CANalyzer.CANopen, Version 7.2

CANalyzer Extension for CANopen Users

CANopen® is an open, CAN-based communication protocol. Its wide range of application includes areas such as transport and control technology, measurement systems, medical technology, railway technology, maritime applications, and much more – in short everywhere a great amount of flexibility and rapid data transmission are in demand. The specification drew from the experience of several component manufacturers and users, resulting in a well established standard that is maintained by the CAN in Automation (CiA®) user organization.

Features and Advantages

CANalyzer's powerful functionality together with the functional expansions for CANopen gives the user a tool that can be used both in the development of CANopen systems as well as their startup and maintenance. CANalyzer.CANopen Option makes high-performance, CANopen-specific functions for analysis, simulation, testing, and configuration available to the developer. This is a prerequisite for reliable and efficient interaction with CANopen networks.

Functions

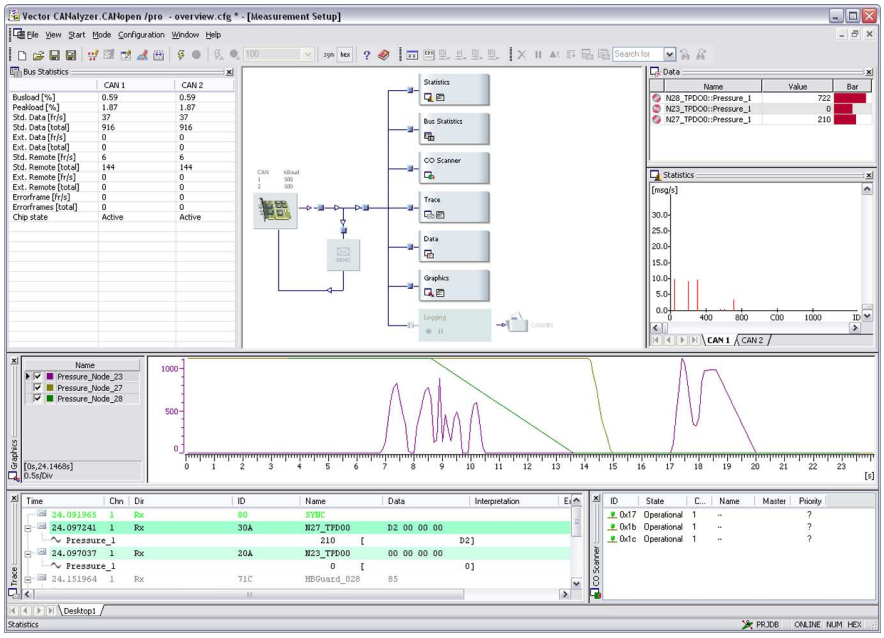
CANalyzer.CANopen extends the functionality of the CANalyzer to include:

- Protocol-specific representation in the Trace Window
- Protocol monitoring
- Graphical representation of devices
- Network management, guarding, heartbeat
- Setting the baud rate and node ID via Layer Setting Services
- Use of the CiA standard file formats EDS/DCF
- Configuration of CANopen devices

These functions will be described below in detail.

Application Areas

The primary application for the .CANopen Option for CANalyzer is the development of CANopen systems. In this environment the tool is primarily used to analyze and stimulate the CANopen-specific message traffic on one or more CAN buses. Additional areas are diagnostics, testing, service, and start-up.



CANalyzer.CANopen enables CANopen-specific interpretation of data in the Trace Window

Hardware Interfaces
CANalyzer.CANopen is available for all current hardware platforms of Vector.

Communication Monitoring
In a Trace Window the CAN message traffic is displayed while simultaneously interpreting the protocol information it contains. The user not only sees the service that is currently being executed, but can also see all relevant service parameters at a glance. This information is displayed in clear text. To simplify analysis, CANopen services are distinguished by different font and background colors. It gives the user a quick overview of the chronological order of individual protocol sequences for the observed CANopen services thereby making it significantly easier to localize errors in a real system.

Protocol Monitoring
CANalyzer.CANopen also monitors for the correctness of individual protocol sequences. This involves detection of whether the protocol information contained within a protocol sequence is properly entered in the CAN message.

Analysis
Not only is the Trace Window provided for the analysis of CANopen data traffic, but also the familiar Data and Statistics Windows of the standard version of CANalyzer.
The CANopen Scanner evaluates CAN messages and shows the active nodes in a list. Other node-specific information is also output, such as the node state and device name.

"CANopen" Interactive Generator Block
Dialog-based creation of CANopen-specific message sequences is offered. From a project-specific list of existing messages, the user selects and configures the desired messages (PDOs, SDOs, etc.) and assembles them into a sequence. The sequence can be replayed once or periodically. This means that message sequences can be sent to the connected devices as often as desired.

Programming
By inserting programmable function blocks into the measurement setup, the user can extend the CANalyzer functionality in virtually any way. This method allows the user to create CANopen-specific test programs or simple node simulations.

New functions of Version 7.2:

Color highlighting of CANopen services accelerates analysis

> Bus analysis is simplified considerably by using separate colors for different CANopen services in the Trace Window. A font and background color may be defined for each category to recognize messages easier and faster. Message categories include: SDOs, PDOs, EMCY-, SYNC- and NMT messages. CAN messages that are part of a SDO protocol, for example, are detected and shown in the appropriate color.

Optimized Network Management in CANalyzer

> NMT commands are now sent directly from the CANalyzer toolbar. You can specify whether the command should be sent globally or to a specific node here.

Signal support provided in device access

> Per CiA311, EDS files can be saved in XML format. One aspect of this standard is that it permits subdividing an object's value into signals. Accordingly, a digital 8-bit input value can be described by 8 separate signals. In accessing a device it is possible to perform specific manipulations and display object values via signals, provided that they have been described in the EDS file.

Node-specific saving of databases improves organization

> Besides the existing global saving of databases, you can now also save them node-specifically. In node-specific saving, a separate CAN database is created for each node. It contains all messages that can be assigned to a node. This leads to considerably better organization, making it easier to select messages or signals.

Graphical Representation

The individual devices of a CANopen network are shown in a summary graphic that shows each device's name, Node-ID, and optionally a bitmap. They can be combined into groups, for example to represent their functional relationship to one another.

It is easy to read-in a network that already exists. This involves reading objects from the object dictionary and comparing them to existing EDS files. The assignment of an EDS file in a real device in the network is performed automatically. If no EDS is available the objects existing in the object dictionary are found and displayed according to their device type. Modular devices are also supported.

Object Directory

The object directory for an individual device is shown in a tree structure that is structured as a function of user inputs. The objects to be shown are taken from the EDS file for the relevant device. Attributes are shown for each individual object in an easy to read format. It is easy to read-out and modify device parameters that are mapped into a device by object dictionary entries. This is how the user can configure necessary settings in a device. In the modification of PDO parameters the access dialog considers the access order specified by the DS301 communications profile.

If no EDS file exists for a device, it is still possible to access the object dictionary by a special dialog. Changes to device parameters are stored separately for each device in a Device Configuration File (DCF).

Creating and changing objects:

Unfortunately it is often the case in practice that existing EDS files for devices are incomplete. CANalyzer.CANopen therefore makes it possible to create new objects or to assign the correct attributes to existing objects. However such changes do not modify the original EDS file, rather only the DCF.

Checking of EDS files and configured PDOs:

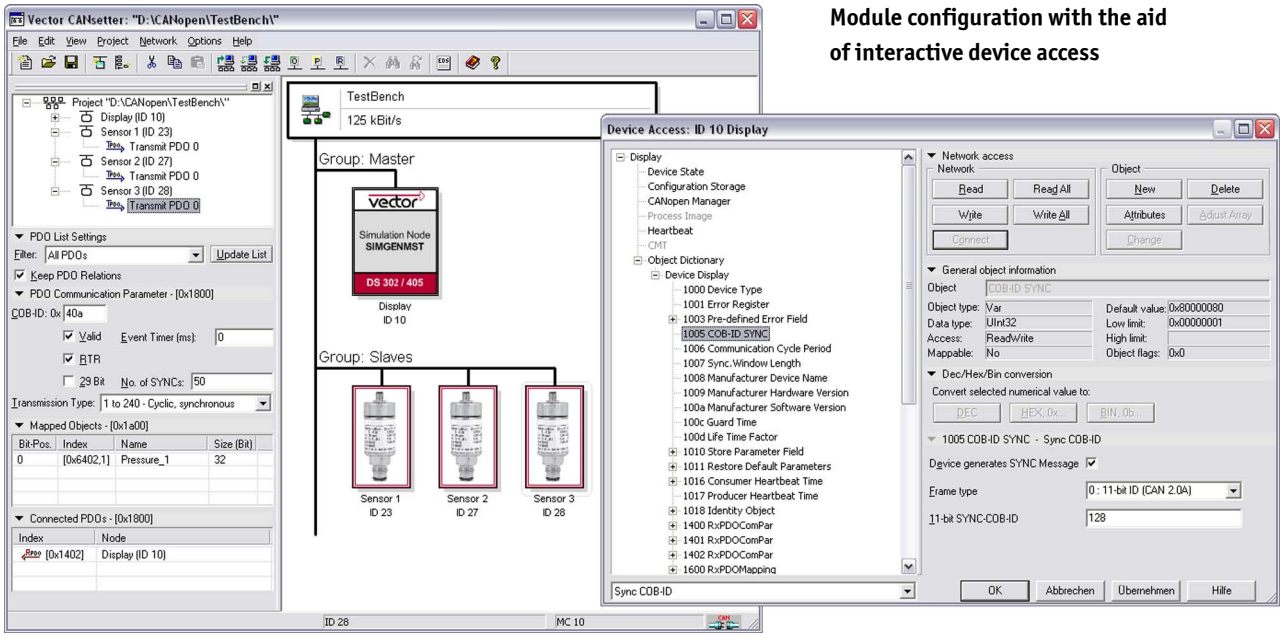
EDS files can be tested with the built-in EDS Checker. This is the same test program that is also used for the CANopen Conformance Test. Additionally, a check is also made to determine whether PDOs can be configured as they are specified by attributes in the object dictionary.

SDO Client:

Access to the object dictionary is generally performed by the SDO (Service Data Object). Besides the "Expedited" transfer mode, CANalyzer.CANopen also supports "Segmented" and "Block" modes. Objects of any size can be transmitted by SDO.

Network Management

The communication states of connected devices can be influenced by certain NMT commands. Guarding or Heartbeat messages are used to evaluate a node's state.



Module configuration with the aid of interactive device access

Training

As part of our training program, we offer a range of classes and workshops on CANopen in our classrooms and on-site at our customers. For more information and the dates of our training courses, please visit our homepage on the Internet at:
www.vector-academy.com

Configuration Saving in the Network

CANalyzer.CANopen supports central saving of all configuration data in a CANopen Manager. This involves sending the configuration data from the individual DCFs to the object dictionary of the CANopen Manager. The "Concise Format" defined by CiA e.V. is used for this. When the CANopen system is started the CANopen Manager then uses these configuration data to configure each individually connected node.

For CANopen devices that save changes internally in the object dictionary, it is possible to work with the predefined memory commands "Store" and "Restore". These commands cause a CANopen device to save the configuration and restore default values, respectively.

Layer Setting Services

Devices in which the baud rate and Node-ID can only be configured by software are also supported by the integrated LSS services.

Standard Formats

CANalyzer.CANopen uses the standardized file formats EDS and DCF to store data. Simple data exchange with any other CANopen tool such as ProCANopen or CANoe.CANopen is guaranteed. Both the conventional INI format and XML format specified in CiA311 are supported.

Add-on Functions

Dialog-based configuration of CANopen safety systems per CiA304 (framework for safety-relevant communication) is offered, provided that a device has SRDO (safety relevant PDO).

Moreover, the following application profiles are supported:

- Application profile for building door control (CiA 416)
- Application profile for special-purpose car add-on devices (CiA 447)
- Application profile for lift control systems (CiA 417)
- Device profile for battery modules and battery chargers (CiA 418, CiA 419)
- FireCAN (not specified by CiA)