework			✓
CiA 402 Framework			✓
CiA 402 Application note			✓

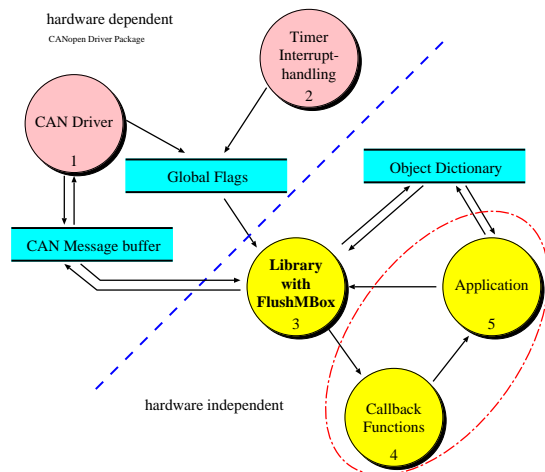
All hardware specific parts are decoupled from the CANopen protocol stack through a defined driver interface. This provides easy adaptation to different hardware platforms.

The user application communicates with the *CANopen Library* through function calls and call-back functions.

Configuration and scaling of the *CANopen Library* is done with the help of the *CANopen Design Tool*,

which is delivered as *CANopen Design Tool Light* version. With it the *CANopen Library* can be tailored to an optimum to the available resources of the application. Besides the creation of the object directory all settings for the hardware can be carried out with it.

## Description



The *CANopen Library* consists of a hardware independent and a hardware dependent part that communicate with message queues. The hardware dependent part consists of controlling software for the CAN controller and timer functions. A detailed description can be found in the *CANopen Driver Package*.

The application communicates only with the hardware independent part of the *CANopen Library*. That way it is possible to exchange drivers without any influence on the functionality of the application. The initialization of CANopen services is done with function calls within the application. During the execution of the application the *CANopen Library* executes all necessary communication tasks autonomously and informs the application about CANopen messages with the help of callback functions.

Communication requests from other devices as well as necessary periodical tasks and time out monitoring is handled within the *CANopen Library*. All requests are proved for correctness (access rights, data types etc.). The application is notified after completion of the communication and occurrence of failures, respectively, through service oriented callback functions. In these callback functions appropriate actions can be carried out from the application.

The object directory is designed to contain references to the variables in the user application. Consequently it is possible to take over variables from an existing software without any changes in the object dictionary.

The high degree of scalability of the *CANopen Library* is of particular importance for devices with limited resources. On one hand, this is achieved by the modularity in individual service groups, like *sdo.c*, *pdo.c*, *...sync.c*, and on the other hand, through the use of compiler directives in the respective modules.

Thus, the code size is proportional to the utilized CANopen services.

Furthermore, variants of the *CANopen Library* for the support of multiple CAN lines (max.255) are available. Therefore it is possible to serve several independent CAN networks on devices without an operating system or with an operating system. The usage of an operating system requires it to provide means of resource protection mechanisms. Each line holds its own object directory and can be used as master and slave, respectively, independent of the other lines. Owing to the separation of the protocol stack and the hardware drivers, the individual lines can also be operated with different CAN controllers.

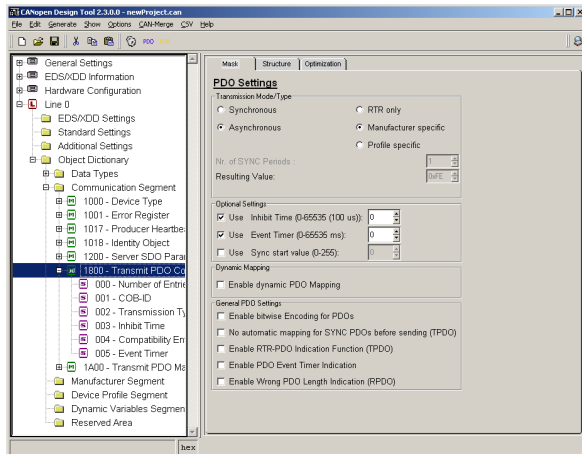
Delivery of the *CANopen Library* comprises different example programs that describe the usage of the various CANopen services. All examples contain a complete implementation of a CANopen device including the object directory as well as application code. These are ready to be compiled and run.

Among the detailed documented source code there is the reference manual and an a printed user manual, numbering 200 pages, as documentation of the *CANopen Library* available.

The *CANopen Library* is constantly improved and adopted to customer requirements. In order to keep up with the latest version of the *CANopen Library* **port** provides its customers an update service.

The support engineers of **port** are ready to answer all questions by email, phone or fax regarding the initial operation of the *CANopen Library* or further questions to the topic CANopen. This service is exempt from charges up to 45 days for requests by email and for requests by phone.

For development, test and initial operation of CANopen devices **port** provides a comprehensive tool chain. The creation of the object directory is simplified with the *CANopen Design Tool* (*CANopen Design Tool Light* version belongs to the scope of delivery).



The graphical *CANopen Design Tool* is available for the creation of the object directory, the electronic data sheet (EDS file) and for the documentation of the device in HTML or text format.

The *CANopen Device Monitor* can be utilized for commissioning, but also for the implementation and test phase. As a fully implemented CANopen master the *CANopen Device Monitor* with a graphical user interface assures access to all services in the network and also allows analysis of the bus traffic.

For the development of CAN application monitoring and analyzing the CAN bus traffic is essential. The service oriented display of CANopen messages of the *CAN-REport* allows easy and quick interpretation of CANopen messages.

Besides the communication objects of the CiA 301 several application objects in different device profiles are also specified in CANopen. These definitions guarantee a defined behavior of the corresponding device class and enable the interchangeability of CANopen devices.

For these different device profiles *port* provides extension modules as an add-on to the *CANopen Library*. These modules allow the customization of an application based on the CANopen standards.

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

Please have a look at the information of the data sheet *CANopen-Library Extension modules*.

## System environment

The *CANopen Library* runs on targets with and without operating systems. It supports many of the available CAN controllers and many microcontrollers/processors. For detailed information see data sheet *CANopen Driver Package*.

For systems based on Windows™ the *CANopen Library* is also available a Dynamic Link Library (DLL). This DLL is available for special hardware boards and contains all services of the *Master/Slave*

*standard* edition of the library. It can be used to create master or slave applications.

Furthermore a *CANopen-Starter Kit (Windows)* is available.

## Delivery scope

- CANopen library with separate driver interface
- CPU/CAN driver
- numerous, immediately compilable examples
- *CANopen Design Tool Light*
- detailed user manual
- reference manual containing descriptions of all functions including parameters and return values

### *CANopen Design Tool Light*

- generates an object dictionary and an initialization function in C-code, an Electronic Data Sheet and the documentation are produced automatically

## Licensing conditions (excerpt)

For the CANopen Library a one-off license fee in the form of a product / project or site license is collected. Further license fees do not arise from the deployment of the software within the same company (no runtime licenses). It is not allowed to hand over the software and the implementation, respectively, towards a third party.

## Further Services

### Maintenance Agreement

All changes of current standards as well as extensions through new developed standards are constantly incorporated into the *CANopen Library*. In order to take profit of the changes *port* offers all its customers a maintenance agreement with the following conditions:

- updates free of charge for the contracted period
- free of charge support for the *CANopen Library* and topics of CANopen

- Support for initial operation  
In order to provide a quick and effective access to the development of CANopen devices we recommend to do the initial operation of the CANopen device together on the target platform. Customer experiences of his/her hardware and the used compiler and the experience of our engineers with CANopen and the *CANopen Library* can complement each other. This leads to reduced development times and a CANopen conform device.
- Conformance Test  
In preparation of the charged Conformance test of the CiA we provide you our service, knowledge and equipment. With the preparation and execution of preliminary tests possible problems can be detected and removed.
- System analysis, consulting and support  
Take profit of our Know How during planning and development of your CANopen devices and networks. An ideal design requires knowledge about the used protocols and the system environment. Our competent engineers create together with you a cost-efficient solution that fits your needs to an optimum.
- Software development services for CAN, CANopen and DeviceNet
- Training  
To simplify the first steps with a new technology *port* provides one-day and several-days trainings for the topics of CANopen and its services as well as for the *CANopen Library* and its tools. These trainings can be given at *port* or at your companies rooms and are exclusively carried out for your workers.
- CANopen User Manual  
You want to get to know about the advantages of our *CANopen Library*? Then our user manual is the best starting point to get an overview about the functionality and deployment possibilities of the *CANopen Library*.

- Starter Kit

For a quick and uncomplicated start with CANopen the *CANopen Starter Kit* provides everything you need. It comes with a pre-compiled *CANopen Library* for your target hardware and several simple examples that provide a first insight and that ease the start with CANopen. As hardware interfaces several possibilities are available, like PC cards or USB interfaces. With these setting up a CAN network is as easy as a children's game.

The CANopen Starter Kit comes with hardware (USB/CAN interface) you can use for your first steps in CANopen. The special IO-Box module allows controlling and monitoring of most common features of CANopen IO modules. Besides that you get a CAN monitoring device which can be useful for developing, testing and commissioning of CANopen devices.

Further information can be found at *Windows™ Starter Kit*.

## Ordering Information

0564/12	CANopen-SRCLIB Slave Single Line
0564/10	CANopen-SRCLIB Master/Slave Single Line
0564/15	CANopen-SRCLIB Slave Multi Line
0564/13	CANopen-SRCLIB Master/Slave Multi Line
0910/xx	CANopen-DLL Master/Slave
0564/90	CANopen-SRCLIB User Manual
0571/10	CANopen-SRCLIB Software Maintenance Agreement
0569/10	CANopen Training Course
0570/10	CANopen Integration Support
0572/02	CANopen Starter Kit for WINDOWS

### 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.