



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 checked carefully and is considered to be entirely reliable. esd electronics makes no warranty of any kind regarding the material in this document and assumes no responsibility for any errors that may appear in this document. In particular, the 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.

© 2024 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_11.docx
Date of print:	2024-06-18
Document-type number:	DOC0800

Software version:	0.3.0
-------------------	-------

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
1.1	all	All chapters revised, Names and icons of the VIs inserted, if applicable, figure of defines added	2024-06-18
	1.1	Figure 1 updated	
	2.1	Overview of Vis added	
	1.1, 2.3.1, 2.3.4, 2.4.1, 2.4.2	Note on the required CAN driver version included	
	3	Chapter revised	

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><i>open()</i></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	Overview of Vis	10
2.2	Starting the Network.....	11
2.2.1	co Tiny open.vi	11
2.2.2	co Tiny close.vi	12
2.2.3	co Tiny nmt.vi	13
2.3	Network Monitoring	14
2.3.1	co Tiny guarding Register.vi.....	14
2.3.2	co Tiny heart Beat Register.vi.....	15
2.3.3	co Tiny heart Beat Con.vi.....	15
2.3.4	co Tiny heart Beat Pro.vi.....	16
2.4	Network Synchronization.....	17
2.4.1	co Tiny start Sync.vi.....	17
2.4.2	co Tiny stop Sync Functions.vi.....	18
2.5	SDO (Service Data Object)	19
2.5.1	co Tiny sdo Transfer.vi.....	19
2.6	PDO (Process Data Object)	20
2.6.1	co Tiny write Pdo.vi.....	20
2.6.2	co Tiny read Pdo Register.vi.....	21
2.6.3	co Tiny read Pdo.vi	22
2.7	Return Values	23
3	License Terms.....	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
Figure 4: <i>CanOpen_Tiny_Manager</i>	10

1 Overview

1.1 Description of CANopen Tiny Manager

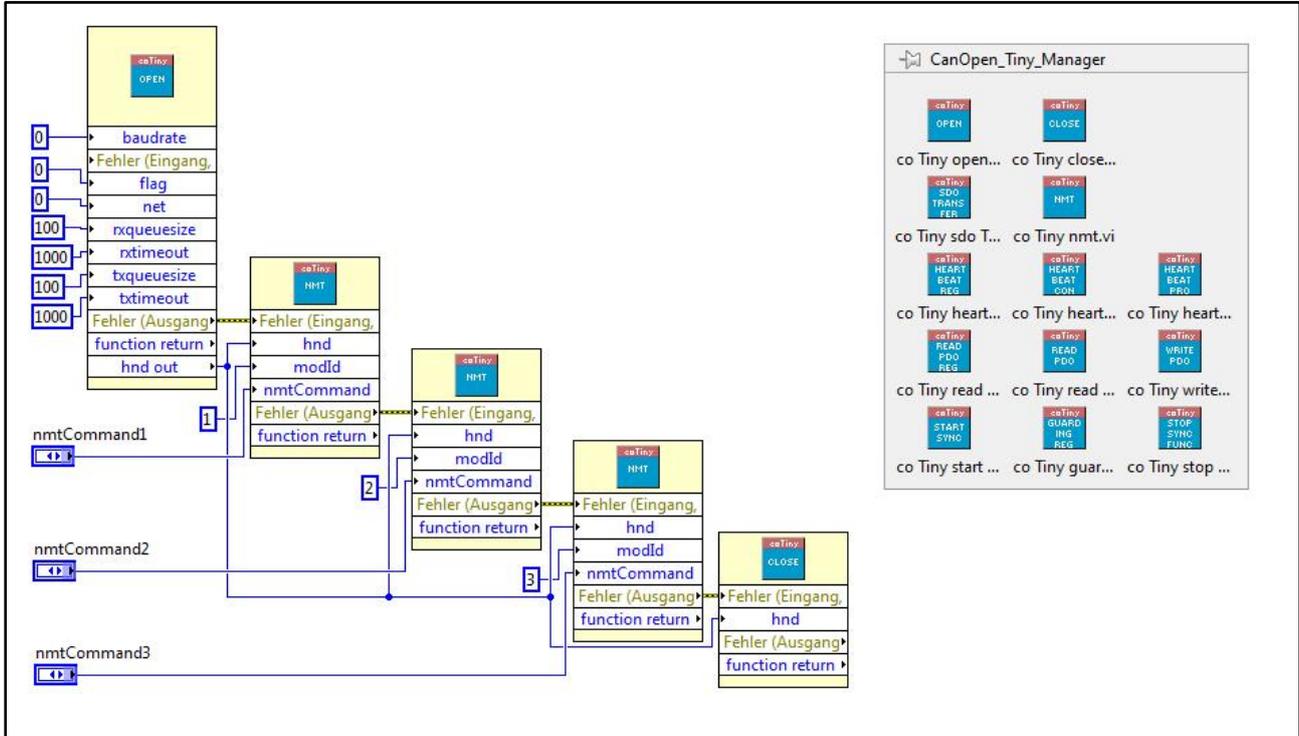


Figure 1: CANopen Tiny Manager with example VIs

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®.

i

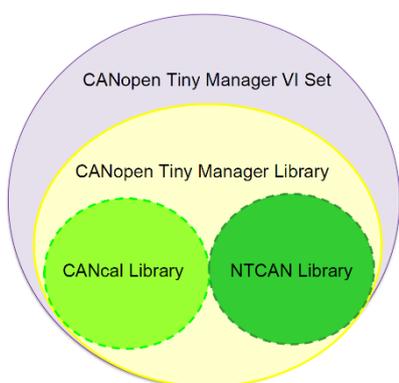
INFORMATION

Please note, that some of the VIs only work with CAN drivers from version 3.X. (I.e. not with CAN driver V2 due to the lack of Tx scheduling functionality):

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.

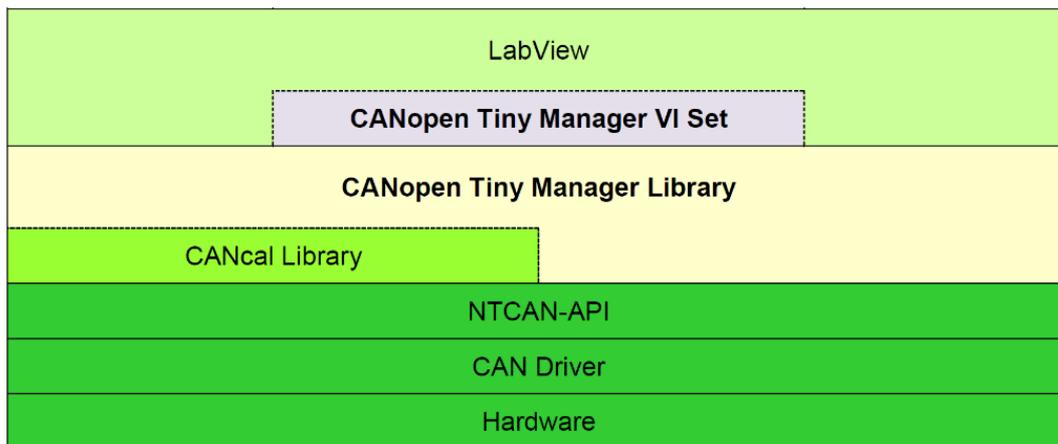


Figure 3: Software structure

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 chapter License Terms, page 24

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 chapter License Terms, page 24

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 Overview of Vis

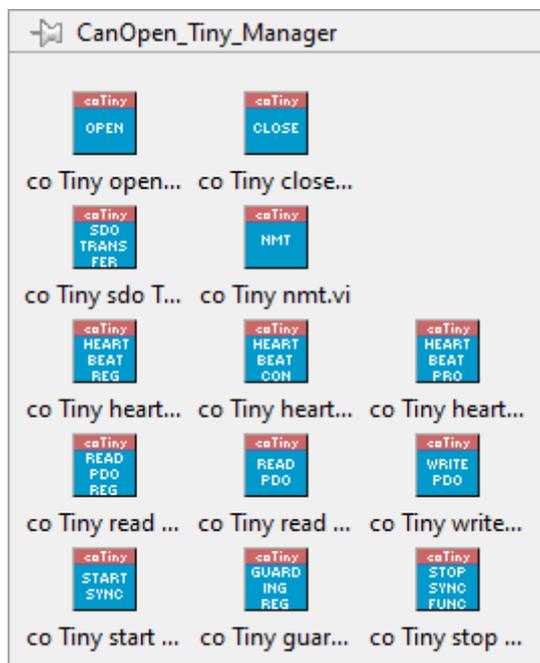


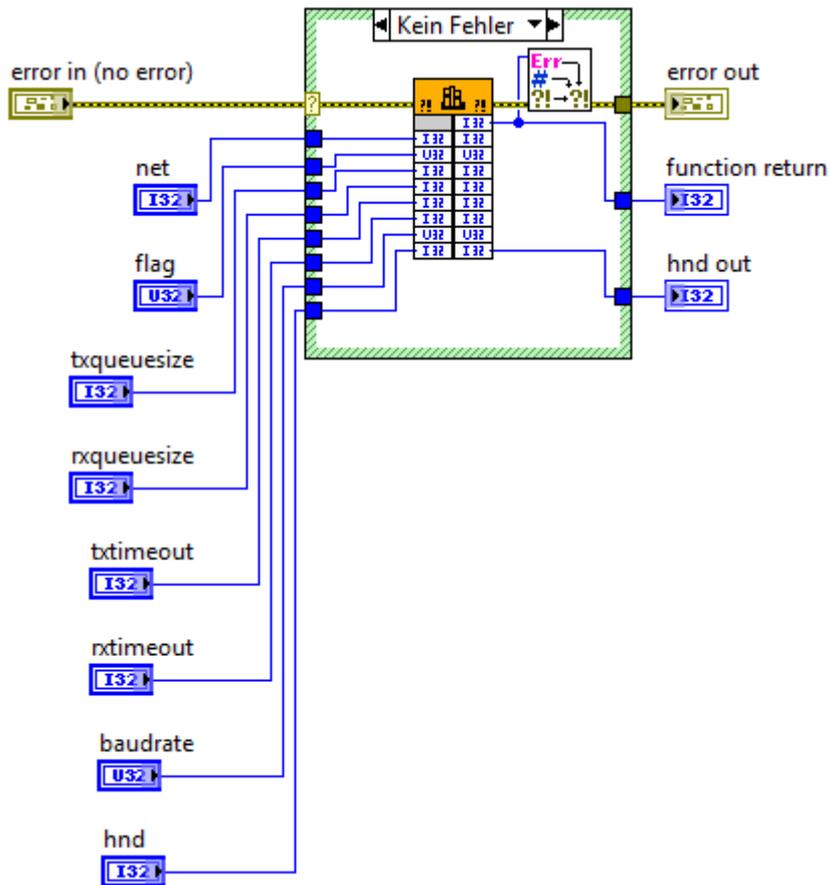
Figure 4: CanOpen_Tiny_Manager

VI Name	Functionality	See page
co Tiny close.vi	Close the CANopen Tiny Manager Library	12
co Tiny guarding Register.vi	Launch the Node Guarding function	14
co Tiny heart Beat Con.vi	Monitor the received Heartbeat messages	15
co Tiny heart Beat Pro.vi	Register a Heartbeat Producer with a given time interval and NMT status	16
co Tiny heart Beat Register.vi	Register a node as Heartbeat Consumer	14
co Tiny nmt.vi	Handle the network management	13
co Tiny open.vi	Open the CANopen Tiny Manager Library	11
co Tiny read Pdo Register.vi	Read PDO register	21
co Tiny read Pdo.vi	Read PDOs	22
co Tiny sdo Transfer.vi	Enable SDO access	19
co Tiny start Sync.vi	Send SYNC messages	17
co Tiny stop Sync Functions.vi	Stop sending a defined SYNC message	18
co Tiny write Pdo.vi	Write PDOs	20

2.2 Starting the Network

2.2.1 co Tiny open.vi

Use this VI 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);
```

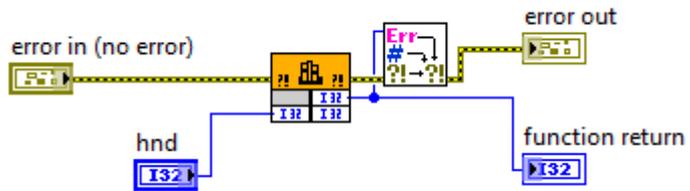


INFORMATION

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).

2.2.2 co Tiny close.vi

Close the CANopen Tiny Manager Library with this VI.

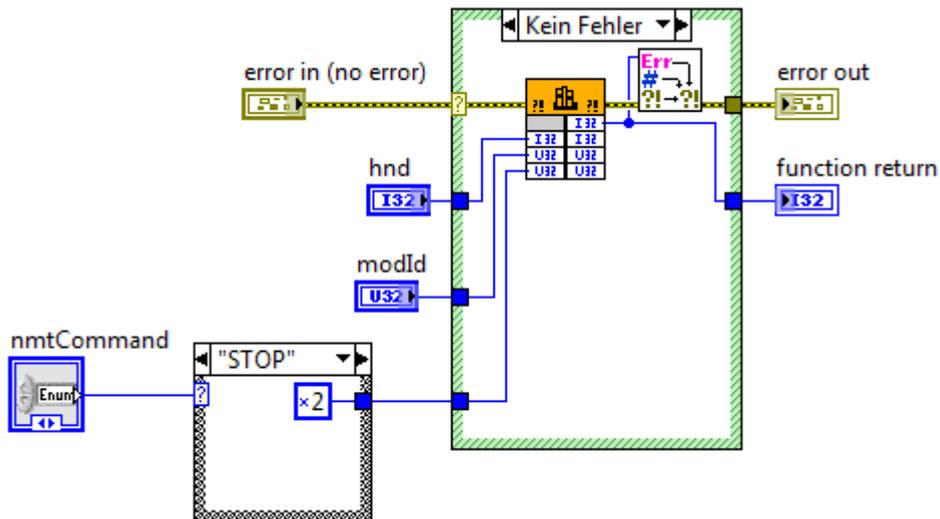
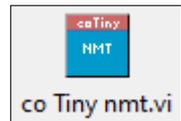


Function

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

2.2.3 co Tiny nmt.vi

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

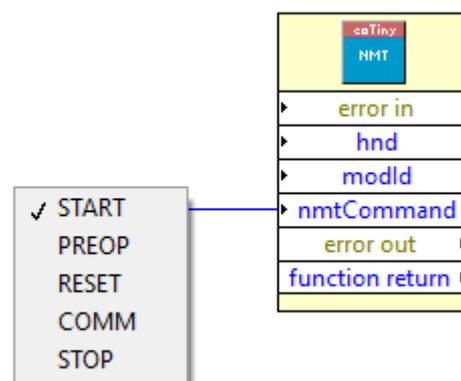


Function

```
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
};
```



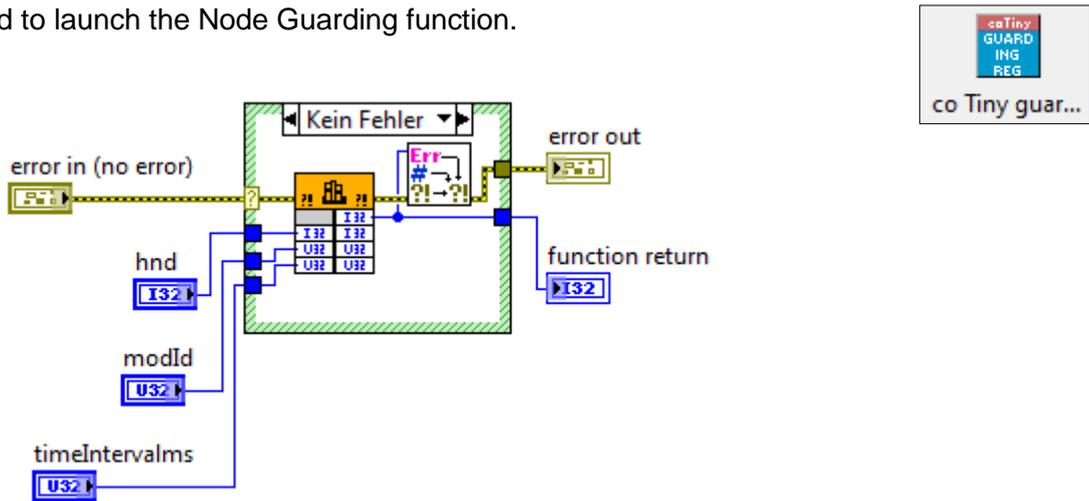
2.3 Network Monitoring

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).

2.3.1 co Tiny guarding Register.vi

This VI is used to launch the Node Guarding function.

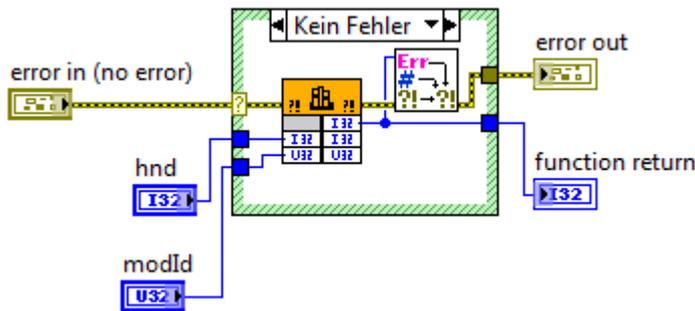


Function
<pre>int coTiny_guardingRegister(COTINY_HANDLE hnd, uint32_t modId, uint32_t timeIntervalMs);</pre>

i **INFORMATION** This VI only works with CAN drivers from version V3.X. (I.e. not with CAN driver V2 due to the lack of Tx scheduling functionality).

2.3.2 co Tiny heart Beat Register.vi

Register a node as Heartbeat Consumer with this VI. The function `coTiny_heartBeatRegister()` must be called before the Heartbeat Consumer function `coTiny_heartBeatCon()` can be called via `coTiny_heartBeatCon.vi`.



Function

```
int coTiny_heartBeatRegister(COTINY_HANDLE hnd,
                             uint32_t modId);
```

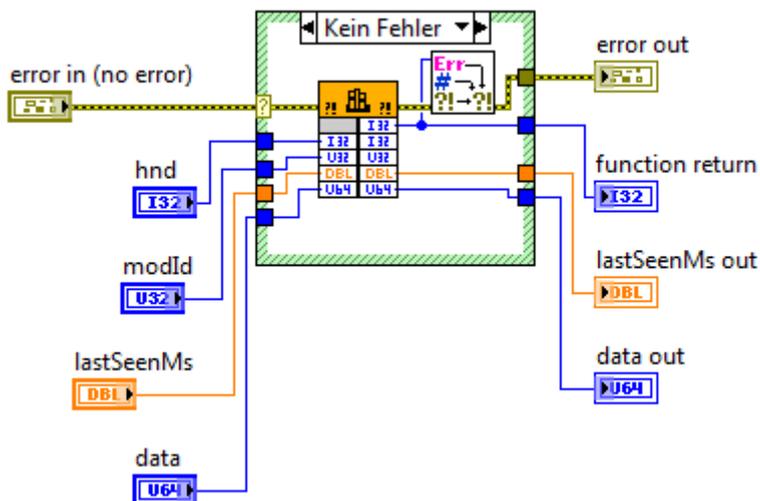
2.3.3 co Tiny heart Beat Con.vi



NOTICE

Before using this VI, the Heartbeat Register function (`coTiny_heartBeatRegister()`) must be used (see `coTiny_heartBeatRegister.vi`).

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

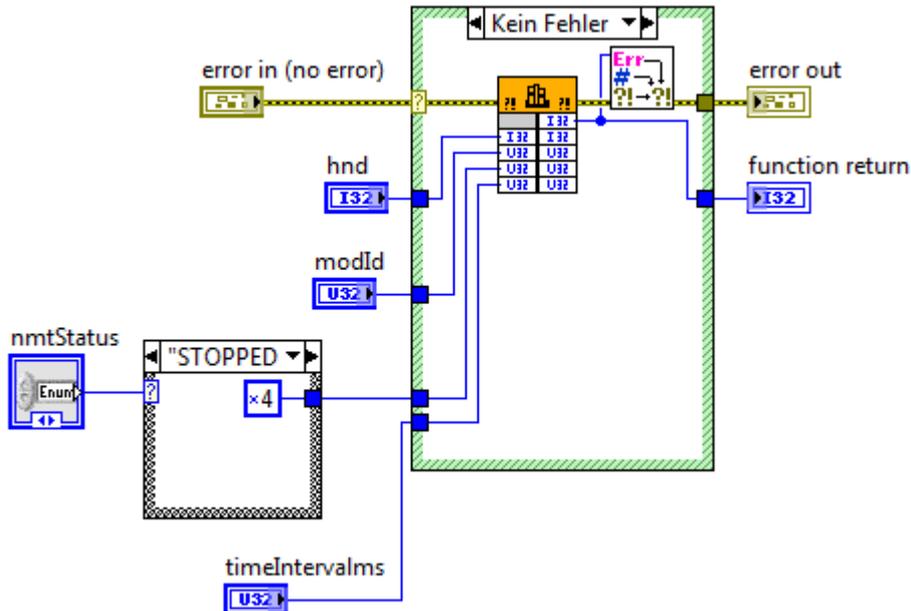


Function

```
int coTiny_heartBeatCon(COTINY_HANDLE hnd,
                        uint32_t modId,
                        double *lastSeenMs,
                        uint64_t *data);
```

2.3.4 co Tiny heart Beat Pro.vi

Register a Heartbeat Producer with a given time interval and NMT status via this VI. The Heartbeat message will be sent every time interval defined in *timeIntervalMs*.



Function

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

Defines

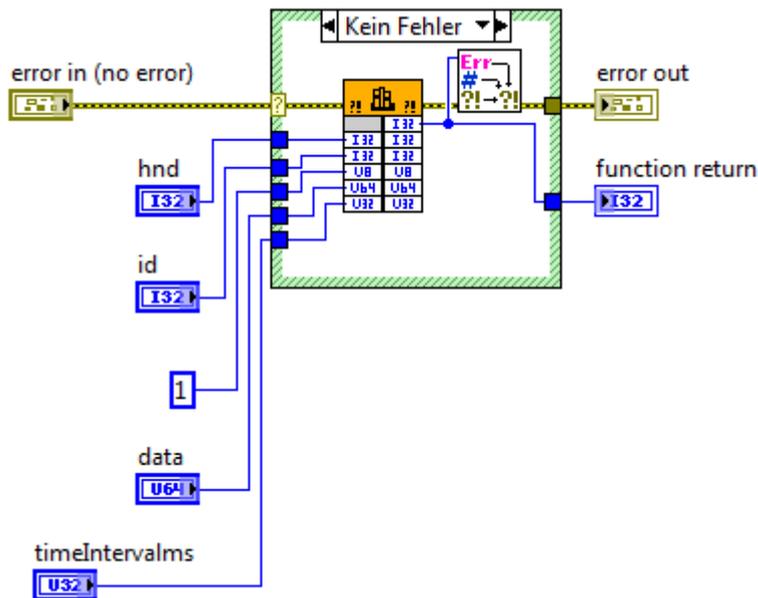
```
enum NMT_STATUS {
  NMT_BOOTUP = 0x00,
  NMT_OPERATIONAL = 0x05,
  NMT_PREOPR = 0x7f,
  NMT_STOPPED = 0x04
};
```

INFORMATION
 This VI only works with CAN drivers from version V3.X.
 (I.e. not with CAN driver V2 due to the lack of Tx scheduling functionality).

2.4 Network Synchronization

2.4.1 co Tiny start Sync.vi

With the help of this VI, 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);
```

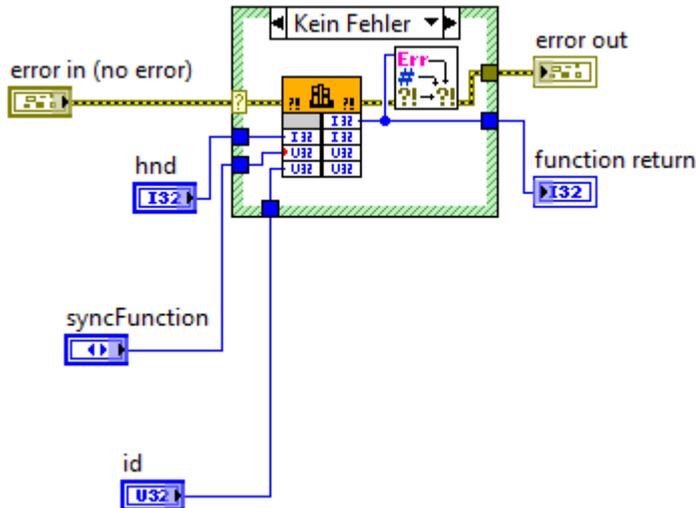


INFORMATION

This VI only works with CAN drivers from version V3.X.
(I.e. not with CAN driver V2 due to the lack of Tx scheduling functionality).

2.4.2 co Tiny stop Sync Functions.vi

This VI 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
};
```

i INFORMATION
 This VI only works with CAN drivers from version V3.X.
 (I.e. not with CAN driver V2 due to the lack of Tx scheduling functionality).

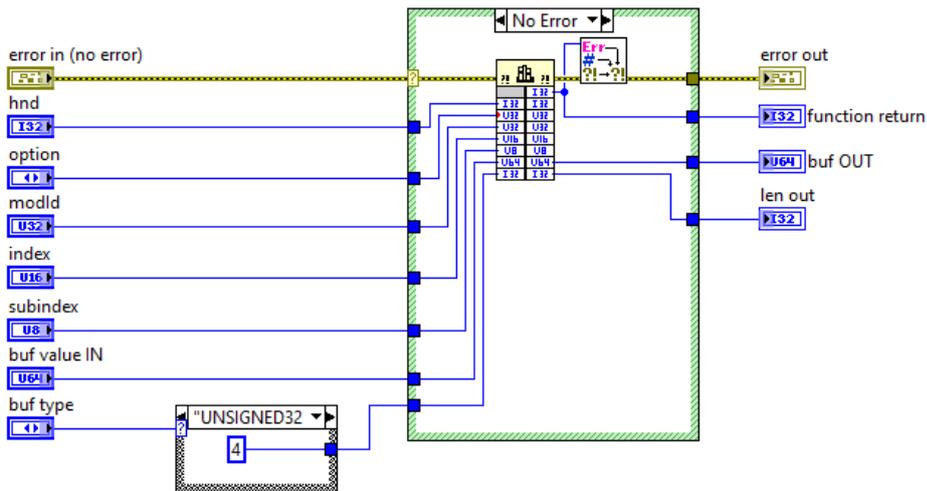
2.5 SDO (Service Data Object)

2.5.1 co Tiny sdo Transfer.vi

This VI 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.



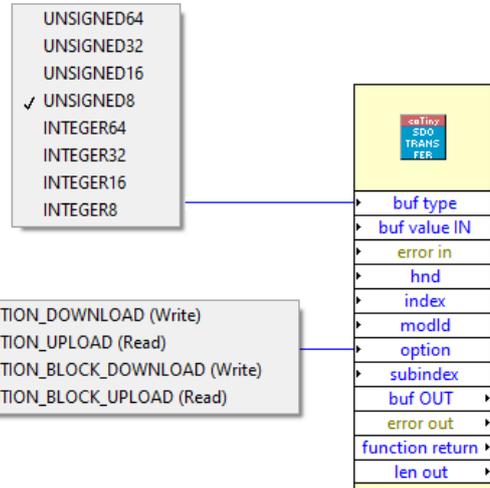
Function

```
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_DOWNLOAD 2
#define SDOOPTION_BLOCK_UPLOAD 3
```



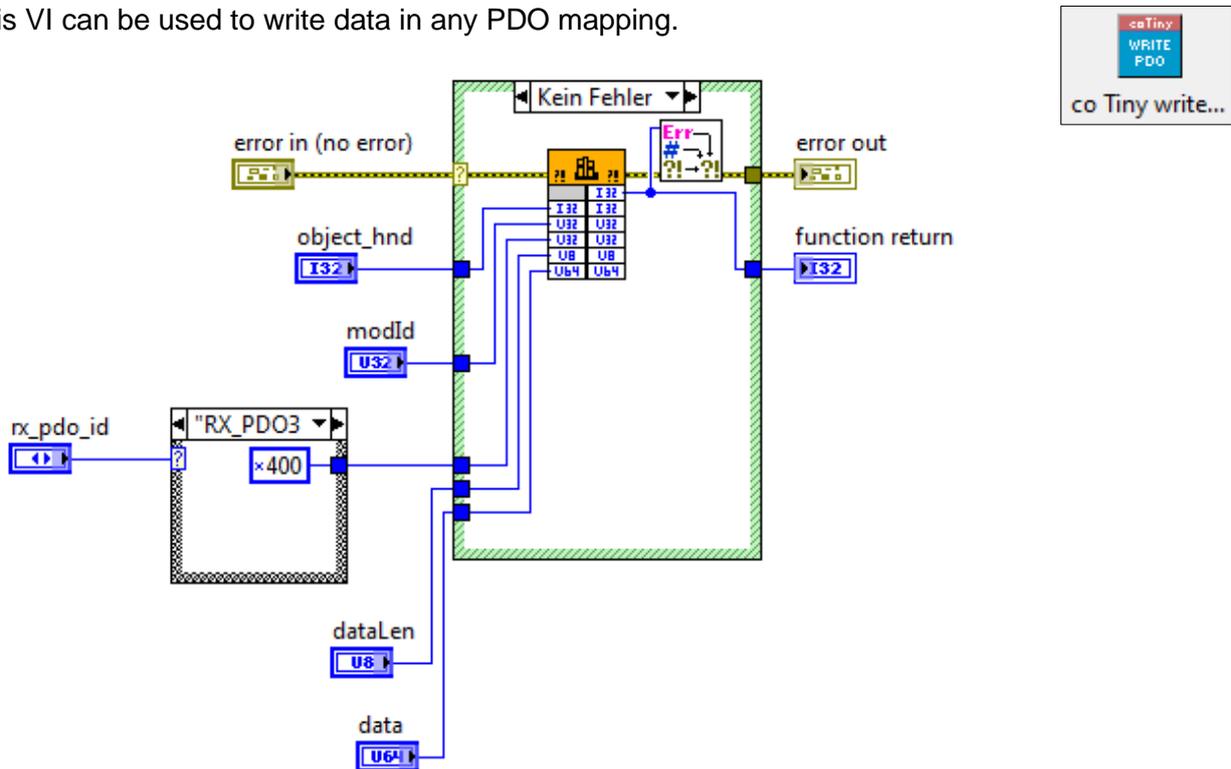
2.6 PDO (Process Data Object)

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.

2.6.1 co Tiny write Pdo.vi

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

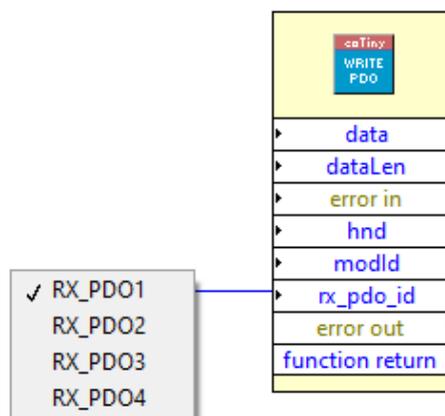


Function

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

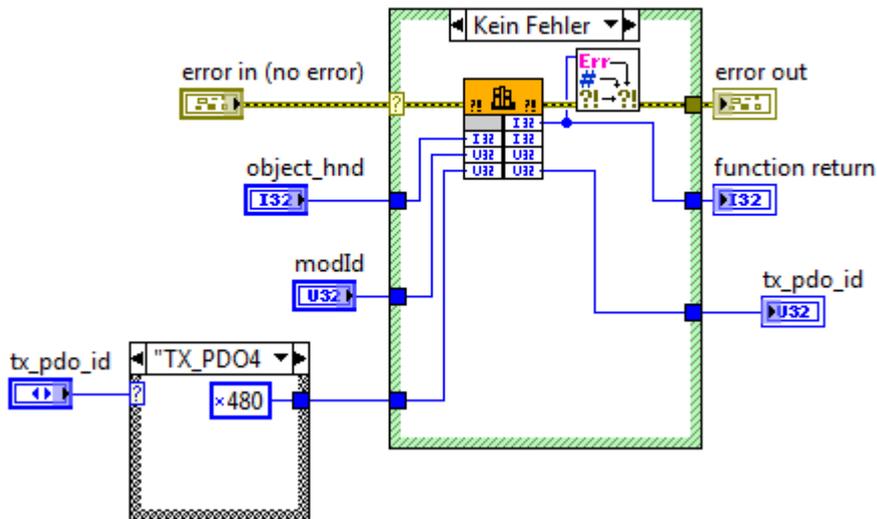
Defines

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



2.6.2 co Tiny read Pdo Register.vi

Before you can read PDOs by means of the *co Tiny* read Pdo function, this VI must be used.



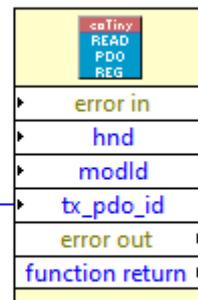
Function

```
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
};
```

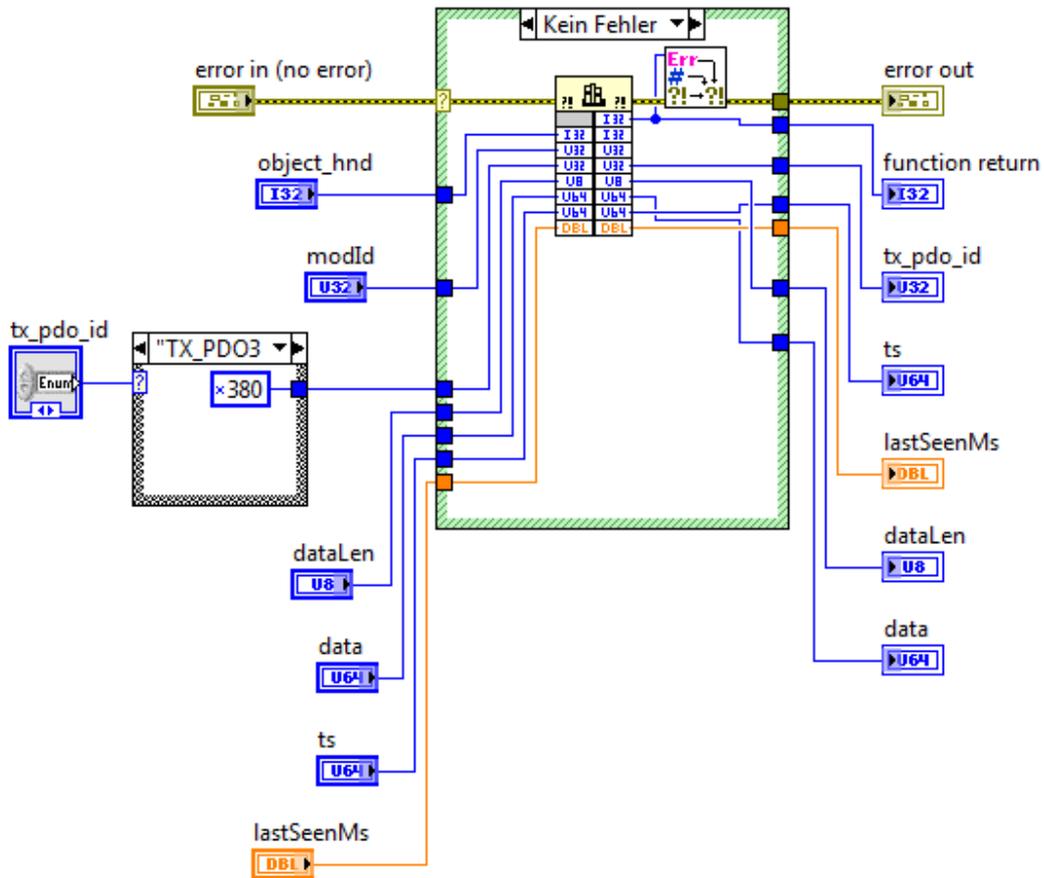
- ✓ TX_PDO1
- TX_PDO2
- TX_PDO3
- TX_PDO4



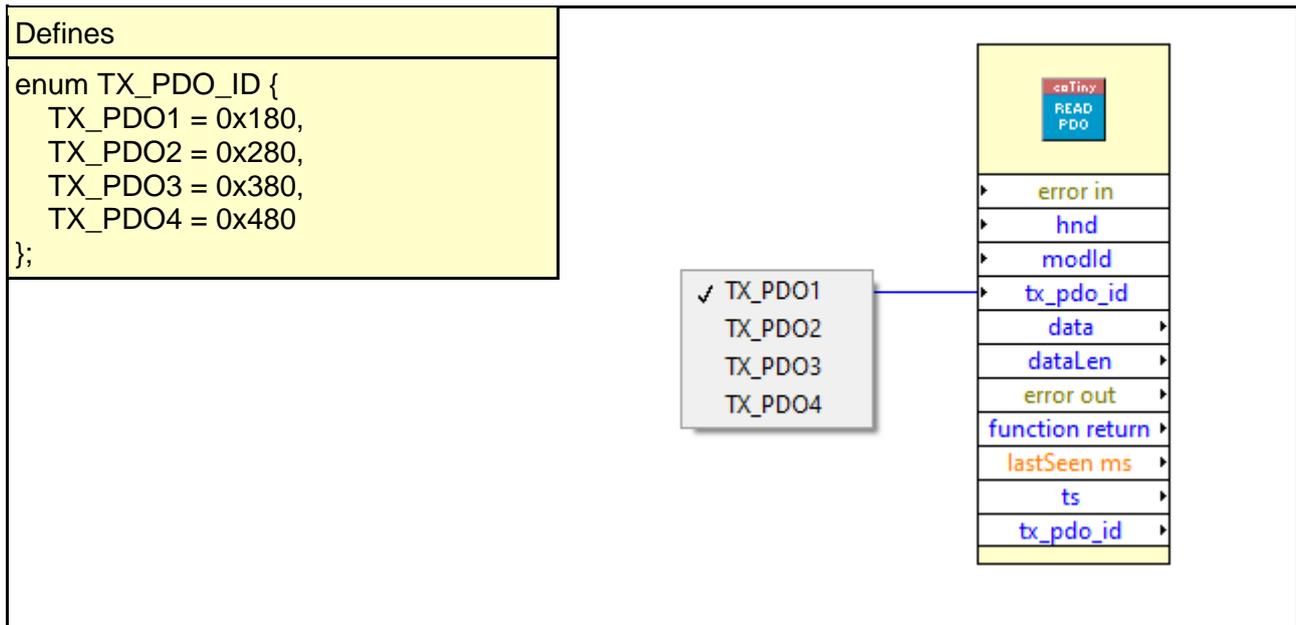
2.6.3 co Tiny read Pdo.vi

NOTICE
 Before using this VI, use co Tiny read Pdo Register.vi. The *co Tiny read Pdo Register()* function must be used before the function *coTiny_readPdo()* can 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);



2.7 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



NOTICE

The software from esd used in the CANopen Tiny Manager is subject to the license terms of the respective authors or rights holders. CANopen Tiny Manager may only be used in accordance with these license terms!

By using the CANopen Tiny Manager you agree to the terms of these software licenses.

The license terms of esd electronics gmbh (Software Licence Agreement) is displayed and installed on your system during installation of the CAN SDK.

You can also download the Software Licence Agreement from our website:

https://esd.eu/fileadmin/esd/software/licenses/esd/license_binary.txt

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 7/8/10/11 (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.