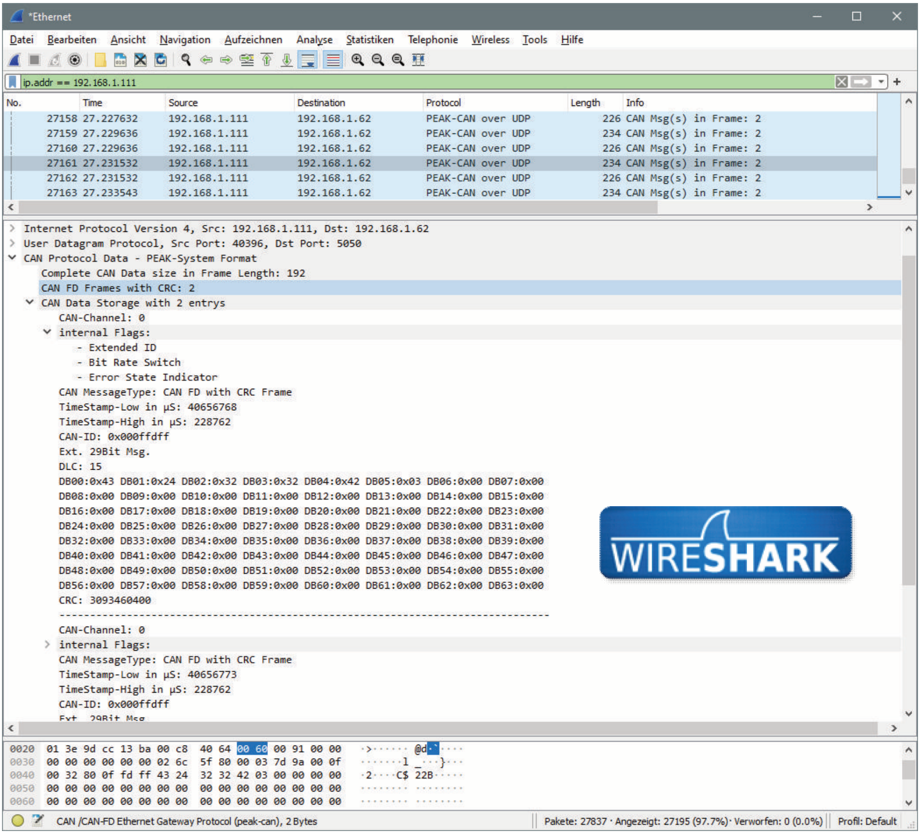
Wireshark Dissector

Support PCAN-Gateways IPEH-004010, IPEH-004011, IPEH-004012,IPEH-004020



Document version 1.2 (2022-01-24)

Relevant products

Product name	Model	Part number
PCAN-Ethernet Gateway DR	CAN to LAN Gateway in DIN Rail Plastic Housing	IPEH-004010
PCAN-Wireless Gateway DR	CAN to WLAN Gateway in DIN Rail Plastic Housing	IPEH-004011
PCAN-Wireless Gateway	CAN to WLAN-Gateway	IPEH-004020 IPEH-004020-A
PCAN-Ethernet Gateway FD DR	CAN FD to LAN Gateway in DIN Rail Plastic Housing	IPEH-004012

PEAK-System Technik GmbH
Otto-Roehm-Strasse 69
64293 Darmstadt
Germany

Phone: +49 (0)6151 8173-20
Fax: +49 (0)6151 8173-29

www.peak-system.com
info@peak-system.com

Document version 1.2 (2022-01-24)

Contents

1 Introduction 4

1.1 Properties at a Glance 5

2 Installing the Software 6

3 Capture some traffic 7

3.1 Setup Wireshark™ for Decode CAN Data 7

3.2 Start Analyzing Network Traffic 9

3.3 Fragmented CAN Packages 11

3.5 IP Frame Format Description 12

3.5.1 Optional CRC32 Checksum 17

3.6 Notes about the License 18

1 Introduction

The PEAK-System Wireshark™ Dissector is a simple LUA AddIn for the free available Network Analyzer Wireshark™.

Wireshark™ is the world's foremost network protocol analyser. It lets you see what's happening on your network at a microscopic level. It is the de facto (and often de jure) standard across many industries and educational institutions.

Wireshark development thrives thanks to the contributions of networking experts across the globe. It is the continuation of a project that started in 1998.

See `https://www.wireshark.org`

PEAK-System offer a free of charge LUA AddIn that could be easily used to decode Network Traffic from/to the PEAK-System Network to/from CAN Interfaces via Wireshark™

1.1 Properties at a Glance

- Easy to integrate into an existing Wireshark™ installation
- Open-Source LUA code
- Basic Setup could be done with the Wireshark™ preference menu
- Will work with all Wireshark Implementations, at all available Operating Systems. See https://www.wireshark.org for more details.

Note: This manual describes how to use of the Wireshark Dissector with Wireshark™ Version 3.6.1 on a Windows™ System.

2 Installing the Software

This chapter covers the installation of the LUA Dissector. Up in front you must install the Wireshark Software. https://www.wireshark.org/#download.
For Windows™ is also a Portable Version available.

The Wireshark installer contains also the latest WinPcap installer. If you don't have WinPcap installed, you won't be able to capture live network traffic, but you will still be able to open saved capture files. By default, the latest version of WinPcap will be installed. If you don't wish to do this or if you wish to reinstall WinPcap you can check the Install WinPcap box as needed.

For more information about WinPcap see https://www.winpcap.org/ and https://wiki.wireshark.org/WinPcap.

Do the following to install the Dissector.:

1. Copy the Start `pcan_gateway.lua` from the supplied Mediao or Download to the root of your Wireshark™ installation.
2. Goto the `x:\WiresharkPortable\App\Wireshark\` Directory and search for a file called `init.lua` (depending on your selected Wireshark™ Installation Path).
3. Open the `init.lua` file with a Text Editor (for Example Notepad)

   Copy this Text block at the end of the File:

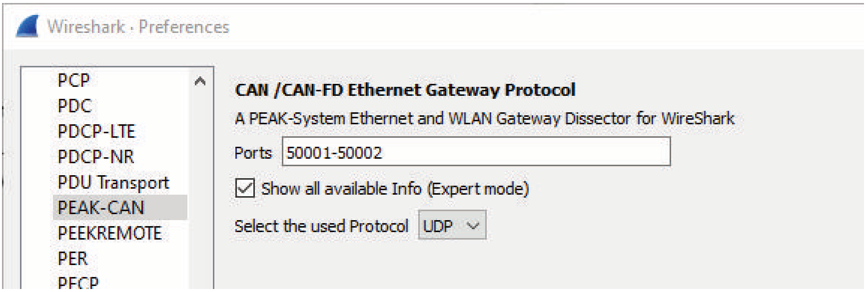
   ```
   -- Add PEAK Network CAN Protocol at startup
   PEAK_PROTO_SCRIPT_PATH="x:\\WiresharkPortable\\"
   dofile(PEAK_PROTO_SCRIPT_PATH.."pcan_gateway.lua")
   ```

   Change the PATH for your need.
4. Start the Wireshark Tool and check if the Dissector was loaded successfully. For that simply open the Wireshark™ preferences Dialog by going to the main menu and select "***Edit – Preferences***" (Ctrl+Shift+P). In the section ***Protocols*** you should now see a entry "PEAK-CAN"

3 Capture some traffic

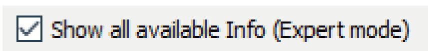
3.1 Setup Wireshark™ for Decode CAN Data

When you have startup the Wireshark Tool, open the Preferences and go to Protocols. Search for *PEAK-CAN*.



In the first Edit field, you can set a Port Range, or a single Port Value where the Dissector should listen to.

The CheckBox allow you to set/unset a enhanced decoding. When the Check Box is set, the Dissector also decode the not used BitFields in the Data Package.



Here a samples of a CAN-FD Frame transmittet with UDP, with Experet mode off and on.

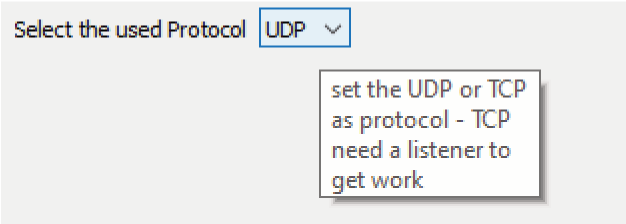
Expert mode off:

```
CAN Protocol Data - PEAK-System Format
    Complete CAN Data size in Frame Length: 192
    CAN FD Frames with CRC: 2
    CAN Data Storage with 2 entrys
        TimeStamp-Low in µS: 40656768
        TimeStamp-High in µS: 228762
        CAN-ID: 0x000ffdff
        Ext. 29Bit Msg.
        DLC: 15
        DB00:0x43 DB01:0x24 DB02:0x32 DB03:0x32 DB04:0x42 DB05:0x03 DB06:0x00 DB07:0x00
        DB08:0x00 DB09:0x00 DB10:0x00 DB11:0x00 DB12:0x00 DB13:0x00 DB14:0x00 DB15:0x00
        DB16:0x00 DB17:0x00 DB18:0x00 DB19:0x00 DB20:0x00 DB21:0x00 DB22:0x00 DB23:0x00
        DB24:0x00 DB25:0x00 DB26:0x00 DB27:0x00 DB28:0x00 DB29:0x00 DB30:0x00 DB31:0x00
        DB32:0x00 DB33:0x00 DB34:0x00 DB35:0x00 DB36:0x00 DB37:0x00 DB38:0x00 DB39:0x00
        DB40:0x00 DB41:0x00 DB42:0x00 DB43:0x00 DB44:0x00 DB45:0x00 DB46:0x00 DB47:0x00
        DB48:0x00 DB49:0x00 DB50:0x00 DB51:0x00 DB52:0x00 DB53:0x00 DB54:0x00 DB55:0x00
        DB56:0x00 DB57:0x00 DB58:0x00 DB59:0x00 DB60:0x00 DB61:0x00 DB62:0x00 DB63:0x00
        CRC: 3093460400
```

Expert mode on:

```
CAN Protocol Data - PEAK-System Format
    Complete CAN Data size in Frame Length: 192
    CAN FD Frames with CRC: 2
    CAN Data Storage with 2 entrys
        CAN-Channel: 0
        internal Flags:
            - Extended ID
            - Bit Rate Switch
            - Error State Indicator
        CAN MessageType: CAN FD with CRC Frame
        TimeStamp-Low in µS: 40656768
        TimeStamp-High in µS: 228762
        CAN-ID: 0x000ffdff
        Ext. 29Bit Msg.
        DLC: 15
        DB00:0x43 DB01:0x24 DB02:0x32 DB03:0x32 DB04:0x42 DB05:0x03 DB06:0x00 DB07:0x00
        DB08:0x00 DB09:0x00 DB10:0x00 DB11:0x00 DB12:0x00 DB13:0x00 DB14:0x00 DB15:0x00
        DB16:0x00 DB17:0x00 DB18:0x00 DB19:0x00 DB20:0x00 DB21:0x00 DB22:0x00 DB23:0x00
        DB24:0x00 DB25:0x00 DB26:0x00 DB27:0x00 DB28:0x00 DB29:0x00 DB30:0x00 DB31:0x00
        DB32:0x00 DB33:0x00 DB34:0x00 DB35:0x00 DB36:0x00 DB37:0x00 DB38:0x00 DB39:0x00
        DB40:0x00 DB41:0x00 DB42:0x00 DB43:0x00 DB44:0x00 DB45:0x00 DB46:0x00 DB47:0x00
        DB48:0x00 DB49:0x00 DB50:0x00 DB51:0x00 DB52:0x00 DB53:0x00 DB54:0x00 DB55:0x00
        DB56:0x00 DB57:0x00 DB58:0x00 DB59:0x00 DB60:0x00 DB61:0x00 DB62:0x00 DB63:0x00
        CRC: 3093460400
```

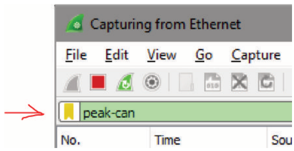
The Last Option is for selecting the Protocoll which should be used by the Dissector. TCP or UDS. Please be sure that you select the correct Type!



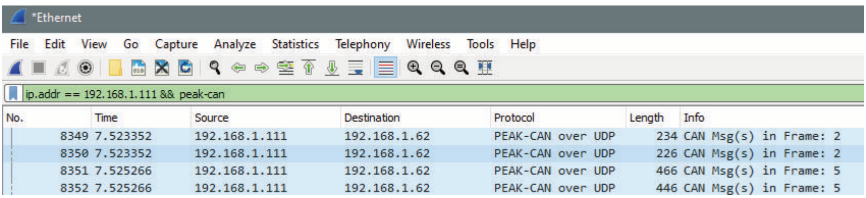
3.2 Start Analyzing Network Traffic

After selected the correct Network Interface (be sure you know on which Network you Gateways are running) you could now start capturing the Network traffic. If you run on the same Network Interface other Application, or you have only one physical Network Interface on your PC, you better setup a filter. Otherwise, you also see all the other Network Traffic in the Wireshark Tool.

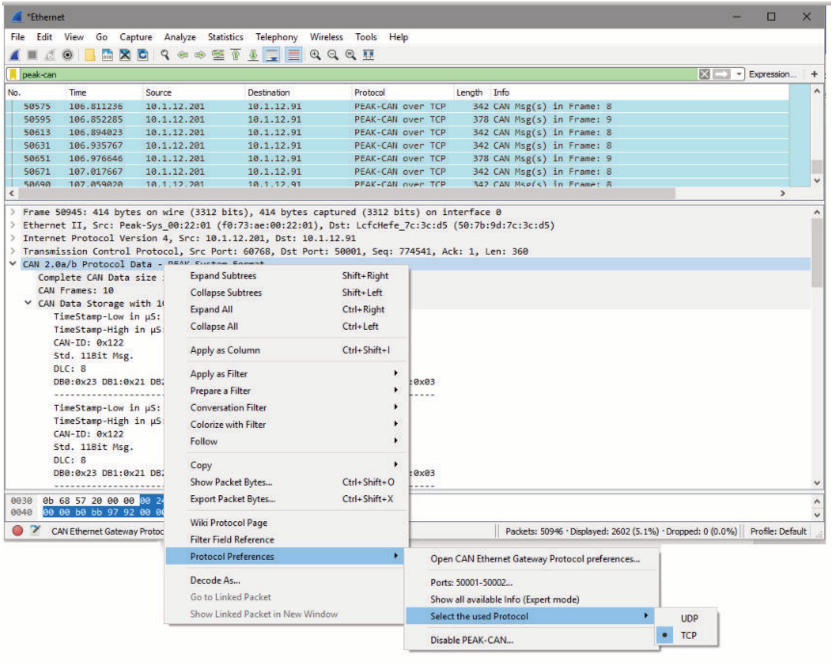
Simply add “peak-can” in the Filter line to be sure that you only see Ethernet Packages that are send over the ports you have selected in the PORT settings. Or simply filter by the IP Address of the Gateway. Both is possible, also in combination.



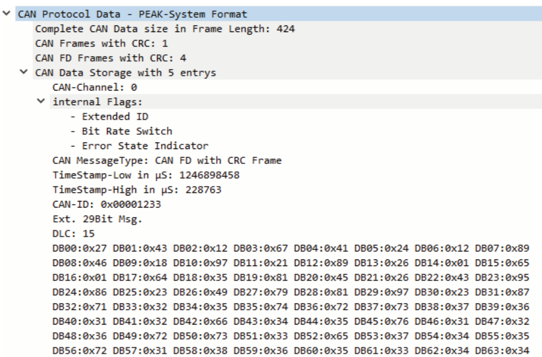
Sample: ip.addr == 192.168.1.111 && peak-can



Also check if you have selected the correct protocol type.



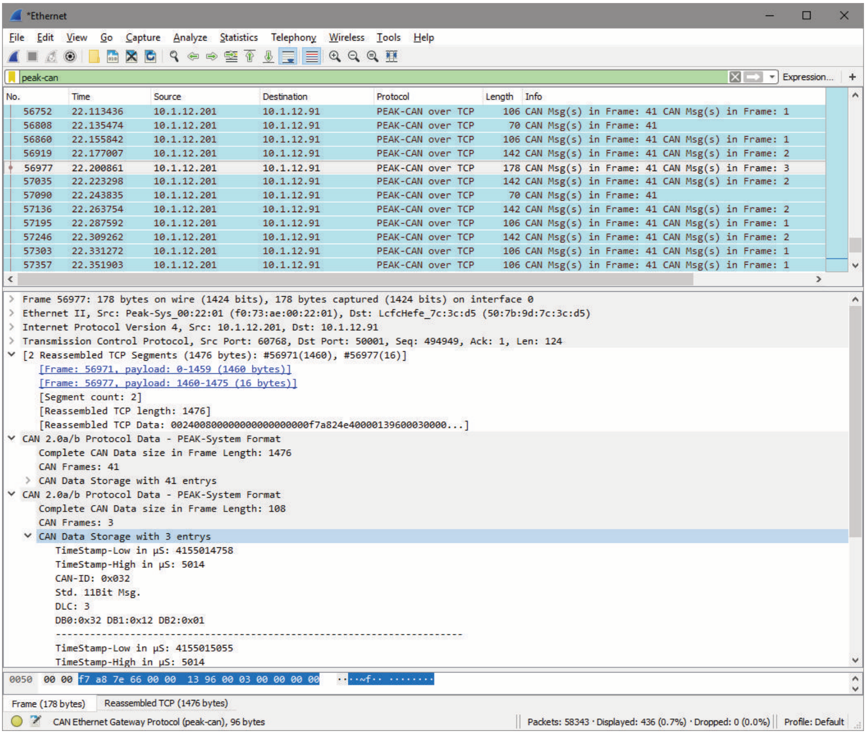
If all is setup correct, you now could see the decoded CAN Data inside the Ethernet Packages.



3.3 Fragmented CAN Packages

When using UDS, you could configure the CAN Data count (max.) inside an Ethernet Package on the Gateways. With TCP this is not possible. The size is fixed defined by the MTU of 1500 which the Ethernet Gateway(s) use. So, it could be possible that the Package will be send meanwhile the CAN Frame is not completely.

The TCP Protocol itself reassemble these packages.



3.5 IP Frame Format Description

PCAN-Gateways allow the connection of various CAN busses over IP networks. For this CAN frames are wrapped in TCP or UDP messages packets and transmitted over the IP network from one device to another. If all the precautions are taken, you can also use a socket to send and receive CAN data via UDP or TCP.

Depending on the type of the CAN message and if the CRC feature is used, the CAN data is transmitted with a different structure in the IP frame. The possible data structures are shown in the following tables.

The values are stored in Network Byte order. The CAN data is stored as single bytes in ascending order. Whether you send or receive, the structure remains the same. With simple TCP/UDP implementations such as in embedded applications, it is possible to receive the TCP or UDP header in addition.

We recommend to study the extra document “*PCAN-Gateways - Developer Documentation*“ which is available for download from the Download Section of the Product pages of the Ethernet Gateway.

Data structure for CAN 2.0 A/B frames

Length	Field Name	Meaning
2 Byte	Length	This field specifies the total length of the packet including this Length field in bytes. The maximum length of a classic CAN frame is 0x24, decimal 36.
2 Byte	Message Type	This field specifies the type of the message. The value **0x80** represents a classic CAN frame.
8 Byte	Tag	Not used in the current version.
4 Byte	Timestamp Low	Timestamp of CAN messages in µs. The value has no effect on the transmission of frames. This information is purely informative.
4 Byte	Timestamp High	
1 Byte	Channel	Not used in the current version. **Note:** The CAN channel is determined by the route configuration.
1 Byte	DLC	The Data Length Count (DLC) indicates the length of the CAN data in bytes.
2 Byte	Flags	The following flags are defined for this frame type: • 0x01 - Message is a Remote Transmission Request (corresponds to bit 30 in the ID field) • 0x02 - Message has an Extended ID (corresponds to bit 31 in the ID field)
4 Byte	CAN ID	Bit 0 - 28: ID Bit 29: Fixed value 0 Bit 30: RTR Bit 31: 1 for Extended frame, 0 for Standard frame.
8 Byte	CAN Data	This field always contains 8 x 8 data bits. **Note:** Use only as many bytes as the DLC indicates. All the following bytes are available but invalid.

Data structure for CAN FD frames

Length	Field Name	Meaning
2 Byte	Length	This field specifies the total length of the packet including this Length field in bytes. The length of the packet for CAN FD frames is variable in contrast to the classic CAN frame. Only as many bytes as necessary are transmitted. The maximum length of a CAN FD frame is 0x60, decimal 96.
2 Byte	Message Type	This field specifies the type of the message. The value **0x90** represents a CAN FD frame.
8 Byte	Tag	Not used in the current version.
4 Byte	Timestamp Low	Timestamp of CAN messages in µs. The value has no effect on the transmission of frames. This information is purely informative.
4 Byte	Timestamp High	
1 Byte	Channel	Not used in the current version. **Note:** The CAN channel is determined by the route configuration.
1 Byte	DLC	The Data Length Count (DLC) indicates the length of the CAN data in bytes.
2 Byte	Flags	The following flags are defined for this message type: • 0x02 - Message has an Extended ID (corresponds to bit 31 in the ID field) • 0x10 - Message with Extended Data Length • 0x20 - Message with activated Bit Rate Switch • 0x40 - Message with set Error State Indicator bit
4 Byte	CAN ID	Bit 0 - 28: ID Bit 29: Fixed value 0 Bit 31: 1 for Extended frame, 0 for Standard frame.
N Byte	CAN Data	This field contains the CAN data bytes. The number of bytes transmitted corresponds to the length specified in the DLC field.

Data structure for CAN 2.0 A/B frames with CRC

Length	Field Name	Meaning
2 Byte	Length	This field specifies the total length of the packet including this Length field in bytes. The maximum length of a classic CAN frame with additional CRC checksum is 0x28, decimal 40.
2 Byte	Message Type	This field specifies the type of the message. The value **0x81** represents a classic CAN frame with additional CRC checksum.
8 Byte	Tag	Not used in the current version.
4 Byte	Timestamp Low	Timestamp of CAN messages in µs. The value has no effect on the transmission of frames. This information is purely informative.
4 Byte	Timestamp High	
1 Byte	Channel	Not used in the current version. **Note:** The CAN channel is determined by the route configuration.
1 Byte	DLC	The Data Length Count (DLC) indicates the length of the CAN data in bytes.
2 Byte	Flags	The following flags are defined for this frame type: • 0x01 - Message is a Remote Transmission Request (corresponds to bit 30 in the ID field) • 0x02 - Message has an Extended ID (corresponds to bit 31 in the ID field)
4 Byte	CAN ID	Bit 0 - 28: ID Bit 29: Fixed value 0 Bit 30: RTR Bit 31: 1 for Extended frame, 0 for Standard frame.
8 Byte	CAN Data	This field always contains 8 x 8 data bits. **Note:** Use only as many bytes as the DLC indicates. All the following bytes are available but invalid.
4 Byte	CRC32	CRC checksum. See the following chapter for details.

Data structure for CAN FD frames with CRC

Length	Field Name	Meaning
2 Byte	Length	This field specifies the total length of the packet including this Length field in bytes. The length of the packet for CAN FD frames is variable in contrast to the classic CAN frame. Only as many bytes as necessary are transmitted. The maximum length of a CAN FD frame with additional CRC checksum is 0x64, decimal 100.
2 Byte	Message Type	This field specifies the type of the message. The value **0x91** represents a CAN FD frame with additional CRC checksum.
8 Byte	Tag	Not used in the current version.
4 Byte	Timestamp Low	Timestamp of CAN messages in µs. The value has no effect on the transmission of frames. This information is purely informative.
4 Byte	Timestamp High	
1 Byte	Channel	Not used in the current version. **Note:** The CAN channel is determined by the route configuration.
1 Byte	DLC	The Data Length Count (DLC) indicates the length of the CAN data in bytes.
2 Byte	Flags	The following flags are defined for this message type: • 0x02 - Message has an Extended ID (corresponds to bit 31 in the ID field) • 0x10 - Message with Extended Data Length • 0x20 - Message with activated Bit Rate Switch • 0x40 - Message with set Error State Indicator bit
4 Byte	CAN ID	Bit 0 - 28: ID Bit 29: Fixed value 0 Bit 31: 1 for Extended frame, 0 for Standard frame.
N Byte	CAN Data	This field contains the CAN data bytes. The number of bytes transmitted corresponds to the length specified in the DLC field.
4 Byte	CRC32	CRC checksum. See the following chapter for details.

3.5.1 Optional CRC32 Checksum

CAN frames can be transmitted over the IP network with an additional CRC32 checksum in the IP packet. It is created from DLC, flags, CAN ID, and data of the CAN frame together with a CRC start value and polynomial. This option was introduced with software version 2.8.1 for classic CAN frames and for CAN FD frames with software version 1.0.0 of the PCAN-Ethernet Gateway FD DR.

The CRC option for routes can be configured via the JSON interface or the INI file. For a valid connection with CRC option, the CRC settings of the Send and Receive route must match.

CRC32 example for an incoming message:

```
0028 0081 0000000000000000 060DE1E10005BD06 01       08  ...
|    |    |                |                |        |
Size Type Tag              Timestamp        Channel  DLC

...  0000  00000111 FFEEDDCCBBAA9988 49198620
     |     |        |                |
     Flags ID       Data             CRC32
```

The colored values DLC, flags, CAN ID, and CAN data are used for the calculation.

```
crc32([0x08, 0x00, 0x00, 0x00, 0x00, 0x01, 0x11, 0xFF, 0xEE, 0xDD,
0xCC, 0xBB, 0xAA, 0x99, 0x88]) = 0x20861949
```

There is no XOR with 0xFFFFFFFF at the end of the calculation. The CRC32 result is added to the message in little endian byte order.

3.6 Notes about the License

Device drivers, the interface DLL, and further files needed for linking own applications are property of the PEAK-System Technik GmbH and should be used only in connection with a hardware component purchased from PEAK-System or one of its partners. If a CAN hardware component of third-party suppliers should be compatible to one of PEAK-System, then you are **not allowed** to use or to pass on the driver software of PEAK-System.

If a third-party supplier develops software based on the PCAN-Basic and problems occur during the use of this software, consult the software provider.