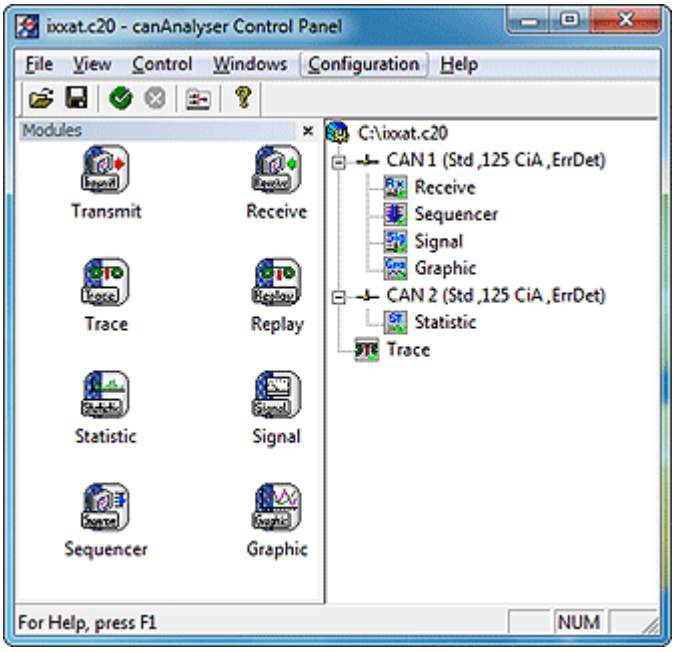
canAnalyser

The canAnalyser is a powerful, versatile tool for the development, testing and servicing of Controller Area Network based systems. The canAnalyser is already used extensively, especially in the area of industrial automation, and, is also being used more and more in the automotive industry.



The software package is based on a modular concept which combines exceptional openness and versatility. Customer specific functions can be easily integrated via an open .NET programming interface in the form of individual modules. By using a powerful CAN interface from IXXAT, the canAnalyser achieves the reception of CAN messages and time oriented buffering even when dealing with very high bus loads and baudrates.

In the standard version the canAnalyser offers functions covering many areas of application:

- Online monitoring of bus traffic
- Transmission of one-off or cyclic messages and entire message sequences
- Parallel monitoring of several CAN buses
- Recording of CAN messages with various trigger conditions
- Static evaluation of the message traffic
- Recording and display of bus load
- Graphic display of message contents over the time axis
- Creation of command controlled message sequences

An integral part of the canAnalyser is the processing of message databases. With this, each CAN identifier can be allocated a message name and the signals transmitted in the data field can be interpreted and displayed as physical parameters in different ways. The canAnalyser also processes the widely used CANdb format. In all modules the relevant message name from the database is displayed in addition to the CAN identifier.

Additional functions are provided by optional modules, such as the protocol specific display of messages of CANopen, DeviceNet or J1939 based systems.

Highlights

- Multilingual software for Windows 2000 /XP / Vista / Windows7 (32/64 bit)
- Support of all CAN interfaces supplied by IXXAT
- Support of 11- and 29-bit identifiers (CAN 2.0A/2.0B)
- Timestamp for receive objects with a resolution of 1 µsec (depending on the used hardware)
- Passive mode (no transmission of Ack bit and error frames, thus no interference of the CAN system by the canAnalyser)
- Detection and display of error frames
- Integrated support of project databases
- Online trace on hard disk with various trigger conditions
- Support of all CAN controllers available on the interface board (multi-line mode)
- Display of the CAN controller status and bus load
- Documented .NET programming interface for extension by user-specific modules
- Execution of Scripts

Functions

Configuration and Operation

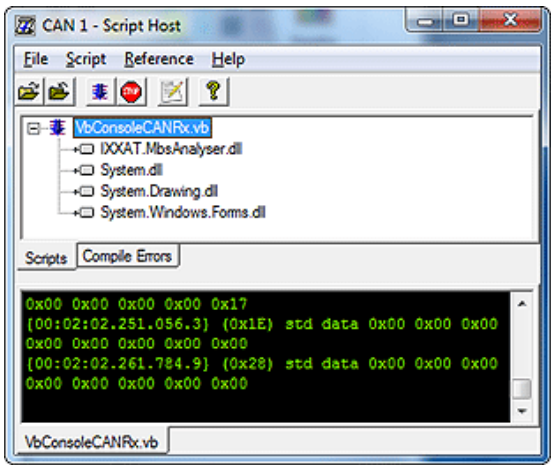
The control panel of the canAnalyser is the central element for configuration. Supported by a Wizard, the required CAN interface is selected and the CAN controller on the board is configured. It also defines which function modules are assigned to the CAN controllers. The configuration is clearly displayed in the form of a tree and the module assignment occurs intuitively by means of drag and drop. Each CAN bus can be assigned a database.

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), new, independent modules can be developed and added to the canAnalyser. It is possible for users to create interfaces for their systems or for certain devices or tools with system specific analysis functions.

Scripting Host

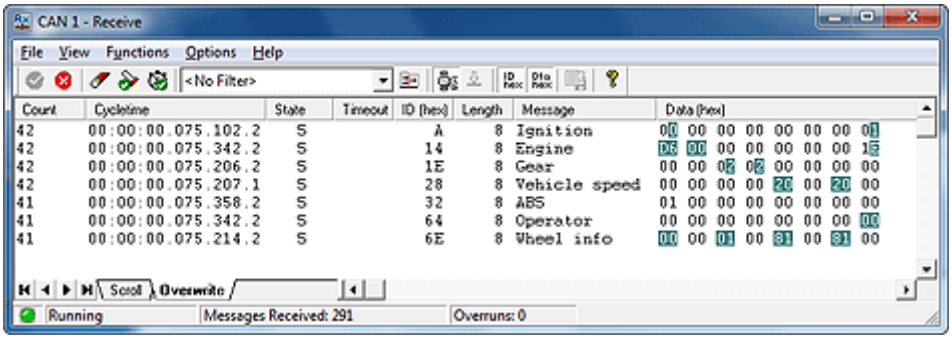
The Scripting Host provides a powerful interface that combines the advantages of graphic Windows programs with the flexibility of scripts. By using the Scripting Host the canAnalyser can be quickly and easily adapted to specific measuring and analysis tasks. This allows the user to simulate devices and protocols or to test existing devices in the simulated restbus and to put them into operation. Specific test environments can be easily created using any Windows interface components. The Scripting Host supports the standard script languages C# and Visual Basic .NET. The incorporation of DLLs also enables the integration of further modules.



Receiving and displaying CAN messages (Receive Module)

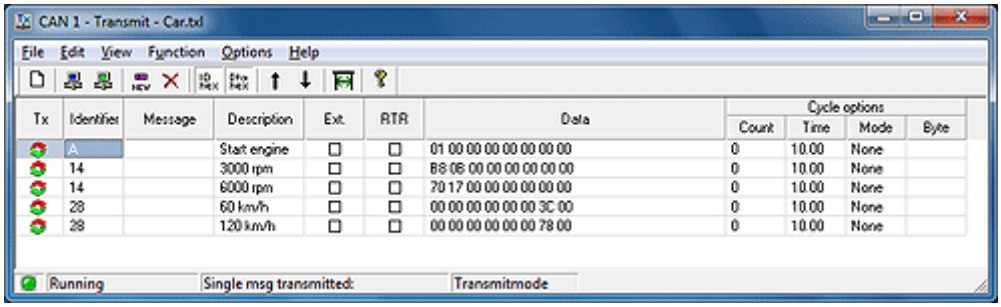
The CAN messages transmitted on the CAN bus can be displayed online in different ways. Either all messages or only certain ones, selected via an adjustable filter, are displayed. Messages can be displayed in two ways. In scroll mode, the messages are displayed together with the time of reception one after the other in a list. This form of display is particularly suitable for monitoring message sequences. On the other hand, in the overwrite mode the messages received are listed according to the identifier and permanently overwritten with the incoming data. Each message is assigned a counter, which displays the commonness of its transmission. Here the altered byte is highlighted in color. With the additional cycle time monitoring, the regularity of reception is observed.

For further monitoring of certain message groups, the receive module can be started multiple times, where each instance can display one section of the CAN data flow.



Transmission of CAN-messages (Transmit Module)

Messages to be transmitted can be arranged by the user in a message table. Individual entries from this table can be transmitted once or cyclically. The table contains both the definition of the message (identifier, data bytes, RTR bit) and a description of the message. The data entry can be either decimal or hexadecimal. For messages to be transmitted cyclically, cycle times of 250 us (depending on the used hardware) to 100 s can be specified. In cyclic transmission mode, identifiers or data contents can be incremented automatically.



Recording CAN messages(Trace Module)

With the trace module all received messages and error frames are recorded directly onto the hard disk. The recording can be started and stopped via the trace control. In addition, trigger conditions for starting and stopping as well as filters for the CAN messages to be recorded can be defined for each bus. A trace can be viewed at any time and can be reloaded into a system offline for analysis by specifically configured analysis modules, or, online with the aid of the sequence module (limited number of messages).

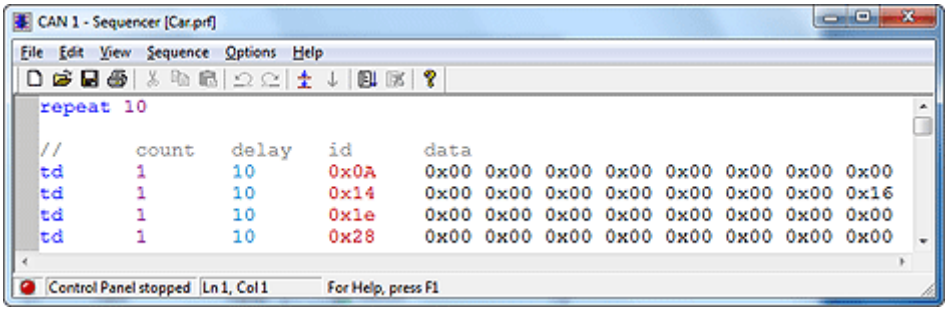
Play-back of Trace Files (Replay Module)

The Replay Module enables the play-back of trace files. In online mode the messages can be send to the CAN network and received via self reception, in offline mode the

messages can be distributed to the connected canAnalyser modules.

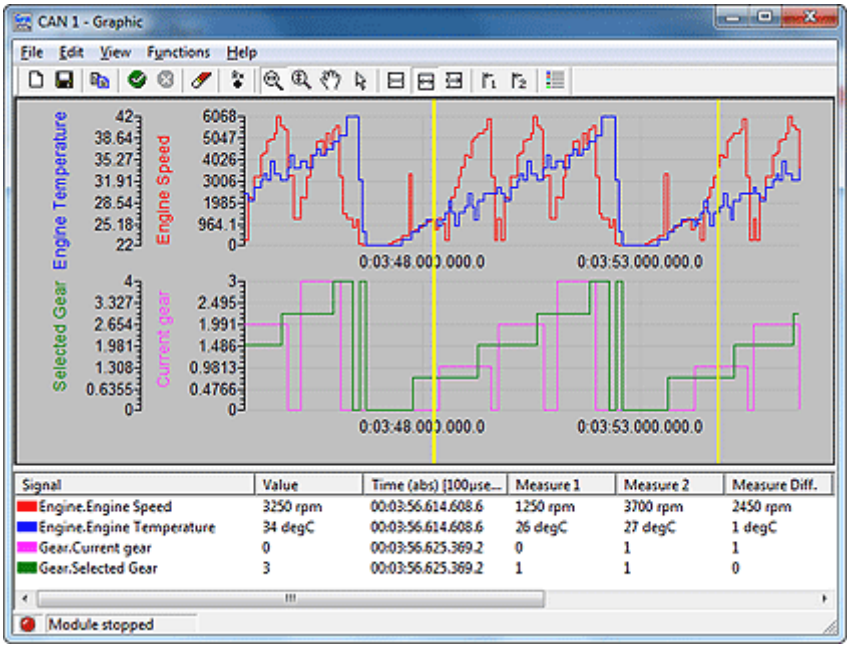
Transmitting message sequences (Sequencer Module)

During the development of CAN devices, functions, protocols and complete systems situations can be tested by transmitting message sequences. Unavailable devices can also be simulated. The message sequences are created with a few easy to learn commands (such as transmitting a message, waiting for a message, pause with specified duration, repeat, user input) via an integrated editor and then executed at the push of a button.



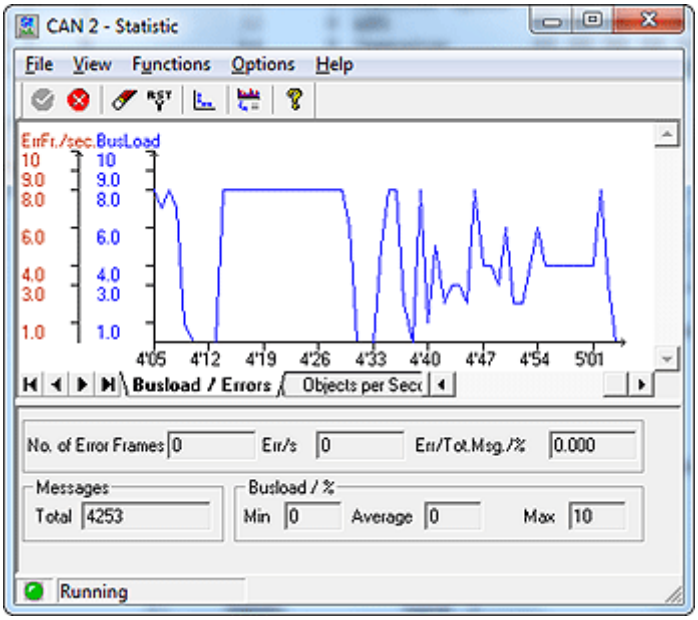
Graphic display of data (Graphic Module)

Data contents of CAN messages are displayed in the graphic module in a window over the time axis. The information to be displayed, such as name and unit is automatically taken from the database allocated to a CAN bus. The data is displayed in real time, where a maximum of 16 signals per graphic window are distributed over up to 4 time axes. In addition, the current value of a signal is displayed numerically. With the aid of a metering bar, specific values can be determined and evaluations carried out.



Statistic analysis of network variables (Statistic Module)

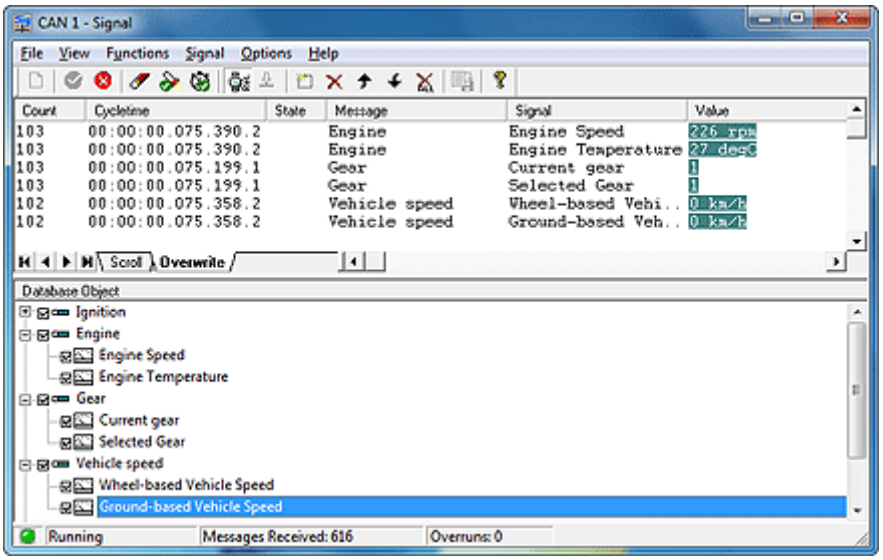
With the statistic module, important variables of a CAN network can be both statistically recorded and displayed graphically. The module determines the current bus load of the system and displays this over the time. In addition to general data such as the number of error frames or the total number of transmitted CAN messages, the frequency of the individual CAN messages is displayed as a histogram.



Symbolic interpretation and display of transmitted process variables (Signal Module)

Beyond the display of the receive module, not only is the identifier of a CAN message with a symbolic name displayed, but the complete contents of the message are also extracted and displayed as signals (physical values) based on the interpretation rules stored in the database.

Whether in scroll mode or in overwrite mode, the signals contained in a CAN message can be displayed with the signal module. Messages and signals can be activated and deactivated individually, i.e., excluded from interpretation.

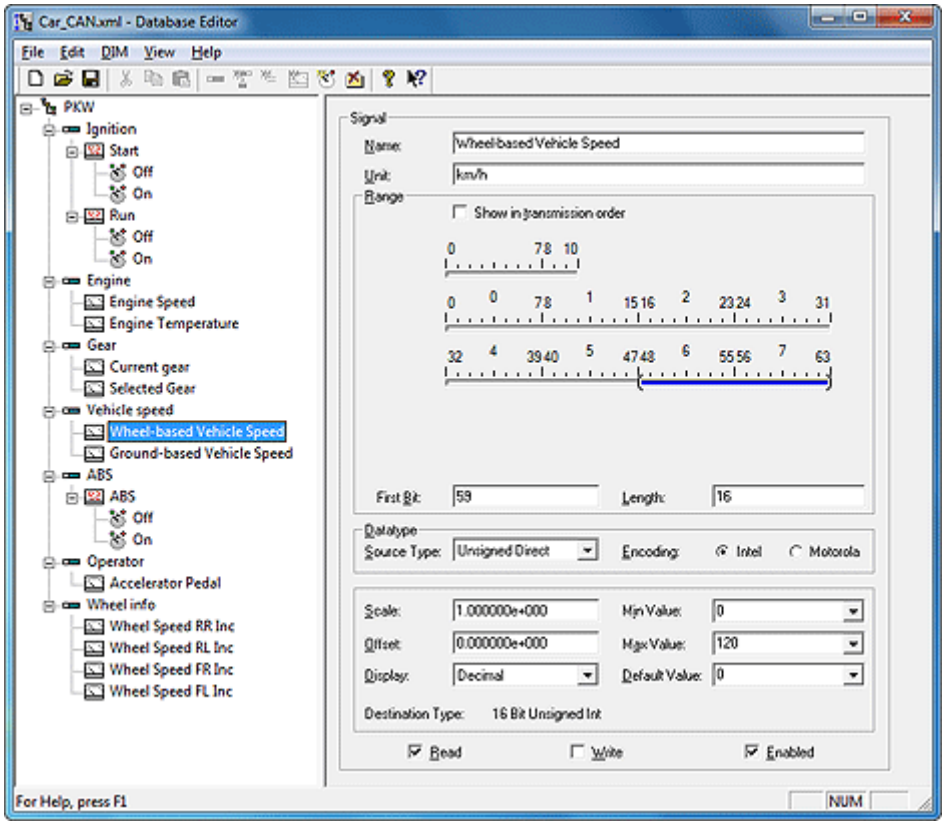


Editor for the project databases

The basis for the interpretation and symbolic display of the data transmitted in the CAN message is the project database. In this, a CAN message is first assigned a name according to its identifier. The message can contain up to 64 individual data (signals). The file format used is XML. In addition, there exists an import filter for the CANdb format.

Each signal can be interpreted as an analog, digital or string value.

The structure of the project database is displayed in a two part window as a hierarchical tree with signal type dependent icons, with separate, clear input masks existing for each area.



Additionally available software

LIN analysis: With the LIN2CAN device it is possible to monitor and transmit LIN messages with the canAnalyser. The LIN2CAN is configured as a gateway and converts the LIN messages into CAN messages and vice versa.

Order number

1.02.0133.00000	canAnalyser

canAnalyser Bundles Order number

1.03.0133.00001	canAnalyser, **CANopen Module**, USB-to-CAN II Industrial galv. decoupled
1.03.0133.00002	canAnalyser, **DeviceNet Module**, USB-to-CAN II Industrial galv. decoupled
1.03.0133.00003	canAnalyser, **SAE J1939 Module**, USB-to-CAN II Industrial galv. decoupled