

canAnalyser

The universal CAN tool for development, testing and service

The canAnalyser version 2 from IXXAT is an optimized, enhanced version of the universal analysis tool for development, testing and maintenance of CAN networks. The new canAnalyser has been thoroughly revised in terms of flexibility and extendibility.

Flexibility through modularity

The main component of the canAnalyser is a central communication server, which acts as an interface between the CAN bus (CAN board) and the analysis modules. The actual analysis functions

are provided by separate function modules, which can be connected to the communication server in any combination.

In addition to the modules provided by IXXAT, specific modules created by the user can also be used. Due to the open programming interface of the canAnalyser, customized functionalities in the form of individual modules can easily be integrated. This unique concept guarantees almost unlimited extendibility.

The canAnalyser allows to statistically analyze and graphically display message traffic and bus load (statistic module).

In addition, the canAnalyser enables monitoring of cycle times and a clear display of modified data contents.

Further powerful features are the interpretation (signal module) and graphic display of message contents on a time axis (graphic module), the simultaneous execution of several instances of a module and the integration of own modules and programs by means of an open programming interface (.NET API) or the Scripting Host.

The standard version of the canAnalyser also allows parallel analysis of two CAN systems (multi-line mode). The time-synchronous analysis of the systems is enabled by means of graphic markers.

	canAnalyser	canAnalyser lite
Receive Module	✓	✓
Transmit Module	✓	✓
Trace Module	✓	✓
Sequencer Module	✓	✓
Graphic Module	✓	
Statistic Module	✓	✓
Signal Module	✓	
CANopen Module	○	○
DeviceNet Module	○	○
Multi Channel Capability	✓	
Multiple Module Instances	✓	
Open Programming Interface	✓	
Scripting Host	✓	

Modules and functions of the canAnalyser
○ = optional

Functions and versions

The canAnalyser offers a wide range of functions and covers various areas of applications. It allows online monitoring of the bus traffic (receive module), transmission of single-shot or cyclic messages (transmit module) and complete message sequences (sequencer module), tracing of CAN messages (Trace module) with many different configurable trigger conditions and offline analysis of the recorded data with all available modules.

Configuration and operation

The central element for configuration is the control panel, with which the desired CAN interface is selected for analysis and the analysis modules are assigned to the various CAN buses.

The CAN baudrate can be individually set for each CAN controller. In addition to the standard baudrates, individual entries are also possible. The individual modules are started and managed via the control panel.

In addition to the CAN controller parameters, the control panel also stores the window arrangement and

all other module settings in a central configuration file. Furthermore, status or warning messages are recorded centrally for all modules.

Scripting Host

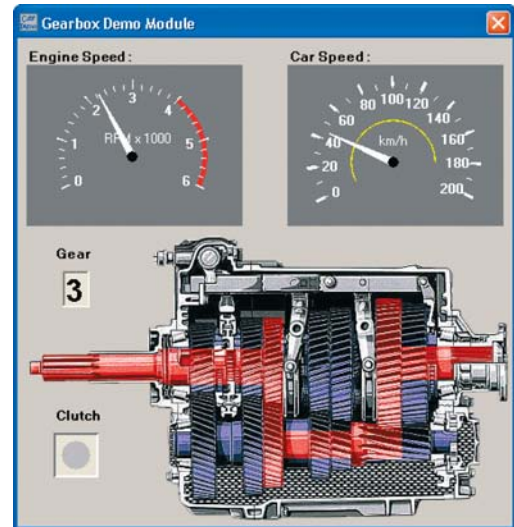
The Scripting Host provides the canAnalyser user with a powerful interface that combines the advantages of graphic Windows programs with the flexibility of scripts.

By using the Scripting Host the canAnalyser can be quickly and easily adapted to specific measuring and analysis tasks. This allows to simulate devices and protocols or to test existing devices in the

simulated restbus and to put them into operation. Specific test environments can be simply created using any Windows interface components.

The Scripting Host supports the standard script languages C#, Visual Basic-Script and Java. The incorporation of DLLs also enables the integration of further modules.

The Scripting Host can be used to manage an unlimited number of scripts, which are started manually or automatically.



Example: Gear simulation based on the Scripting Host



Scripting Host

Programmability

Due to the open programming interface, the canAnalyser can be extended by the user's own modules or user interfaces.

With common Windows development systems (e.g. Visual Studio .NET, Delphi), new, independent modules can be developed and added to the canAnalyser. It is possible, for example, to create user interfaces for own systems or for certain devices or tools with system-specific analysis functions.

Scripting Host Example

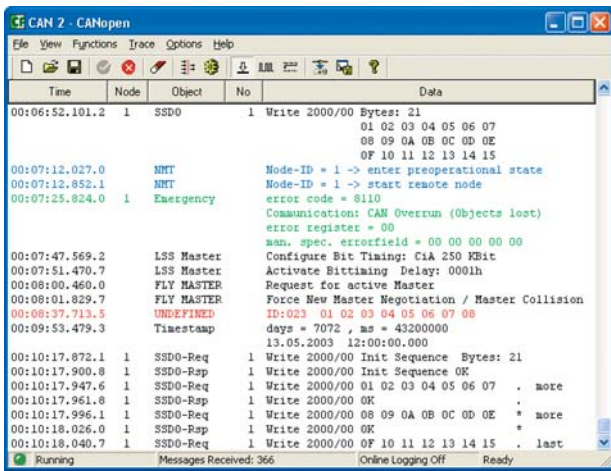
```

/*****
Function:      OnMessageReceived
Description:   Handler function of IMbaMessageSource. MessageReceived event.
                This event is fired for every received message.
Arguments:    sender [in] The source of the event.
                e [in] A MessageReceivedEventArgs that contains the received
                message.
Results:      -

*****/
private void OnMessageReceived(object sender, MessageReceivedEventArgs e)
{
    IMbaCanDataMessage rxDataMessage = e.Message as IMbaCanDataMessage;
    if (null != rxDataMessage)
    {
        if (0x100 == rxDataMessage.Identifier)
        {
            MbaCanDataMessage gatewayMessage = new MbaCanDataMessage(rxDataMessage);
            gatewayMessage.Identifier = 0x200;
            can1Transmitter.PostMessageObj(gatewayMessage);
        }
    }
}

```

Function: After the reception of a message with the Identifier 0x100, a message with the Identifier 0x200 is generated and the data bytes are copied to the new message.



CANopen Module

CANopen and DeviceNet

Higher layer CAN protocols can be analyzed via supplementary modules such as the CANopen or DeviceNet analysis modules.

The **CANopen module** interprets all received CAN messages in accordance with the CANopen specifications DS-301, DS-305, DS-401, DS-402, DS-405 and DS-406. The messages are recognized as SDOs, PDOs (also multiplexed), NMT, Emergency, Sync and Timestamp objects and interpreted accordingly.

In addition, the Error Control, LSS and Flying Master protocols are interpreted. These are displayed in plaintext, color-coded according to the message type. Interpretation of the data received is based on the so-called configuration set. This either assigns an EDS/DCF file to each of the 127

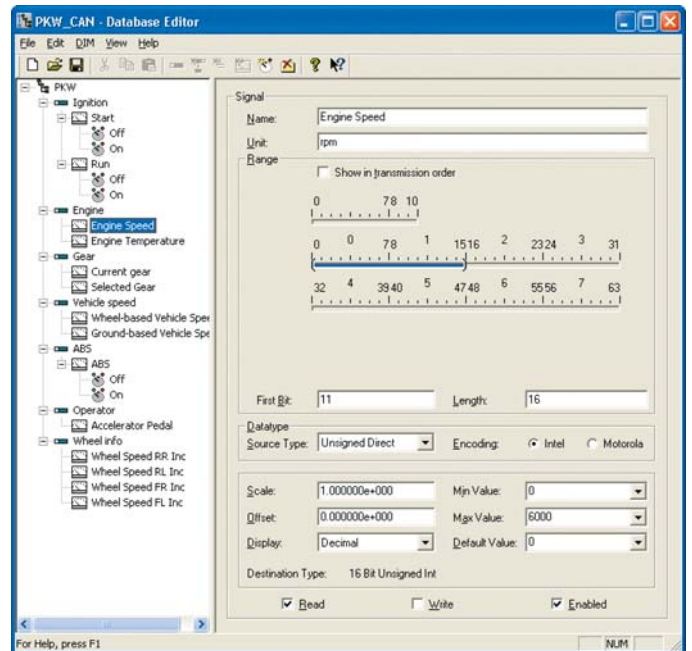
possible network nodes or defines its device profile. The configuration set can be loaded, edited and saved. The profiles used for interpretation are integrated via external text files, whereby the CANopen module can be very easily extended by new profiles. Due to its flexible structure,

the CANopen module can thus be used universally in all CANopen systems.

The **DeviceNet module** interprets all received CAN messages according to their DeviceNet significance and displays the transmitted parameters in plaintext, classified according to message group, MAC-ID and message-ID. The incoming information are divided into unconnected message, explicit message, I/O message, duplicate MAC-ID check message, device heartbeat message, device shutdown message,

offline connection set as well as reserved and invalid CAN messages and their content decoded according to the type of message.

Interpretation of the received data is also based on the configuration set. It defines the message format for each node (8/8, 8/16, 16/8 or 16/16) as well as the basic interpretation texts to be displayed and edited. Due to its flexible filter configuration and the comprehensive interpretation basis, the DeviceNet module is suitable for all DeviceNet systems.



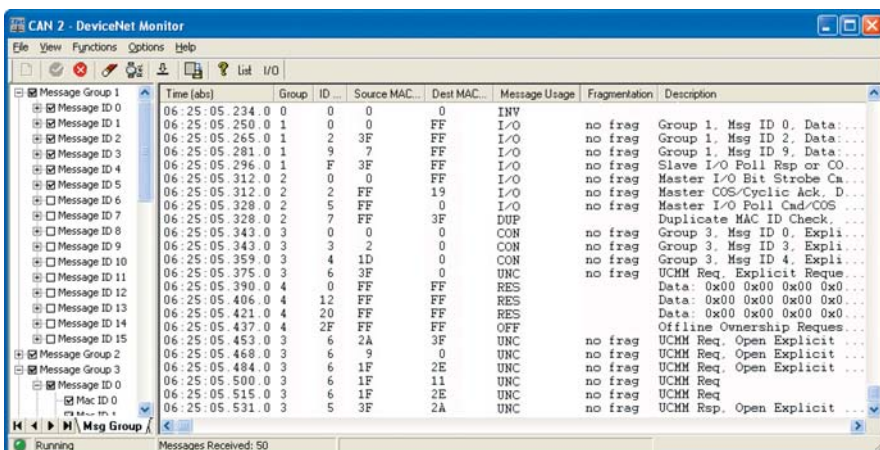
Database Editor

Support of project databases

The interpretation and symbolic display of the data transmitted in the CAN message is based on a project database, in which a CAN message is assigned a name according to its identifier.

The message can contain up to 64 individual data objects (signals, max. 64 bit data per CAN message), with support for multiplexed messages.

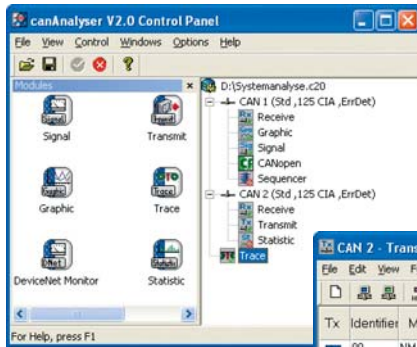
... continued on the backside



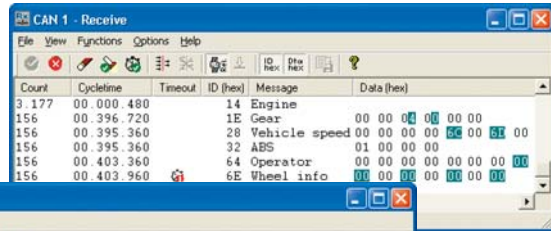
DeviceNet Module

Standard Version

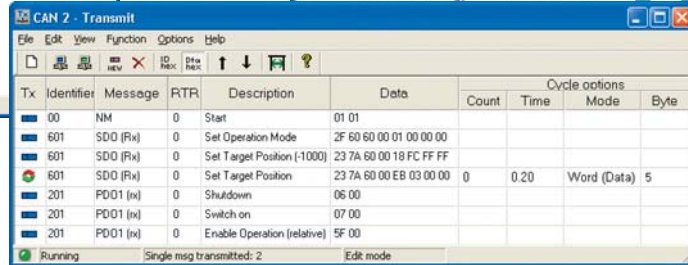
lite Version



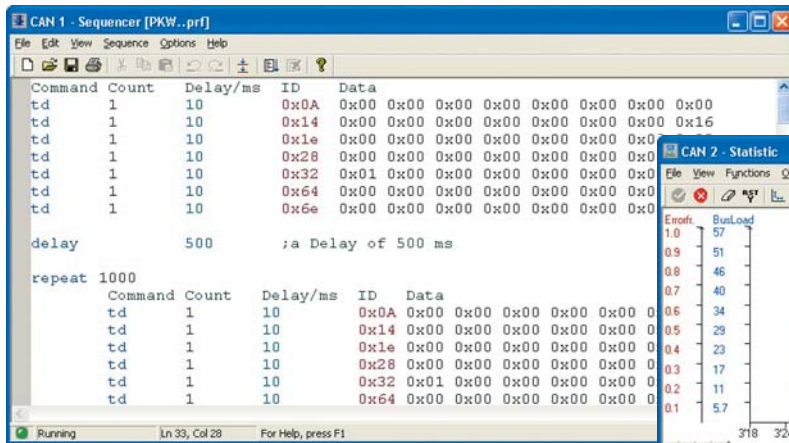
Control Panel



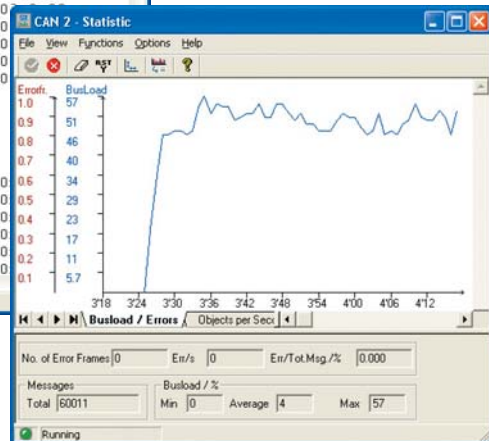
Receive Module



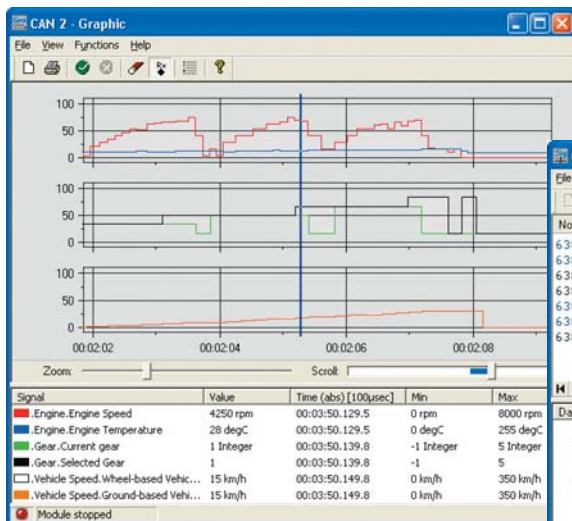
Transmit Module



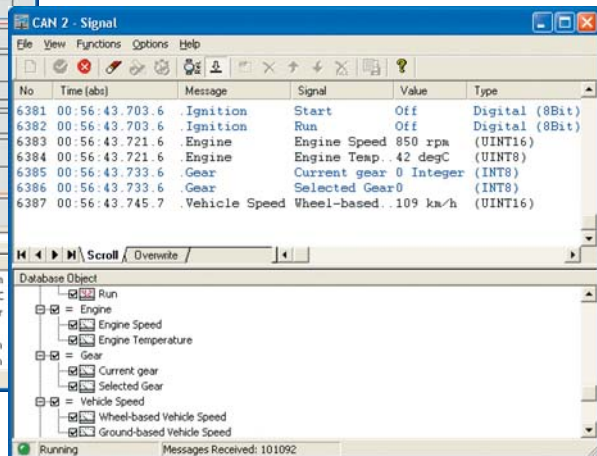
Sequencer Module



Statistic Module



Graphic Module



Signal Module

Overview of functions

- ☑ Transmission of individual CAN messages using transmit lists
- ☑ Transmission of message sequences with flexible control commands (repeat, delay, etc.)
- ☑ Display of received messages in scroll and overwrite mode
- ☑ Interpretation of data contents and display as plaintext/line diagram
- ☑ Support of 11 or 29 bit identifiers (CAN 2.0A / 2.0B)
- ☑ Timestamp for receive objects with a resolution of 100µsec
- ☑ Support of high-speed/low-speed CAN (depending on CAN interface)
- ☑ Passive mode (no transmission of Ack-bit and Errorframes, therefore no influence on the CAN system)
- ☑ Detection/display of Errorframes
- ☑ Support of project databases
- ☑ Online trace on hard disk, versatile trigger conditions (pre-triggering/post-triggering, etc.)
- ☑ Off-line analysis of the recorded messages by using all canAnalyser modules
- ☑ Flow control for off-line analysis (break points, start/stop)
- ☑ Support of all on-board CAN controllers (multi-line mode)
- ☑ Time-synchronous analysis of several buses
- ☑ Automatic cycle time monitoring
- ☑ Display of modified data
- ☑ Cyclic real-time transmission of CAN messages on board
- ☑ Multiple execution of modules at the same time
- ☑ Programming interface for customized extensions
- ☑ Scripting Host to set-up test and node emulation applications
- ☑ Support of all CAN interface boards supplied by IXXAT

... continued from the inside

The file format used is XML. In addition import filters for CANdb and for CANopen DCF files are provided.

Each signal can be interpreted as an analog, digital or string value: Analog signal values are defined by their bit position and length in the CAN message, their data type and coding (Intel/ Motorola), scaling and offset, value range and physical unit. With digital variables, the values are given symbolic names, which are then displayed as plaintext. Each CAN bus can be assigned its own database.

In addition to the signal and graphic modules, which display received messages in plaintext with the aid of the project database, other modules, such as the transmit and receive modules, also use the data base to assign symbolic names to the messages.

Supported CAN hardware

The canAnalyser can be used with all IXXAT CAN interface boards. IXXAT offers interface boards for all conventional PC interfaces, e.g. USB, PCI, Ethernet, Bluetooth, etc.



A selection of the CAN interfaces offered by IXXAT, e.g. for PCI, USB and Bluetooth

System requirements

- PC with Windows 2000, XP
- IXXAT CAN interface board with SJA1000 CAN controller and CAN driver VCI, version 2.16 or higher

canAnalyser.com

Further information, such as FAQs, a detailed tutorial, a demo version and support are provided on the canAnalyser product homepage www.cananalyser.com.

