

# CAN Tutorial

This leaflet contains our tutorial on the CAN bus system along with details of a typical PC to CAN interface using USB and its associated free supporting software.

Also included are details of Explorer our software package that converts an interface into a CAN analyser, logger and control centre.

In this leaflet:

[The CAN Tutorial](#)

[Using CAN](#)

[CAN to USB Interface](#)

[Simple CAN management interface PCAN-View](#)

[CAN DLL for PC based applications](#)

[Explorer – CAN data logging, control and display package](#)



Distributes the Peak System range of CAN interfaces, adapters, I/O modules, data acquisition systems and supporting software.

Follow these web links for more information on:

[Other CAN PC Interfaces](#)

[CAN Data Acquisition Systems](#)

[CAN I/O Modules](#)

[Software Support for FMS](#)

[Cables and Adapters](#)

Most of these items are available from our [Web Shop](#) for next day delivery.

# CAN - a brief tutorial

The CAN bus (Controller Area Networking) was defined in the late 1980 by Bosch, initially for use in automotive applications. It has been found to be very useful in a wide variety distributed industrial systems as it has the following characteristics:

- Uses a single terminated twisted pair cable
- Is multi master
- Maximum Signal frequency used is 1 Mbit/sec
- Length is typically 40M at 1Mbit/sec up to 10KM at 5Kbits/sec
- Has high reliability with extensive error checking
- Typical maximum data rate achievable is 40KBytes/sec
- Maximum latency of high priority message <120  $\mu$ sec at 1Mbit/sec

CAN is unusual in that the entities on the network, called nodes, are not given specific addresses. Instead, it is the messages themselves that have an identifier which also determines the messages' priority. Nodes then depending on their function transmit specific messages and look for specific message. For this reason there is no theoretical limit to the number of nodes although in practice it is ~64.

Two specifications are in use:

- 2.0A sometimes known as Basic or Standard CAN with 11 bit message identifiers which was originally specified to operate at a maximum frequency of 250Kbit/sec - ISO11519.
- 2.0B known as Full CAN or extended frame CAN with 29 bit message identifier which can be used at up to 1Mbit/sec - ISO 11898.

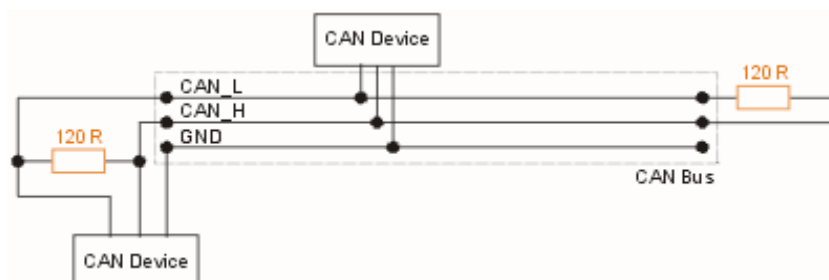
## Nuts and Bolts

From the systems and design viewpoint the detailed management of sending and receiving CAN messages will normally be done by dedicated hardware, on or off chip, (e.g. SJA1000) but an overview of these functions will be useful in order to design, setup and control a CAN system.

## Signal Characteristics

CAN may be implemented over a number of physical media so long as the drivers are open-collector and each node can hear itself and others while transmitting (this is necessary for its message priority and error handling mechanisms). The most common media is a twisted pair 5v differential signal which will allow operations in high noise environments and with the right drivers will work even if one of the wires is open circuit. A number of transceiver chips are available the most popular probably being the Philips 82C251 as well as the TJA1040.

When running Full CAN (ISO 11898-2) at its higher speeds it is necessary to terminate the bus at both ends with 120 Ohms. The resistors are not only there to prevent reflections but also to unload the open collector transceiver drivers. We recommend that you terminate the bus correctly in all circumstances.



## Message formats

The CAN protocol uses a modified version of the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) technique used on Ethernet. Should two messages determine that they are both trying to send at the same time then instead of both backing off and re-trying later as is done with Ethernet, in the CAN scheme, the transmitters detect which message has the highest priority and only the lower priority message gets delayed. This means that a high priority message is sure of getting through.

|     |            |     |         |      |     |     |     |
|-----|------------|-----|---------|------|-----|-----|-----|
| SOF | MESSAGE ID | RTR | CONTROL | DATA | CRC | ACK | EOF |
|-----|------------|-----|---------|------|-----|-----|-----|

## Data Frames

These are the normal message frames used to carry data. They contain the following fields -- this is a simplified description as the controller takes care of the detail which is only of interest to those designing controllers (who should consult the spec)

Start of frame (SOF)

Message Identifier (MID) the Lower the value the Higher the priority of the message  
its length is either 11 or 29 bits long depending on the standard being used

Remote Transmission Request (RTR) = 0 ----- see "Remote Frames" para below for no zero value

Control field (CONTROL) this specifies the number of bytes of data to follow (0-8)

Data Field (DATA) length 0 to 8 bytes

CRC field containing a fifteen bit cyclic redundancy check code

Acknowledge field (ACK) an empty slot which will be filled by any and every node that receives the frame it does NOT say that the node you intended the data for got it, just that at least one node on the whole network got it.

End of Frame (EOF)

The way in which message collision is avoided is that each node as it transmits its MID looks on the bus to see what everyone else is seeing. If it is in conflict with a higher priority message identifier (one with a lower number) then the higher priority messages bit will hold the signal down (a zero bit is said to be dominant) and the lower priority node will stop transmitting.

If you are writing diagnostic code and wish to not "exist" on the network as a node, just to spy on what is happening, then you will need to ensure that the interface you use can be set to a mode where it does not automatically set the ACK bit. The [Peak interfaces](#) and their [Explorer diagnostic package](#) can be set into such a mode.

## Remote Frames

These are frames that are used to request that a particular message be put on the network - of course a node somewhere on the network has to be set up to recognise the request, get the data and put out a Message frame. This mechanism is used in polled networks. The fields are ....

Start of frame (SOF)

Message Identifier (MID) either 11 or 29 bits long depending on the chosen mode.

Remote Transmission Request (RTR) = 1

Control field (CTRL) this specifies the number of bytes of data expected to be returned (0-8).

CRC field containing a fifteen bit cyclic redundancy check code.

Acknowledge field (ACK) an empty slot which will be filled by any and every node that receives the frame it does NOT say that the node you intended the data for got it, just that at least one node on the whole network got it.

End of Frame (EOF)

## Error checking

CAN is a very reliable system with multiple error checks

Stuffing error - a transmitting node inserts a high after five consecutive low bits (and a low after five consecutive high). A receiving node that detects violation will flag a bit stuffing error.

Bit error - A transmitting node always reads back the message as it is sending. If it detects a different bit value on the bus than the one it sent, and the bit is not part of the arbitration field or in the acknowledgement field, an error is detected.

Checksum error - each receiving node checks CAN messages for checksum errors.

Frame error - There are certain predefined bit values that must be transmitted at certain points within any CAN Message Frame. If a receiver detects an invalid bit in one of these positions a Form Error (sometimes also known as a Format Error) will be flagged.

Acknowledgement Error - If a transmitter determines that a message has not been ACKnowledged then an ACK Error is flagged.

## Variants

By defining only the physical and data link levels of the OSI communications model the CAN specification has become the basis for a wide number of industry and manufacture specific variants (and the source of much confusion as all the users will tell you they are using CAN). If you are trying to clarify a CAN systems status the first thing to find out is the transceivers in use - the most common "normal 5v" CAN uses the Philips 82C251 or the TJA1040.

**TJA 1054** is a low power, low speed physical layer that is mostly used in automotive applications. It employs the PCA82C252, TJA1053 or TJA1054 transceivers.

**AU5790** also known as "Single Wire CAN" is a low power, low speed physical layer that is used in automotive applications and an increasing number of industrial applications. It employs the AU5790 transceiver.

**DeviceNet** - Developed for use in industrial process control it is based on the standard Full CAN - ISO 11898-2 5v bus. However DeviceNet rigorously defines the physical interconnect, has a more restrictive transceiver specification, 11 bit identifiers only, allows 125, 250 and 500KBaud operation only and regulates the message content in order to more easily support interoperability of different manufacturers units.

**CANopen** - Also designed with control applications in mind, it is a software standard based on the standard Full CAN - ISO 11898-2 5v bus. It limits the number of nodes to 127 and allocates them IDs. Profiles are specified for each type of device by CiA to simplify using similar units (eg motor drives) from different manufacturers. Some standard network commands are defined that allow modules to be automatically identified and allocated a node ID. The spec also defines a way to handle synchronised data reads and writes as well as providing a standard way in which large blocks of data can be read and written. We can supply CANopen **diagnostic and network management software**, Embedded drivers and **I/O modules**.

**TTCAN** - Time Triggered CAN - The Time-Triggered Protocol has nodes reporting in predefined time windows that have to be planned and synchronised but which then ensure that an overload on the bus is not possible even in a worst case situation.

**J1939** - A whole family of industry specific standards (agriculture, marine, truck & bus etc) are built on the basic communication services of the J1939 protocol specification ( itself based on Full CAN - ISO 11898-2) with industry-specific documents defining the particular combination of layers for that industry. PEAK provide a full database of J1939 mnemonics with their [J1939 option to PCAN-Explorer](#)

**B10011S** is the Transceiver used in a very restricted version of CAN (ISO 11992-1) that has only two nodes normally a truck and its trailer - not to be confused with.....

**FMS** is a message protocol subset of J1939 defined for the Bus and Truck/Trailer market. For a software packages that knows the meaning of all the FMS messages and can test/emulate and display them in a meaningful way see our [FMS Toolkit](#).

**MilCAN** - is defined for use in military land vehicles where a deterministic protocol is require. It sets up some rules for use and a software layer on top of a conventional CAN network. A Pseudo Hardware Sync is created by one node "the SyncMaster" that sends Sync CAN Frames with a "sync slot number".

MilCAN A uses 29 bit Identifiers. It allows both periodic and event driven data to be transmitted via the bus.

MilCAN B uses 11 bit identifiers. It allows only periodic data to be transmitted via the bus

| Standard    | Common Name        | Baud Rate                           | Max nodes          | Max Length | Adapter for PCAN interfaces                   |
|-------------|--------------------|-------------------------------------|--------------------|------------|---|
| ISO 11783   | ISOBUS             | 250 KBit/s                          | 30                 | 40m        | None required                                 |
| ISO 11898-2 | High speed-CAN     | max. 1 MBit/s                       | 110                | 6500 m     | None required                                 |
| ISO 11898-3 | Fault Tolerant CAN | max. 125 KBit/s                     | 32                 | 500 m      | <a href="#">PCAN TJA1054</a>                  |
| ISO 11992   | Truck/Trailer CAN  | max. 125 KBit/s                     | 2 (Point to Point) | 40 m       | <a href="#">PCAN-BD10011S</a>                 |
| ISO 15765   | Diagnostics On CAN | max 1 MBit/s                        | 110                |            | <a href="#">PCAN-OB</a> connector             |
| SAE J1939   | J1939              | 250 KBit/s                          | 30 ECUs            | 40m        | <a href="#">J1939 option to PCAN-Explorer</a> |
| SAE J2284   |                    | max. 1 MBit/s                       | 110                |            | None required                                 |
| SAE J2411   | Single Wire CAN    | 33.3 KBit/s<br>83.3KBit/s in HSMODE | 32                 |            | <a href="#">PCAN-AU579</a>                    |

## LIN

Local Interconnect Network is simpler than CAN and is often used in automotive "body functions" - eg windows, where performance is not critical but cost is. CAN is then often used to integrate the operation of multiple LIN sub networks. LIN is a single master, multiple slave system that uses a 12V single wire physical layer and a UART/SDI with master driven self synchronisation. It is capable of running at data rates of up to 20Kbits per second over a maximum distance of 40 Meters. We can supply a [USB to LIN interface](#) and a [LIN to CAN gateway](#) which simplify developing LIN and mixed CAN/LIN systems.

## Interface between CAN and PCs

We can supply a number of different interfaces to allow a PC to talk to CAN. The most popular is the [PCAN-USB interface](#) but we also have the many form factors of PCI including PC104 and PCMCIA as well as ISA and RS232. [See our range](#). These are then all supported by extensive

## Software

Embedded CAN chip manufacturers will provide examples of how to drive their chips, usually written in assembler or C. PC software to drive the USB, PCI or other interfaces to CAN will also be supplied by the interface manufacturer. Peak provide both low level drivers for Windows and a [free API](#) for their interfaces to allow CAN and LIN to be driven from a range of application languages on the PC. Also available are CAN analyser packages from the simple but powerful [PCAN-view](#) to the sophisticated [PCAN-Explorer](#) which provides data plotting with strip charts, user defined message names and data conversions for ease of analysis as well as extensive macro and script support for data collection and control. Add-in packages include J1939 support, a GUI interface that can be used for both display and control and a replay facility for Simulation. A specific software package is available to cover the [FMS](#) Bus and Truck standard so that it can be easily used by engineers not familiar with CAN.

A Windows standard API has been developed for communicating between C code on PCs and CAN - its called [RP1210](#) and a driver for it is available for the PCAN range of interfaces so that they can be used with applications supporting that standard..

CAN-Open is necessarily more complex but we can supply both PC based CANopen [diagnostic and network management software](#) and embedded drivers.

## Further reading

[Staffan Nilssons excellent introduction to CAN](#)

[Bosch started CAN and include many useful links on their site](#)

[CAN in Automation \(CiA\) is the CAN trade association](#)

[The LIN Consortium is the LIN standards guardian](#)

[Kvaser have a good CAN description area with details of available embedded interface chips and screen shots of messages](#)

[COMSOL have a wide range of CAN and LIN interfaces, CAN I/O units and Data Acquisition systems available with supporting software.](#)

If you have found this tutorial useful you might also be interested in our tutorials on Embedded TCP/IP and USB or in tutorials on a range of microprocessor and microcontroller families.

If so you can find them at [Embedded Tutorials](#)



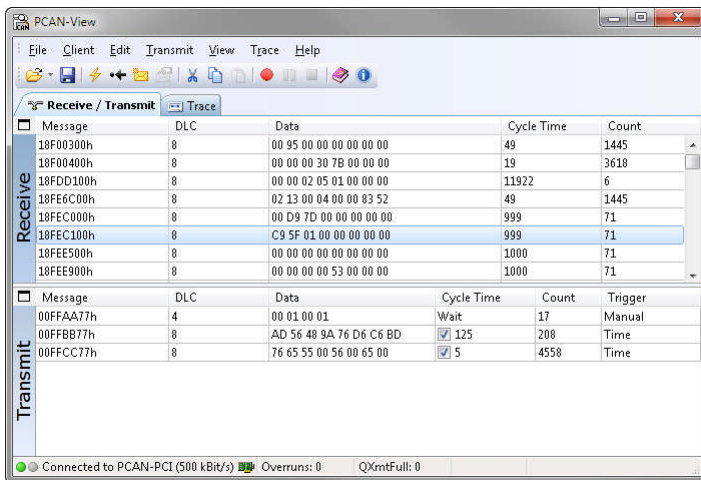
# Using CAN

## PEAK CAN Controllers

The Peak range of CAN interfaces provides simple and cost effective connections between PCs and CAN-networks and includes routers, extenders and adapters to the many CAN variants.



**PCAN interfaces support both 11 bit ID and 29 bit ID CAN specifications with a maximum speed of 1Mbaud.** They use the SJA1000 CAN-controller and the 82C251 driver. The CAN-bus connection is via a 9-pin SUB-D plug, whose pin assignments conform to the CiA recommendation. No termination is included in the interfaces. Optoisolated versions are available if required and most interfaces can be supplied with one or two ports.



Drivers and supporting DLL's are included to allow operation under XP, Vista, Windows 7 and Linux in 32 and 64\* bit modes. CE6.x support\* for ARM and x86 is also available. (\* most interfaces). Language support is provided for C++, C#, C++/CLR, Delphi, VB.NET, Java and Python 2.6.

A Windows package PCAN-View is included with all interfaces to allow the user to view messages on the CAN bus. All data is displayed in Hex and messages are timed and counted. A trace buffer allows messages to be recorded and saved to disk. Errors such as over-run and baud rate problems are reported. Messages can be user created and

then sent as one-shot, repeating periodically or in response to a remote request (RTR).

## Industrial I/O Modules

These Industrial I/O modules are available in a number of useful configurations including signal conditioning and termination in an industrial case. Up to 8 10 bit analogue inputs, 4 PWM/frequency outputs (to 20KHz) and 8 digital ins and outs are available. At 51 x 60 x 20mm, they are suitable for a wide variety of applications. The electrical connections provide snap-in termination; with a screw connection as an option. A windows package is provided so that the user can set message ID and data layouts and conversion constants as well as setting report rates or reporting on change.

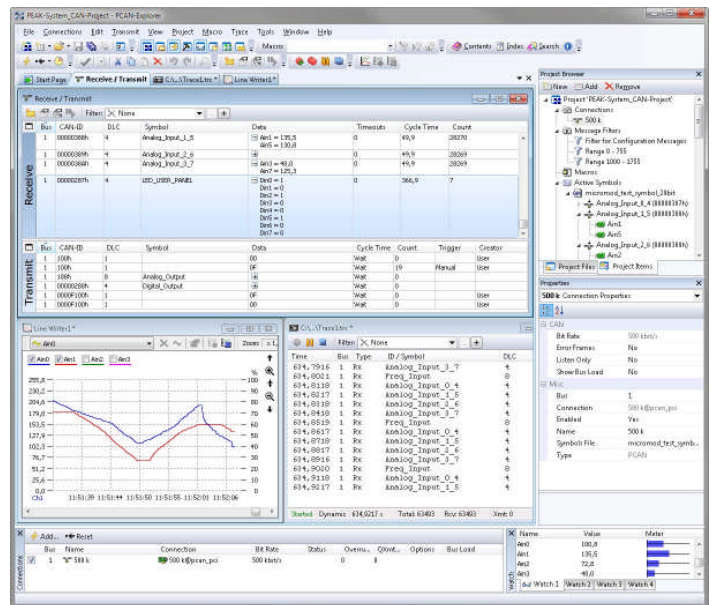
CAN Open firmware is also available if required.



## PCAN-Explorer

This tool can be used as an advanced CAN bus traffic monitor. It provides the following features: Message identifiers can be given names avoiding having to remember each messages HEX value. Message data can be displayed in a wide range of formats including text, hex, signed and unsigned and floating point.

An extensive conditional macro language allows complex tests and simulations to be performed. Optional packages provide sophisticated graphing, CANdb data exchange, a visual GUI to control and display CAN data collection and a full J1939 symbol database simplifying control of ECU's.



## PCAN-FMS Toolkit

A special software package is available to support FMS and Bus-FMS standards. It logs and translates the CAN messages in real time providing the user with a "Dashboard display". The log can be replayed to a CAN bus or values can be set manually from a convenient Windows display for system simulation.

## Data Acquisition Systems

Peak have a growing range of units designed to suite a variety of data acquisition requirements such as multi Thermocouple, A/D and D/A conversions and digital I/O. As well as data logging and mobile GPRS links there are also CAN Routers and filters. Some have full C programmability others only need simple windows configuration.

## CAN MicroMod Boards

A flexible, small format, Analogue and Digital I/O board with a CAN-Bus interface.

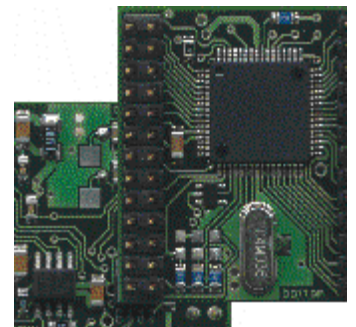
The PCAN-MicroMod board can be plugged into an application to provide control and monitoring services via the CAN-Bus. It provides the following

- 8 analogue inputs, 10-bits resolution, Vref 5 V
- 8 digital inputs & 8 digital outputs
- 4 PWM / frequency outputs, with a range 1 Hz-20 kHz

The integrated firmware provides simple configuration of the target system via a Windows utility program, the configuration data being sent to the board via CAN. No embedded programming skills are required to set up a system. Up to 32 PCAN-MicroMod boards can be put onto a single CAN network.

An optional evaluation board simplifies development of user designed carrier boards and also makes the MicroMod an ideal CAN evaluation tool.

Call or email us with your requirements.





# PCAN-USB

## USB to CAN Interface

The PCAN-USB adapter enables simple connection to CAN networks. Its compact plastic casing makes it suitable for mobile applications.

The opto-decoupled version guarantees galvanic isolation of up to 500 Volts between the PC and the CAN sides.

The package is also supplied with the CAN monitor PCAN-View for Windows and the programming interface PCAN-Basic.



| D-Sub | Pin | Pin assignment               |
|-------|-----|------------------------------|
|       | 1   | Not connected / optional +5V |
|       | 2   | CAN-L                        |
|       | 3   | GND                          |
|       | 4   | Not connected                |
|       | 5   | Not connected                |
|       | 6   | GND                          |
|       | 7   | CAN-H                        |
|       | 8   | Not connected                |
|       | 9   | Not connected / optional +5V |

### Specifications

- ☐ Adapter for USB connection (USB 1.1, compatible with USB 2.0)
- ☐ USB voltage supply
- ☐ Bit rates up to 1 Mbit/s
- ☐ Time stamp resolution approx. 42 µs
- ☐ Compliant with CAN specifications 2.0A (11-bit ID) and 2.0B (29-bit ID)
- ☐ CAN bus connection via D-Sub, 9-pin (in accordance with CiA® 102)
- ☐ NXP SJA1000 CAN controller, 16 MHz clock frequency
- ☐ NXP PCA82C251 CAN transceiver
- ☐ 5-Volts supply to the CAN connection can be connected through a solder jumper, e.g. for external bus converter
- ☐ Extended operating temperature range from -40 to 85 °C (-40 to 185 °F)

Optionally available:

- ☐ Galvanic isolation on the CAN connection up to 500 V

### Ordering information

| Designation             | Art. No.    |
|-------------------------|-------------|
| PCAN-USB                | IPEH-002021 |
| PCAN-USB opto-decoupled | IPEH-002022 |

### Scope of supply

- ☐ PCAN-USB in plastic casing
- ☐ Device drivers for Windows 7/Vista/XP/Linux (32/64-bit)
- ☐ Device driver for Windows CE 6.x (x86 and ARMv4 processor support)
- ☐ PCAN-View CAN monitor for Windows
- ☐ PCAN-Basic programming interface consisting of an interface DLL, examples, and header files for all common programming languages
- ☐ Manual in PDF format

# PCAN-USB Pro

## High-Speed USB 2.0 to CAN/LIN Interface

The PCAN-USB Pro adapter enables simple connection of a PC to CAN and LIN networks. Two field busses can be connected at the same time, with up to four connections available using appropriate adapter cables (2 x CAN, 2 x LIN). Its robust aluminum casing makes the PCAN-USB Pro adapter suitable for mobile applications.

The supplied monitor applications PCAN-View and PLIN-View Pro as well as the PCAN-Basic and PLIN programming interfaces round off the range.



### Specifications


- ☐ Adapter for USB connection (USB 2.0)
- ☐ Transmitting and receiving of CAN and LIN messages using 2 D-Sub connections (both with pin assignment for CAN and LIN bus)
- ☐ Time stamp resolution 1  $\mu$ s
- ☐ Extended operating temperature range from -40 to 85 °C (-40 to 185 °F)

#### CAN operation properties:

- ☐ Bit rates up to 1 Mbit/s
- ☐ Fulfills CAN specifications 2.0A and 2.0B
- ☐ MAX3057ASA CAN transceiver
- ☐ Each CAN channel is separately opto-decoupled against USB and LIN
- ☐ Measurement of bus load including error frames and overload frames
- ☐ Induced error generation for incoming and outgoing CAN messages

#### LIN operation properties:

- ☐ Bit rates of 1 – 20 kbit/s
- ☐ AMIS-30600 LIN transceiver
- ☐ Both LIN channels (common ground) are opto-decoupled against USB and CAN
- ☐ Can be used as a LIN master or slave (1 ms master task resolution)
- ☐ Automatic bit rate, frame length, and checksum type recognition
- ☐ Autonomous scheduler with support for unconditional, event, and sporadic frames
- ☐ Hardware can work through a schedule table (up to 8 schedule tables can be configured with a total of 256 slots)

| D-Sub  | Pin | Pin assignment   |
|--|-----|------------------|
|  | 1   | +5V              |
|  | 2   | CAN-L            |
|  | 3   | CAN-GND          |
|  | 4   | LIN              |
|  | 5   | LIN-GND          |
|  | 6   | LIN-GND          |
|  | 7   | CAN-H            |
|  | 8   | Not connected    |
|  | 9   | V <sub>BAT</sub> |

### Ordering information

| Designation  | Art. No.    |
|--------------|-------------|
| PCAN-USB Pro | IPEH-002061 |

#### Scope of supply

- ☐ PCAN-USB Pro in aluminum casing
- ☐ Device drivers for Windows 7/Vista/XP/Linux (32/64-bit)
- ☐ PCAN-View CAN monitor for Windows
- ☐ PLIN-View Pro LIN monitor for Windows
- ☐ PCAN-Basic programming interface consisting of an interface DLL, examples, and header files for all common programming languages
- ☐ PLIN-API programming interface consisting of an interface DLL, an example, and header files for all common programming languages
- ☐ Manual in PDF format

# PCAN-USB Hub


All-in-one USB Adapter for Communication through USB, CAN, and RS-232

The PCAN-USB Hub provides multiple hardware interfaces through a USB connection. It offers the user one CAN, two RS-232, and two further USB interfaces.

Its robust aluminum casing makes the PCAN-USB Hub suitable for mobile applications.


The package is also supplied with the CAN monitor PCAN-View for Windows and the programming interface PCAN-Basic.



| D-Sub   | Pin | Pin assignment RS-232 |
|---|-----|-----------------------|
|  | 1   | DCD                   |
|   | 2   | RxD                   |
|   | 3   | TxD                   |
|   | 4   | DTR                   |
|   | 5   | GND                   |
|   | 6   | DSR                   |
|   | 7   | RTS                   |
|   | 8   | CTS                   |
|   | 9   | RI                    |

## Specifications

- ☐ High-speed USB 2.0 hub with
  - USB to CAN interface, connection through 9-pin D-Sub (in accordance with CiA® 102)
  - Two USB to RS-232 converters using two D-Sub connectors (9-pin)
  - Two High-speed USB 2.0 downstream ports
- ☐ Passive (bus-powered) hub operation through the USB port of a PC enables power consumption of up to 100 mA per USB channel
- ☐ Active (self-powered) hub operation through the optional external hub power supply (9 – 36 V) enables power consumption of up to 500 mA per USB channel
- ☐ Guaranteed high bit rates on all channels if a Full-speed device is connected, thanks to a hub controller with 4 transaction translators
- ☐ Bit rates up to 1 Mbit/s
- ☐ Time stamp resolution approx. 42 µs
- ☐ Compliant with CAN specifications 2.0A and 2.0B
- ☐ NXP PCA82C251 CAN transceiver
- ☐ 5-Volt and external power supply at the CAN connector connectible by solder bridges, e.g. for external bus converter
- ☐ Extended operating temperature range

| D-Sub  | Pin | Pin assignment CAN           |
|--|-----|------------------------------|
|  | 1   | Not connected / optional +5V |
|  | 2   | CAN-L                        |
|  | 3   | CAN-GND                      |
|  | 4   | Not connected                |
|  | 5   | Not connected                |
|  | 6   | PWR-GND                      |
|  | 7   | CAN-H                        |
|  | 8   | Not connected                |
|  | 9   | PWR-OUT                      |

## Ordering information

| Designation  | Art. No.    |
|--------------|-------------|
| PCAN-USB Hub | IPEH-002004 |

## Scope of supply

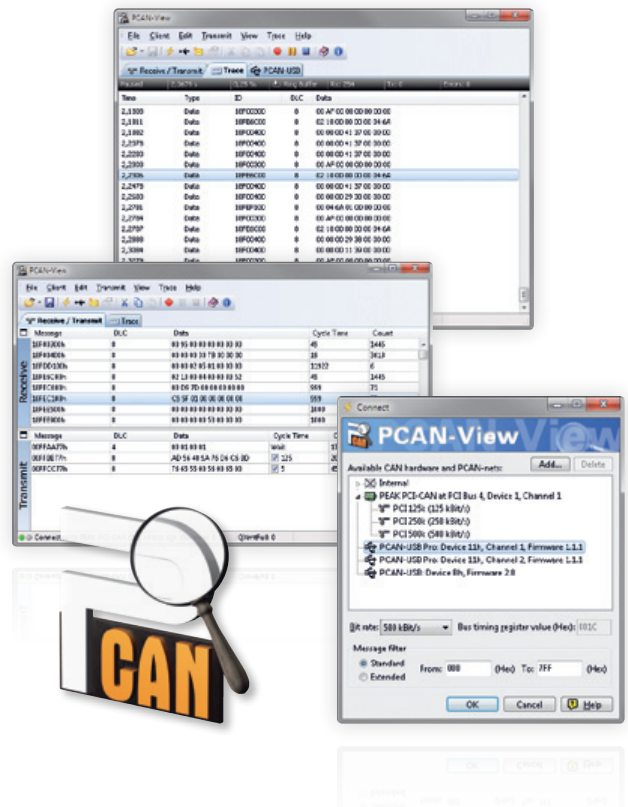
- ☐ PCAN-USB Hub in aluminum casing
- ☐ Mating connector for voltage supply
- ☐ Device drivers for Windows 7/Vista/XP/Linux (32/64-bit)
- ☐ Device driver for Windows CE 6.x (x86 and ARMv4 processor support)
- ☐ PCAN-View CAN monitor for Windows
- ☐ PCAN-Basic programming interface consisting of an interface DLL, examples, and header files for all common programming languages
- ☐ Manual in PDF format

# PCAN-View

## Windows Compatible Software for Displaying CAN Messages

The software PCAN-View for Windows is a simple CAN monitor for viewing, sending, and recording CAN data traffic. Messages can be sent manually and periodically at a user-determined bit rate of up to 1 Mbit/s. Bus system errors and memory overflows in the CAN hardware being controlled are displayed during the process. The trace function can be used to record and save CAN data traffic.

PCAN-View is supplied with every PCAN PC hardware product and enables a fast and simple initial hardware operation. All available PEAK CAN interfaces are listed in the connection dialog. After selecting the hardware and the bit rate, the user can access all the software functions, hardware-specific settings, and information.



### Features

- \_\_\_ Manual and periodic transmission of CAN messages with bit rates up to 1 Mbit/s and a resolution of 10 ms
- \_\_\_ Support for CAN 2.0A (11-bit ID) and 2.0B (29-bit-ID) specifications
- \_\_\_ Saving and reloading of CAN messages as transmit lists
- \_\_\_ CAN message reception with a resolution of 1 ms
- \_\_\_ Recording CAN messages in trace files
- \_\_\_ Display of receive, transmit, and error states
- \_\_\_ CAN controller hardware reset (SJA1000)
- \_\_\_ Access to hardware-specific settings and information

Note: ISA and parallel port CAN interfaces only support 32-bit

### Ordering information

#### Designation

PCAN-View

#### Scope of supply

- \_\_\_ PCAN-View software
- \_\_\_ Documentation in HTML Help format

#### The current version ...

... is available on the product CD  
(supplied with our hardware)  
... can be downloaded from the Internet free of charge at [www.peak-system.com](http://www.peak-system.com)

#### System requirements

- \_\_\_ Windows 7/Vista/XP (32/64-bit)
- \_\_\_ At least 512 MB RAM and 1 GHz CPU

# PCAN-Basic

## CAN Software API for Windows

The PCAN-Basic API (application programming interface) enables easy development of powerful software with CAN support. It includes all of the functions that a application needs in order to communicate with PCAN PC hardware. The cross-operation system design makes it possible to easily port software projects between platforms.

PCAN-Basic consists of the actual device driver and an interface DLL, which provides the API functions.

As the successor to PCAN-Light, PCAN-Basic offers increased functionality and extended language support. It provides various functions for developers under C++, C#, C++/CLR, Delphi, VB.NET, Java, and Python 2.6.



### Features

- \_\_\_ Supports Windows 7/Vista/XP (32/64-bit) and Windows CE 6.x operating systems
- \_\_\_ Multiple PEAK-System applications and your own can be operated on a physical CAN channel at the same time
- \_\_\_ Use of a single DLL for all supported hardware types
- \_\_\_ Use of up to 8 channels for each hardware unit (depending on the PEAK CAN interface used)
- \_\_\_ Simple switching between channels of a PEAK CAN interface
- \_\_\_ Driver-internal buffering of 32,768 messages per CAN channel
- \_\_\_ Precision of time stamps on received messages up to 1  $\mu$ s (depending on the PEAK CAN interface used)
- \_\_\_ Access to specific hardware parameters, such as listen-only mode
- \_\_\_ Notification of the application through Windows events when a message is received
- \_\_\_ Extended system for debugging operations
- \_\_\_ Multilingual debugging output
- \_\_\_ Output language depends on operating system
- \_\_\_ Debugging information can be defined individually

Note: ISA and parallel port CAN interfaces only support 32-bit

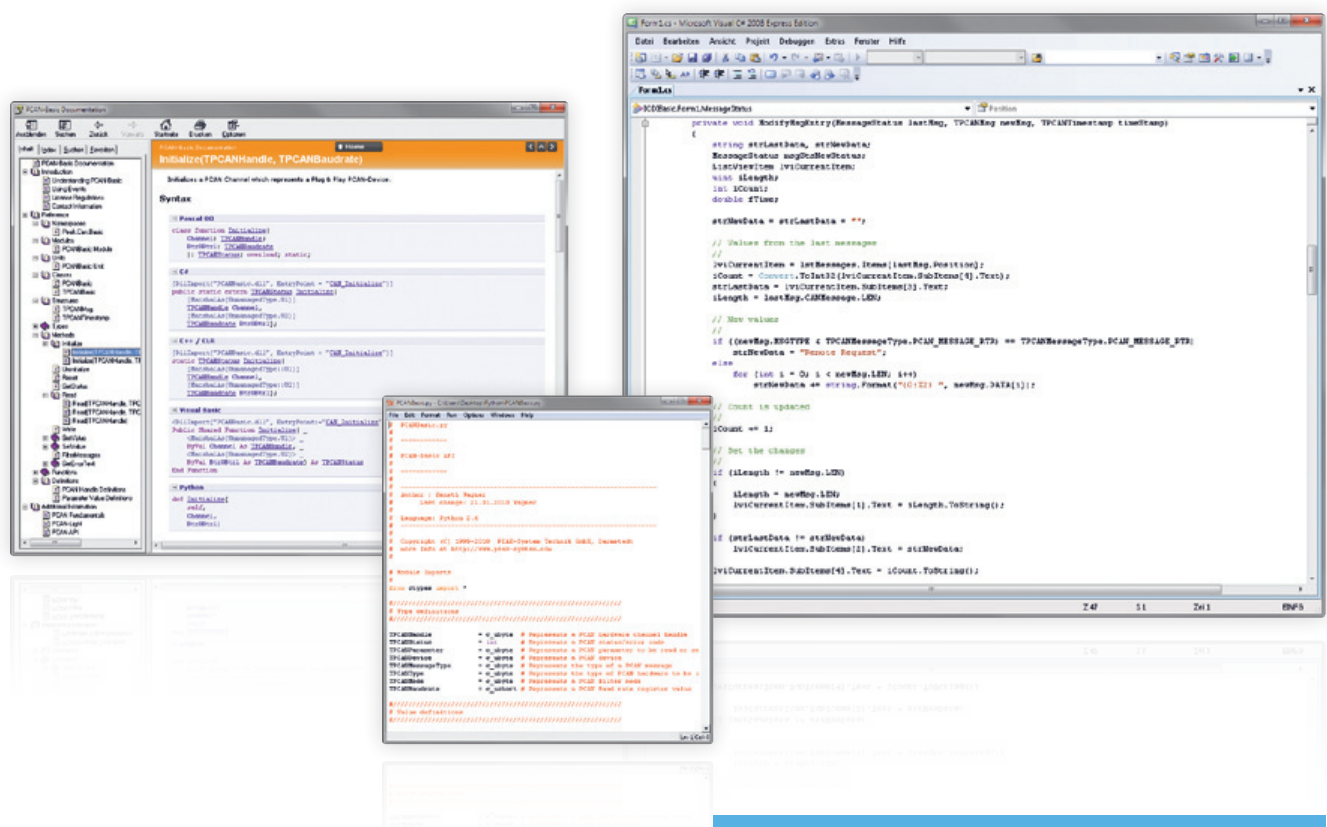
### PCAN-Basic for Windows CE

PEAK-System provides PCAN-Basic API to allow the development of your own CAN applications for Windows CE 6.x. Programmers can also use the languages C++, C# and VB.NET.

### PCAN-Basic for Linux

A Linux version of the PCAN-Basic API is currently being developed and will be available soon.





## Functions

- Connection:
- Initialize: Initialize hardware, set the bit rate, drivers log on
- Uninitialize: Drivers log out
- Configuration:
- SetValue: Setup of hardware parameters such as debug log, listen-only, and auto-reset
- FilterMessages: Register messages to be received
- Information:
- GetValue: Read out DLL and API information
- GetStatus: Read out CAN bus status information
- GetErrorText: Get an error code description
- CAN communication:
- Read: Read a CAN message or status inc. time stamp
- Write: Transmit a CAN message (11/29-bit ID and RTR possible)
- Reset: Clear the transmit and receive buffer

## Ordering information

### Designation

PCAN-Basic

### Scope of supply

- \_\_\_ Interface DLL, examples, and header files for all common programming languages
- \_\_\_ Documentation in HTML Help format

### The current version ...

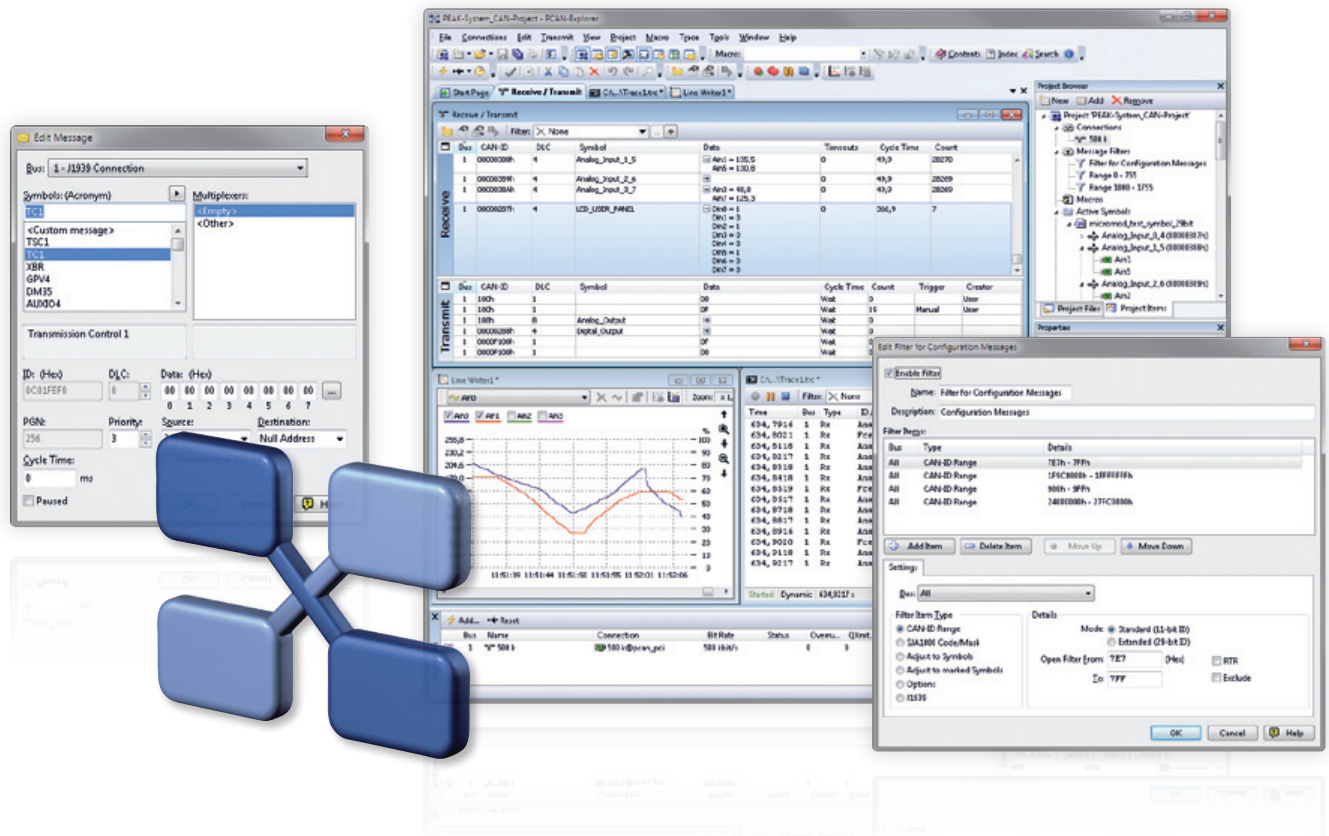
- ... is available on the product CD (supplied with our hardware)
- ... can be downloaded from the Internet free of charge at [www.peak-system.com](http://www.peak-system.com)

### System requirements

- \_\_\_ Windows 7/Vista/XP (32/64-bit) or Windows CE 6.x
- \_\_\_ At least 512 MB RAM and 1 GHz CPU

# PCAN-Explorer 5

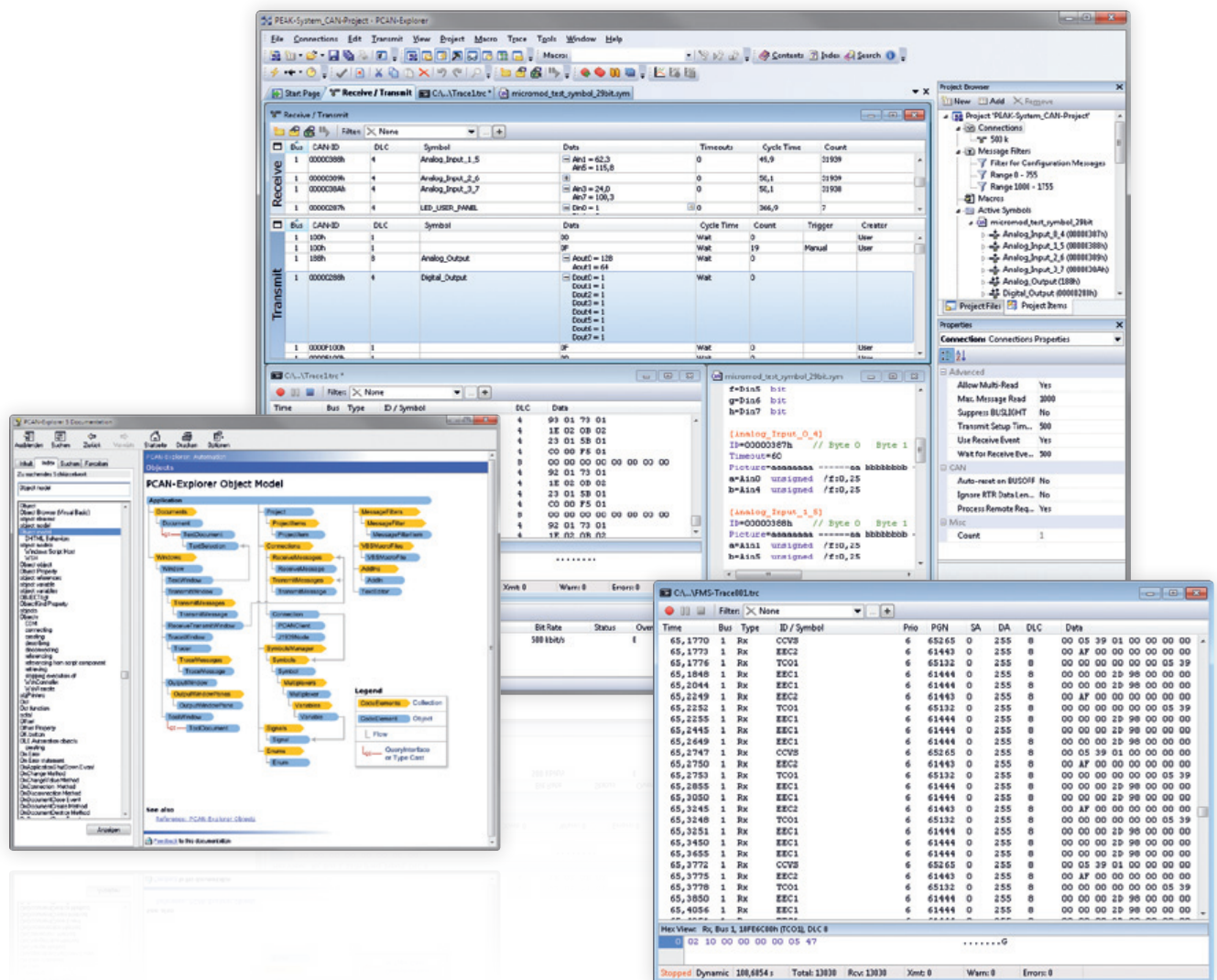
Comprehensive CAN Monitor for Windows



PCAN-Explorer is a universal tool for monitoring data traffic on a CAN network. For easy and clear allocation of the individual messages, these can be identified as so-called symbols. The integrated VBScript support allows the creation of macros to automate complex tasks. The integrated data logger means that the data traffic of a bus can be recorded, analyzed, and stored. PCAN-Explorer is designed as automation server and can therefore be remote controlled through scripts.

## Features

- \_\_\_ All files and elements can be managed and saved in projects
- \_\_\_ Project components such as CAN connections, symbol files and macros are clearly displayed and laid out in the project browser
- \_\_\_ The new start page allows fast access to the most recently opened projects or files
- \_\_\_ Simultaneous connections with multiple networks/CAN interfaces of the same hardware type
- \_\_\_ Connection window with an overview of all connections, complete with status, error counters, bus load, etc.
- \_\_\_ All parameters of all elements in the user interface can be examined using a property window and edited if necessary.
- \_\_\_ Multiple flexible filters can be configured and, for example, assigned to the transmit/receive window or the various different tracers
- \_\_\_ Tabs to switch between the different Windows
- \_\_\_ Flexible arrangement of the user interface using the dockable windows
- \_\_\_ User-defined column display and arrangement in transmit/receive window
- \_\_\_ J1939 support with the relevant add-in
- \_\_\_ Display of received messages showing the ID, length, data bytes, number of messages received and receiving interval
- \_\_\_ Simultaneous hexadecimal and symbolic representation of the details
- \_\_\_ Display of remote frames, status reports of the CAN controller and, as option, CAN-bus error frames also
- \_\_\_ Logging of time-outs
- \_\_\_ Sending of messages at fixed intervals of time, manually or as reply to remote frames

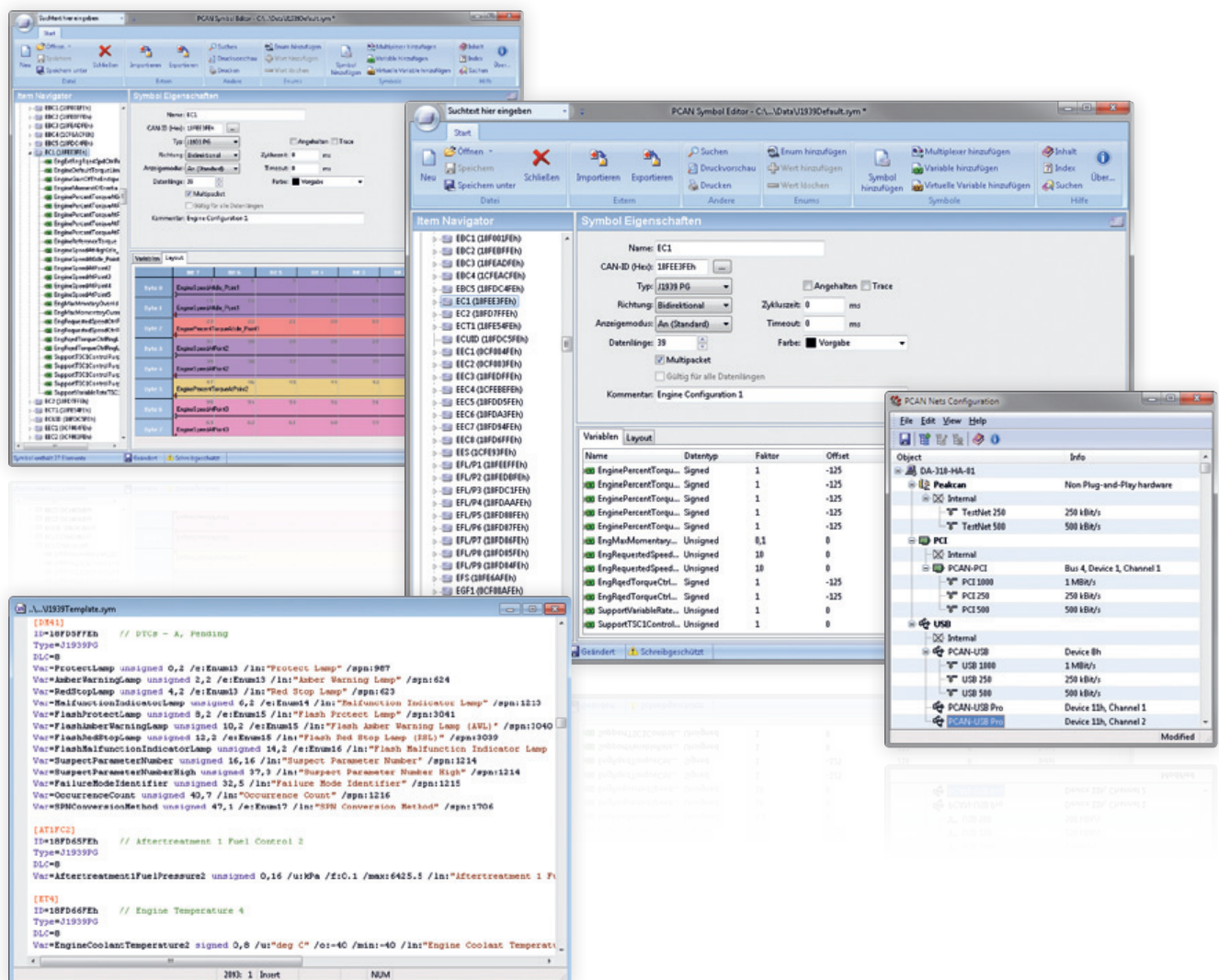


- \_\_\_ Messages can be created as send lists, stored and loaded as desired, in order to e. g. emulate CAN nodes
- \_\_\_ Periodical sending with up to 1 ms precision
- \_\_\_ Easy creation of symbol files and macros using the integrated text editor with syntax highlighting
- \_\_\_ User-friendly real-time monitoring of several signals via the watch window
- \_\_\_ Extensive improvements to user guidance and interface compared to PCAN-Explorer version 4
- \_\_\_ User interface language can be switched (German/English)
- \_\_\_ Simple integration of external tools
- \_\_\_ Integration of Add-Ins to upgrade functionality

Properties of the integrated, configurable PCAN-Explorer data logger:

- \_\_\_ Operation of multiple tracers at the same time
- \_\_\_ Variable buffer size
- \_\_\_ Optional linear buffer or circular buffer
- \_\_\_ Representation of the logged messages with time stamp, type, ID, length and data bytes
- \_\_\_ Logging of errors that have occurred is possible
- \_\_\_ Flexible storage possibility for the logged data in text form for importing into Excel or similar
- \_\_\_ Filtering of the messages for logging through symbol definitions
- \_\_\_ Subsequent examination of the logged data in the buffer via different symbol files





Function upgrade of the PCAN-Explorer with the integrated VBScript language:

- \_\_\_ Creation of macros in VBScript with the integrated text editor
- \_\_\_ Access with macros and scripts to almost all program elements via the PCAN-Explorer object model
- \_\_\_ Ideal for creating test tools to implement or develop CAN systems
- \_\_\_ Examples: sending of e-mails when a temperature is exceeded, starting of a test tool when a particular message is received, opening of an Excel sheet when an event occurs and saving of data in the individual cells
- \_\_\_ Assignment of function keys with individual send messages or macros
- \_\_\_ VBS scripts run in the background even without the PCAN-Explorer interface

## Ordering information

| Designation     | Art. No.    |
|-----------------|-------------|
| PCAN-Explorer 5 | IPES-005028 |

## Scope of supply

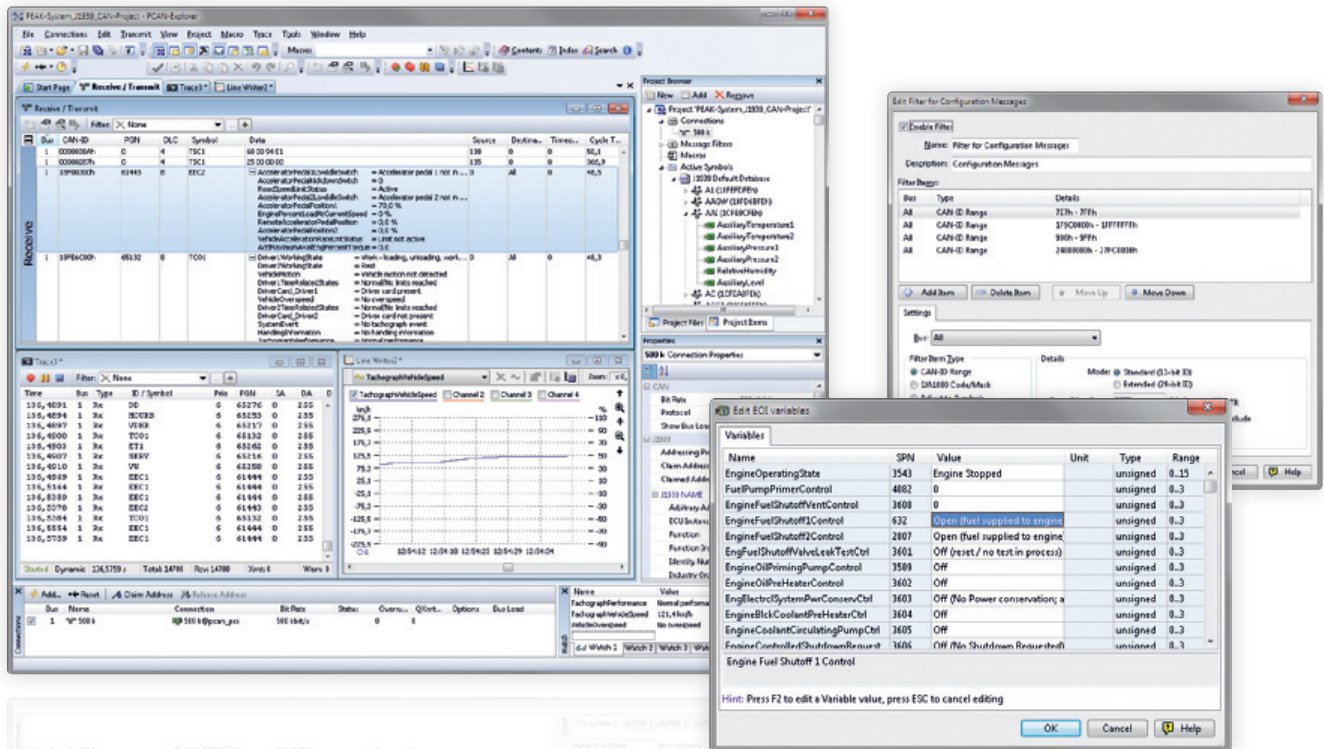
- \_\_\_ PCAN-Explorer installation CD including PCAN-Explorer Line Writer Add-in
- \_\_\_ PCAN Symbol Editor for Windows (details on page 62)
- \_\_\_ Documentation in HTML help format

## System requirements

- \_\_\_ Windows 7/Vista/XP (32/64-bit)
- \_\_\_ At least 512 MB RAM and 1 GHz CPU

# PCAN-Explorer Add-ins

Optional Function Upgrades for PCAN-Explorer



## J1939 Add-in

The SAE J1939 network protocol describes communication on a CAN bus in utility vehicles for the transmission of diagnostics data and control information. It contains a complete network definition using 29-bit CAN-IDs (CAN 2.0B Extended Frame).

The J1939 add-in for PCAN-Explorer 5 supports all definitions established by the standard's parameter groups and provides a simple means of accessing the parameters. A complete database of all the definitions and the parameters contained is also supplied.

### Features

- \_\_\_ Support for all functions of the SAE J1939 network protocol
- \_\_\_ CAN messages can be sent in broadcast form or targeted to individual control units (ECUs)
- \_\_\_ Addressing of up to 254 ECUs
- \_\_\_ Supports multi-packet messages

## Ordering information

### Designation

J1939 Add-in

### Art. No.

IPES-005089

### Scope of supply

- \_\_\_ J1939 Add-in software
- \_\_\_ Documentation in HTML help format

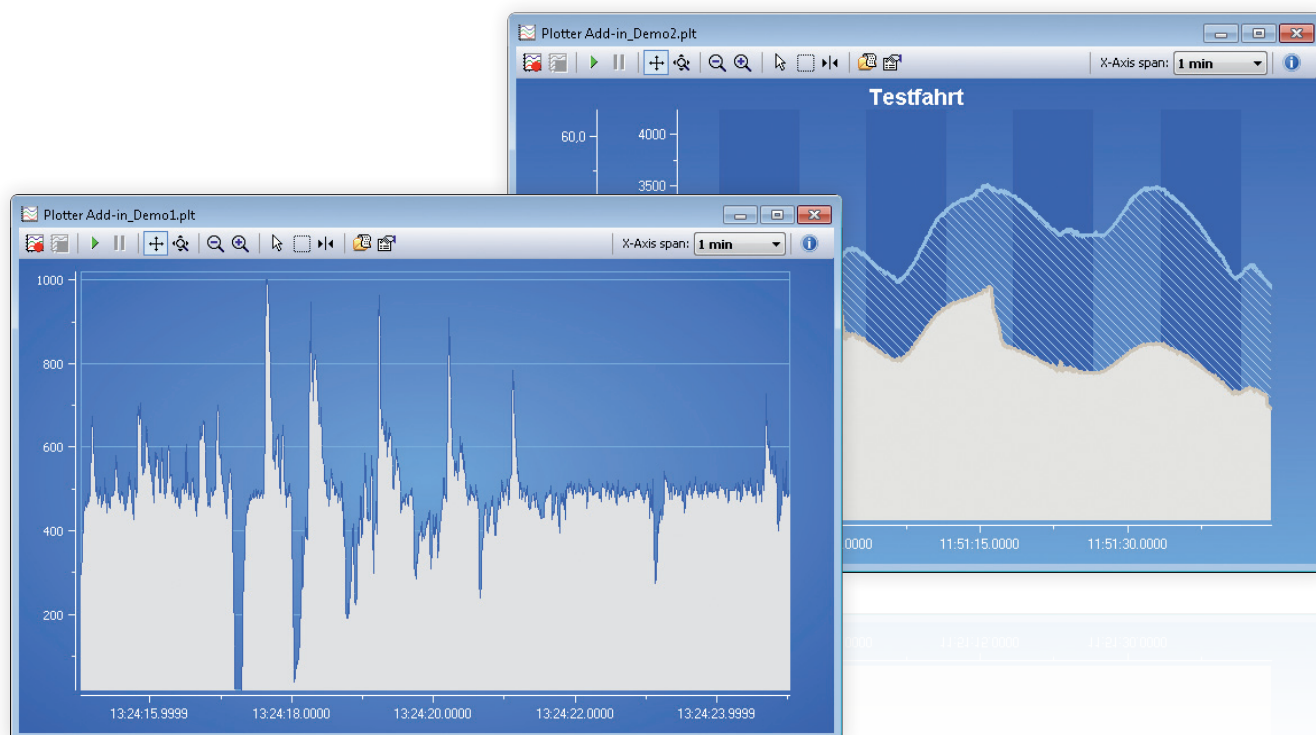
### System requirements

- \_\_\_ PCAN-Explorer 5
- \_\_\_ Windows 7/Vista/XP (32/64-bit)
- \_\_\_ At least 512 MB RAM and 1 GHz CPU



# PCAN-Explorer Add-ins

Optional Function Upgrades for PCAN-Explorer



## Plotter Add-in 2

The plotter allows the graphical representation of CAN data using any number of channels.

### Features

- ☐ Real-time display
- ☐ Unlimited number of channels
- ☐ Unlimited number of Y-axes
- ☐ X-axis and Y-axes can be zoomed and scrolled quite freely, even during recording
- ☐ Labelling of time axis with absolute or relative time stamps
- ☐ Facility for automatic adaptation of axes to plots
- ☐ Reversible Y-axes
- ☐ Logarithmic scales
- ☐ Cursor display for plot measurement
- ☐ Export to EMF-, PNG-, BMP-, JPEG graphical formats
- ☐ Data import from the PCAN-Explorer Tracer

- ☐ Representation of limiting values and value ranges
- ☐ Comprehensive formatting options for representing the curves, axes and the plotter layout

### Ordering information

| Designation      | Art. No.    |
|------------------|-------------|
| Plotter Add-in 2 | IPES-005087 |

### Scope of supply

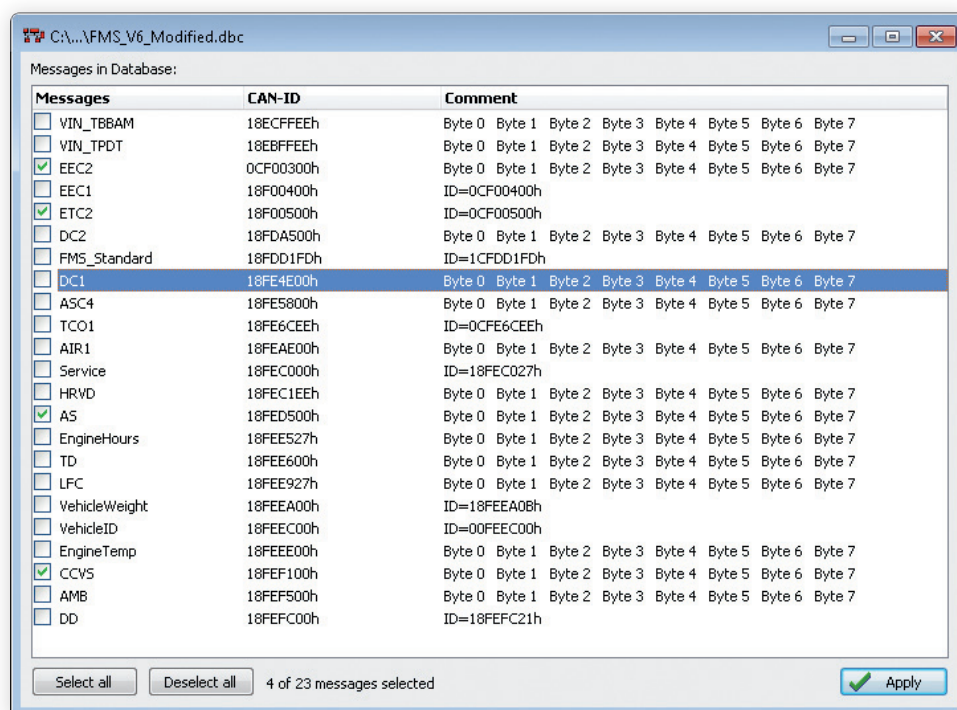
- ☐ PCAN-Plotter Add-in software

### System requirements

- ☐ PCAN-Explorer 5
- ☐ Windows 7/Vista/XP (32/64-bit)
- ☐ At least 512 MB RAM and 1 GHz CPU

# PCAN-Explorer Add-ins

Optional Function Upgrades for PCAN-Explorer



## CANdb Import Add-in 3

The CANdb format is a common data description format for CAN bus information in the car industry.

CANdb Import allows the import of CANdb files. This is a useful function for all those who do not want to manually transcribe their database into the PCAN-Explorer symbol file format.

### Features

- ☐ Opening of CANdb files (.dbc)
- ☐ Selecting of the messages for importing in a CANdb file
- ☐ Saves data using the project administration function in PCAN-Explorer
- ☐ Storing in the PCAN-Explorer symbol file format

### Ordering information

| Designation           | Art. No.    |
|-----------------------|-------------|
| CANdb Import Add-in 3 | IPES-005086 |

### Scope of supply

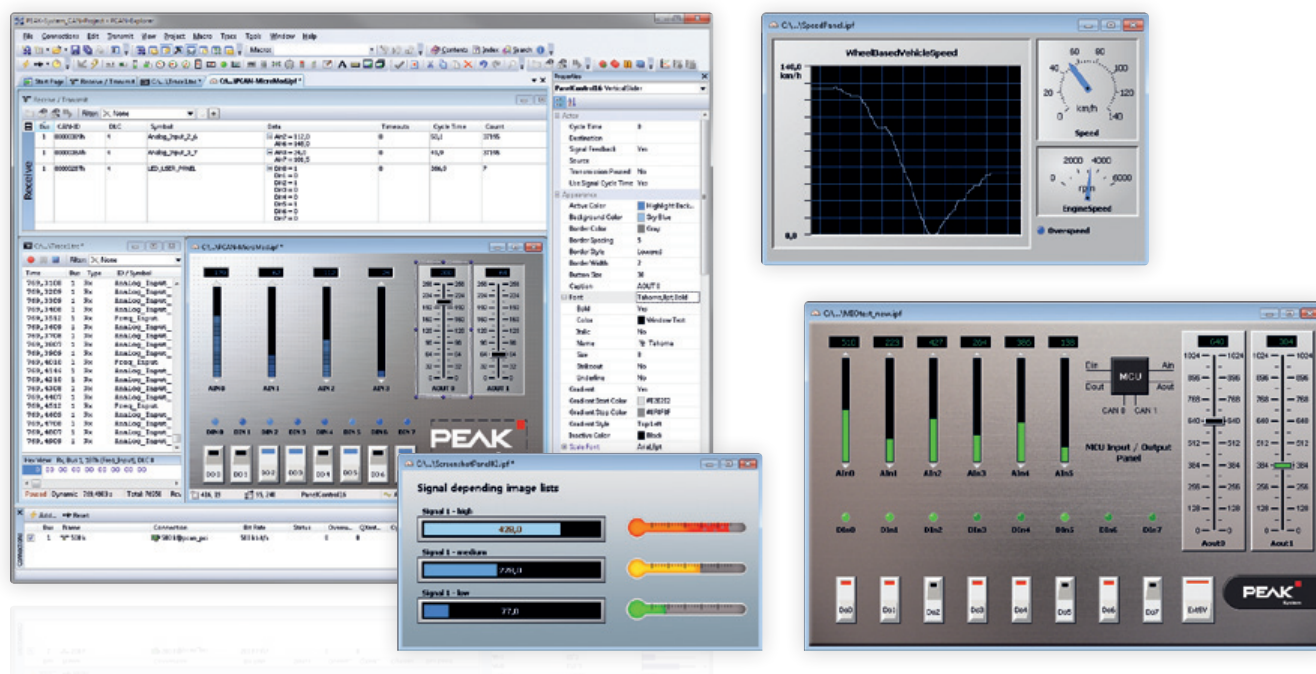
- ☐ CANdb Import Add-in software
- ☐ Documentation in HTML help format

### System requirements

- ☐ PCAN-Explorer 5
- ☐ Windows 7/Vista/XP (32/64-bit)
- ☐ At least 512 MB RAM and 1 GHz CPU

# PCAN-Explorer Add-ins

Optional Function Upgrades for PCAN-Explorer



## Instruments Panel Add-in 3

The Instruments Panel Add-in allows the graphical representation of digital and analog signals using different display instruments. The integrated input options and controllers mean that signals can also be produced on the CAN bus, allowing easy simulation of complex CAN applications.

- \_\_\_ Free positioning of the instruments using drag & drop, or numerical inputs for spot-on positioning
- \_\_\_ Loading and storing of complete panel configurations

## Features

- \_\_\_ Representation of analog and digital signals from received CAN messages using different display instruments
- \_\_\_ In addition to potentiometers, switches, and sliding controllers input fields can be used to generate CAN messages
- \_\_\_ Selection and configuration of multiple elements at the same time
- \_\_\_ Extensive configuration of the properties of one or more elements using the new property window
- \_\_\_ The new Instruments Panel object model enables complete automation using COM and scripts
- \_\_\_ Representation of different scenes on the same panel during running time
- \_\_\_ Signal-dependent display of image lists and scenes

## Ordering information

| Designation                | Art. No.    |
|----------------------------|-------------|
| Instruments Panel Add-in 3 | IPES-005088 |

## Scope of supply

- \_\_\_ Instruments Panel Add-in software
- \_\_\_ Documentation in HTML help format

## System requirements

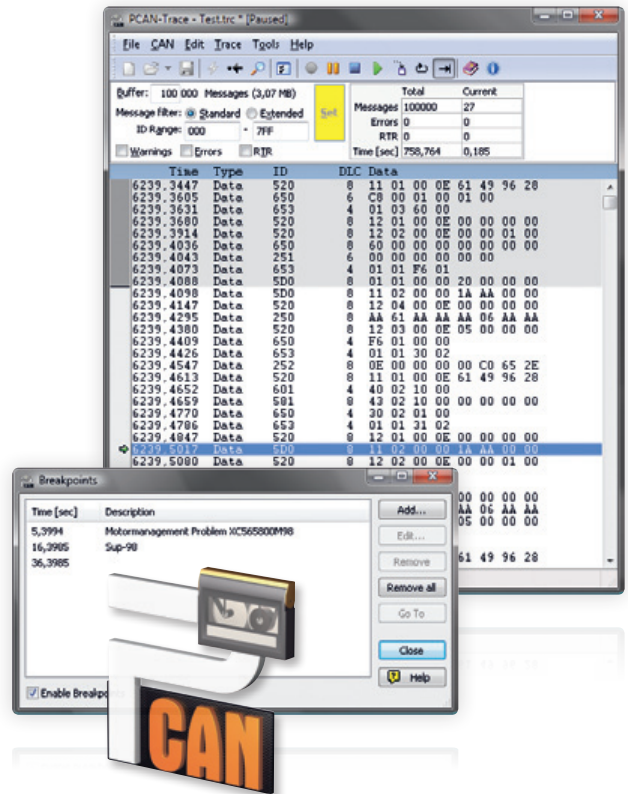
- \_\_\_ PCAN-Explorer 5
- \_\_\_ Windows 7/Vista/XP (32/64-bit)
- \_\_\_ At least 512 MB RAM and 1 GHz CPU

# PCAN-Trace

## Comprehensive Data Logger for CAN Messages

The PCAN-Trace program is a data logger for up to 9,999,000 CAN messages. It enables CAN messages to be quickly recorded, saved, and even played back on the CAN bus. The program displays the number of received messages, and identifies the types of the messages (data frame, error frame, RTR frame).

CAN messages can be recorded or replayed in linear or ring buffer mode. PCAN-Trace also provides an option to play back CAN messages in single step mode. You can also simplify analysis and tracing by setting playback mode breakpoints.



### Features

- ☐ Log facility for up to 9,999,000 CAN messages
- ☐ Choice of linear buffer or ring buffer (in receive and playback mode)
- ☐ Displays number and type of received CAN messages
- ☐ Adjustable message filter
- ☐ Support for 11-bit and 29-bit IDs
- ☐ Facility to play back CAN messages that have been recorded using PCAN-Explorer, even in single-step mode
- ☐ Breakpoints can be used in playback mode
- ☐ Integrated online help
- ☐ Received data can also be viewed in a text editor

### Ordering information

| Designation | Art. No.    |
|-------------|-------------|
| PCAN-Trace  | IPES-002027 |

### Scope of supply

- ☐ PCAN-Trace installation CD (in English)
- ☐ Documentation in HTML help format

### System requirements

- ☐ Windows 7/Vista/XP (32-bit)
- ☐ At least 512 MB RAM and 1 GHz CPU