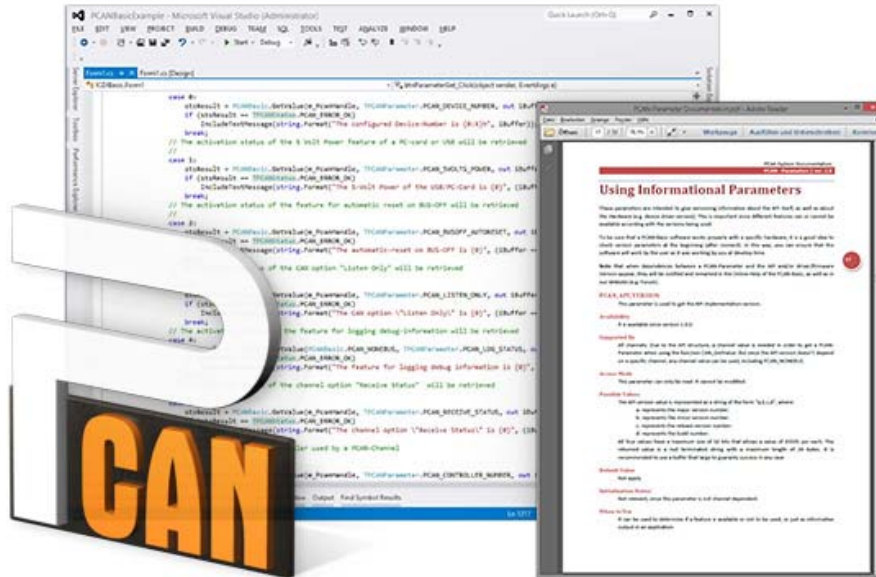# PCAN-Basic

Product Overview

The PCAN-Basic API (application programming interface) allows easy development of powerful software with CAN and CAN FD support. It includes all of the functions that an application needs in order to communicate with PCAN PC hardware. The cross-operating-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 3.x.

Features

- API for developing applications with CAN and CAN FD connections
- Support for CAN specifications 2.0 A/B and FD
- Application development for Windows 11 (x64/ARM64), 10 (x86/x64), and Linux
- Multiple PEAK-System applications and your own can be operated on a physical channel at the same time
- Use of a single DLL for all supported hardware types

- Use of up to 16 channels for each hardware type
- Simple switching between channels of a PEAK CAN interface
- Access to CAN channels of a PCAN-Gateway via the PCAN-LAN hardware type
- Driver-internal buffering of up to 32,768 CAN messages per channel
- Precision of time stamps up to 1 µs
- Supports PEAK-System's trace formats v1.1 and v2.0 (for CAN FD)
- Access to hardware parameters such as listen-only mode
- Notification of received messages via Windows events
- Support of CAN error frames
- Extended debugging system

System Requirements
- Windows 11 (x64/ARM64), 10 (x64), or Linux
- For CAN bus connection: PC CAN interface from PEAK-System

Ordering Information
Designation

PCAN-Basic

Scope of Supply

- PCAN-Basic DLLs (x86/x64/ARM64)
- PCAN-Basic.NET Assembly
- Przykłady i pliki nagłówkowe dla popularnych języków programowania
- Dokumentacja w formacie PDF oraz HTML Help

The current version can be downloaded free of charge from www.peak-system.com

Functions
Connection

- Initialize: Initialize CAN hardware, set the bit rate, log on at the driver
- InitializeFD: Initialize CAN FD hardware
- Uninitialize: Log off from the driver

Configuration

- SetValue: Setup hardware parameters (debug log, listen-only, auto-reset)
- FilterMessages: Register messages to be received

Information

- GetValue: Read DLL and API information
- GetStatus: Read CAN bus status information
- GetErrorText: Get error code description

CAN Communication

- Read: Read CAN message or status with timestamp
- ReadFD: Read CAN FD message
- Write: Transmit CAN message (11/29-bit ID, RTR possible)
- WriteFD: Transmit CAN FD message
- Reset: Clear transmit and receive buffer