



CANopen Tiny Manager

CANopen[®] Manager with CANopen Basic
Functions for LabVIEW[™] - CAN Applications

Software Manual

to Product C.1101.09

Notes

The information in this document has been carefully checked and is believed to be entirely reliable. esd electronics makes no warranty of any kind with regard to the material in this document and assumes no responsibility for any errors that may appear in this document. In particular descriptions and technical data specified in this document may not be constituted to be guaranteed product features in any legal sense.

esd electronics reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

All rights to this documentation are reserved by esd electronics. Distribution to third parties, and reproduction of this document in any form, whole or in part, are subject to esd electronics' written approval.

© 2021 esd electronics gmbh, Hannover

esd electronics gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

Tel.: +49-511-37298-0
Fax: +49-511-37298-68
E-Mail: info@esd.eu
Internet: www.esd.eu



This manual contains important information and instructions on safe and efficient handling of the CANopen Tiny Manager. Carefully read this manual before commencing any work and follow the instructions.
The manual is a product component, please retain it for future use.

Trademark Notices

CANopen® and CiA® are registered EU trademarks of CAN in Automation e.V.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

LabVIEW™ is a trademark of National Instruments.

All other trademarks, product names, company names or company logos used in this manual are reserved by their respective owners.

Document Information

Document file:	I:\Texte\Doku\MANUALS\PROGRAM\CAN\C.1101.09_CANopen_Tiny_Manager_LabVIEW\CANopen-Tiny-Manager_for_LabVIEW_software-manual_en_10.docx
Date of print:	2021-10-28
Document-type number:	DOC0800

Software version:	0.2.0 from 2019-12-20
-------------------	-----------------------

Document History

The changes in the document listed below affect changes in the hardware as well as changes in the description of the facts, only.

Rev.	Chapter	Changes versus previous version	Date
1.0	-	First English manual	2021-10-28

Technical details are subject to change without further notice.

Classification of Warning Messages and Safety Instructions

This manual contains noticeable descriptions for a safe use of the CANopen Tiny Manager and important or useful information.

NOTICE

Notice statements are used to notify people on hazards that could result in things other than personal injury, like property damage.



NOTICE

This NOTICE statement contains the general mandatory sign and gives information that must be heeded and complied with for a safe use.

INFORMATION



INFORMATION

Notes to point out something important or useful.

Data Safety

This software can be used to establish a connection to data networks. This may allow attackers to compromise normal function, get illegal access or cause damage.

esd does not take responsibility for any damage caused by the device if operated at any networks. It is the responsibility of the device's user to take care that necessary safety precautions for the device's network interface are in place.

Intended Use

The intended use of the CANopen Tiny Manager is the operation as CANopen® Manager with CANopen Basic Functions for LabVIEW™ - CAN Applications

Typographical Conventions

Throughout this manual the following typographical conventions are used to distinguish technical terms.

Convention	Example
File and path names	<code>/dev/null</code> or <code><stdio.h></code>
Function names	<code>open()</code>
Programming constants	<code>NULL</code>
Programming data types	<code>uint32_t</code>
Variable names	<code><i>Count</i></code>

Number Representation

All numbers in this document are base 10 unless designated otherwise. Hexadecimal numbers have a prefix of 0x. For example, 42 is represented as 0x2A in hexadecimal format.

Table of Contents

1	Overview	6
1.1	Description of CANopen Tiny Manager	6
1.2	Basic Functions.....	7
1.3	CANopen Tiny Manager Library	8
1.4	CANopen Tiny Manager VI Set	8
1.5	Glossary.....	9
2	Description of VI Set.....	10
2.1	Starting the Network.....	10
2.1.1	coTiny_open	10
2.1.2	coTiny_Close	11
2.1.3	coTiny_nmt.....	12
2.2	Network Monitoring	13
2.2.1	coTiny_guardingRegister	13
2.2.2	coTiny_heartBeatRegister	14
2.2.3	coTiny_heartBeatCON.....	14
2.2.4	coTiny_heartBeatPro	15
2.3	Network Synchronization.....	16
2.3.1	coTiny_startSync	16
2.3.2	coTiny_stopSyncFunctions	17
2.4	SDO (Service Data Object)	18
2.4.1	coTiny_sdoTransfer	18
2.5	PDO (Process Data Object)	19
2.5.1	coTiny_writePdo	19
2.5.2	coTiny_readPdoRegister	20
2.5.3	coTiny_readPdo.....	21
2.6	Return Values	22
3	License Terms.....	23
3.1	Software licensed by Third Parties	23
3.2	Software licensed by esd electronics.....	23
3.2.1	Rights of Use	23
3.2.2	Additional Rights for the “C Source Code”	23
3.2.3	Limited Warranty.....	23
3.2.4	Limitation of Liability	24
3.2.5	Support and Updates	24
3.3	Field Bus Protocols, Vendor-IDs	24
3.4	Trademark Rights of Third Parties.....	24
4	Order Information	25

List of Tables

Table 1: Order information.....	25
Table 2: Available Manuals	25

List of Figures

Figure 1: CANopen Tiny Manager with example VIs	6
Figure 2: Software components	7
Figure 3: Software structure.....	7

1 Overview

1.1 Description of CANopen Tiny Manager

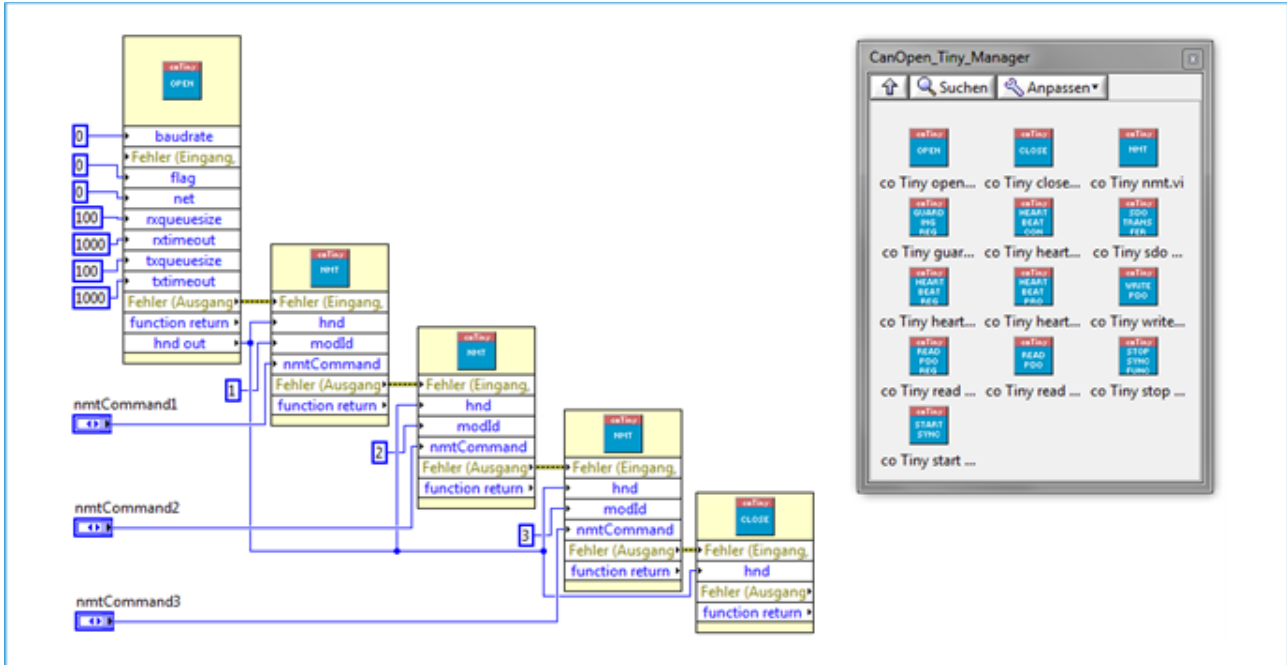


Figure 1: CANopen Tiny Manager with example VIs

The CANopen Tiny Manager for LabVIEW is a library, that facilitates the use of CANopen basic functions in LabVIEW™. In addition, a new VI Set is designed to make it easy to use the Tiny Manager library functions in LabVIEW.

The functions of the CANopen Tiny Manager Library are available in LabVIEW as a set of "Virtual Instruments" (VI).

The CANopen Tiny Manager for LabVIEW includes functions to support network management.

The Heartbeat and Guarding mechanisms are available to monitor for CANopen device failure. If the Heartbeat function is enabled, the nodes transmit their status on their own. Whereas the nodes are cyclically queried for their status by data request telegram (remote frame) if the Guarding function is used.

As a SYNC producer the CANopen Tiny Manager for LabVIEW can send the synchronization object (SYNC) periodically to all network nodes.

Service Data Objects and Process Data Objects (SDO & PDO)

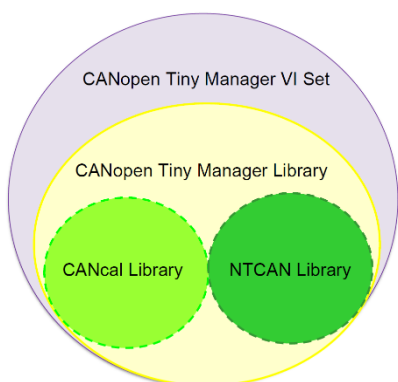
Transmission and reception of data in expedited, segment and block mode are possible. Functions for transmitting process data are provided.

This product is compatible with all esd CAN interfaces, supported by Windows®.

1.2 Basic Functions

The software comprises the following software components:

- CANopen Tiny Manager Library
- CANopen Tiny Manager VI Set



Name	Description
CANopen Tiny Manager Library	The CANopen Tiny Manager Library provides new functions based on the NTCAN Library and the CANcal Library.
CANopen Tiny Manager VI Set	The VI Set is used in LabVIEW to put the CANopen Tiny Manager Library functions into operation

Figure 2: Software components

The CANopen Tiny Manager Library provides the required functionality for the CANopen Tiny Manager VI Set. The CANopen Tiny Manager Library is based on the CANcal Library and the NTCAN Library.

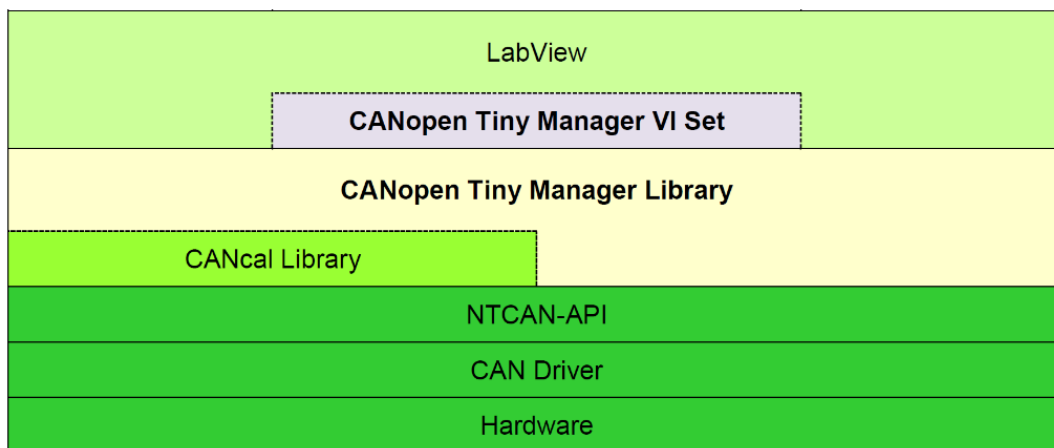


Figure 3: Software structure

1.3 CANopen Tiny Manager Library

CANopen Tiny Manager Library	CANopen Tiny Manager Library provides new functions based on NTCAN Library und CANcal Library.
------------------------------	--

Intention	Software interface between CANcal, NTCAN and LabVIEW.
Dependency	esd NTCAN library and esd CANcal library
Author / license holder	esd electronics gmbh
Delivery	Binary
Documentation	API description
License terms	See Licence Terms (chapter 3.2)

1.4 CANopen Tiny Manager VI Set

CANopen Tiny Manager VI Set	The functions of the CANopen Tiny Manager Library can be used in in LabVIEW by means of "Virtual Instruments" (VI). The CANopen Tiny Manager VI Set (VI) supports all functions of the CANopen Tiny Manager Library.
-----------------------------	--

Intention	CANopen Tiny Manager VI Set (VI) is provided for LabVIEW
Dependency	CANopen Tiny Manager library
Author / license holder	esd electronics gmbh
Delivery	Binary
Documentation	VI description
License terms	See Licence Terms (chapter 3.2)

1.5 Glossary

Abbreviations

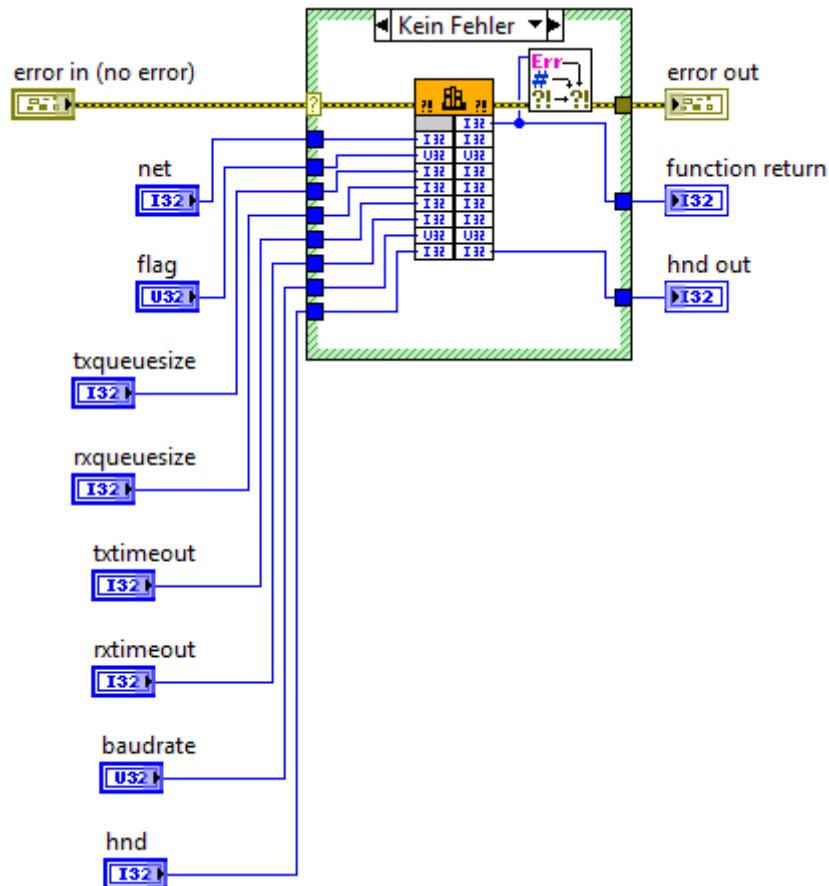
Abbreviation	Term
API	Application Programming Interface
CAN	Controller Area Network
CPU	Central Processing Unit
CiA	CAN in Automation
HW	Hardware
I/O	Input/Output
LSB	Least Significant Bit
MSB	Most Significant Bit
n. a.	not applicable
OS	Operating System
PDO	Process Data Object
SDK	Software Development Kit
SDO	Service Data Object

2 Description of VI Set

2.1 Starting the Network

2.1.1 coTiny_open

Use this function to open the CANopen Tiny Manager Library.



Function

```
int coTiny_open(int32_t    net,
                uint32_t   flag,
                int32_t    txqueuesize,
                int32_t    rxqueuesize,
                int32_t    txtimeout,
                int32_t    rxtimeout,
                uint32_t   baudrate);
COTINY_HANDLE *hnd);
```

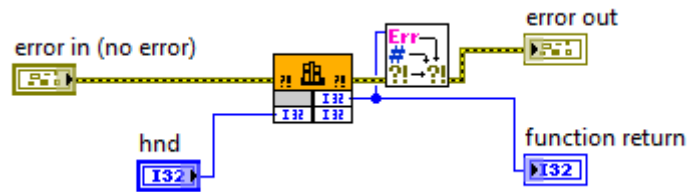


INFORMATION

For further information about the parameters of `coTiny_open` see `canOpen()` and `canSetBaudrate()` in the Application Developers Manual: NTCAN Part 1: Structure, Function and C/C++ API" (CAN-API-ME, see Order Information, page 25).

2.1.2 coTiny_Close

Close the CANopen Tiny Manager Library with this function.

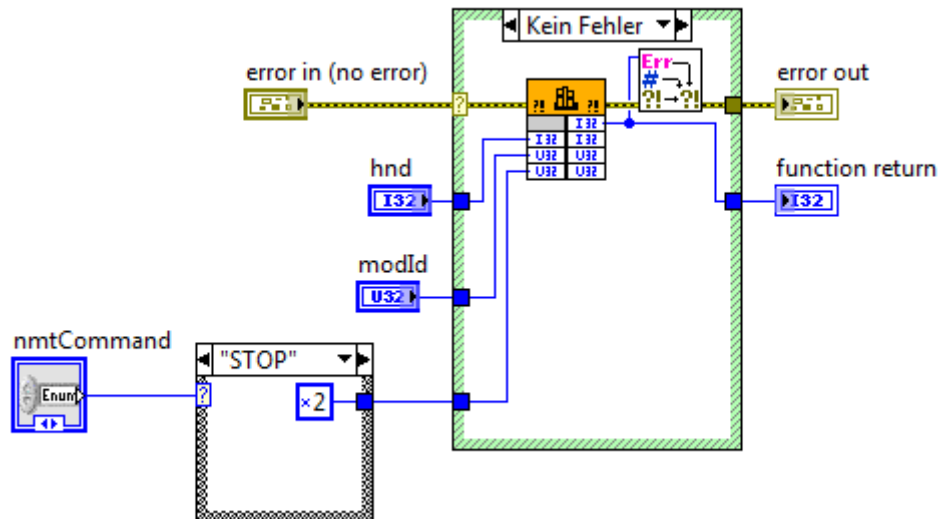


Function

```
int coTiny_close(COTINY_HANDLE hnd);
```

2.1.3 coTiny_nmt

This function handles the network management. The NMT-object can be sent.
This function allows the NMT-master to change the status of the other nodes in the network.



Function

```
int coTiny_nmt(COTINY_HANDLE hnd,  
              uint32_t modId,  
              uint32_t nmtCommand);
```

Defines

```
enum NMT_COMMAND {  
    NMT_START = 0x01,  
    NMT_PREOP = 0x80,  
    NMT_RESET = 0x81,  
    NMT_COMM = 0x82,  
    NMT_STOP = 0x02  
};
```

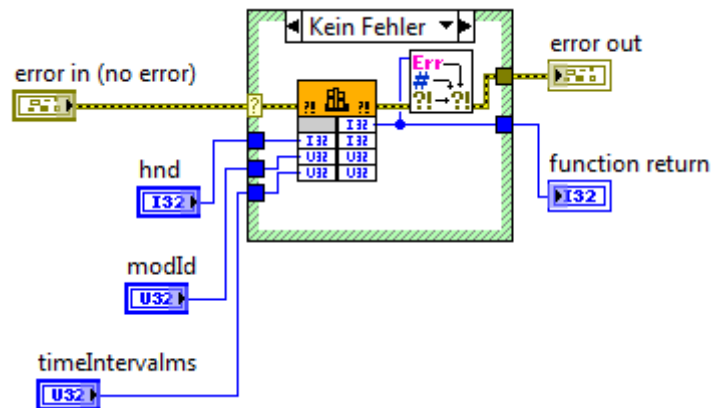
2.2 Network Monitoring

The Heartbeat- and the Guarding mechanism can be used for monitoring of the CANopen modules, especially to detect connection failures.

When using Guarding, the devices will cyclically be requested for their status (by remote frames). When Heartbeat is used, the nodes cyclically transmit a heartbeat message without using remote frames (recommended by CiA).

2.2.1 coTiny_guardingRegister

This function is used to launch the Node Guarding function.

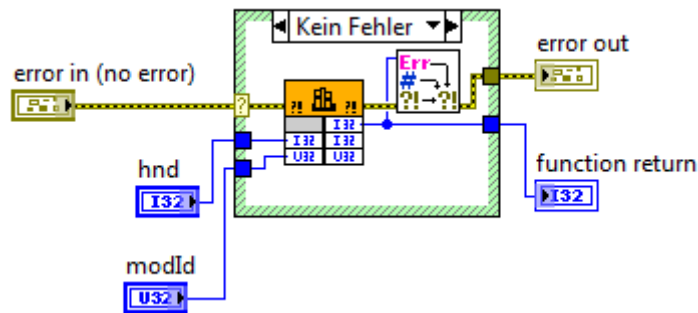


Function

```
int coTiny_guardingRegister( COTINY_HANDLE hnd,
                           uint32_t      modId,
                           uint32_t      timeIntervals);
```

2.2.2 coTiny_heartBeatRegister

Register a node as Heartbeat Consumer with this function. Call this function before the Heartbeat Consumer function *coTiny_heartBeatCON()* is called.

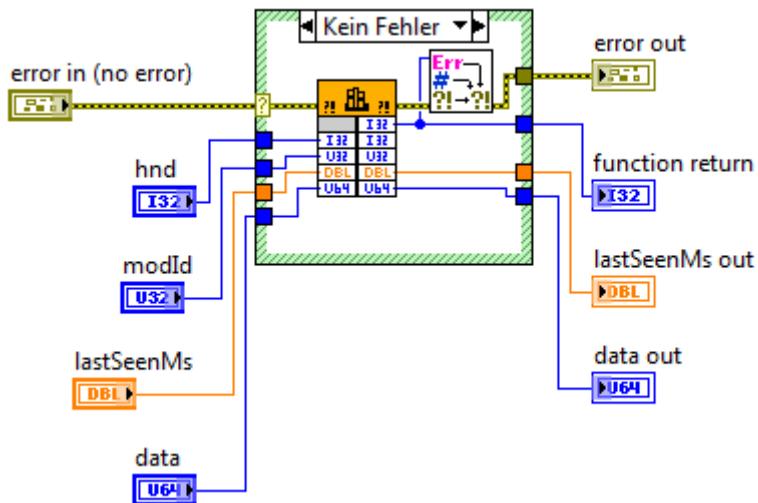


Function
<pre>int coTiny_heartBeatRegister(COTINY_HANDLE hnd, uint32_t modId);</pre>

2.2.3 coTiny_heartBeatCON

Monitor the received Heartbeat messages from the registered nodes with this function.

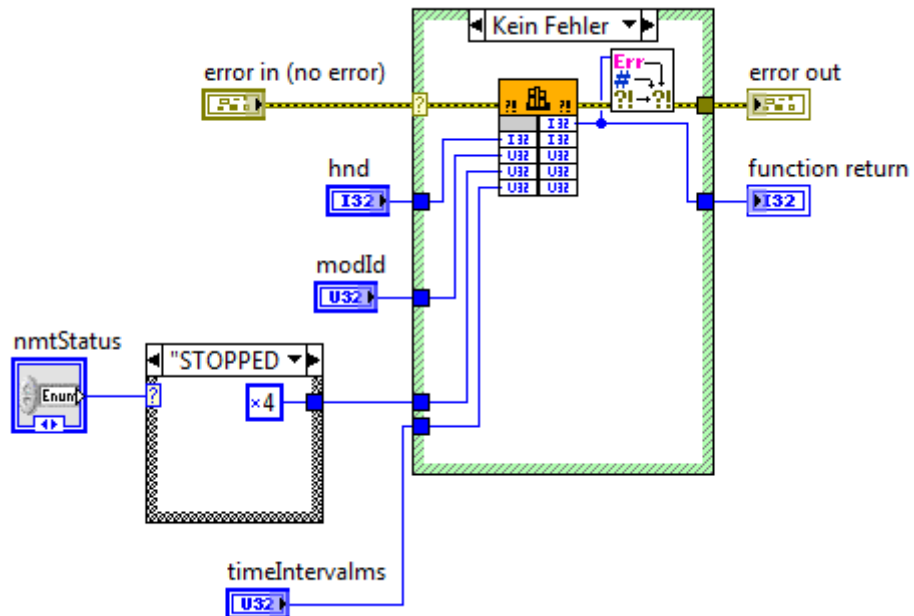
NOTICE
 Before using this function, the Heartbeat Register function must be used (see *coTiny_heartBeatRegister()*).



Function
<pre>int coTiny_heartBeatCon(COTINY_HANDLE hnd, uint32_t modId, double *lastSeenMs, uint64_t *data);</pre>

2.2.4 coTiny_heartBeatPro

Register a Heartbeat Producer with a given time interval and NMT status. The Heartbeat message will be sent every `timeIntervalMs`.



Function

```
int coTiny_heartBeatPro( COTINY_HANDLE hnd,
                        uint32_t modId,
                        uint32_t nmtStatus,
                        uint32_t timeIntervalMs);
```

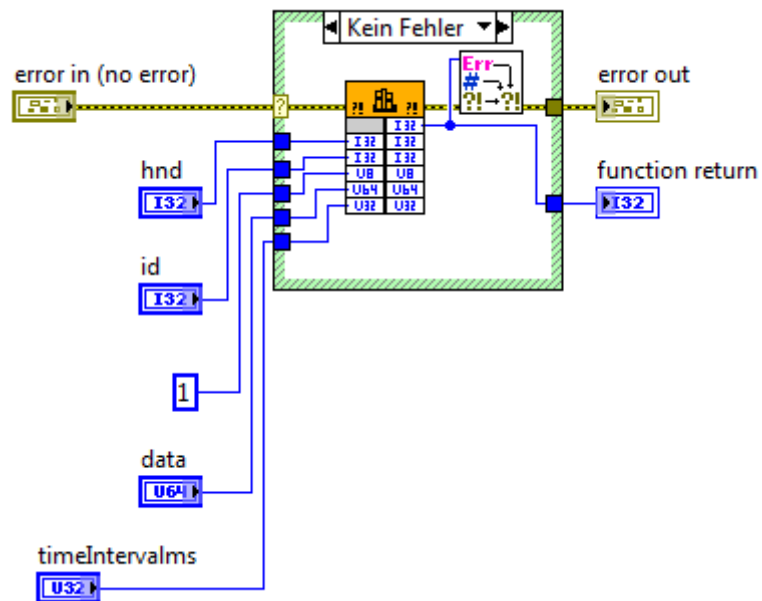
Defines

```
enum NMT_STATUS {
    NMT_BOOTUP = 0x00,
    NMT_OPERATIONAL = 0x05,
    NMT_PREOPR = 0x7f,
    NMT_STOPPED = 0x04
};
```

2.3 Network Synchronization

2.3.1 coTiny_startSync

With the help of this function, SYNC messages can be sent with different IDs to all nodes of the network.

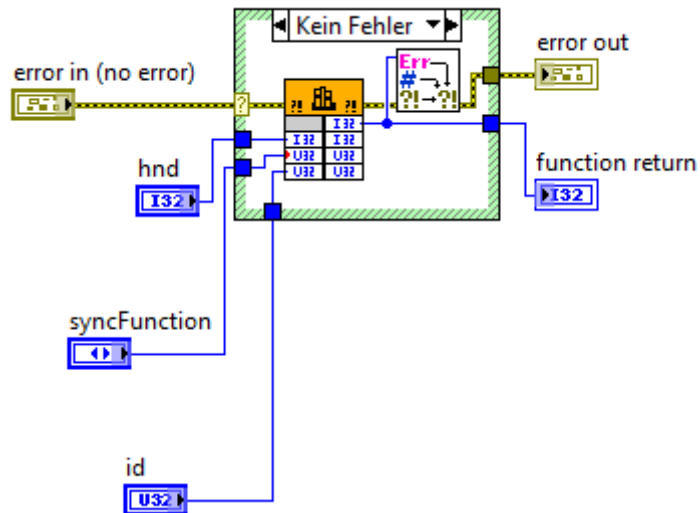


Function

```
int coTiny_startSync( COTINY_HANDLE  hnd,
                    int32_t          id,
                    uint8_t          dataLen,
                    uint64_t          data,
                    uint32_t          timeIntervalMs);
```


2.3.2 coTiny_stopSyncFunctions

This function is required to stop the SYNC message. For this, it must be precisely defined which sync message with which ID must be stopped.



Function

```
int coTiny_stopSyncFunctions(COTINY_HANDLE hnd,
                             uint32_t syncFunction
                             int32_t id);
```

Defines

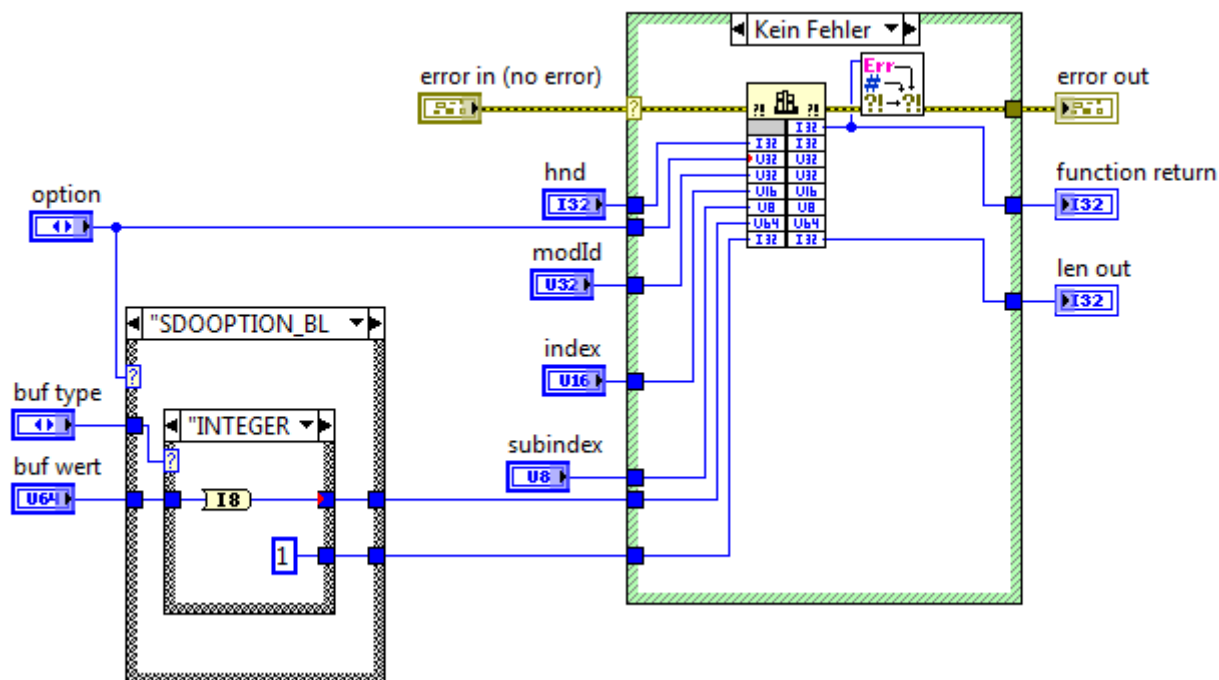
```
enum syncFunction {
    syncMessage,
    heartBeatCon,
    heartBeatPro,
    guarding
};
```

2.4 SDO (Service Data Object)

2.4.1 coTiny_sdoTransfer

This function enables SDO access.

The SDOs can be used for reading and writing of CANopen objects, for example for the configuration. The CANopen Tiny Manager Library uses the functions *ClientUpload* and *ClientDownload* of the CANcal Library for the access of the SDO-Mode.



Function

```
int coTiny_sdoTransfer(COTINY_HANDLE hnd,
                      uint32_t options,
                      uint8_t modId,
                      uint16_t index,
                      uint8_t subindex,
                      void *buf,
                      int32_t *len);
```

Defines

```
SDO_Flags:
#define SDOOPTION_DOWNLOAD 0
#define SDOOPTION_UPLOAD 1
#define SDOOPTION_BLOCK 2
```

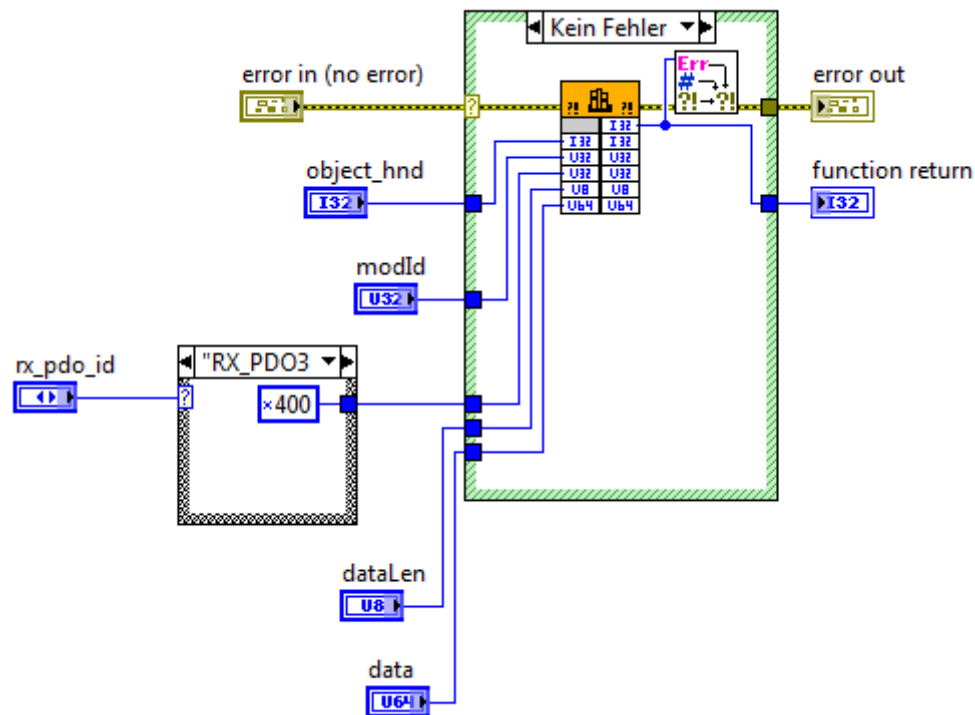
2.5 PDO (Process Data Object)

PDOs are used for the transmission of the process data.

A TX-PDO (TPDO) sends data on the CANopen network. Therefore, the NTCAN-Write function is used. The NTCAN object mode is used to receive Receive-PDOs (RPDO) from the CANopen network

2.5.1 coTiny_writePdo

This function can be used to write data in any PDO mapping.



Function

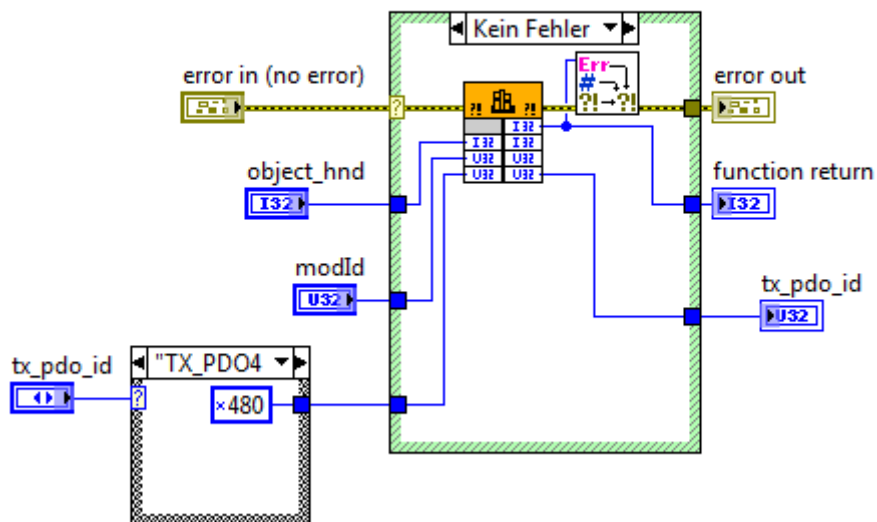
```
int coTiny_writePdo(COTINY_HANDLE  object_hnd,
                   uint32_t        modId,
                   uint32_t        rx_pdo_id,
                   uint8_t         dataLen,
                   uint64_t        data);
```

Defines

```
enum RX_PDO_ID {
    RX_PDO1 = 0x200,
    RX_PDO2 = 0x300,
    RX_PDO3 = 0x400,
    RX_PDO4 = 0x500
};
```

2.5.2 coTiny_readPdoRegister

Before you can read PDOs by means of the *coTiny_readPdo* function, this function must be used.



Function

```
int coTiny_readPdoRegister(COTINY_HANDLE object_hnd,
                          uint32_t      modId,
                          uint32_t      tx_pdo_id);
```

Defines

```
enum TX_PDO_ID {
    TX_PDO1 = 0x180,
    TX_PDO2 = 0x280,
    TX_PDO3 = 0x380,
    TX_PDO4 = 0x480
};
```

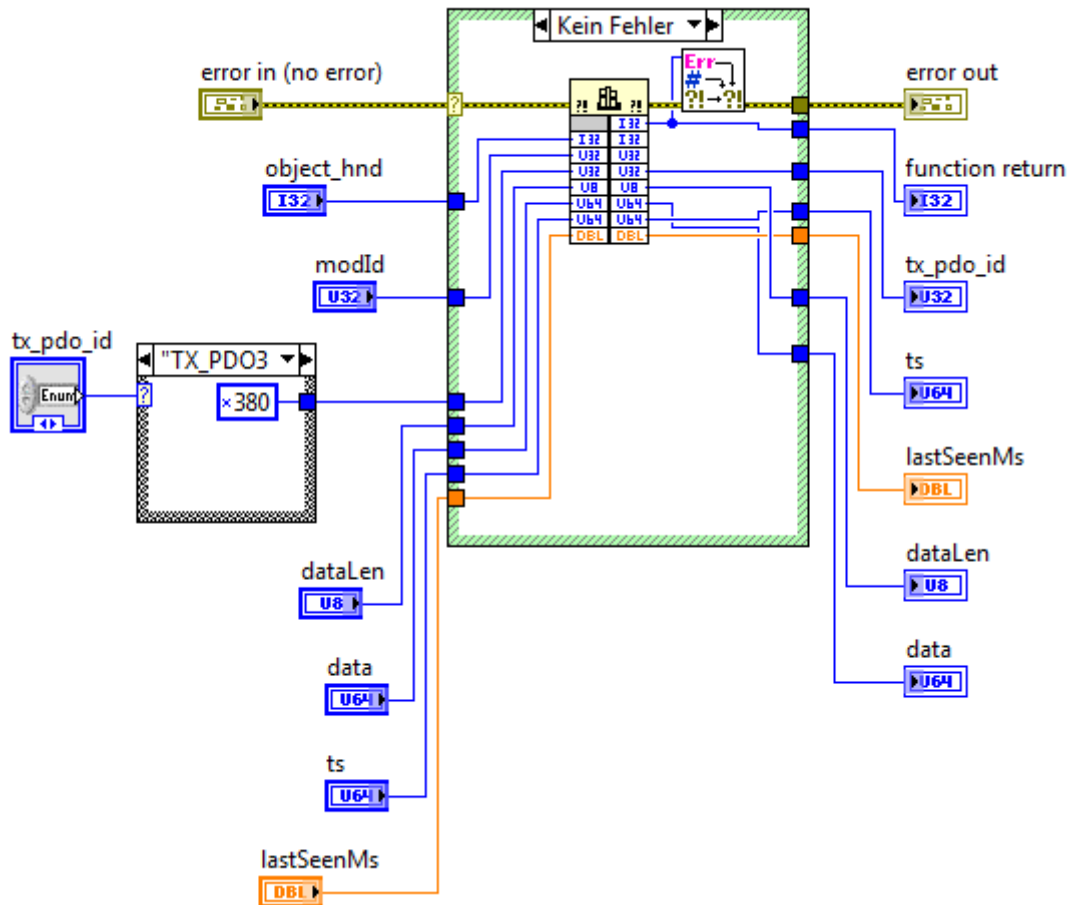
2.5.3 coTiny_readPdo



NOTICE

Before using this function, use the *coTiny_readPdoRegister* function must be used.

This function can be used to read PDOs.



Function

```
int coTiny_readPdo(COTINY_HANDLE object_hnd,
                  uint32_t      modId,
                  uint32_t      tx_pdo_id,
                  uint8_t       *dataLen,
                  uint64_t      *data,
                  uint64_t      *ts,
                  double        *lastSeenMs);
```

Defines

```
enum TX_PDO_ID {
    TX_PDO1 = 0x180,
    TX_PDO2 = 0x280,
    TX_PDO3 = 0x380,
    TX_PDO4 = 0x480
};
```

2.6 Return Values

All NTCAN-API functions return a status starting with the prefix 'NTCAN_', which should always be evaluated by the application. If the call returns an error code, the content of all returned values referenced by pointers are undefined and must not be evaluated by the application. The constants for the returned values are defined in <ntcan.h>.

The general return codes are described in the Application Developers Manual: NTCAN Part 1: Structure, Function and C/C++ API" in chapter "Return Codes" (CAN-API-ME, see Order Information, page 25).

3 License Terms

3.1 Software licensed by Third Parties

For software modules provided by third parties the license terms of these authors or licence holders do apply as specified in their software description. These license terms do apply to the product and must be observed by the CAN Customer.

3.2 Software licensed by esd electronics

The following rights of use (“License Terms”) shall apply to the software modules provided by esd electronics as the author and license holder.

3.2.1 Rights of Use

esd electronics grants CAN Customers the perpetual, non-exclusive right to use the software exclusively in the project CANopen Tiny Manager for LabVIEW.

The software may be copied to an unlimited number of CANopen Tiny Manager for LabVIEW devices and may be used there for an indefinite time.

Distribution and sale of the software by CAN Customers as well as the usage of the software modules in other projects than CANopen Tiny Manager for LabVIEW is prohibited. If the software is shipped as “C source code”, the usage in other projects, even in modified form, is prohibited.

3.2.2 Additional Rights for the “C Source Code”

If the software is supplied as “C source code”, esd electronics grants CAN Customers the following additional rights for this software:

The CAN Customer may modify, extend, adapt, and recompile (“process”) the software, as far as the software is still solely used on the CANopen Tiny Manager for LabVIEW only. The usage of the source code in other projects than CANopen Tiny Manager for LabVIEW as well as the distribution and sales to third parties, even in a modified form, is prohibited.

Processing the software by a service provider entrusted by the CAN Customer is permitted if the CAN Customer does not grant this service provider any own usage rights. This agreement between the CAN customer and the service provider must be agreed in writing.

3.2.3 Limited Warranty

The software supplied by esd electronics is in accordance with the latest technological standards and complies with the product information and product specification provided by esd electronics.

esd electronics points to the fact that with respect to the current state of the art, even with a maximum of care and diligence it is not possible to avoid software faults.

Indicated software faults will be fixed on esd electronics' choice either by rectification of the fault (“rectification”) or by supplying a defect-free software (“Substitute Delivery”) within a reasonable period of time. The costs for the remedial action are paid by esd electronics. If the software is not explicitly agreed to be supplied as “source code”, the CAN customer is in no way entitled to claim the right to receive the source code regarding due to software faults.

3.2.4 Limitation of Liability

esd electronics is liable for all damage which happens to CAN Customer caused in the case of gross negligence or malicious conduct by esd electronics or caused in the case of any culpable injury to life, physical integrity, or health by esd electronics for which esd electronics is liable under the German Product Liability Act. In all other cases esd electronics' liability for damages is limited to the case of culpable breach of major contractual obligations by esd electronics. The liability for consequential damage of any kind is excluded.

3.2.5 Support and Updates

Within 12 months after delivery of the first prototypes esd electronics provides support (hotline via phone or email) free of charge and supplies software updates free of charge.

After these 12 months further support and updates can be agreed in a separate support contract.

3.3 Field Bus Protocols, Vendor-IDs

If necessary, the "Vendor-IDs", specified for CAN Customer, for the implementation of the specific bus protocols CANopen, PROFINET, EtherCAT and Ethernet/IP in the development results are provided by CAN Customer.

3.4 Trademark Rights of Third Parties

Furthermore, third party property and trademark rights, CANopen here (CAN in automation (CiA)), must be observed. The CAN Customer takes care for the correct usage of labels and logos at his own and, if necessary, bears the costs that may appear to maintain the products conformance at future modified requirements (for example conformance tests, changes of user-group specifications) of the third party as well as the costs resulting in the usage of labels and logos for some other reason.

4 Order Information

Type	Properties	Order No.
CANopen Tiny Manager for LabVIEW	CANopen basic functions for NI LabVIEW Available for NI LabVIEW 2013 for Windows 7/8/10 (32-/64-bit) operating systems	C.1101.09

Table 1: Order information

PDF Manuals

For the availability of the manuals see table below.

Please download the manuals as PDF documents from our esd website <https://www.esd.eu> for free.

Manuals		Order No.
CANopen Tiny Manager-ME	Software manual in English	C.1101.10
CAN-API-ME	NTCAN-API Part 1: Application Developers Manual NTCAN-API Part 2: Driver Installation Guide	C.2001.21
CANopen-ME	CANopen Manuals in English	C.2002.21

Table 2: Available Manuals

Printed Manuals

If you need a printout of the manual additionally, please contact our sales team (sales@esd.eu) for a quotation. Printed manuals may be ordered for a fee.