COBview

CANopen Tool for Analyzing and Diagnosing CANopen Nodes

Software Manual

esd electronic system design gmbh
Vahrenwalder Str. 207 • 30165 Hannover • Germany
www.esd-electronics.com • Fax: 0511/37 29 8-68
Phone: 0511/37 29 80 • International: +49-5 11-37 29 80

N O T E

The information in this document has been carefully checked and is believed to be entirely reliable. **esd** 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. **esd** reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

esd assumes no responsibility for the use of any circuitry other than circuitry which is part of a product of **esd** gmbh.

esd does not convey to the purchaser of the product described herein any license under the patent rights of **esd** gmbh nor the rights of others.

esd electronic system design gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

Phone: +49-511-372 98-0
Fax: +49-511-372 98-68
E-mail: info@esd-electronics.com
Internet: www.esd-electronics.com

USA / Canada:
esd electronics Inc.
525 Bernardston Road
Suite 1
Greenfield, MA 01301
USA

Phone: +1-800-732-8006
Fax: +1-800-732-8093
E-mail: us-sales@esd-electronics.com
Internet: www.esd-electronics.us

Document file:	I:\texte\Doku\MANUALS\PROGRAM\CAN\CAN-Tools\COBview\englisch\COBview_11.en9
Date of print:	2007-06-20

Changes in the chapters

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

Chapter	Changes versus previous version
-	Minor changes.

Technical details are subject to change without further notice.

This page has intentionally been left blank.

Contents

1. Introduction 6
1.1 Installation 6

2. Functions of the Interface Elements 7
2.1 Display of the CANopen ObjectView Window 7
2.2 Description of Buttons 8
CAN-field 8
Info field 9
Module field 9
NMT-Management NMT 10
Status Changes in Accordance with CANopen DS-301 11
Index field 12
Read/Write 12
Data type 14
List View 15

1. Introduction

This document describes the program **C**ANopen **Ob**ject **View**. **COBview** is an effective CANopen tool for the analysis/diagnostics of CANopen nodes.

The program offers:
- detecting and displaying CANopen devices in networks
- basic CANopen NMT functionality
- the support of read and write accesses, i.e. that individual objects can be written via SDO.

The program must not be run during normal operation in a CAN-network, because it conflicts with CAN-master or CAN-manager.

1.1 Installation

The tool COBview is contained in the CAN SDK, which is distributed with the **esd**-CAN-CD or can be downloaded from the esd-homepage (www.esd-electronics-com). At the installation of the SDK (Software Development Kit) COBview is automatically installed.
Start the SDK-installation file `Can_sdk\setup.exe` on the **esd**-CAN-CD and carry out the installation.

If not defined at the installation differently, after successful installation COBview can be started under Windows by selecting the menu items *Start / Program files / CAN / CANopen Viewer*.

You can open the program window several times in order to analyze various nodes in the network simultaneously, for instance.

2. Functions of the Interface Elements

2.1 Display of the CANopen ObjectView Window

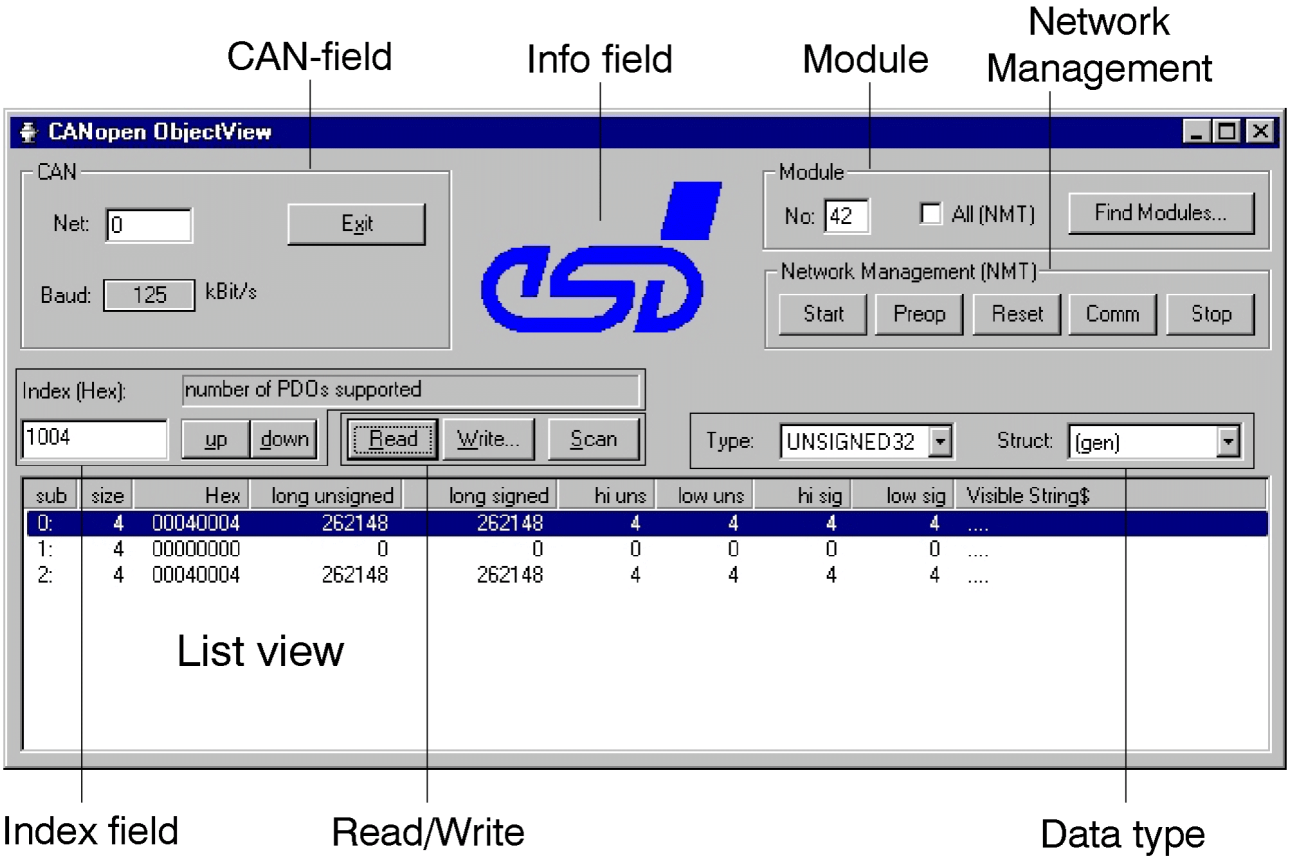
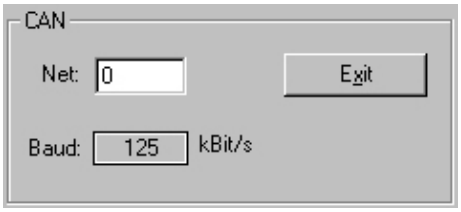


Fig. 1: COBview user interface

2.2 Description of Buttons

CAN-field



Net:
Input box *Net:* please enter the esd-network number of the module used here.

Baud:
In order to set the baud rate click the button next to *Baud:*
If no baud rate was selected, the button indicates *Set...*
If there was already a preset baud rate, e.g. from another application in this network, the button indicates the value of the entered baud rate in kBit/s or the entered index, accordingly.

The input window shown on the right is opened. Here you can enter the baud rate of the CAN network.
The baud rate can be either entered in kBit/s or as index, according to the following table. After confirming the entered value with *OK* it is automatically shown in kBit/s or accordingly indicated as index.

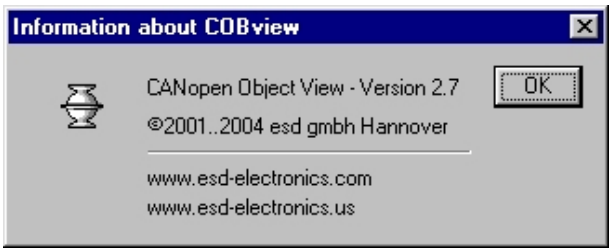
Index	0	1	2	3	4	5	6	7	8	9	n.i.	n.i.	n.i.	n.i.
Baud rate	1000	666	500	333	250	166	125	100	66	50	33	20	12	10

n.i. ... (no index) this baud rate cannot be selected via the index.

Table 1: Index/baud rate

Exit
Via this button the application is closed.
The current settings in the program window are automatically saved when exiting and are displayed again at the following start.
You can open several program windows simultaneously, however, only the values of the program window closed last will be saved.
The program can also be exited via the key combinations *Alt + x* or *Alt + F4*.

Info field



By clicking the esd logo in the program window a window is opened which contains information about COBview. You can close the window by confirming via *OK*.

Module field

No:

The input box *Module No* contains the module number. It can be directly entered or selected via *Find Modules*. If *All(NMT)* is active, *No:* is set to '0', i.e. the NMT-functions are executed for all modules.

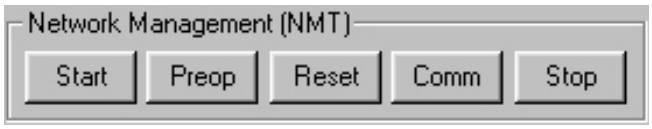
Find Modules...

By clicking the button *Find Modules..* a list of all found modules is automatically generated under list view. By clicking a module in list view the desired module is then selected and the module number is entered under *No:*.

All(NMT)

If *All(NMT)* is not activated, all NMT-functions are only executed for the module number of the selected module.
If *All(NMT)* is activated, the NMT-functions are executed for all modules. Under *Module No.* the value '0' is entered, then.

NMT-Management NMT



In this field basic NMT-functions can be executed.
If *All (NMT)* is activated in the *Module* field, the commands are valid for all modules in the network. If *All (NMT)* is not activated, the commands are only valid for the module selected under *Module No:*.
Please also refer to the diagram on status changes in accordance with CANopen for an explanation of CANopen modes (see page 11).

Start

This button switches the module or modules to CANopen status *Operational*.

Preop

This button switches the module or modules to CANopen status *Pre-Operational*.

Reset

Via *Reset* the module or modules is/are set to CANopen initialization status *Reset Application*. Then the module automatically gets into *Pre-Operational* status via *Reset Communication*.

Comm

Comm sets the module or modules to CANopen initialization status *Reset_Communication*. Then the module automatically gets into *Pre-Operational* status.

Stop

This button sets the module or modules to CANopen status *Stopped.*
In *Stopped* status the module cannot be listed under *Find Modules...* .

Status Changes in Accordance with CANopen DS-301

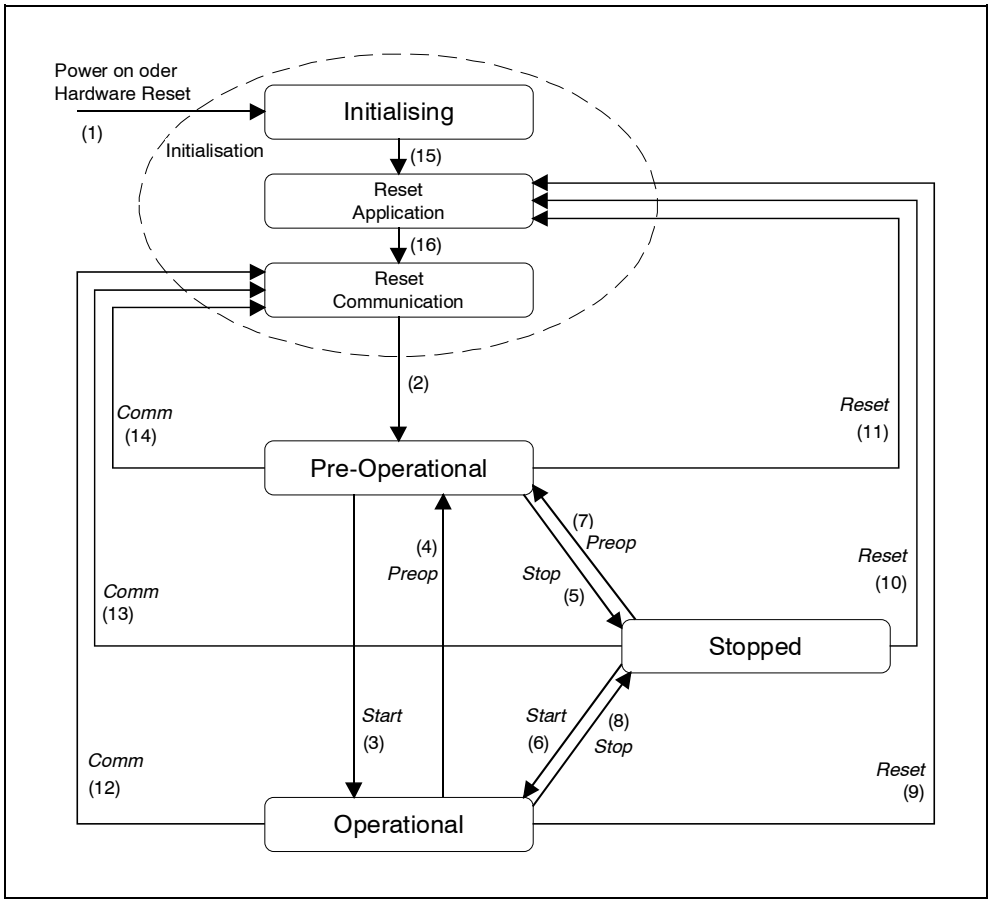


Fig. 2: Status diagram of a module according to CANopen standard DS-301

Status change	Button	Meaning
(1)	-	Initialization is automatically started following 'power on'
(2)	-	Initialization is finished, module is automatically set to *Pre-Operational* mode
(3), (6)	*Start*	**Start**_Remote_Node message,
(4), (7)	*Preop*	Enter **Pre-Op**erational_State message,
(5), (8)	*Stop*	**Stop**_Remote_Node message,
(9), (10), (11)	*Reset*	**Reset**_Node message,
(12), (13), (14)	*Comm*	Reset_**Comm**unication message,
(15)	-	Initialization finished, module is automatically set to *Reset Application*
(16)	-	Reset_Application finished, module is automatically set to *Rest Communication*.

Table 2: Status changes in accordance with CANopen standard DS-301

Index field

Index (Hex)

You can either enter the index of the desired object directly into the lower input box or select it via the two buttons *up* and *down*. If the index is entered directly, the entry has to be confirmed by pressing the 'enter' key.

In the display panel above the name of the selected object is shown for some indices in the 'thousand' range (1000_h-$1FFF_h$, DS-301 range), this applies only for standard 'thousand' objects, however.

When the program is started the index is set to 1000_h by default.

up / down

By clicking the *up* button the index (Hex) next to it is raised by '1' each time.

By clicking the *down* button the index (Hex) next to it is lowered by '1' each time.

Every time *up/down* is clicked *Read* is automatically executed. This deletes the display of the previous index from the list view and all subindices of the new index are shown in the list view. The index can also be increased by entering the key combinations Alt + *u* and decreased via *Alt +d*.

Read/Write

Read

After clicking the *Read* button the program reads all subindices (0-255) of the selected index until the first error is returned by the module. The read subindices are listed under list view.

If a subindex is not assigned, this will be interpreted as error and the subindex and all following subindices will not be read further and will not be displayed.

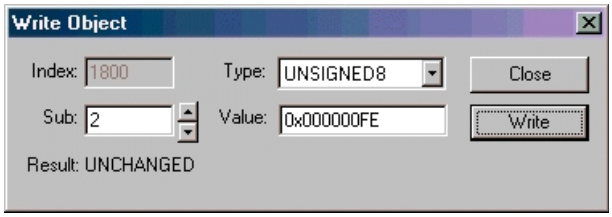
If a time out occurs during readout of a subindex, this will be shown under list view and readout will be aborted.

Readout of the subindices can also be started via the key combination *Alt + r*.

Via the ESC-key *Read* can be aborted. The column Visible String under list view then shows the entry: 'ESCAPE PRESSED'.

Write

Click this button to open the *Write* dialog. Via *Write* a selected subindex of the index can be changed. The *Write* dialog can also be opened by double-clicking the desired subindex under list view or the context menu (see page 16).



Index Currently selected index. The index cannot be changed here.

Type Data type *Type* is taken over like it was opened, i.e. as it was defined under list view.
If the data type was not set to the correct type, the write access is incorrect. The values cannot be written then and the module returns an error message.

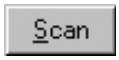
Sub *Sub* shows the subindex which is to be written. This value is also preset.

Value Here you can enter the value which is to be written into the subindex. Numerical values are written either in a decimal or a hexadecimal (with 0x) form. With data type *String* text can be entered under *Value*.

Close The *Close* button closes the dialog window without invoking the *Write* function.

Write Starts the *Write* function. The value defined under *Value* is entered via SDO into the subindex *Sub* of the displayed index. A message (*Result: Changed*) whether the write access was successful is returned. The window then remains open and further subindices can be written.

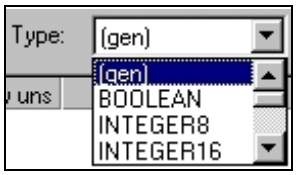
Scan



When clicking the *Scan* button the program reads out all subindices (0-255) of the selected index. The read subindices are listed under list view. If a subindex was not assigned, this is interpreted as error and will be shown under list view for this subindex. The readout of the following subindices will be continued. Only a time out results in an abort. The subindices can also be scanned via the key combination *Alt + s*.
Via the ESC-key *Scan* can be aborted. The column Visible String under list view then shows the entry: ‘ESCAPE PRESSED’.

Data type

Type:

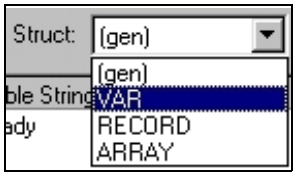


In this selection box you can assign a data type to a subindex selected under list view. Not all data types shown in the selection list are being supported, however.
According to the data length *size* readout by the module a data type is assigned to the subindex at read-in.

size	data type
1	unsigned8
2	unsigned16
4	unsigned32
8	unsigned64
3, 5, 6, 7, >8	Visible String

Table 3: Data type

Struct:



Via *Structure* the number of read-in subindices can be limited.

Struct:	Meaning
(gen)	general, via *Read* all subindices are read-in until the first error and displayed under list view. Via *Scan* all subindices are listed
VAR	only Subindex 0 is being read
RECORD/ ARRAY	Only as many entries are read-in as reported by the module under subindex 0. If an error occurred before, read-in is aborted with *Read*, with *Scan* read-in is continued, however.

Table 4: Data structure

List View

List view contains a list of read subindices or, when *Find Modules..* was selected, a list of modules found. Below, only the list of read subindices will be described.

sub	size	Hex	long unsigned	long signed	hi uns	low uns	hi sig	low sig	Visible String$
0:	4	00040004	262148	262148	4	4	4	4	....
1:	4	00000000	0	0	0	0	0	0	....
2:	4	00040004	262148	262148	4	4	4	4	....

sub

This line shows the respective subindex in decimal form.

size

This line contains the data lengths reported by the module in bytes. If the data length was not reported, size is automatically set to ‘4’ (unsigned32).

size	data type
1	unsigned8
2	unsigned16
4	unsigned32
8	unsigned64
3, 5, 6, 7, >8	Visible String

Table 5: Data type

Hex

Display of data in hexadecimal form (xxxx yyyy, accordingly xxxx - high, yyyy - low).
Read data is interpreted as hexadecimal numerical value and displayed. The order of bytes of the CANopen standard is accordingly turned around for data types.
If the data type is Visible String, the individual bytes are shown in hexadecimal form, divided by points.

long unsigned / long signed

Data is interpreted as *Long* data and shown interpreted without sign under *long unsigned* and with sign under *long signed*.
The complete number is always shown in decimal form.

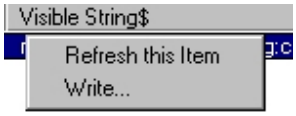
hi uns / low uns / hi sig / low sig

The data is divided in the centre and the higher and lower part of it is interpreted separately. The data is given as unsigned values ***high unsigned, low unsigned*** and signed values, ***high signed*** and ***low signed***. 1-byte values (8-bit values) are not divided, values under high are then '0'.

Visible String

Visible String shows a text interpretation of data. When displaying String no numerical values are interpreted.

Refresh



By clicking a marked list entry with the right mouse key this context menu is opened. This function is not available in the list of found modules via *Find Modules*.

Refresh this Item
The selected subindex is refreshed.

Write The *Write*-dialog is opened (see also page 13).