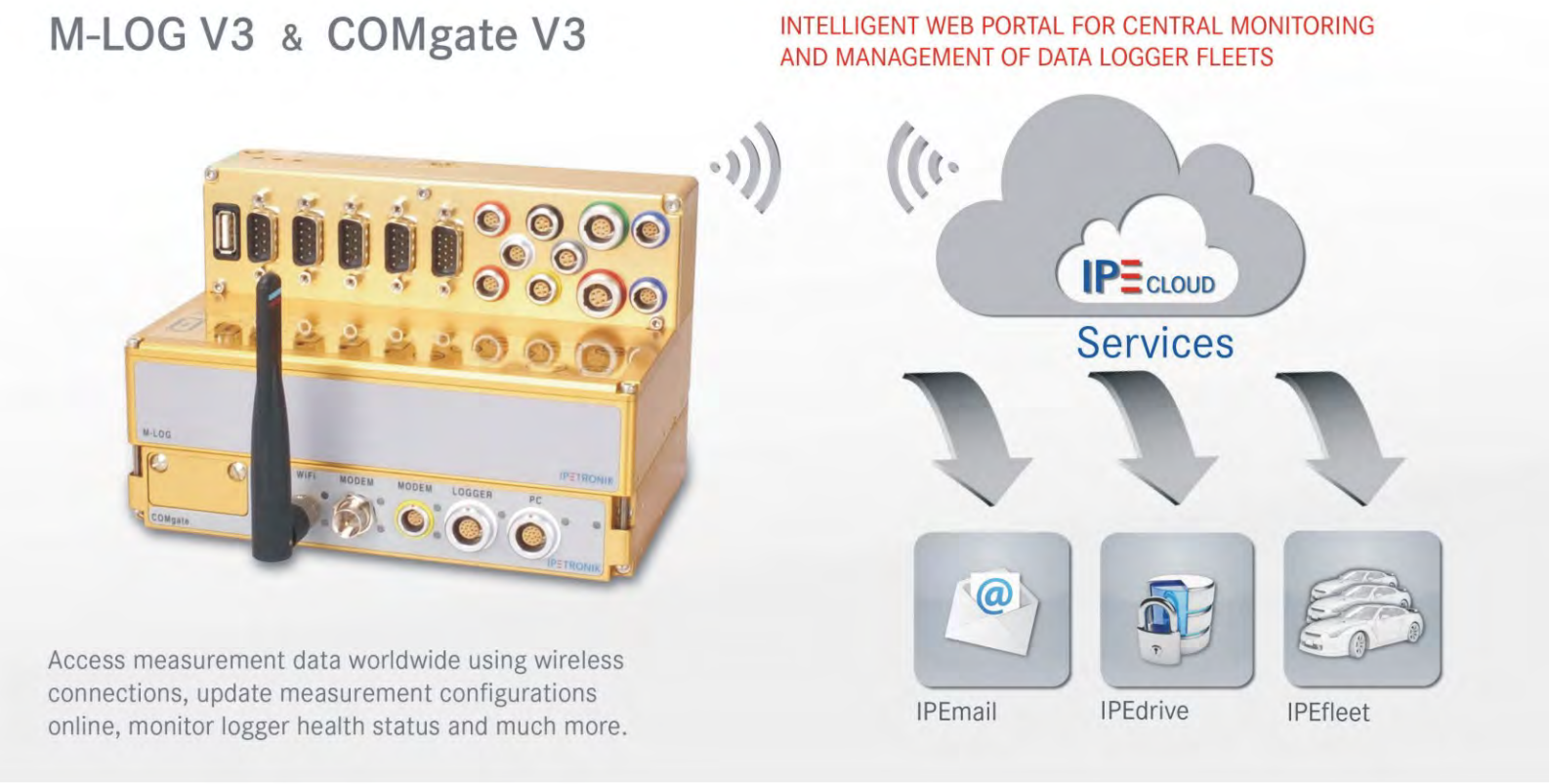
IPETRONIK




IPE MEASURE
Instrumentation

# IPEmotion PlugIn IPETRONIK-LOG V03.60.00

**February 2018**

# Table of contents

# 1 Important and general information

## 1.1 Important information

**Please follow these instructions before and during the use and application on any IPETRONIK product!**

### 1.1.1 Safety and Warning instructions

Please follow the instructions **and** information as contained in the user manual!

1. The user can **influence an electronic system by applying the IPETRONIK product**. This might cause risk of personal injury or property damages.
2. The **use and application of the IPETRONIK product is permitted only to qualified professional staff**, as well as, only in appropriate manner and in the designated use.
3. **Before using an IPETRONIK measurement system** in the vehicle it **has to be verified** that **no function of the vehicle, which is relevant for secure operation, might be influenced**:
   - by the installation of the IPETRONIK measurement system in the vehicle,
   - by an potential malfunction of the IPETRONIK system during the test drive.

   In order to avoid possible danger or personal injury and property damages, appropriate actions are to be taken; such actions have to bring the entire system into a secured condition (e.g. by using a system for emergency stop, an emergency operation, monitoring of critical values).

   Please check the following points to avoid errors:

   - Adaption of sensors to components of the electrical system / electronics, brake system, engine and transmission control, chassis, body.
   - Tap of one or several bus systems (CAN, LIN, ETHERNET) including the required electrical connection(s) for data acquisition.
   - Communication with the vehicle's control units (ECUs), especially with such of the brake system and/or of the engine and transmission control (power train control system).
   - Installation of components for remote data transmission (mobiles, GSM/GPRS modems, WiFi and Bluetooth components).

4. IPETRONIK devices are designed for applications in **extended temperature ranges** > 70 °C (158 °F). A high environmental temperature and the module's self-heating may cause burns of the skin when touching the hot surface. In order to avoid the risk of injury we recommend to take care for appropriate safety precautions (e.g. contact protection, covering/enclosure, warning sign, ... )..
5. **Before** directly or indirectly using **the data acquired by an IPETRONIK measurement system to calibrate control units, please review the data regarding to plausibility**.
6. With regard to the application of IPETRONIK products in vehicles during use on public roads the manufacturer and/or registered user of the vehicle **has to ensure that all changes/modifications have no influence concerning the license of the vehicle or its license of operation.**
7. **User does agree to the instructions and regulations as mentioned above.** In case the user does not agree with the instructions and regulations as mentioned above, he has to notify this expressly and immediately in writing to IPETRONIK before confirming the sales contract.

### 1.1.2 Liability, Warranty, Copyright, License agreement

Please refer to http://www.ipetronik.com/en/terms-conditions for detailed contract information:

- Limitation of liability
- Warranty
- Copyright and Duplication
- Software license agreement

# 1.2 General information

## 1.2.1 About this manual

The manual describes the structure of the IPEmeasue data logger devices M-LOG / M-LOG V3, S-LOG, FLEETlog / FLEETlog2 and IPElog / IPElog2, as well as, peripheral devices and accessories components.



### IPEmotion PlugIn IPETRONIK-LOG

Descriptions in this documentation refer to the current release. Please note, that the logger requires the corresponding application software TESTdrive.

*To run this PlugIn an release ≥ IPEmotion 2017 R1 has to be installed on your computer.*

### IPEmotion

Contents described in this document relates to the release versions IPEmotion 06.00.00 (2016) to IPEmotion 07.02.00 (2017 R3).

IPEmotion 2017 requires Microsoft .NET 4.5.1 Framework. This version is no longer supported by Windows XP.

## 1.2.2 Legend of used icons

***Tip*** *This icon indicates a useful tip that facilitates the application of the software.*

***Information*** *This icon indicates additional information for a better understanding.*

***Attention!*** *This icon indicates important information to avoid potential error messages.*

### 1.2.3 New features, Changes

**Please also refer to the latest release notes at:**
**c:\Program Files (x86)\IPETRONIK\IPEmotion PlugIn IPETRONIK LOG V03.5x.xx\Help**

| PlugIn IPETRONIK-LOG V03.60 | | Release January 2018 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | PlugIn LOG / X in parallel | Activating the PlugIn IPETRONIK-LOG and the PlugIn IPETRONIK-X (simultaneous operation) is supported. |
| 2 | openABK functions | Implemented functions of the logger display & control concept openABK buttons, openABK DHCP server, openABK unique names |

| PlugIn IPETRONIK-LOG V03.59 | | Release November 2017 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | IPElog2 access point | WiFi access point support |
| 2 | IPEwifi V3 | External device for WiFi data transfer with M-LOG V3 |
| 3 | Transfer category FTP server | Up to 3 categories can be assigned to an FTP server |
| 4 | Transfer category LOG files | Up to 3 categories can be assigned to the log file |
| 5 | XCP: Disconnect Second Tester | Delay time after the the second tester will be disconnected |
| 6 | ECU CCP parameters | Optionally use CCP parameters from ECU |
| 7 | UDS security access | Security access for the UDS protocol |
| 8 | SFTP resume | Resume is active: The data transfer will be continued where it has been stopped with the previous transfer.<br>Resume inactive: A corresponding and already existing file on the FTP server will be overwritten. |
| 9 | Transmit traffic messages | Transmit messages in traffic groups |
| 10 | IPElog2 gyro sensor | Support of IPElog2 internal gyro sensor |
| 11 | Data transfer parameters | Default values for data transfer parameters have been adapted |
| 12 | Project properties | Project properties are no longer stored in mea_conf |
| 13 | PlugIn LOG / X in parallel | PlugIn IPETRONIK-LOG and IPETRONIK-X activated in parallel might cause software problems |

| PlugIn IPETRONIK-LOG V03.58 | | Release February 2017 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | Cisco VPN removed | Cisco VPN settings will be deleted when loading a former configuration (GUI access no longer supported). |
| 2 | SW filter frequency selection | Software filter frequency now as drop-down menu selection |
| 3 | Export of channel comments | Activation at Options > PlugIns > IPETRONIK LOG > IPEmotion settings > Options > Extended > IPETRONIK CAN, default = deactivated |
| 4 | New OBD channels | Supported PIDs: 102 - 106, 108 - 110, 112 - 124, 127, 131 |
| 5 | Radius calculation (polar) | Radius = SQRT( "phi_y" * "phi_y" + "phi_z" * "phi_z" ) |
| 6 | Angle calculation (polar) | Angle = IF( "phi_z" >=0; ACOS("phi_y" / "Radius" ) * 180/PI; -1 * |

| | | |
|---|---|---|
| | | ACOS("phi_y" / "Radius" ) * 180/PI + 360) |
| 7 | Calculation ISNOVSALUE | ISNOVALUE (x;y), NoValue detection with delay |
| 8 | External video data storage | Storing video data on external USB medium |
| 9 | Sign-of-live | Connecting status between logger and web interface |
| 10 | Measurement status file with wake up reason | New entry wake up reason:<br>Unknown/ Remote/ Wake On CAN/ No Message Lost/ Wake On RTC/ Modem (SMS)/ low voltage |

| PlugIn IPETRONIK-LOG V03.58 | | Release February 2017 (continuation) |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 11 | TESTdriveCmd DataTransfer | The new parameter „transfer“ controls the data post processing. Depending on its status (true/ false) zipped and splitted files will be copied to the USB stick subsequently. |
| 12 | Remote2 (M-LOG V3, IPElog2) | 2nd input in addition to the control input Remote 1 (e.g. ignition line 15) as OR-conjunction solely supported with the Power-IN/Remote connector of M-LOG V3 and IPElog2 |

| PlugIn IPETRONIK-LOG V03.57 | | Release July 2016 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | IPElog2 support | Hardware type 10x CAN, 6x LIN, 2x ETH, WLAN<br>Hardware type 16x CAN, 2x ETH, WLAN |
| 2 | Quickstart data options | Off, During boot-up, During boot-up + Measurement Stop-Start |
| 3 | Available WLAN networks | Cyclic SSID scan provides latest WLAN status information (logger with built-in WLAN unit required) |
| 4 | New UDS super job | PST_LESEN_UDS_2 |
| 5 | UDS check string | Comparing the „SearchString“ of the UDS job and the ECUs reply |
| 6 | Indication of CAN/LIN bus activity | Creating status entry in Log file / Measurement status file once the timeout delay of the respecitve CAN/LIN input expired |
| 7 | Hardware description file | New backup of the HW_descr.xml prevents from illegal access |
| 8 | Data post processing | User configurable Start Delay / Retry Delay time (range 10 s … 5 min) to trigger the parallel data post processing |
| 9 | Use last value from previous measurement | Using the last value from previous measurement as start value for a signal calculation within the current measurment |
| 10 | Category overview | Selection list for the data transfer category |
| 11 | XCPonUDP import | Import of A2L description files using an USB2ETH adapter |

| PlugIn IPETRONIK-LOG V03.56 | | Release April 2016 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | Multi CAN-Send | CAN node supporting multiple CAN-Send blocks |
| 2 | GPS channels added | GPS latitude in degrees, GPS longitude in degrees |
| 3 | openABK capable display | Support of the openABK protocol V1.0 for EMBU-SYS displays |
| 4 | Logger status info | Web interface showing logger status information and signal list |
| 5 | IPEconnect access point | IPEhub2 supports access point operation for online data visualization with mobile devices (smartphone, tablet) |
| 6 | J1939 extension | Event triggert measurement in signal mode |
| 7 | UDS extension Second Tester | Configuration of an additional tester ID |

| PlugIn IPETRONIK-LOG V03.55 | | Release August 2015 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | M-LOG V3 | Support of new data logger M-LOG V3 |
| 2 | FLEETlog2-03 | Support of FLEETlog2 with D Sub D connector (CAN, DI/O) |
| 3 | IPElog 6x CAN, 6x LIN | Data logger IPElog providing 6 CAN and 6 LIN inputs available |
| 4 | M-VIEWfleet, scaling | Multipoint scaling for M-VIEWfleet supported |
| 5 | USB video | - Support of multiple USB cameras via USB hub<br>- Support of Logitec QuickCAM VisionPro (DID 0x09A6) |
| 6 | Mail groups, recipients list | Separate list of mail recipients per mail group |
| 7 | Serial number specific program/configuration update (mcf, fcf, rtb, prg) | Verification of the logger serial number for program/configuration update (mcf, fcf, rtb, prg). In case of missmatch, the latest valid file is used. |
| 8 | Import of PDX files | One or more PDX files can be imported for each CAN node |
| 9 | NoValue group | Extended NoValue monitoring with with new data group „NoValue group“ (Logger processing). User defined formulas for trigger conditions are supported. |
| 10 | Send category „NoTransfer“ | Additional send category „NoTransfer“ in order to exclude user defined data from transfer. |
| 11 | CCP, XCP second tester | Check for second tester connected to the bus before starting ECU communication. |
| 12 | CCP, XCP EPK missmatch | Selectable behaviour in case of EPK missmatch. (default setting: proceed measurement) |
| 13 | New parameter <meaNumber> for TESTdriveCmd.xml | The service <dataTransfer> now provides the new paramater <meaNumber> in order to send defined mea files in advance (per modem), instead of at the end of the test drive trip. |
| 14 | IPEconverter | Refer to manual IPEconverter V03.55 for latest extensions |

| PlugIn IPETRONIK-LOG V03.54 | | Release November 2014 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | M-LOG 6 CAN<br>M-LOG 3 CAN / 1 LIN | Support of new M-LOG measurement interfaces |
| 2 | New CAN devices | Support of M-SENS2 250 Hz, M-UNI2, M-THERMO2 HV, CANpressure 150 bar |
| 3 | Diagnostic measurement | Set trigger conditions to start diagnostic jobs |
| 4 | CAN Timeout value | User defined value for the timeout. Without a valid signal value after the timeout has expired, the system will indicate “NoValue” |
| 5 | DAQ polling groups (polling list) | ECU signals can be allocated to different polling groups. Up to 4 groups can be defined by user |
| 6 | Seed&Key file using the XCP protocol | Set file location for the Seed&Key file. |
| 7 | XCPonCAN extension | New Protocol status channel with XCP, CCP, KWP, UDS |
| 8 | CAN-Send extensions | New columns bit count and data format added |

| PlugIn IPETRONIK-LOG V03.54 | | Release November 2014 (continuation) |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 9 | Statistic Group (Min/Max list) (STG file) | Activation of the min/max list calculation has been modified. Requested signals can be assigned to max number of 4 statistic groups. |
| 10 | File transfer categories | Storage, traffic and statistic groups can be asigned to a file transfer category. Each of 3 categories can uses a data transfer medium (LAN, WLAN or COMgate). |
| 11 | Disable USB stick data exchange | Automatic data download / configuration upload can be enabled/disabled with the PlugIn options settings. |
| 12 | Logger reset | Reset the logger to factory default settings (DIN 01 = active) |
| 13 | IPEcloud | Define access parameters for data download from FTP server. Measurement data can be iported directly using IPEmotion‘s Data manager and Analysis tools. |
| 14 | J1939 extensions | Activate J1939 protocol information<br>DM1 messages conversion methods |
| 15 | DLM | Optional path for user operations (DLM)<br>C:\Users\Public\Documents\IPETRONIK\IPEmotion\Custom\User Operation |
| 16 | XCPonCAN and XCPonUDP with additional storage rates | 30/min, 12/min, 6/min, 1/min, 2/h, 1/h |
| 15 | IPElog extensions<br>WakeOnSMS<br>Rescue configuration<br>Provider blacklist | <br>Start IPElog per SMS from standby mode<br>Export rescue configruation<br>Create provider blacklist to exclude them for data transfer |
| 16 | Time zone status information | Enable the use of time zone status information (UTC status) with the PlugIn options settings |
| 17 | exFAT format of the SSD card used with IPElog | Format SSD drive using the exFAT data format which support memory sizes up to 512 Terra bytes |
| 18 | S.M.A.R.T. | Support of Self-Monitoring, Analysis and Reporting Technology with the appropiate storage medium |
| 19 | Event based measurements in MDF4 data format | Channels of a CAN measurement with disabled cyclic acquisition (Event controlled) will be stored using the MDF4 format. |
| 20 | Count ECU data requests | All ECU data requests (means successful and unsuccessful attempts) will be logged. |
| 21 | Mail groups | Support of sending e-mails parallel to data post processing. |
| 22 | IPEconverter extensions<br>ASAM ATF/XML export<br>DIAdem TDM/DAT export<br>CSV, Excel2003/2010 export | <br>NoValue handling, grouping mode<br>new data format types<br>V-TAB scaling, rounded floating values |

| PlugIn IPETRONIK-LOG V03.53 | | Release April 2014 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | LOG2PC (ETH communication) IP 239.192.0.5 at port 7302 | The UPD based protocol uses multicasts now, Firewall settings must not refuse the use of this port/IP. |
| 2 | TESTdrive update to V03.53.xx | Execute inital update of the logger software by USB stick. Following updates are supported by LOG2PC as usual. |

| PlugIn IPETRONIK-LOG V03.52 | | Release December 2013 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | FLEETlog2 support | Data logger FLEETlog2 is supported now |
| 2 | CAN-Send extensions | Send counter with individual settings for start bit, bit length, and data format, Signal output with individual settings for start bit, bit length, and data format in expert mode |
| 3 | J1939 extensions | J1939 protocol including DM1 signals<br>refer to separate documentation J1939_DM1 |
| 4 | OBD 2 extensions | Single PID, status channel, trigger conditions |
| 5 | External storage medium (USB) | Selectable for signal and traffic storage groups<br>TESTdriveCmd.xml required |
| 6 | Traffic measurement extensions | Ring buffer, Filter functions |
| 7 | CAN ID assignment | Automatic assignment during system identification > IPEmotion Options > PlugIn specific settings |
| 8 | Measurement stop with automatic restart | StopStart event (logger processing) stops the current measurement and restarts immediately |
| 9 | New status signals | Video file size, Measurement number, OBD-2 processing |
| 10 | New calculation | INT_ADD() |
| 11 | Hardware license information | Logger context menu Extras shows hardware license information from the description file (hw_descr.xml) |
| 12 | Licensing of traffic groups | License verification of storage and traffic groups |

| PlugIn IPETRONIK-LOG V03.51 | | Release July 2013 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | PIN assignment | FLEETlog and IPElog added |
| 2 | Traffic storage groups | Traffic measurement now support different storage groups |
| 3 | Extension for IPElog<br>(requires PIC ≥ V01.01.07,<br>FPGA ≥ V01.02.08 | NoMessageLost function (NML)<br>WakeOnCAN can be activated for each channel<br>up to 6 ID triggers<br>Logger restart using WakeOnCAN or remote signal<br>CAN-ID trigger now StartNotStopTrigger (instead StartOnly)<br>PIC update per job executable<br>Status indication of the flap for the storage medium with M-VIEWfleet |
| 4 | CCP | Command CCP_DISCONNECT implemented |
| 5 | Upload and download | Supported with different transfer media (Modem, WiFi) |
| 6 | Event controlled measurement | CAN data acquisition based on traffic measurement but handled as signal measurement (DAT format) |
| 7 | Masks | for CAN identifiers are supported now |
| 8 | File name length | Up to 260 characters are allowed for file names of the external library and configuration files. |
| 9 | Stop date, Stop time | Stored as project property with the traffic file |
| 10 | PreTriggerTime, PostTriggerTime | Supported as project property |

| PlugIn IPETRONIK-LOG V03.50 | | Release October 2012 |
|---|---|---|
| **No.** | **Feature** | **Description** |
| 1 | **Notes**<br>Version reference<br>SC1200<br>IPElog | <br>Logger PlugIn ≥ 3.5x requires IPEmotion ≥ 2.xx<br>M-LOG with SC1200 is no longer supported<br>Requires TESTdrive / PlugIn IPETRONIK-LOG ≥ 3.5x |
| 2 | Status signal "FIFO overrun" | Indicates the data processing status.<br>Output 0 = OK, Ouput 1 = processor overloaded |
| 3 | Debounce time of the remote signal | Configurable delay time within a range of 0 ... 5 seconds. The signal status ist valid if the remote signal lasts for this time without interruption. |
| 4 | Extensions TESTdriveCmd.xml | New jobs "OnOK" and "OnError" used for audible alarm of the job "OnConnect" |
| 5 | Differnt init modes for module initialization | Connected IPETRONIK modules will be initialized on command: Never, Once-only, Always. |
| 6 | Status e-mail with snap shot report | Each time the trigger condition is fulfilled, the current signal values (configured in the mail group) will be sent per mail. |
| 7 | Limit display with M-VIEWvga | Indication of limit violation has been revised. |
| 8 | V-TABs enable clear text display | Individual text can be assigned to different signal values (or ranges) which is respectively displayed (instead of digits). |

### 1.2.4 Support

**Headquarter:**

**IPETRONIK GmbH & Co. KG**
Im Rollfeld 28

76532 Baden-Baden, Germany

Phone +49 7221 9922 0
Fax +49 7221 9922 100

info@ipetronik.com
www.ipetronik.com

Limited commercial partnership with its head office in Baden-Baden, registry court HRA No. 201313

IPETRONIK Verwaltungs-GmbH Baden-Baden is an individually liable society, registry court Mannheim HRB No. 202089

CEOs: Andreas Wocke, Christian Buchholz

**Technical support and product information**

www.ipetronik.com e-mail: support@ipetronik.com

### 1.2.5 Related documentation

#### IPEmotion

The documentation IPEmotion.pdf provides you with a description and useful information related to IPEmotion. This documentation is stored in the following standard language dependent directory: C:\Programs\IPETRONIK\IPEmotion Vxx.xx.xx\Help

### 1.2.6 Documentation feedback

At IPETRONIK, the technical publications team strives to produce documentations of the highest quality and values your feedback as a reader and user. If you have any comments or suggestions regarding our product manuals, contact us under support@ipetronik.com .

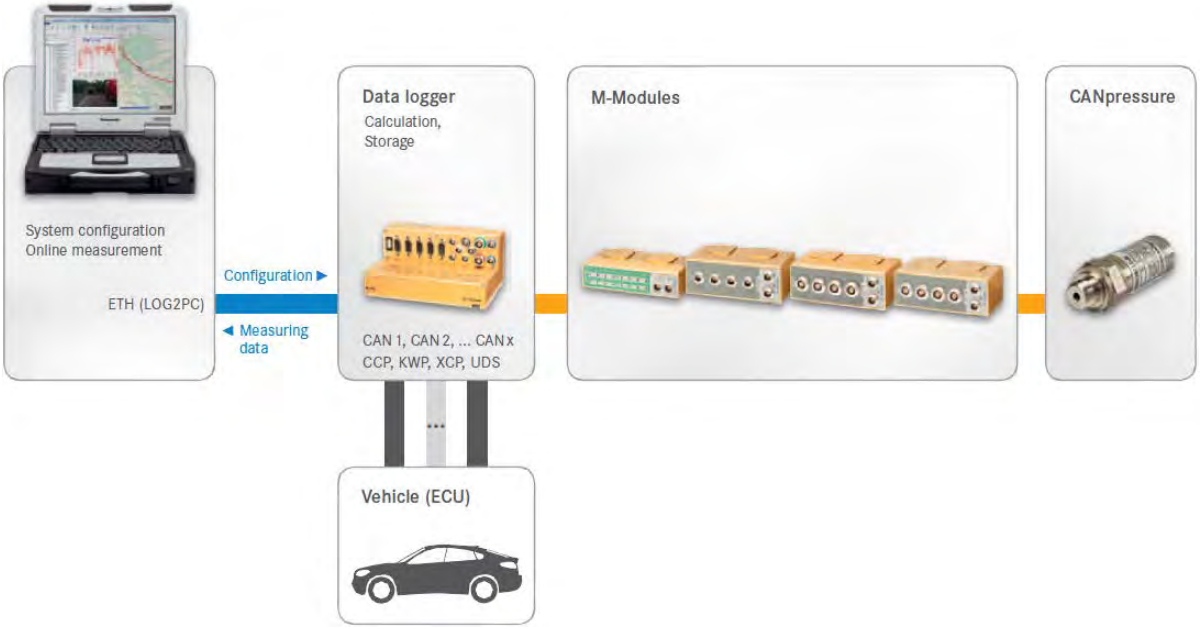
When commenting on our products, please include the following information:

- Version number
- Name of the guide
- Page number or section title
- Brief description of the content (e.g. inaccurate instructions, grammatical errors, or information that require clarification)
- Any suggestions for a general documentation improvement

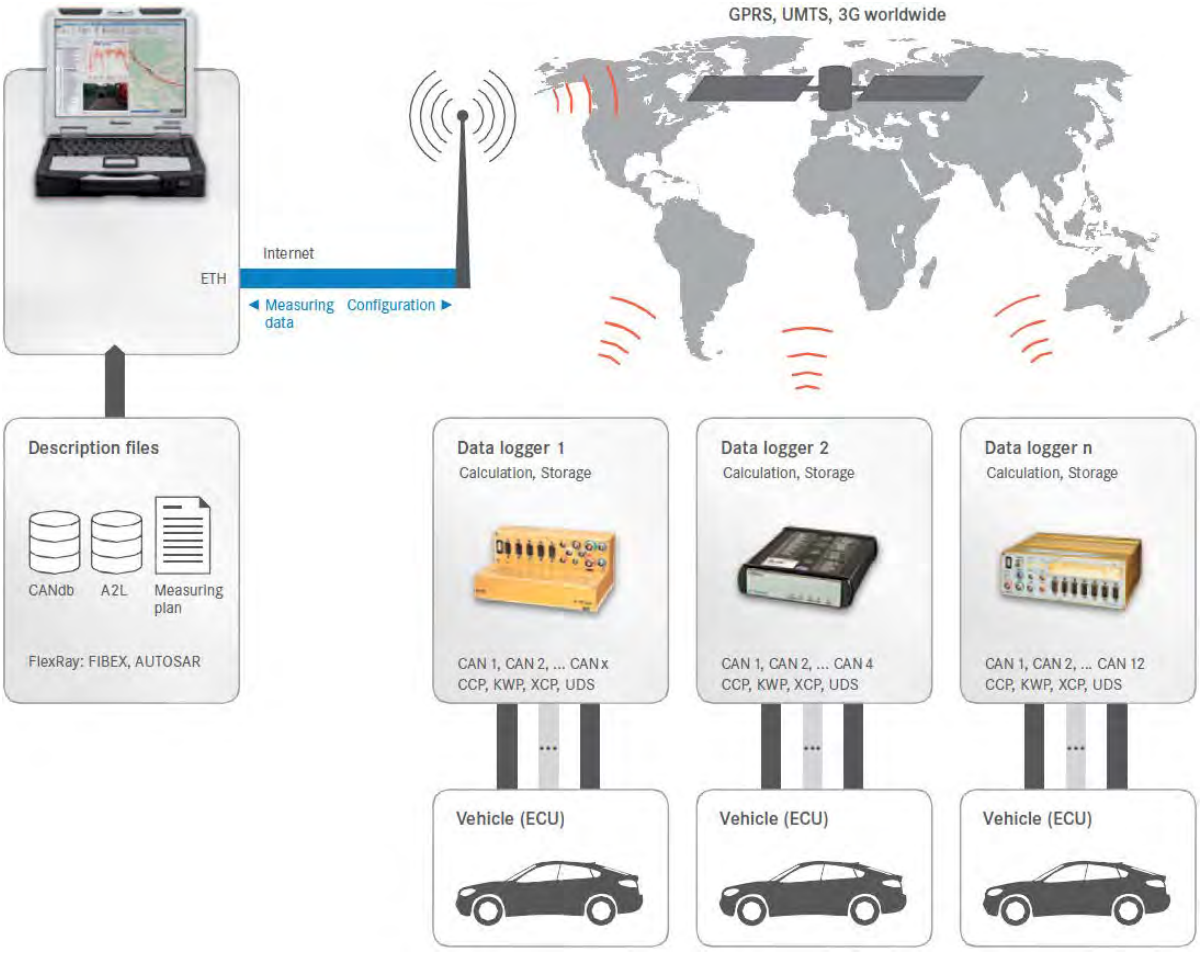
# 2 System basics

## 2.1 Data logger applications (extract)

### 2.1.1 Configuration, Online measurement using Ethernet




### 2.1.2 Fleet data logger



## 2.2 Connecting devices via CAN bus

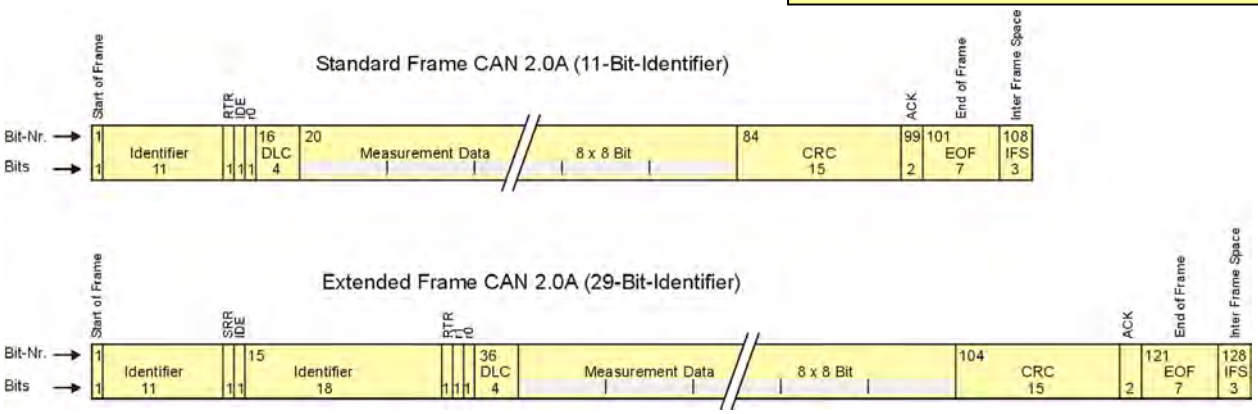
### 2.2.1 CAN bus basics

#### CAN-Standard

The communication of the IPETRONIK SIM and M devices takes place by using the CAN bus according to the CAN 2.0 A (11 Bit Identifier) and CAN 2.0 B (29 Bit Identifier) specification. Each software application, which is able to process CAN data through a suitable interface can acquire measurement data from IPETRONIK CAN devices.

**User data within the CAN message:**
A CAN message can transfer 8 values in Byte format resp. 4 values in Word format.

#### Structure of a CAN message




| CAN 2.0A (11 Bit Identifier) | | | | CAN 2.0B (29 Bit Identifier) | | |
|---|---|---|---|---|---|---|
| **Bits** | | **Description** | | **Bits** | | **Description** |
| 1 | SOF | Start of Frame | | 1 | SOF | Start of Frame |
| 11 | ID | Identifier | | 11 | ID | Identifier |
| 1 | RTR | Remote Transmission Request | | 1 | SRR | |
| 1 | IDE | Identifier Extension (0) | | 1 | IDE | Identifier Extension (1) |
| | | | | 18 | ID | Identifier (extended) |
| | | | | 1 | RTR | Remote Transmission Request |
| | | | | 1 | r1 | |
| 1 | r0 | | | 1 | r0 | |
| 4 | DLC | Number of following data bytes | | 4 | DLC | Number of following data bytes |
| 64 | Data | Data bytes | | 64 | Data | Data bytes |
| 15 | CRC | Error Identification Code | | 15 | CRC | Error Identification Code |
| 2 | ACK | Acknowledge | | 2 | ACK | Acknowledge |
| 10 | EOF | End of Frame, Inter frame space | | 10 | EOFS | End of Frame, Inter frame space |
| | | | | | | |
| 110 | | Total number of bits | | 130 | | Total number of bits |

| Word | Byte | Bit (Message layout in the displaying format „Intel Standard“) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **0** | **0** | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | **1** | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| **1** | **2** | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | **3** | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| **2** | **4** | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| | **5** | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 |
| **3** | **6** | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 |
| | **7** | 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 |

## Access to the CAN bus, Transferring properties

The CAN bus allows a safe and effective data transfer of the connected devices (non-destructive bitwise arbitration = resource distribution to different devices). The CAN bus is therefore used as a standard communication medium in the automotive area and the industrial automation.

**The most important characteristic CAN bus properties are:**

- Every bus participant (node) can send, as well as, receive.
- First of all, the node, which wants to send, needs the authorization. All participants become automatically a recipient (There is no abortion of the data sending process > non-destructive collision).
- No stations are addressed but messages.
- Every message is characterized by its name (Identifier).
- The less the identifier, the higher the message priority.
- A message can transport up to 8 * 8 Bit = 64 Bit (8 Byte) user data, whereas each message requires 110 Bit or 130 Bit (Extended ID).
- Depending on the hardware and the bus line length, up to 1 MBit/ s can be transferred.

**The following important conclusions result from the properties above:**

- The less the bus load, the less the probability of a “Bus access conflict” (you can call this a real- time capable area).
- A high bus load forces stations to loose messages with a high identifier or to send them more slowly. Messages with a high identifier can “get lost”.
- Not sent messages are only registered by the “Recipient node” because data are missing. If no timeout has been defined, the last valid value is generally sent, i.e. a mistakenly constant value.

## Transfer rate, Bus line length

The CAN bus supports a max. transfer rate of 1 MBit/s according to Norm ISO 11898-2.
This value is limited in practice by the following points:

- the bus line length
- the branch line length to the CAN stations
- the bus lines quality and the plug contacts
- the bus line design (twisted, single or two-wire bus)
- bus connection structure and
- type and strength of external perturbations

| **Example** | |
|---|---|
| Data rate on the bus | 1 MBit/s = 1 µs/Bit |
| Data length of a CAN message | 130 Bits total |
| User data in a message | 64 Bit = 4 values with 16 Bit resolution each |
| Time for a CAN message | 130 Bit x 1 µs/Bit = 130 µs/message<br>i.e. 4 values require 130 µs |
| Calculating the total sampling rate | 130 µs match 7.69 kHz |
| Converted to one channel | 4 x 7.69 kHz = 30.76 kHz |
| Theoretical transfer rate | 30 channels with 1 kHz = 30 kHz |
| Practical experiences | 26 channels with1 kHz = 26 kHz<br>(The value is lower at guaranteed synchronity.) |

If CAN messages are not completely used (e.g. only three 16 bit values instead of four per message), less data can be transferred although the sum sampling rate has not yet reached the maximum. This also applies if different sampling rates are defined in one system, because the data division to the CAN messages is not time-optimized (minimum time required).

## 2.3 Ampacity and voltage drop

Besides the fact that the max. bus line length is defined by the desired data transfer rate, the ampacity and the voltage drop in the system have primarily to be checked. This is especially important for systems with a high number of devices and/or long connections lines of the devices (e.g. distributed systems with connection lines of 3 m (9.84 ft) and more between the device groups). Additional actions should be taken accordingly to the situation.

### 2.3.1 Ampacity

The maximum current via the M-CAN system cables (e.g. 620-560.xxx) is 4 A (heat generation by transition resistances of the plug contacts).

The system capacity and therefore the power consumption can approximately be calculated by using the number of devices (including the sensor supply). A direct power acquisition in the real system provides exact values.

We recommend one or several of the following actions if the limit value is exceeded:

- Increasing the supply voltage of the devices (e.g. 24 V DC power supply or 42 V DC instead of 12 V)
- Centered voltage supply via T connection or as close as possible to the devices with high power requirements (rather than at the beginning or end of the system chain)
- Additional system supply via a T connection at a suitable position

### 2.3.2 Voltage drop

Even if the limit value for the ampacity is not reached, long lines in an extensive system can cause perturbations in the acquisition process. This mainly applies to devices at the end of the system chain, because the voltage of the last devices does not exceed the input threshold of 9 V (due to a high voltage drop in the system).

We recommend one or several of the actions mentioned above.

The voltage drop can be calculated by using the following formula:

$$U = R \times I$$

$$R = 2 \times R_{Cable\ [\Omega/m]} \times Length_{Cable\ [m]}$$

$$I = P_{Devices\ [W]} / U_{Devices\ [V]}$$

For estimating the voltage drop, a resistance of

> 50 mΩ/m for the M-CAN cables and
> 35 mΩ/m for the SIM-CAN cables

can be used including the transition resistances of the plug contacts. Systems, which are in the limit range of the voltage drop, should be controlled in individual cases. To do so, our support will be pleased to assist you.

As the power consumption of a device depends on the supply voltage, it is useful to calculate the voltage drop from the chain end to the feeding point. In this case, a minimum voltage of 9 V is set to the last device and the required excitation is calculated. The calculated value should be generously rounded upwards for guaranteeing a safe operation.

Another fact is the variable internal resistance of the input power supplies (low excitation = lower internal resistance).

In practice, this means: If the net excitation decreases (e.g. because of a weak power supply or a high resistivity with long cables), the devices have to readjust to cover the current power requirements. This causes a higher power consumption, which additionally increases the voltage drop.

# 3 Data logger M-LOG, S-LOG, FLEETlog/FLEETlog2, IPElog

## 3.1 Overview

| Device |  IPElog2 |  IPElog |  M-LOG V3 |  FLEETlog2 |  IPEhub2 |
|---|---|---|---|---|---|
| Operating system | RTOS 32 (32 bit) | RTOS 32 (32 bit) | RTOS 32 (32 bit) | RTOS 32 (32 bit) | Linux (32 bit) |
| Processor | Intel ATOM T3805 | Intel ATOM T3805 | Intel ATOM T3805 | LX800 | Arm |
| RAM memory | 2 GB | 2 GB | 2 GB | 256 MB | 256 MB |
| Data loggers software | TESTdrive | TESTdrive | TESTdrive | TESTdrive | IPEhub |
| Removable storage medium | cFast | SSD | cFast | CF | SD |
| Storage capacity | 8/ 16/ 32/ 64 | 8/ 16/ 32/ 64/ 128 | 8/ 16/ 32/ 64 | 8/ 16/ 32 | 1 / 2 / 4 / 8 |
| Software | | | | | |
| Configuration software | IPEmotion | IPEmotion | IPEmotion | IPEmotion | IPEmotion |
| No Message Lost (NML) | ■ | ■ | | | |
| Wake on CAN (WoC) | ■ | ■ | ■ | ■ | |
| Wake on LIN (WoL) | ■ | | | | |
| Wake on FlexRay (WoFR) | | | | | |
| Wake on SMS (WoSMS) | ■ | ■ | | | |
| Wake on Real Time (WoRTC) | ■ | ■ | | | |
| On board scripting | | | | | |
| On board math & logic operations | ■ | ■ | ■ | ■ | |
| File formats for data storage | DAT, BIN, AVI, WAV, MDF4.1 | DAT, BIN, AVI, WAV, MDF4.1 | DAT, BIN, AVI, WAV, MDF4.1 | DAT, BIN, AVI, WAV, MDF4.1 | HRD |
| Interfaces | | | | | |
| Ethernet interface to PC | 1 GigETH | 100 Mbit | 100 Mbit | 100 Mbit | 100 Mbit |
| USB 2.0 ports | 2 | 2 | 2 | 2 | |
| USB 3.0 ports | | | | | |
| Lemo, 9-pin for M-CAN modules | 1 | 1 | 2 | 1 | 1 |
| CAN HS (ISO 11898-2) | ■ | ■ | ■ | ■ | ■ |
| CAN LS (Low Speed) | ■ | ■ | ■ | | |
| CAN FD | | | | | |
| LIN (1.3 & 2.0) | 1 ... 6 | | 1 ... 4 | | |
| ETH (100 Mbit) | 1 | 2 | 2 | | |
| ETH (1 GigETH) | 1 | | | | |
| FlexRay | 1 (By extender) | 1 (By extender) | 1 (By extender) | | |
| Digital I/O | 4/4 | 4/4 | 4/4 | 2/2 | |
| COM (serial RS232) | 1 | | 2 | | |
| Audio input / output | 1 | | 1 | | |

| Device | IPElog2 | IPElog | M-LOG V3 | FLEETlog2 | IPEhub2 |
|---|---|---|---|---|---|
| Driver display system | IPEmotion app / openABK | IPEmotion app / openABK | IPEmotion app / openABK | IPEmotion app / openABK | IPEmotion app |
| Interface option 1 | 10 CAN + 6 LIN + 2 ETH | 12 CAN + 2 ETH | 12 CAN | 4 CAN | 2 CAN |
| Interface option 2 | 16 CAN + 2 ETH | | 8 CAN + 4 LIN | | |
| Interface option 3 | | | 8 CAN + 2 ETH | | |
| Interface option 4 | | | 6 CAN + 2 LIN + 2 ETH | | |
| Interface option 5 | | | 4 CAN + 2 ETH | | |
| Extender boxes | FlexRay-Extender | FlexRay-Extender | FlexRay-Extender , COMgate V3, IPEwiFi | | |
| **Protocols and traffic** | | | | | |
| CCP / XCP on CAN | ■ | ■ | ■ | ■ | |
| J1939 | ■ | ■ | ■ | ■ | |
| GM-LAN | ■ | ■ | ■ | ■ | |
| OBD | ■ | ■ | ■ | ■ | |
| WWH-OBD | ■ | ■ | ■ | ■ | |
| KWP on CAN | ■ | ■ | ■ | ■ | |
| UDS / ODX / PDX | ■ | ■ | ■ | ■ | |
| XCPonETH | ■ | ■ | ■ | ■ | |
| FlexRay / XCP on FlexRay | ■ | ■ | ■ | | |
| MOST (25 / 150) | | | | | |
| openABK | ■ | ■ | ■ | ■ | |
| SOME IP | | | | | |
| FlexRay traffic | | | | | |
| CAN traffic | ■ | ■ | ■ | ■ | ■ |
| LIN traffic | ■ | | | | |
| CAN traffic send | ■ | ■ | ■ | ■ | ■ |
| **Wireless communication** | | | | | |
| GNSS (Global Navigation Satellite System) | 20 Hz (GPS) | 1 Hz (GPS) | 1 Hz (GPS) | 1 Hz (GPS) | |
| Gyro sensor | 100 Hz | | | | |
| WiFi | 2.4 GHz (WiFi 802.11 b/g) | 2.4 GHz (WiFi 802.11 b/g) | via COMgate V3 | 2.4 GHz (WiFi 802.11 b/g) | 2.4 GHz (WiFi 802.11 (b/g) |
| Modem | 3G (UMTS/HSDPA) | 3G (UMTS) | via COMgate V3 | 3G (UMTS) | |
| VPN tunnel | | | via COMgate V3 | | |
| IPEcloud / FTP server | ■ | ■ | ■ | ■ | |
| **Video** | | | | | |
| IP-Camera (RTSP protocol) | | | | | |
| USB Video Class (UVC) | ■ | ■ | ■ | ■ | |

### Data transfer range of built-in modems (FLEETlog, IPElog)

| | |
|---|---|
| Frequency range | Quad-Band EGSM 850 / 900 / 1800 / 1900 |
| Download rate | HSDPA 7,2 Mbps |

UMTS/HSDPA (WCDMA/FDD) 2100 MHz

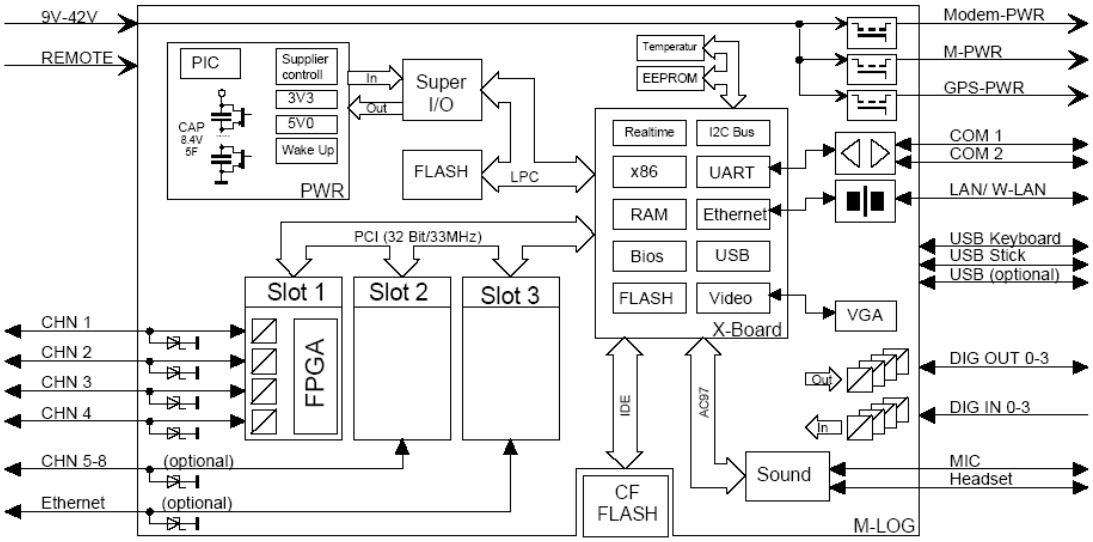
GPRS multi-slot class 12

Edge multi-slot class 12

## 3.2 Hardware

### 3.2.1 Block diagram

The block diagram shows the basic structure of the logger board.




### 3.2.2 Enclosure types


**M-LOG with port replicator PR05** **S-LOG front view** **FLEETlog front and rear view**

The M-LOG enclosure is compatible to all M devices. The devices can be easily connected with each other with a dovetail adapter. M-LOG is available with 6 different port replicators (cable connecting adapter). User specific port replicators are available on request.

The FLEETlog2 hardware concept was designed for using the data logger in vehicle fleets, whereas FLEETlog2 WAN is already equipped with modem, WiFi client, and GPS receiver.

### 3.2.3 Port replicators and cables

Various cables are available for every port replicator. The complete PIN configuration, as well as, the appropriate cables of the PR05 and PR08 port replicators are shown in the appendix.

### 3.2.4 Device panel for changing the internal memory card

S-LOG and M-LOG (M-LOG with Upgrade Kit 300) offer a screwed panel for exchanging the cF memory card if required.

The FLEETlog is equipped with a cF card slot behind the folding front panel.

IPElog has a screwless fixed front panel with open/close contact for exchanging the SSD memory card.

*Please note the advice to only change the memory card in exceptional cases because of a possible intrusion of foreign material, dirt or water into the device and the logger could be damaged at removing the cF card. The regular data transfer should be done with the USB stick or the wireless connection (modem or WiFi option).*

### 3.2.5 External fuse for logger protection (IPElog, M-/S-LOG, FLEETlog)

*Depending on the number of inputs, capacitor charge, and connected devices (M or SIM devices, Modem, GPS receiver), the total power consumption (especially at temperatures of –40 °C / -40 °F) can reach up to approx. 150 Watt. To protect the total system, we recommend an external overload protection with the following tasks:*

- *Protection of the supply line in error case*
- *Overload protection of the current source*
- *Logger protection.*

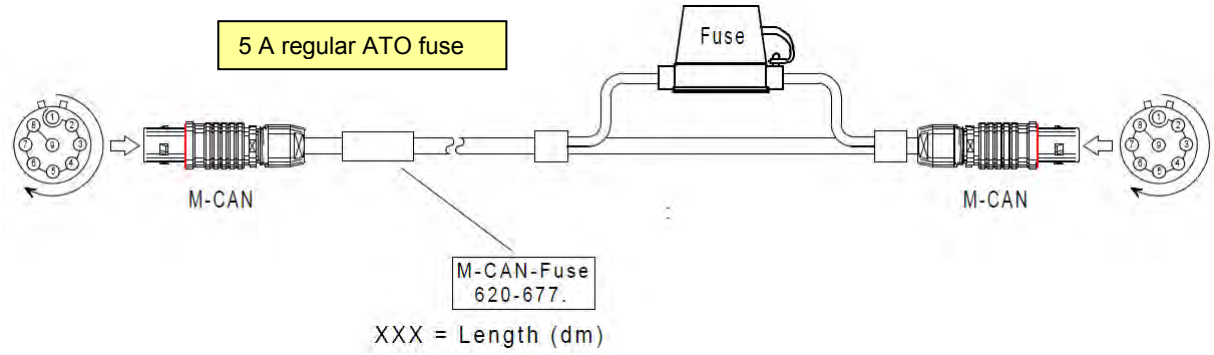
We recommend using a **LittleFuse ATO Fuse Fast Acting Type** with a nominal current of 10 A. At using the maximum charge in ambient temperatures under –40 °C / -40 °F, it can be required to increase the nominal current to 15 A.

*The individual electric circuits (also see M-LOG Port replicators, S-LOG = PR05) are internally protected. The internal resistance of multifuse types increases exponentially and limits the current to a minimum. After clearing the cause of error, the resistance decreases to the normal value (automatic resetting).*

*The motherboard fuse does not work reversibly and must be replaced in the case of a required repair.*

### 3.2.6 Additional short-circuit protection for IPElog and FLEETlog

As the M-CAN connection is not fused by the logger, we recommend to use the cable M-CAN Cable M-CAN/PWR-Fuse (No. 620-677.xxx) in order to prevent the logger from damage caused by a short circuit.




### 3.2.7 LED status display (flashing codes)

| LED display | Mode | Meaning |
|---|---|---|
| **GREEN** | Ready or operating | Device is ready<br>(Operation: see yellow LED) |
| | Warning I<br>Low voltage | The excitation is between 6 V and 9 V. M-LOG shuts down after 2 minutes at unchanged status. |
| | Warning II<br>Low voltage | The excitation is under 6 V. M-LOG is buffered by the internal supply and normally shuts down. |
| **YELLOW** | Measurement running | The logger writes the data to the internal memory. |
| | Data medium access<br>(post processing) | Prepare file transfer (zip, split) |
| | Data transfer | Transfer files via USB, modem, or WiFi |
| **RED** | Error,<br>Emergency operation | e.g. at invalid configuration, at less memory capacity, at emergency shut-down due to less excitation<br>A restart is required.. |

### 3.2.8 Power-down at excitation loss

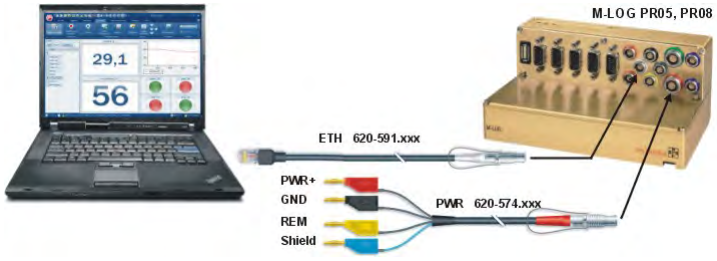
M-LOG, S-LOG, FLEETlog and IPElog are equipped with high-powered capacitors, which guarantee a short excitation in the case of an excitation loss for regularly shutting the system down without data loss. The storage capacity of the capacitors depends on several points (e.g. ambient temperature, aging, charge condition). A data loss can therefore not completely be excluded in the case of a total excitation loss.

M or SIM devices, which are supplied by M-/S-LOG, FLEETlog C2 or IPElog, are immediately switched-off at excitation loss.

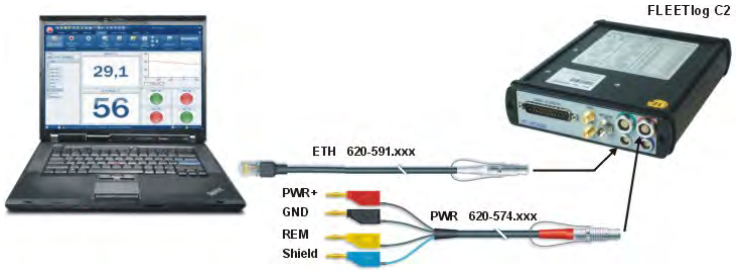
*We recommend to revise the switching and the buffer capacitors every 2 years for guaranteeing a clean functionality. The device calibration every 2 years includes this revision.*

## 3.3 Initial start-up

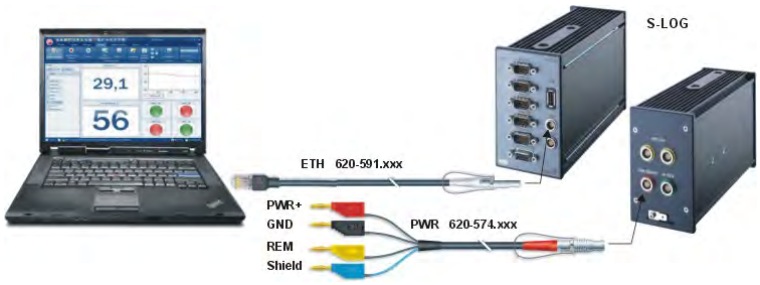
### 3.3.1 Connecting the logger

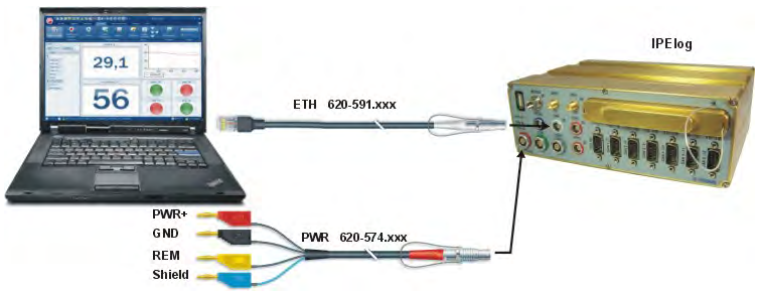



- Connect the red socket using the Power/Remote cable (e.g. 620-574.xxx) with an appropriate power supply (9 $V_{DC}$ ... 36 $V_{DC}$)
  PWR+ = Voltage supply Plus
  GND = Voltage supply Minus
  Shield = Shield or voltage supply Minus




- Connect the white socket using the LAN cable (e.g. 620-591.xxx) with the Ethernet interface of a PCs/Notebook, resp. using the LAN cable (e.g. 620-355.xx) for connection to a network.
- Switch on the logger
  REM = Ignition line 15 or power supply Plus via switch
  → green status LED lights continuously
- Logger starts up and will enter the measurement mode after a short time.
  → yellow status LED lights continuously
- Refer to the **Flow chart of the measuring process** in the appendix for more details.




**Useful notes**

! If the Ethernet connection to the logger will not work properly, at first transfer the correct IP settings to the logger using an USB stick. Refer to ***Add logger system, creating test configuration (USB stick)***.

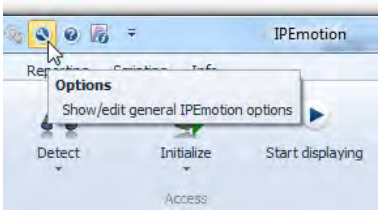
! Using the cable **620-591.xxx LOG Cable ETHERNET (crosslink)** is a point-to-point connection between logger and PC. A fixed IP address is required for both. Refer to ***Static and dynamic IP addresses***.

! When connecting the logger and the PC used for configuration and data acquisition to a network with DHCP server, both have to be set to dynamic IP address, in order to receive the individual IP address by the DHCP server automatically.
In this case use cable **620-355.xxx M-LOG PR05 ETH Cable RJ45**.

Please consider the risk of IP conflicts (same IP address may exist twice), if you use static IP address settings when connected to a network with DHCP support.

## 3.3.2 Detecting the logger, creating test configuration (Ethernet)

- Start IPEmotion at your PC/Notebook.

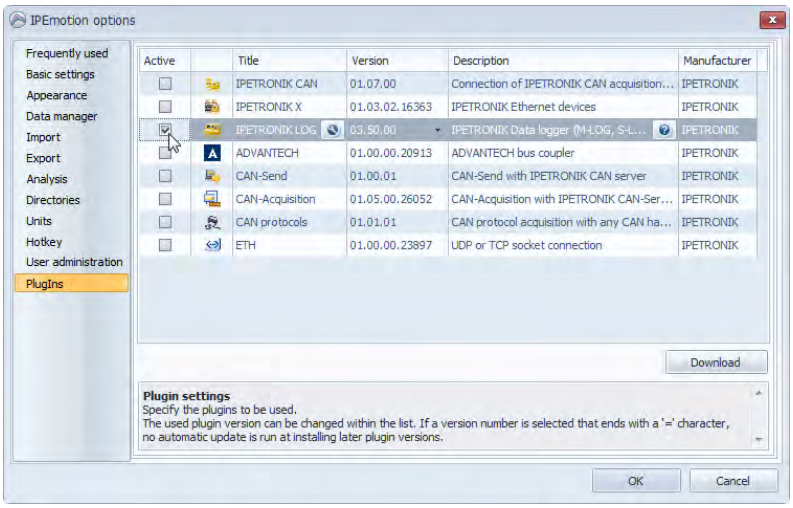



! Using Logger PlugIn version ≥ 3.50, IPEmotion ≥ 2.0 is required.

- Activate the PlugIn IPETRONIK-LOG.

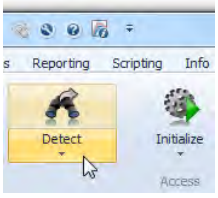
  ***Options*** > ***PlugIns*** > ***IPETRONIK LOG***

! The PlugIn IPETRONIK-LOG version 3.50 and higher requires TESTdrive ≥ 3.50 on the logger. Please update the logger application if necessary!



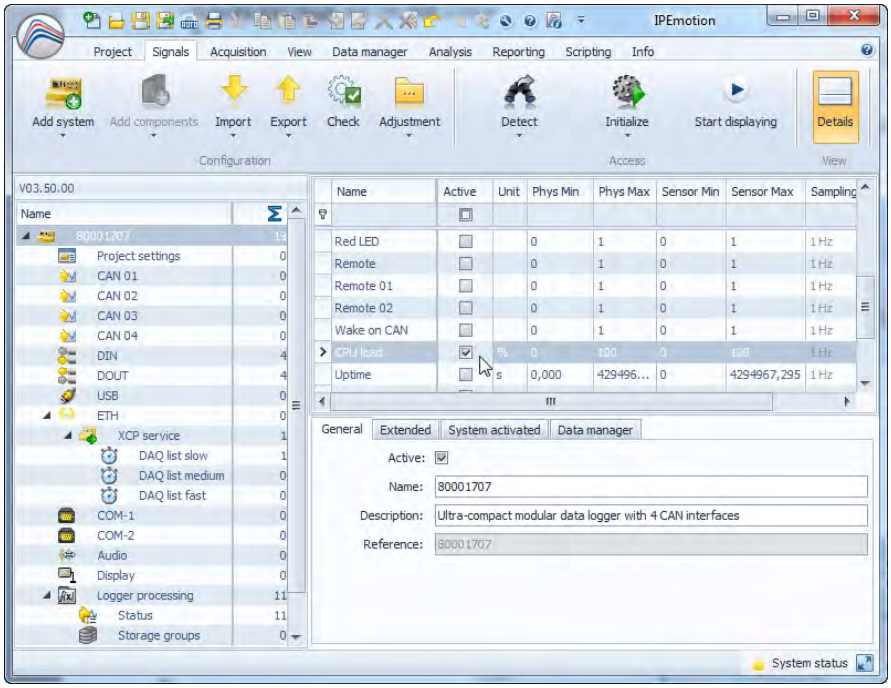

- Select the main tab ***Signals*** .

- Click ***Detect*** to identify the connected hardware components.
  The / all available logger(s) will be displayed in a select list.
  Confirm your selection by activating the corresponding checkbox and click OK. The configuration settings stored on the logger will be readout and displayed.




- Activate the status signal CPU load.

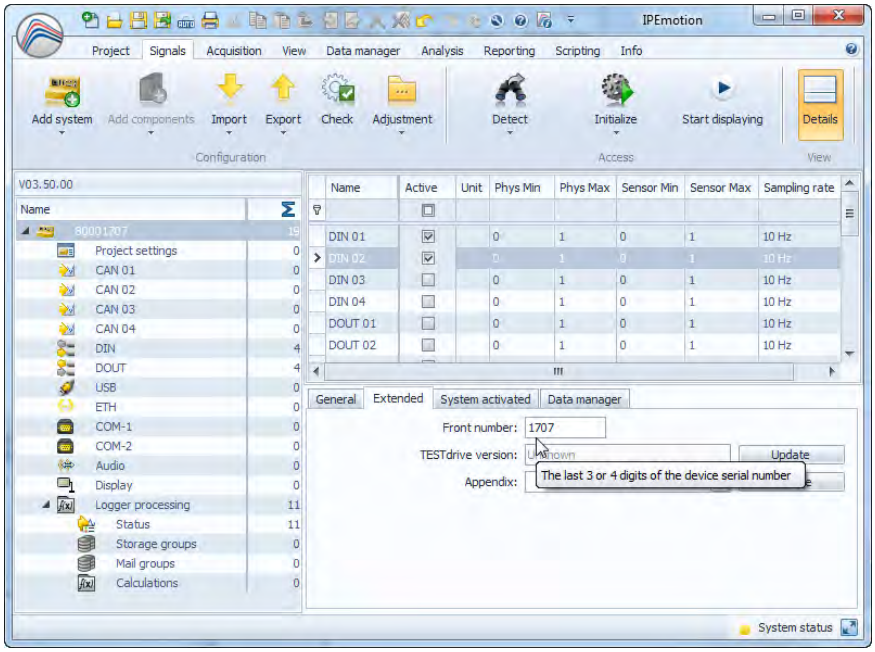
  > Mark the logger at the left hand structure top area and scroll the signal list downwards to CPU load, hook ***Active*** to enable the checkbox.




- Select ***ETH*** from the left hand tree structure, right click and choose > ***Add components*** > **XCP service** from the context menu
- Select ***DAQ list slow*** from the left hand tree structure, right click and choose > ***Add components*** > **Channels** from the context menu
- Mark ***CPU load*** and confirm by clicking ***OK***.
- Mark the logger from the left hand tree structure top area. Click ***Start displaying*** from the tool bar.
- Once the logger has been initialized, the current value of the CPU load is displayed continuously in the signal list. Now logger's ETH communication test and online data streaming via XCP service has been completed succesfully.

### 3.3.3 Add logger system, creating test configuration (USB stick)

In case the data acquisition or the Ethernet connection is not working properly (e.g. when using obsolete / former configurations, incompatible IP address settings, ...), we recommend to transfer a valid (already succefully tested configuration) per USB stick to the logger.




- Create a logger system:
  Select from the main tabs ***Signals*** > ***Add system*** > e.g. **M-LOG (4CAN)**
- Enter the front number of the logger you will use:
  > Mark the logger at the left hand structure top area and choose the ***Extended*** tab to set the last four digits of the logger's serial number using the field ***Front number***.
- Select the ***Data manager*** tab and activate the check box ***Update connection parameters*** and click to ***Configuration***
- Enter valid settings for the IP address used for the Ethernet connection (auto IP or fixed IP) as described at ***Static and dynamic IP addresses***.
- Create a test configuration.
- Save the current configuration settings in the project (*.icf).
- Export the project as measurement configuration (TSTdrive.mcf) to an individual subdirectory located on your Notebook/PC or direct to an USB stick.

- Unplug the USB stick from the configuration PC.
- Switch on the logger.
- Plug in the USB stick while the logger is running in measurement mode (yellow LED lights). The logger will stop measurement an start data postprocessing, existing data will be moved to the USB stick. This process lasts as long as the yellow LED is flashing.
- Then the logger will search for a new measurement configuration on the USB stick. If a new configuration is found, the application on the logger will be updated and the origin TSTdrive.mcf on the USB stick is deleted.
- Wait until the yellow LED lights nor flashes not any longer and unplug the stick. The logger will switch to measurement operation again.

After all necessary connection parameters have been updated with valid settings (logger, PC), a communication via the LAN cable is supported, including these actions:

- transfer a new configuration to the logger,
- detect the logger and read the configuration from the logger,
- transfer measurement data from the logger to the PC (import measurement files).

### 3.3.4 Static and dynamic IP addresses

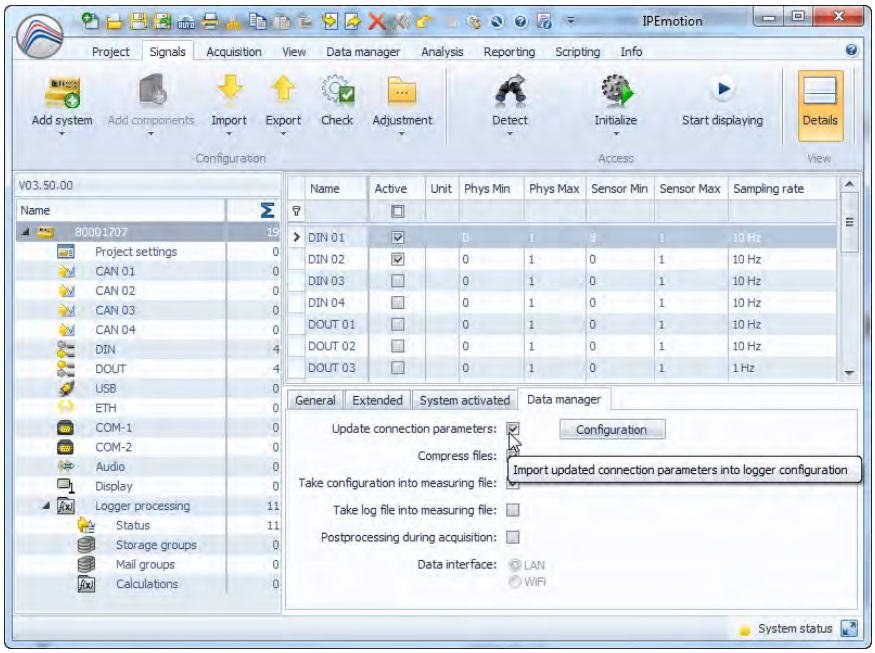
Basically IPETRONIK data loggers can be connected to other network clients via Ethernet in two different ways:

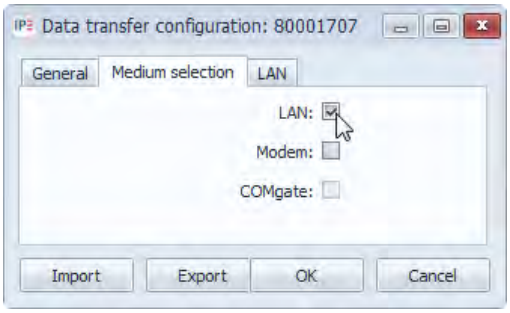
- Point to point connection between the logger and the PC/Notebook.
  Usually neither the logger nor the PC/Notebook supports DHCP (Dynamic Host Configuration Protocol), it is necessary to set a fixed IP address for both.
- Client to client connection between logger and PC/Notebook through a local network. In this case the network server provides DHCP and manages the automatic IP address assignment to all network clients. Therefore logger and PC have to use dynamic IP address settings.

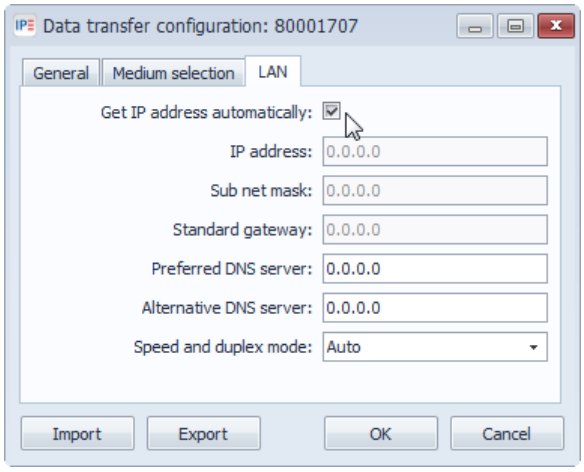
Use the ***Data manager*** tab to enter the settings for the logger:

- Mark the logger at the left hand tree structure top area.
- Select the ***Data manager*** tab from the right hand down area (configuration tabs).and activate the check box ***Update connection parameters***.
- Click ***Configuration***.
- Choose the tab ***Medium selection*** and activate ***LAN***.
- Refer to the dialogs below for dynamic (Get IP address automatically) and static IP address settings.
- When using the static IP address with the logger, it is necessary to adapt the system control settings of the PC/Notebook,

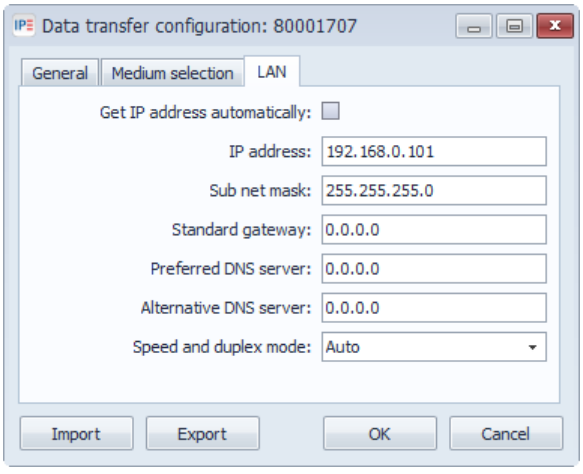
  For example:
  IP address: 192.168.0.100
  Subnet mask: 255.255.255.0



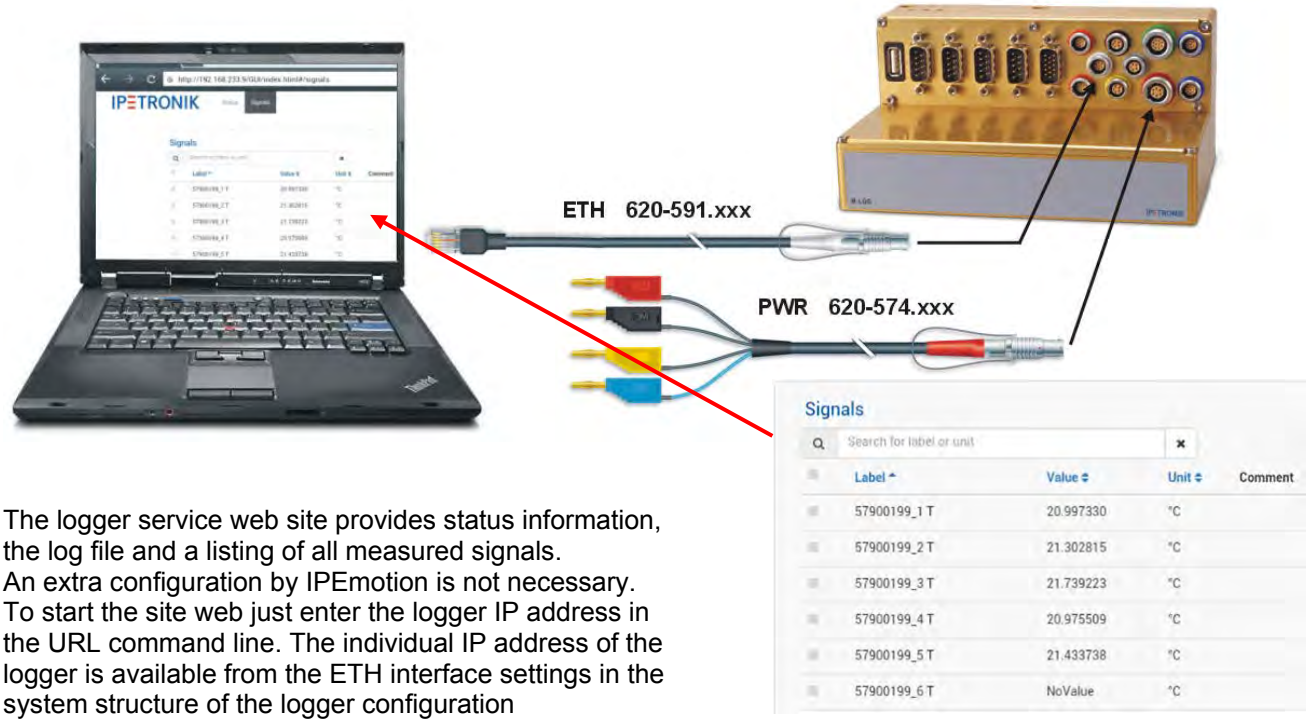




Settings for dynamic IP address




Settings for static IP address

### 3.3.5 Logger web interface for status and signal monitoring




The logger service web site provides status information, the log file and a listing of all measured signals.
An extra configuration by IPEmotion is not necessary.
To start the site web just enter the logger IP address in the URL command line. The individual IP address of the logger is available from the ETH interface settings in the system structure of the logger configuration

Please also refer to **11.1.2 Status information**

Using the logger system with IPEhub2 and a mobile device (smartphone, tablet) or PC with WiFi port, a wireless access to the status web site and a user specific measurement data visualization is supported.

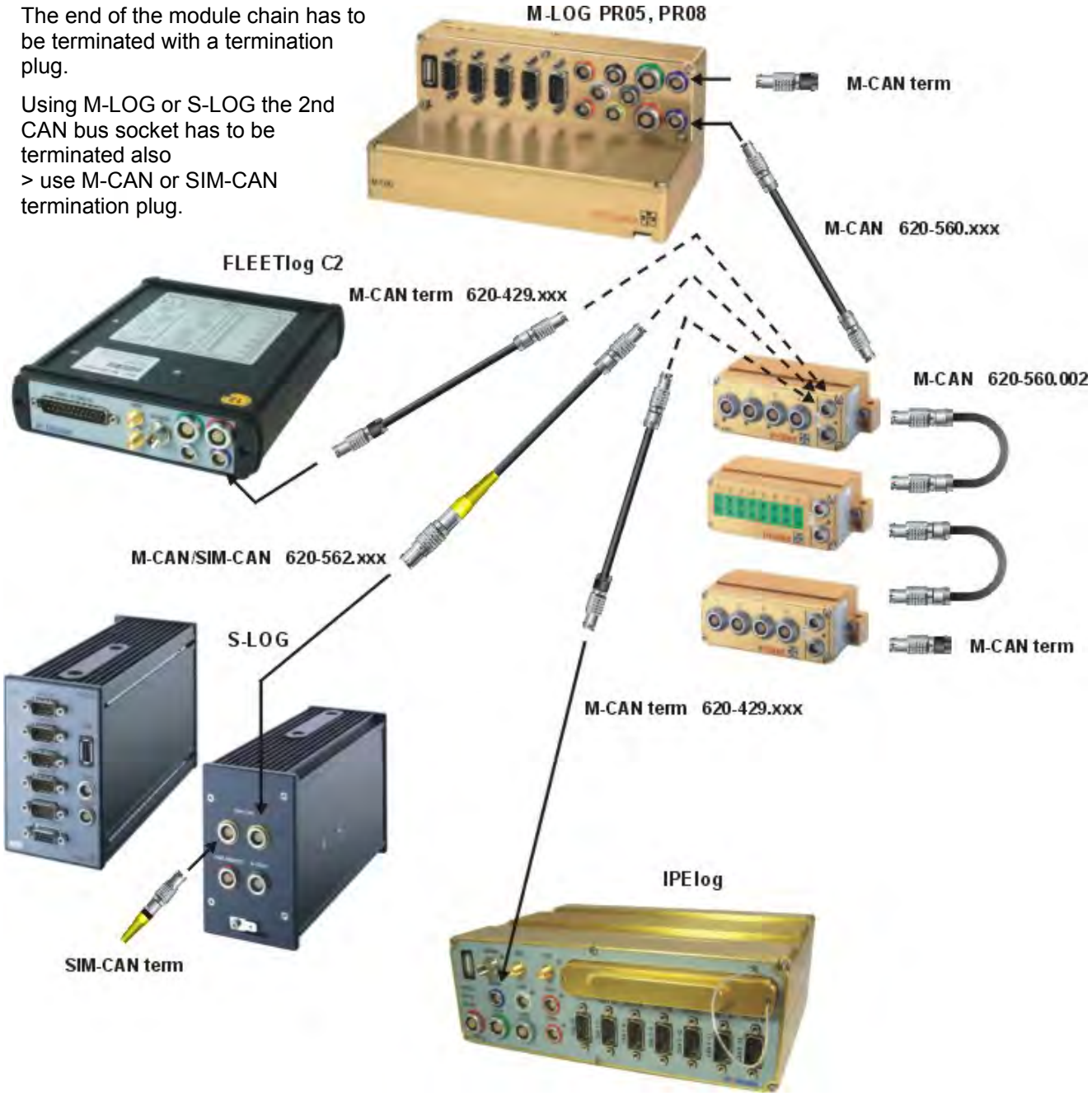
Refer to **9.2 IPEconnect (Online display with Smartphone or tablet)** for details

### 3.3.6 Logger system with CAN modules

All IPETRONIK CAN modules (M-Series, SIM-Series, CANpressure, MultiDAQ, High Voltage Iso DAQ ...) are connected to the data logger with corresponding system cables.

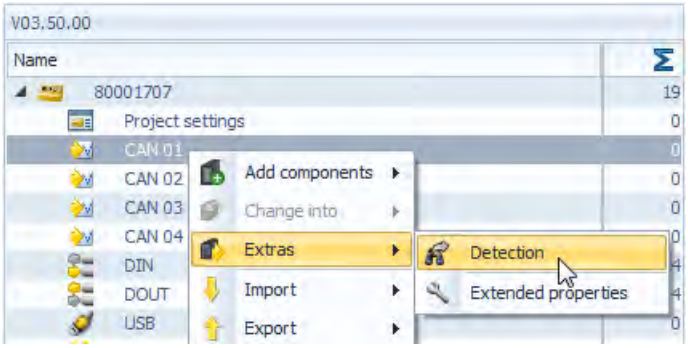
#### Connecting the modules

- Daisy-chain the modules among one another and connect the system with its respective cable to the logger.
- The end of the module chain has to be terminated with a termination plug.
- Using M-LOG or S-LOG the 2nd CAN bus socket has to be terminated also
  > use M-CAN or SIM-CAN termination plug.




*Usually all IPETRONIK CAN modules are connected to the logger through the system connectors M-CAN resp. SIM-CAN with S-LOG. Thus they can be detected and configured by IPEmotion in a user-friendly way. Besides this, it is also possible to connect IPETRONK modules, as well as, non-IPETRONIK CAN modules to a free CAN measurement input. Please note, that you need a separate cable to supply the modules with power for this. Configuration of the modules is done manually or by importing a CANdb file.*

## Detecting modules, creating test configuration




1. Select the CAN input to which the devices are connected to (CAN 01 is recommended because it is connected with the M-CAN or SIM-CAN socket at the logger).
2. Select **Extras** from the context menu (right mouse button).
3. Select **Detection** to detect the connected devices and to transfer the current settings to the configuration.

**MultiDAQ is detected as IPETRONIK CAN device from TESTdrive V03.22!**

- Execute ***Detection*** of connected modules as described.
- Configure the modules and the logger (activate channels, set signal scaling, select sample rates, create storage groups, ...)
- Save the current project.
- Initialize the Logger (working Ethernet connection required) or transfer the configuration per USB stick to the logger.

*To display measurement data acquired by the logger using the Ethernet connection, you have to add an **XCP service** at first. Assign all signals to the DAQ lists of the XCP service which you want to be displayed.*
*The **Options** setting **Automatic service administration** will automatically create an XCP service and assign all active signals to the DAQ lists.*

## 3.3.7 Access to data stored on the logger

The logger stores all data in a Zip archive on internal storage medium (cF card, resp. SD card with IPElog).

| | |
|---|---|
| MEA_xxxx.zip | measurement data + header file + current configuration |
| LOG_xxxx.zip | logfile with status reports of the data acquisition |

*A measurement file (archiv file) is empty, if the sum of the file contents calculated from the files listed below is 0. In this case, the respecitve zip archive will be deleted.*

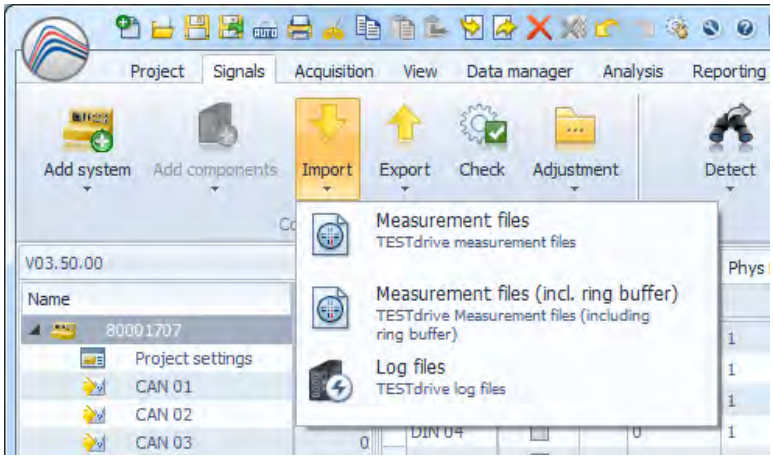
*- all files containing storage groups except the DAT files*
*- all trace files (BD...)*
*- all statistic files except the DAT files*
*- Min/Max statistic file (STG)*
*- Audio/Video files except the DAT files (AVI,WAV,IMG)*
*- Traffic files (BIN, Quickstart)*
*- LOG file*
*- Measurement status file*
*- PPP debug file (PPP...)*

Access (= data transfer to a PC or FTP server) to data stored on the logger is supported:

- per USB stick
- through an Ethernet cable connection
- using COMgate or an internal/external WiFi client
- using COMgate WAN or an internal/external modem

## Using a LAN cable connection

Import measurement files / log files



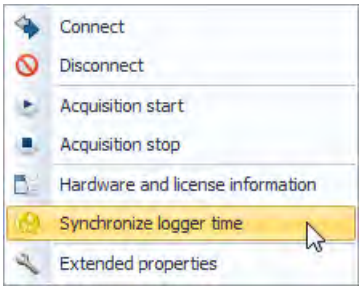

1. Select **Signals** navigation tab.
2. Select logger in the left **Systems** overview.
3. Transfer the current data (internal memory) in the **Signals** navigation tab to PC with **Import > Measurement files**.
4. Transfer the current data (internal memory + pre-/post-trigger data)) in the **Signals** navigation tab to PC with **Import > Measurement files (incl. ring buffer)**.
5. Transfer the current TESTdrive log files in the **Signals** navigation tab to PC with **Import > Log files**.
6. Select the target directory or crate a new folder to save the respective data.

## With USB stick

Plug in the USB stick to the running logger. The logger stops data acquisition, saves data, log files and the related configuration to the stick. If the yellow LED is not permanently on, disconnect the USB stick. The logger runs now in acquisition mode..

*Please note that data can only be imported via USB if no LAN connection to the logger is activated. Disconnect the logger with the logger context menu.*

## 3.3.8 Synchronizing the logger time




- Mark the logger at the left hand structure top area.
- Right click to the logger entry and select **Extras** > **Synchronize logger time** from the context menu.

*Time data of time stamp and log messages have been normalized. Now time stamp output is local time (former UTC). Time stamps reported in XML files always indicate the offset (time lag) to UTC time, e.g. <startdate>2012-04-25T09:30:10+01:00</startdate>*

*In order to ensure a correct time interpretation of recorded data, the IPETRONIK data converter IPEconverter V02.13 is required.*

## 3.3.9 Readout Hardware and license information

### Using the Ethernet connection (LOG2PC)

- Select the data logger on the top left area.
- Select **Extras** > **Hardware and license information** from the context menu by right-clicking.

### Hardware and license information file (hw_descr.xml)

- Select the data logger on the top left area.

- Select **Extras** > **Hardware and license information from file** from the context menu by right-clicking.
- Look for the file **hw_descr.xml** locally or on a USB stick.

*Plug a USB stick into the running data logger to retrieve the license file (together with configuration and acquisition data). Unplug the USB stick from the logger and plug it into a PC. Go to the directory named with the logger's serial number, then go to the **DEVICE** subdirectory. The zip file TSTdrive contains the hardware/license file **hw_descr.xml**.*

## 3.3.10 Updating the measurement application (TESTdrive) per USB stick

### Measurement application TESTdrive

- Copy the file **MLogger.rtb** (standard, inclosed in the TESTdrive_xxx.zip) or **Logger.prg** (for M-VIEWgraph, inclosed in the Setup_xxx.exe of the respective PlugIn) to the subdirectory named with the logger's serial number on the USB stick.
- Power up the logger and wait for the yellow LED which indicates data acqusition.
- Plug in the USB stick to the logger, program update will start automatically.
- Wait until the yellow LED lights no longer and unplug the USB.
  The logger executes a reboot and starts measuring operation with the new application.

### Measurement configuration (logger specific)

- Copy the file **TSTdrive.mcf** to the subdirectory named with the logger's serial number on the USB stick.
- Power up the logger and wait for the yellow LED which indicates data acqusition.
- Plug in the USB stick to the logger, configuration update will start automatically. As soon as the update has been completed successfully, the TSTdrive.mcf will be deleted from the USB stick.
- Wait until the yellow LED lights no longer and unplug the USB.

### Measurement configuration (single USB stick used for different loggers)

- Copy the file **TSTdrive.mcf** to the root directory on the USB stick.
- Power up the logger and wait for the yellow LED which indicates data acqusition.
- Plug in the USB stick to the logger, configuration update will start automatically. The TSTdrive.mcf will remain on the USB stick, whether the update has been completed successfully or not.
- Wait until the yellow LED lights no longer and unplug the USB.
- Repeat this procedure for any other logger update relating to this configuration

### Serial number specific update

When updating via USB/FTP, the system checks, if the file name consists of a 8 digit serial number. Update is executed when the serial number matches with the logger, the file is deleted from the source directory afterwards. In case of missmatch, the latest valid file is used. Files with invalid serial numbers will remain in the source directory.

When the file contains no serial number information, the standard update process is executed (equal to previous TESTdrive versions).

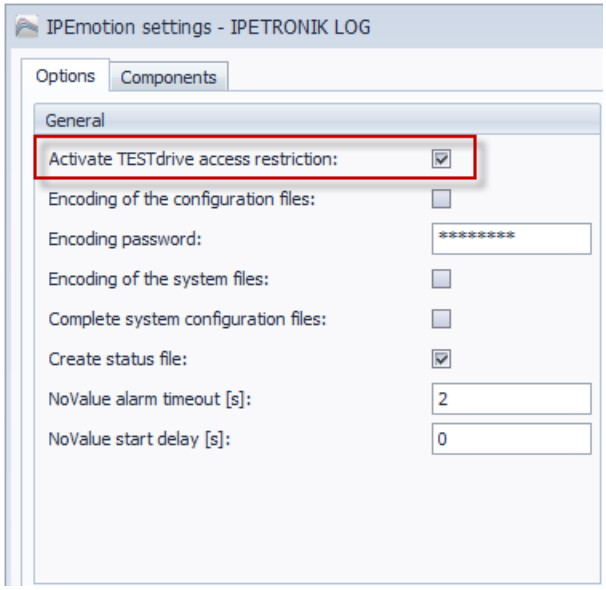
### 3.3.11 TESTdriveCMD.xml

The file TESTdriveCMD.xml controls user access of the logger. If the restriction has been activated, the logger searches for the TESTdrivecmd.xml at the root directory of the USB stick. Measurement will not be started when the file is missing.

Activate the use of the TESTdrive commands at IPEmotion **Options** > **PlugIns** > **IPETRONIK LOG** > **Settings** > **Options** > **Activate TESTdrive access restriction**.

The following templates are located at:

...\IPETRONIK\IPEmotion PlugIn IPETRONIK LOGV03.xx.00\Data\TESTdriveCMD\[language]




**Guest (directory „guest“)**

- Copy data
- External storage

**User (directory „user“)**

- Update measurement configuration
- Update measurement configuration, Copy data
- Update measurement configuration, Copy data incl. PostMortem
- Update measurement configuration, Move data
- Update measurement configuration, Move data, incl. PostMortem
- Start logger in FTP mode
- Set the logger‘s system time

**Administrator (directory „admin“)**

- Update measurement application
- Update measurement application, Delete data
- Update measurement application, Update measurement configuration, move data
- FPGA update

#### Parameter of the “dataTransfer” service

**<meaNumber>**

If this parameter has not been defined, all data will be transferred. Valid values of this parameter are "all" (not case sensitive) and a number between 0 and 9999. This parameter can be used multiple times. The applied sequence has no effect on data transmission. The data transfer starts with the measurement data file named with the lowest number in ascending order.

The file TESTdriveCmd.xml will be deleted from the server as soon as it has been transferred successfully. It will be deleted from the local source if all <meaNumber> entries have been processed. The file name prefix “TESTdriveCmd" is required, the file extension has to be“xml”. If the file name consists an 8 digit number (e.g. TESTdriveCmd_12345678.xml) the logger will take this as serial number. The file will accepted only if the serial number from the file name fits to the loggers serial number. If not, the file will be ignored and remains on the FTP server.

The existing parameters <zip>, <crypt>, <split>, <move> and <useTimeDir> will be ignored. They will be considered only if this service is applied by an USB stick. In this case, the TESTdriveCmd.xml remains unchanged, as usual before and it will not be deleted.

**<transfer>**

This parameter controlls the data transfer following to the data post processing (zipping, splitting, copying). Valid values are "true" and "false". If the parameter is not configured, its value is set to "true" per default.

"true" Zipped and splitted (if required) files are copied to the USB stick.

"false" No measurement data is copied to the USB stick.

# 4 External connection

## 4.1 Grounding

*It is absolutely necessary to connect the logger case with a suitable ground pin within the overall system. Without this grounding, EMV disturbances and uncontrolled voltage interferences can restrict the measuring process.*

### 4.1.1 IPElog, M-LOG grounding connection

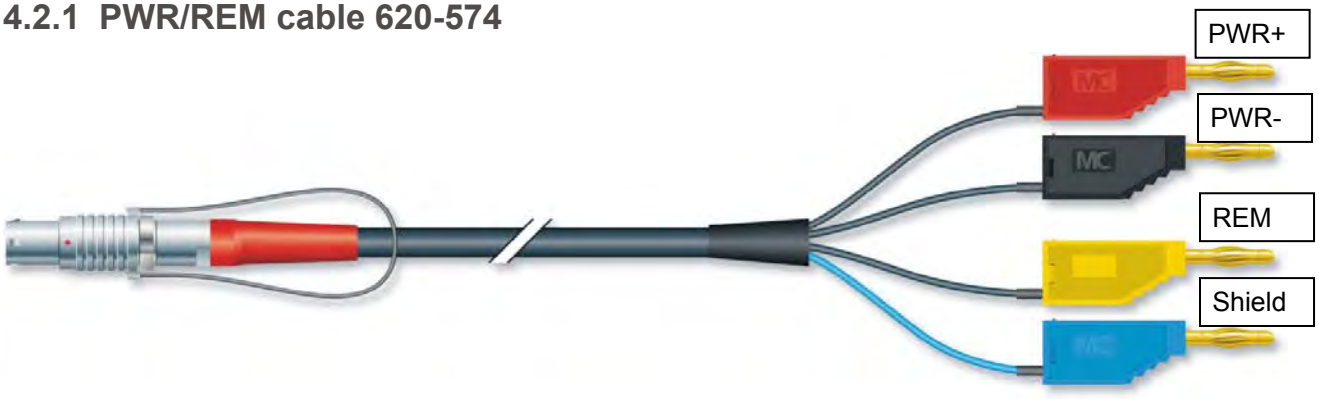
IPElog and M-LOG provide a grounding tap by a blade terminal. Use a 6.3 mm blade receptacle to connect the grounding tap.


### 4.1.2 FLEETlog grounding connection

FLEETlog2 has four M4 threaded holes at each side which are used to tighten the mounting brackets. We recommend to use an appropriate ring tongue terminal with one of the side screws for grounding the logger case. A fork tongue terminal can also be used.

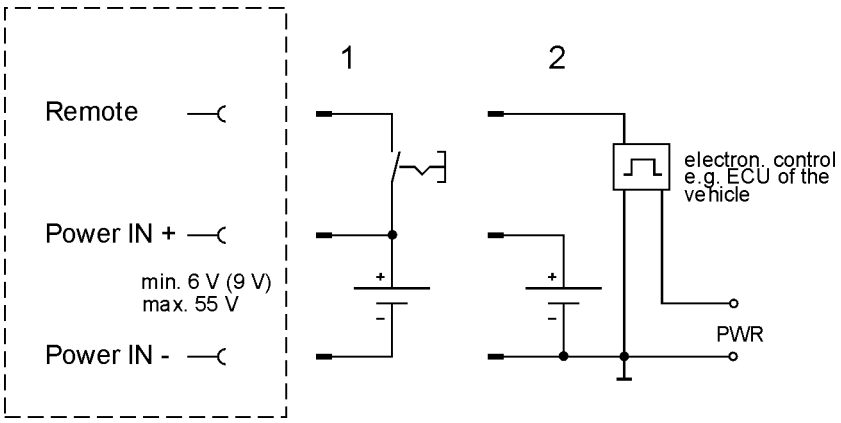

## 4.2 PWR / Remote connection

### 4.2.1 PWR/REM cable 620-574




***Always switch on and off*** *the logger* ***using the REM*** *(ignition line 15)* ***connector****, never by switching the PWR+ line!*
***Never disconnect the PWR- (GND) line*** *while the logger is running, because this will damage electronic circuits.* ***Switiching-on/off the logger via PWR- (GND) is absolutely not allowed!***

### 4.2.2 Remote connection




*The logger has internal buffer capacitors to avoid a data loss due to sudden voltage losses. The regular shut-down is executed via remote and not by switching-off the excitation!*

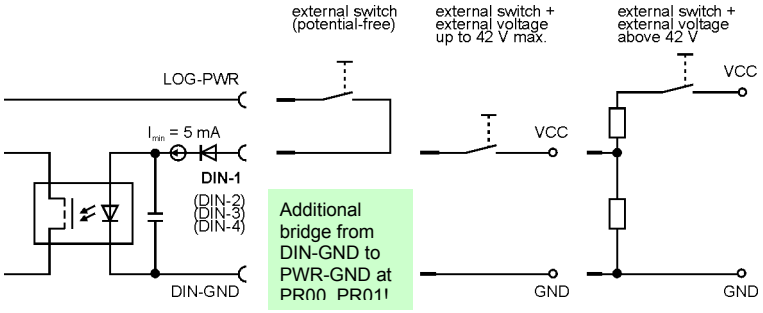
*The activating via a remote impulse (impulse length > 20 ms) is also supported. But the continuous remote signal must be received within 60 s, otherwise the logger is shut down by the watchdog.*

***Example:***
*Start of measuring system with opening the vehicle door. The continuous remote signal is received at motor start and the system continues measuring. If the continuous signal is not received (no motor start), the logger shuts down after 60 s.*

## 4.3 Digital input/output connection

**LOG-PWR** switched logger excitation (not stabilized)
The excitation is supplied, as soon as, the logger is activated.

### 4.3.1 Connecting example digital input




potential free contact with external excitation up to 42 V without any resistor or voltage divider

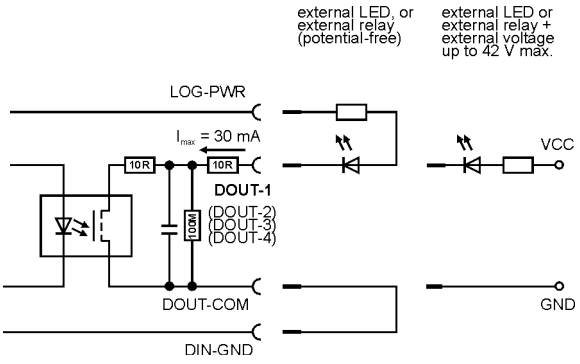
Terminal 15 of the board electrics (all vehicle power supplies up to 42 V without any resistor or voltage divider)

*The digital inputs are electrically isolated but the respective DIN-GNDs are all connected to one PIN of the Sub D socket! This PIN has a direct connection to PWR-IN GND with the PR03, PR04, PR05, PR06 port replicators!*

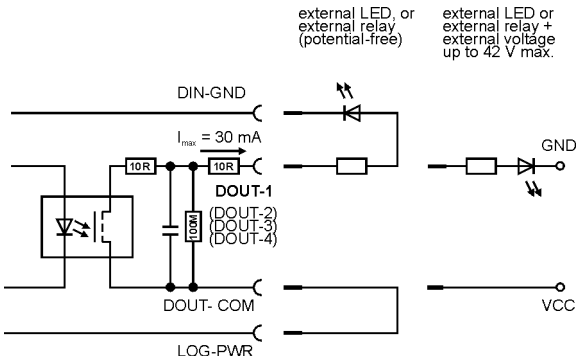
*The inputs are protected by an internal 5 mA current limitation.*

*Required minimum values at the input for a clean functionality:*
*$U_{min}$ = 3 V bzw. $I_{min}$ = 2 mA*

### 4.3.2 Connecting example digital output



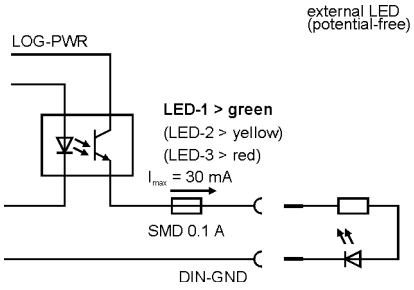

GND (ground) switched
Ri = 20 Ω




V+ (excitation) switched
Ri = 20 Ω

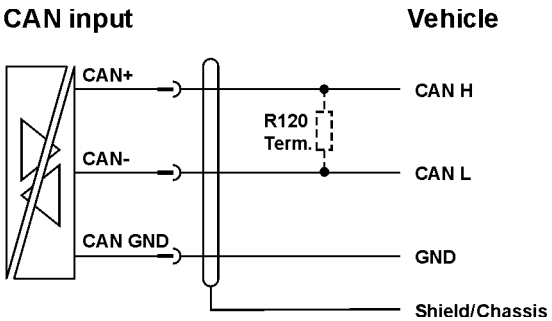
*The digital outputs are electrically isolated and can operate bipolarly but the respective COM ports are all connected to one PIN of the Sub D socket!*

### 4.3.3 Connecting example external status LED




## 4.4 Bus inputs

### 4.4.1 Connecting example CAN bus

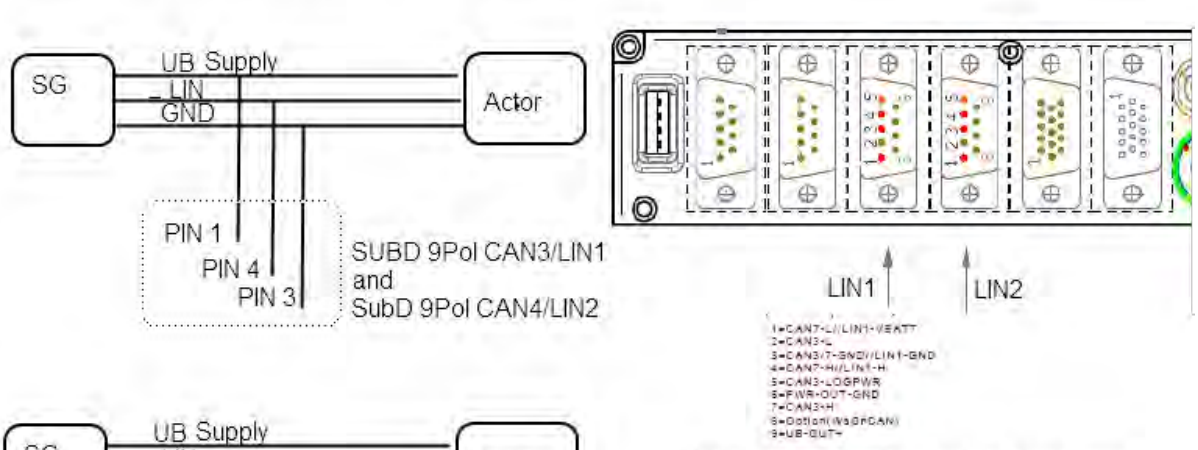


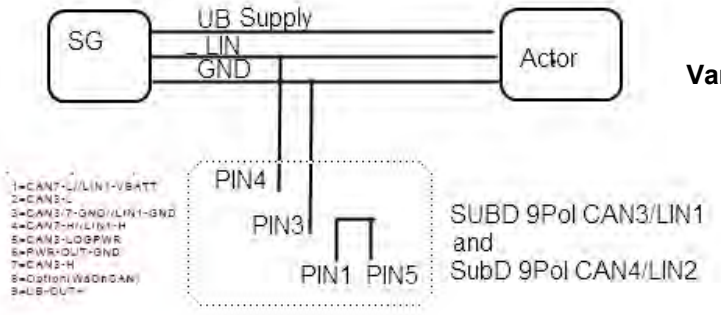

### 4.4.2 Connecting example LIN bus

Due to the galvanic isolation, the transceiver of the LIN measurement input must be power supplied by an external source. This can be:

1. Connection of the vehicle's bus power supply ($V_{Bsupply}$) or
2. Connection of the logger's power supply (bridge from PIN 5 to PIN 1)

**Variant 1:** Connect $V_B$ of the external LIN bus




**Variant 2:** Supply via logger
(bridge from PIN 5 to PIN 1)

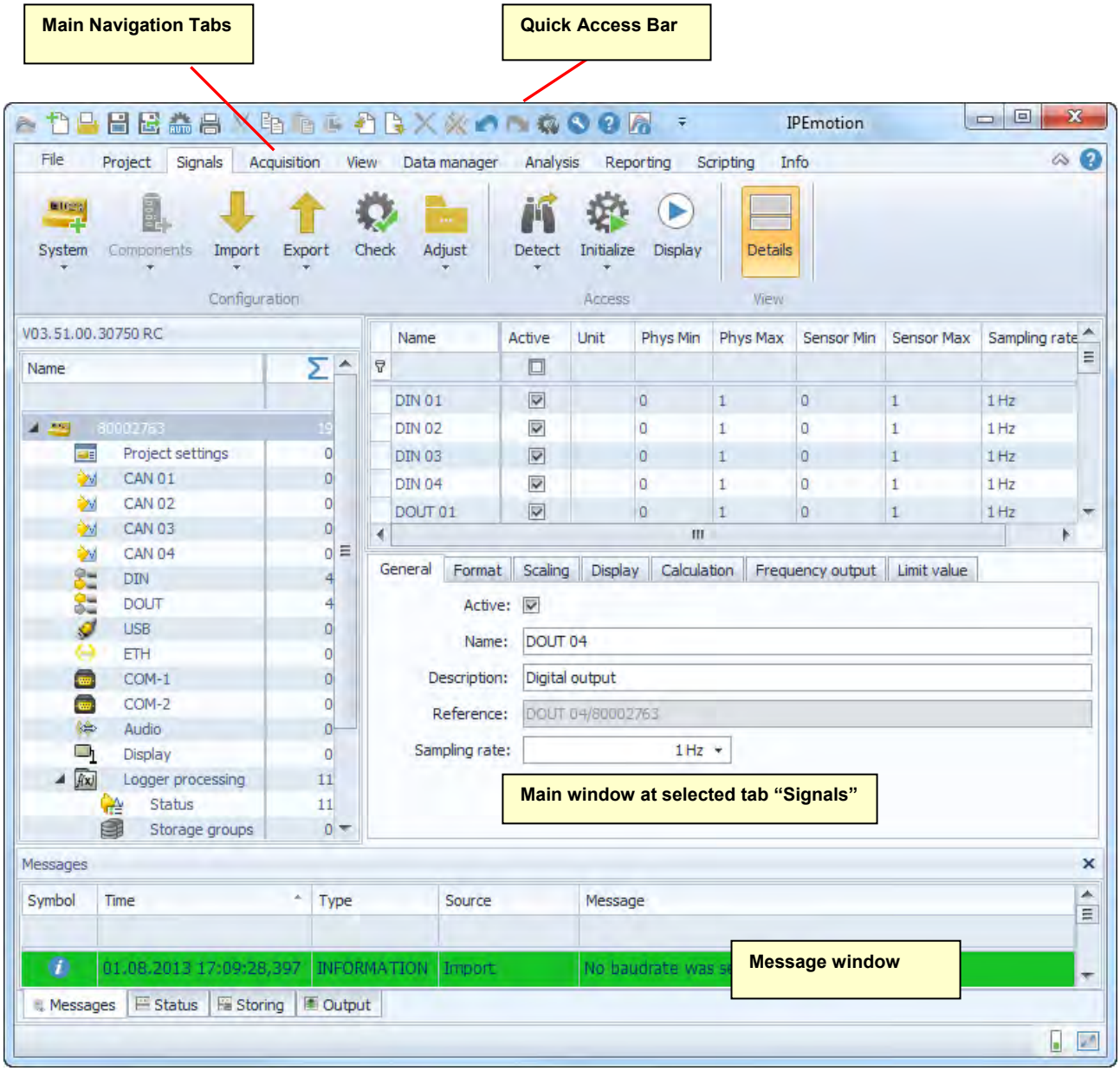
# 5 Configuration with IPEmotion (extract)

The IPEmotion software is required for configuring a logger measuring system and the corresponding IPETRONIK devices. The software must be installed on the notebook/PC, which is used to configure. (See the instructions in Commissioning). IPEmotion also supports the online data measurement with using the XCP service. You can find a detailed description of IPEmotion in the manual, which can be opened in the software as a PDF.

## 5.1 First steps

### 5.1.1 Main dialog

After the start of IPEmotion, the following screen appears.




*IPEmotion automatically detects all available hardware connections at starting.*

*If you want to reduce the required time, select **Options > PlugIns** for deactivating those interfaces, which are not used.*

### 5.1.2 The Title Bar

The title bar contains the quick access bar, the software name, as well as, a tool bar with the following functions:

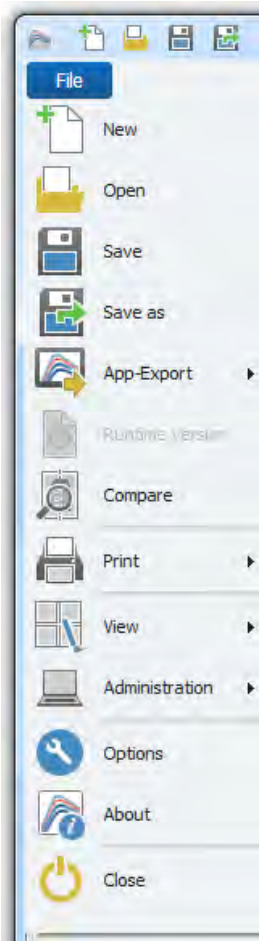

Help – Open the documentation IPEmotion.pdf where you can find useful information for a safe and clean application of the software.

Minimize – Minimize the application window of IPEmotion and place it in the task bar of your desktop.

Maximize – Make the application window visible on your desktop and refit the prior size.

Close – Close the application of IPEmotion.

### 5.1.3 The file menu




The file or application menu contains basic functions as: **New**, **Open**, **Save**, **Save As**, **Runtime version, Print** and **Close**, as well as, further properties such as **View**, **Options**, **Support file** and **About**.

The right partial view of the application menu contains a list of the recently used projects.

The **View** function contains the menu points **Message window** and the **Reset** command. Show or hide the message window and reset the displaying configuration to the default parameters.

### 5.1.4 Using the options

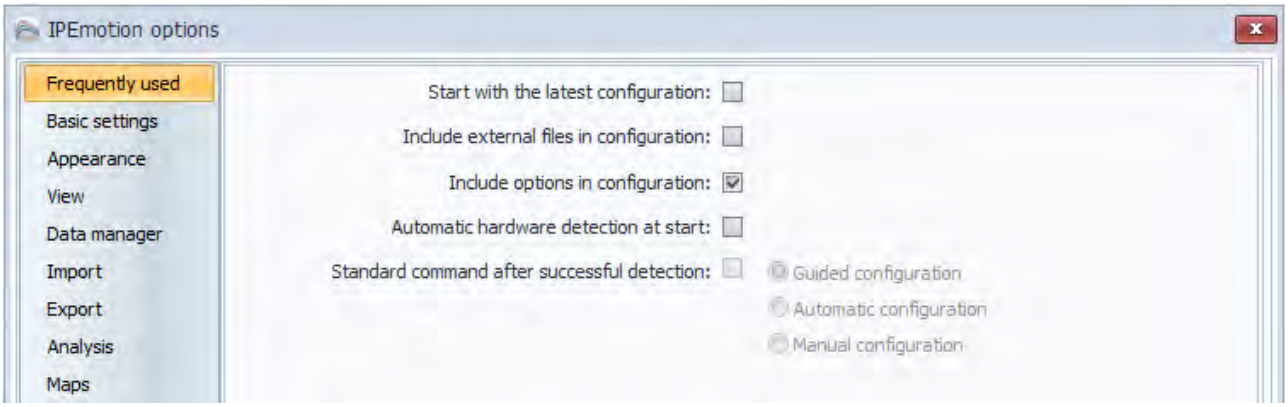
With the Options entry, you have the ability to edit user defined settings. You can define the following options:

- Frequently used
- Basic settings
- View
- Data manager
- Analysis
- Units
- PlugIns

The following section offers you a detailed overview over the available setting options.

## Frequently used

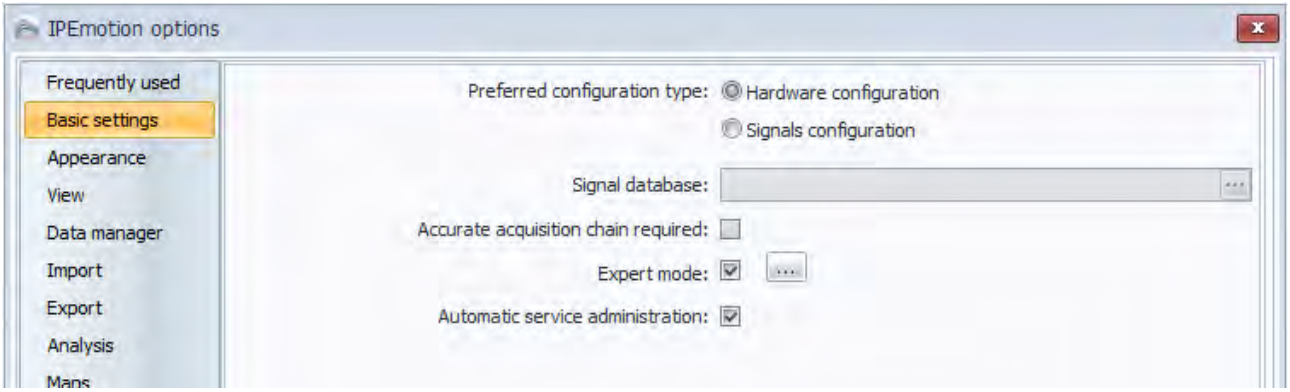
Activate or deactivate **Start with the latest configuration** and define the settings for the **automatic hardware detection**. Activate or deactivate the **automatic hardware detection at start** of IPEmotion and select a possible **standard command after successful detection**:




## Basic settings

Select a **preferred configuration type**: Hardware configuration or signals configuration

Activate or deactivate the options: **Accurate acquisition chain required** and **Expert mode**.




***Expert mode***

*The protocol nodes of imported description files (CANdb, A2L, FIBEX,...) are visible, this means, the user has access to the signal properties.*
*With the protocols CCP and XCP the DAQ lists are displayed and can be processed (delete, move signals).*

*With the release of IPEmotion V01.07.00 internal channels (-> Variables) for temporary storage tasks have been implemented. Managing this variables is only supported with an enabled Expert mode.*

Activate the **Automatic service administration** option to automatically import the active channels into the DAQ lists of the XCP service. These channels are then available in online view.

*Please note the restrictions depending on the Ethernet interface of the PC and the number of active channels. In this case, it is recommended to manually import the required channels into the DAQ lists of the XCP service (deactivated Automatic service administration)*

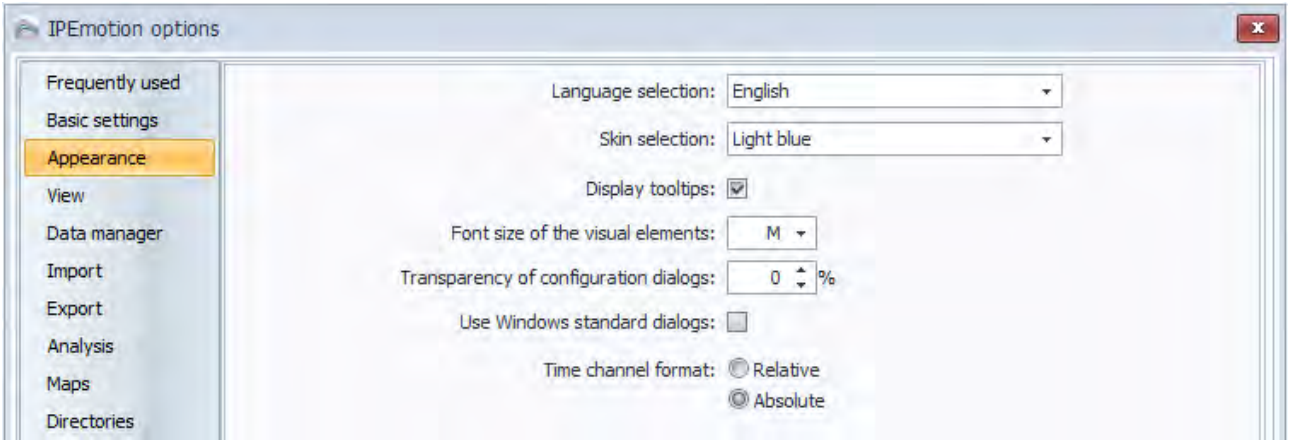
## View

Define your **view settings** according the following listed points:

Language selection Skin selection Displaying tooltips

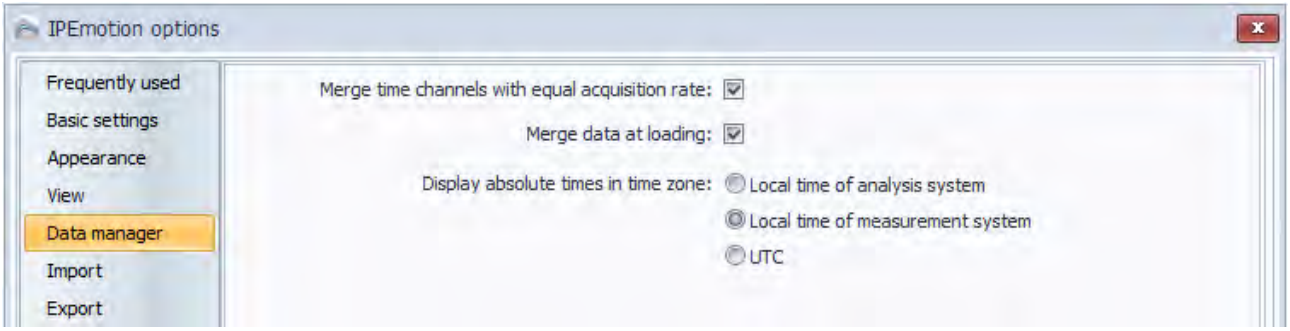
Font size of the visual elements Transparency of configuration dialogs (0 – 30 percent)

Activate or deactivate the use of the **Windows standard dialogs** for the file and directory selection.



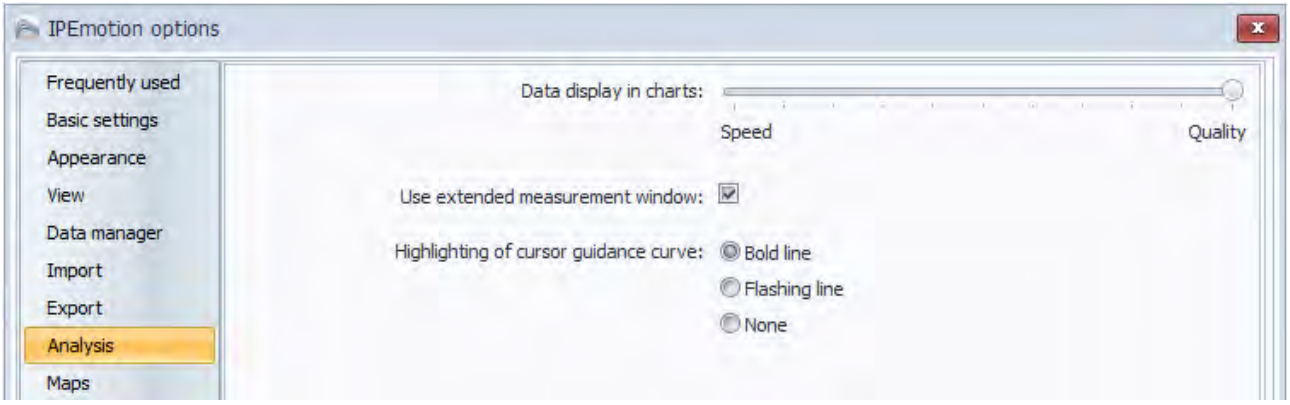

## Data manager

Define the **Time channel format** as *Relative* or *Absolute* (This setting is currently not supported for the export into external formats!) and activate or deactivate the option: **Merge time channels with equal acquisition rate**.



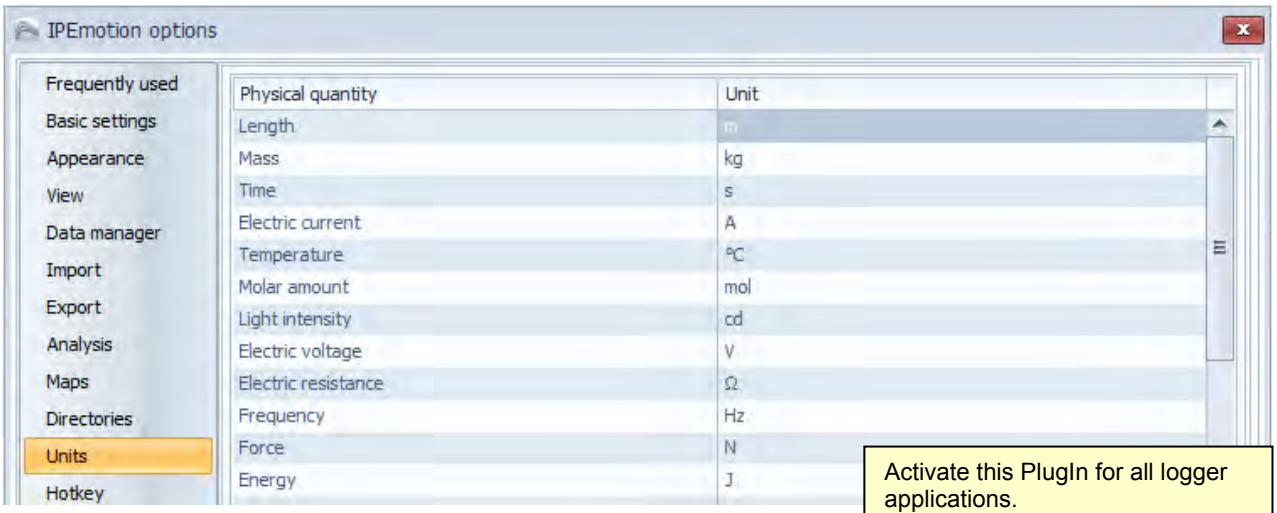

## Analysis

Select the **points per diagram graph**. Define if all signals are considered in the analysis diagrams at drawing the graph or only the samples. Move the bar accordingly to the preferred speed or quality.



## Units

Get an overview according the common physical values and their respective unit and edit them.

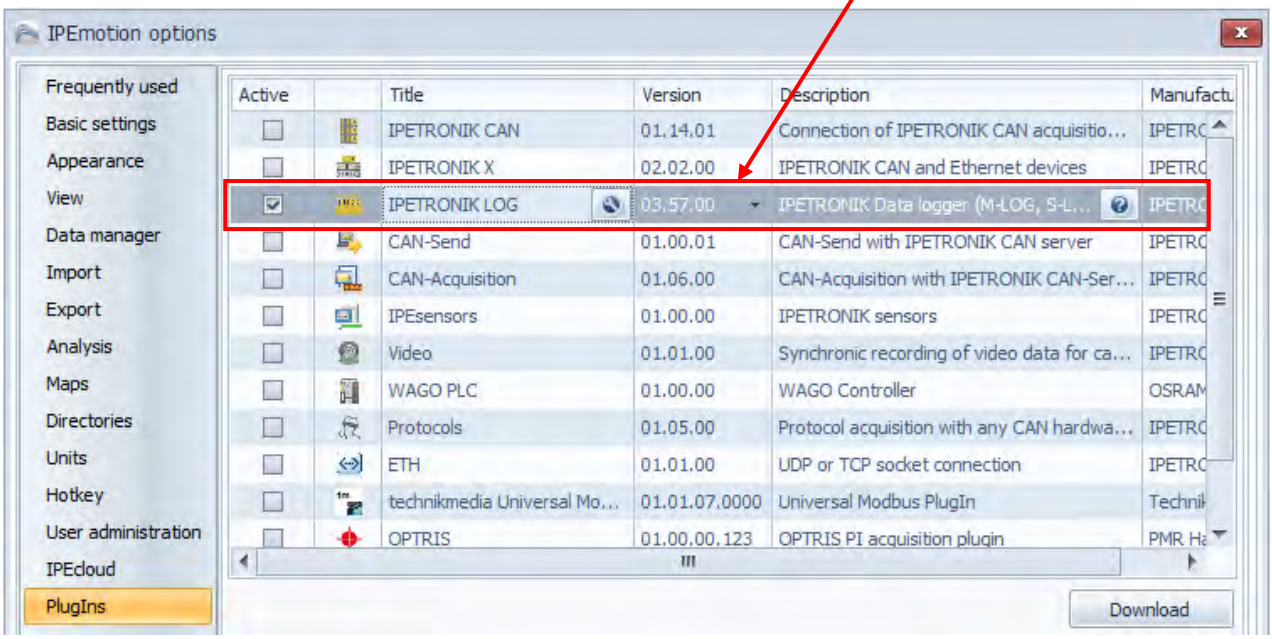



## PlugIns

Activate or deactivate the **hardware systems** to be used.

Activate this PlugIn for all logger applications.

The corresponding TESTdrive version must be available at the logger!




Clicking the wrench symbol  enters the dialog for PlugIn specific settings provided by the tab **Options** and **Components**.

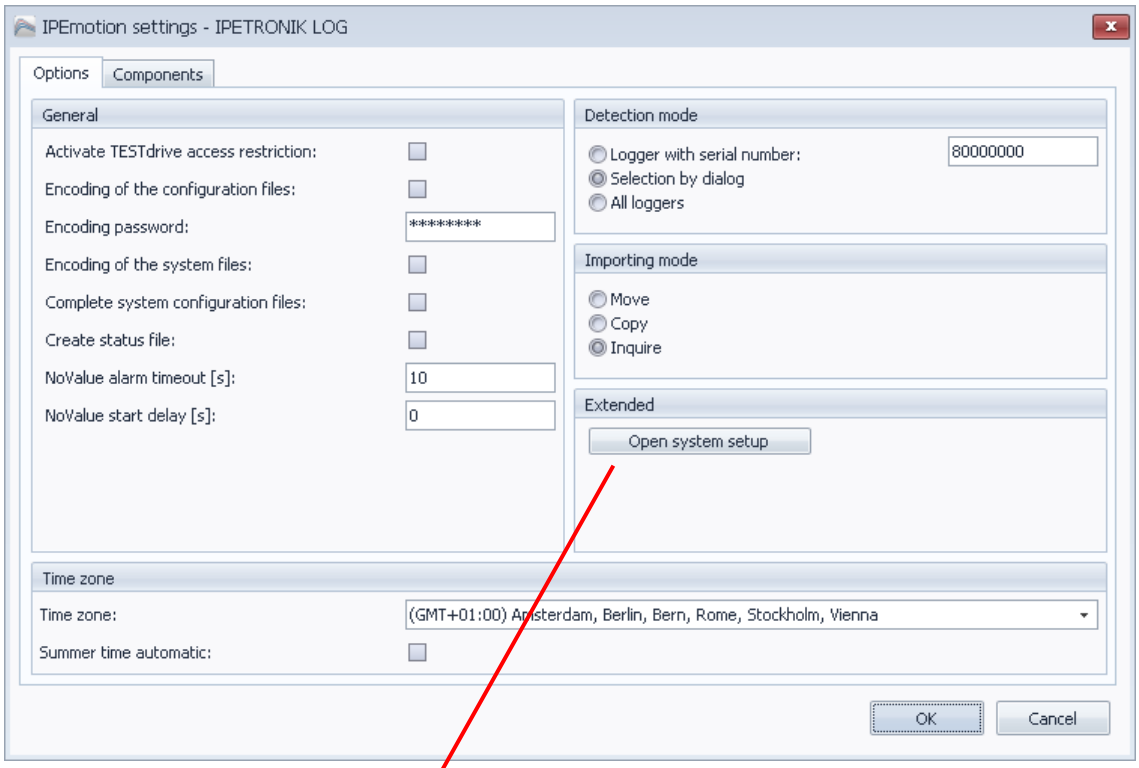
### Components

With the **Settings** button, you have the ability to define the **components** (module type and priority, e.g. for the type selection of the Dry configuration) of the respective hardware system and to edit additional **options** settings.

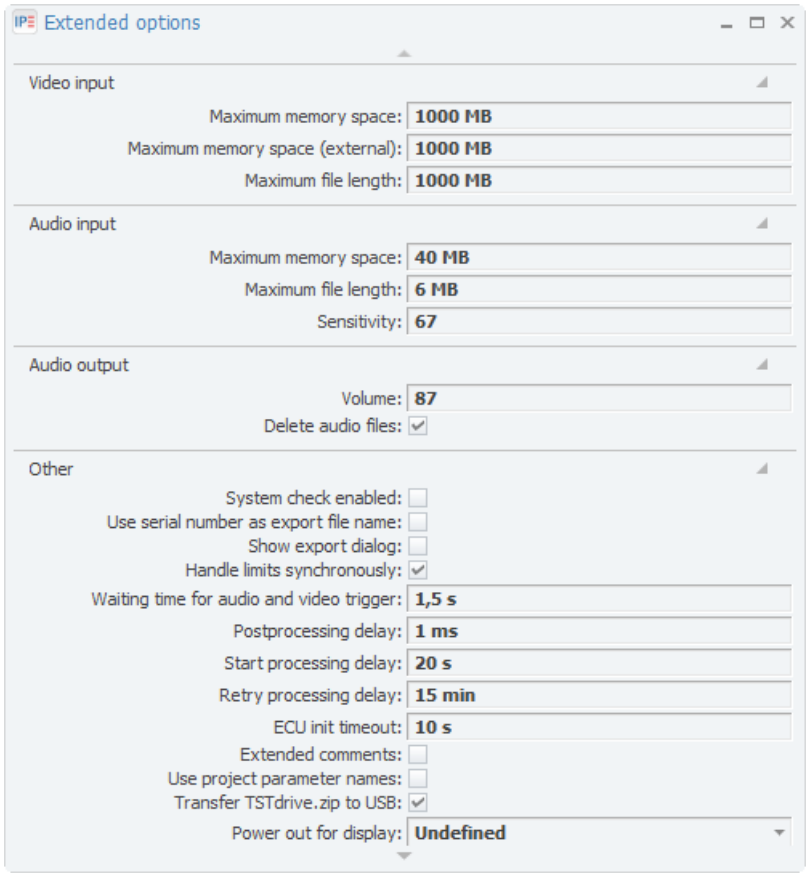
The selection of the hardware components for the configuration by using a signal library is based on the **Priority**. This preselection with a priority assignation of the system components facilitates the device selection and improves the system speed.

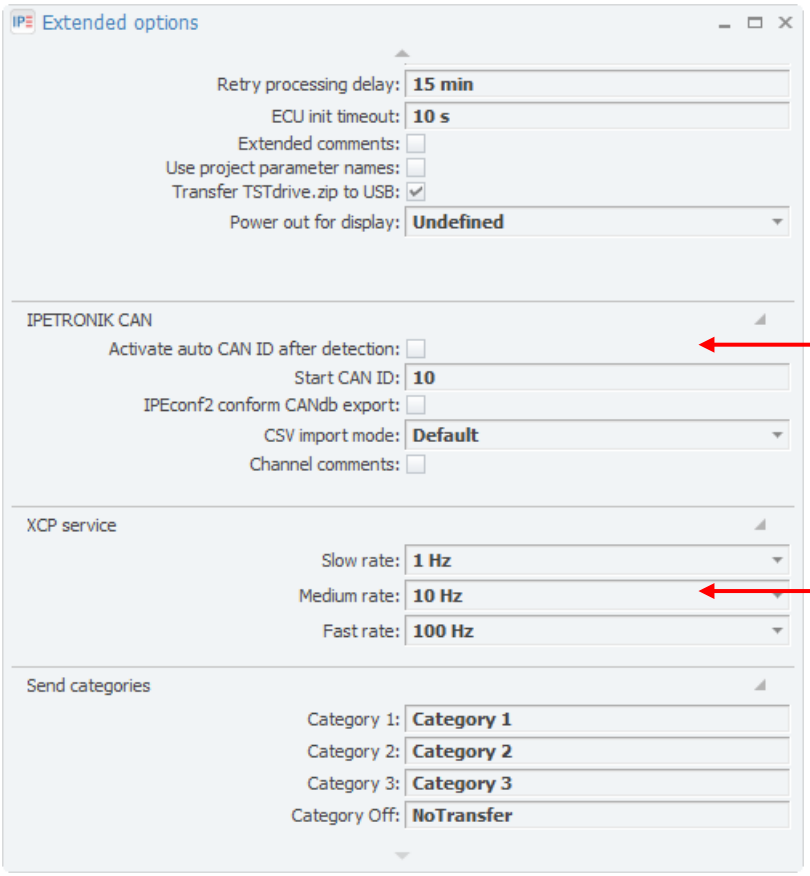
The **High** priority defines a preferred use of the corresponding hardware component at configuring with a signal library. The hardware components, which are defined with the **Not used** priority, cannot be selected for an acquisition.

## Options




Extended options






Activating this check box will execute an automatic CAN ID assignment starting with the predefined ID for connected and identified modules.

Data transmission rate settings

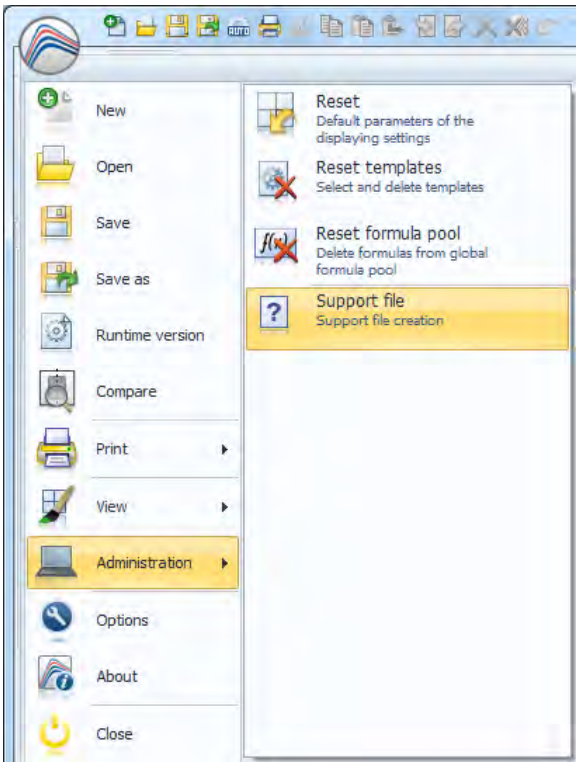
| | |
|---|---|
| slow: | 1 ... 5 Hz |
| medium: | 10 ... 50 Hz |
| fast: | 100 ... 1000 Hz |

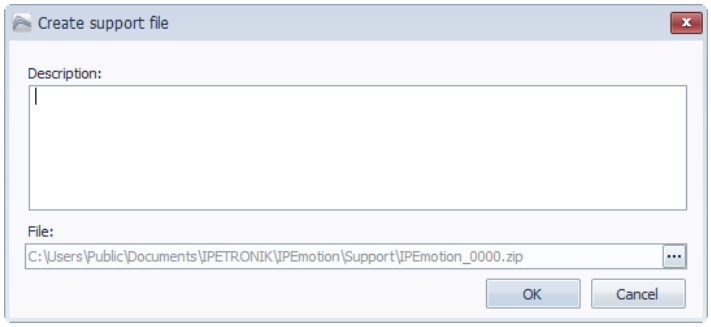
### 5.1.5 Creating a support file

With the **Support file** entry of the application menu, you have the ability to create a support file and add and/or edit own comments and error descriptions.

Enter in the appearing **Create support file** screen an error description. Accept the default location for the file. To select another location click on the ··· symbol.

After you have specified the location and a user defined file name, click **Save** to return to the **Create support file** screen.




After clicking on **OK** a zip file is generated that contains the error description, as well as, the following information:

- System information (Windows version, computer name, free memory on the local drives, …)
- Current configurations (acquisition, online view, script configurations)
- Trace files (.NET, C++)

If you have any problems while working with IPEmotion, send us this support file at support@ipemotion.com.

### 5.1.6 IPEmotion working areas (main navigation tabs)

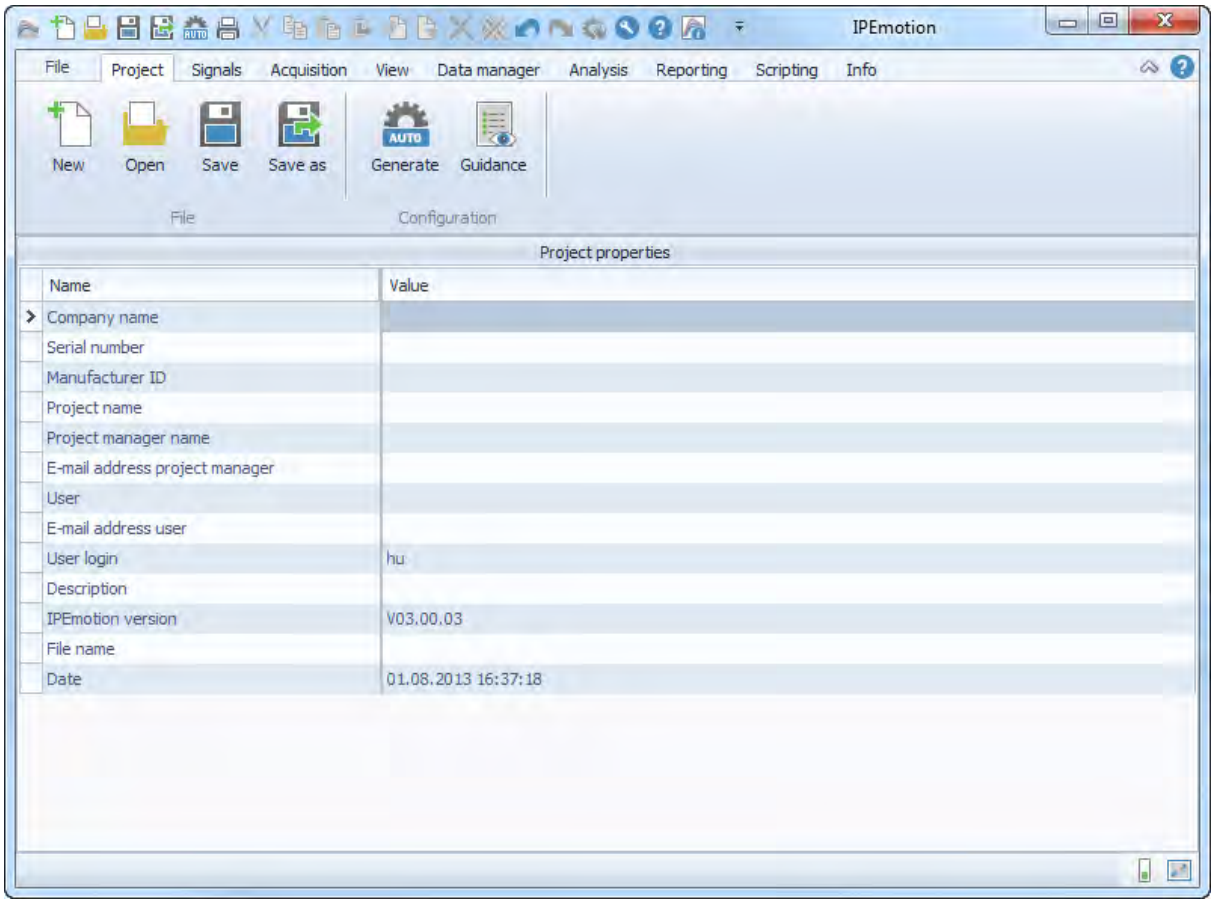
The main navigation tabs allow a quick activation of the different main functions of IPEmotion. A tab displayed in light blue indicates an active function.

IPEmotion is designed to follow the main navigation tabs from left to right. Use this reasonable order like a read thread, which guides you step-by-step to a successful acquisition.

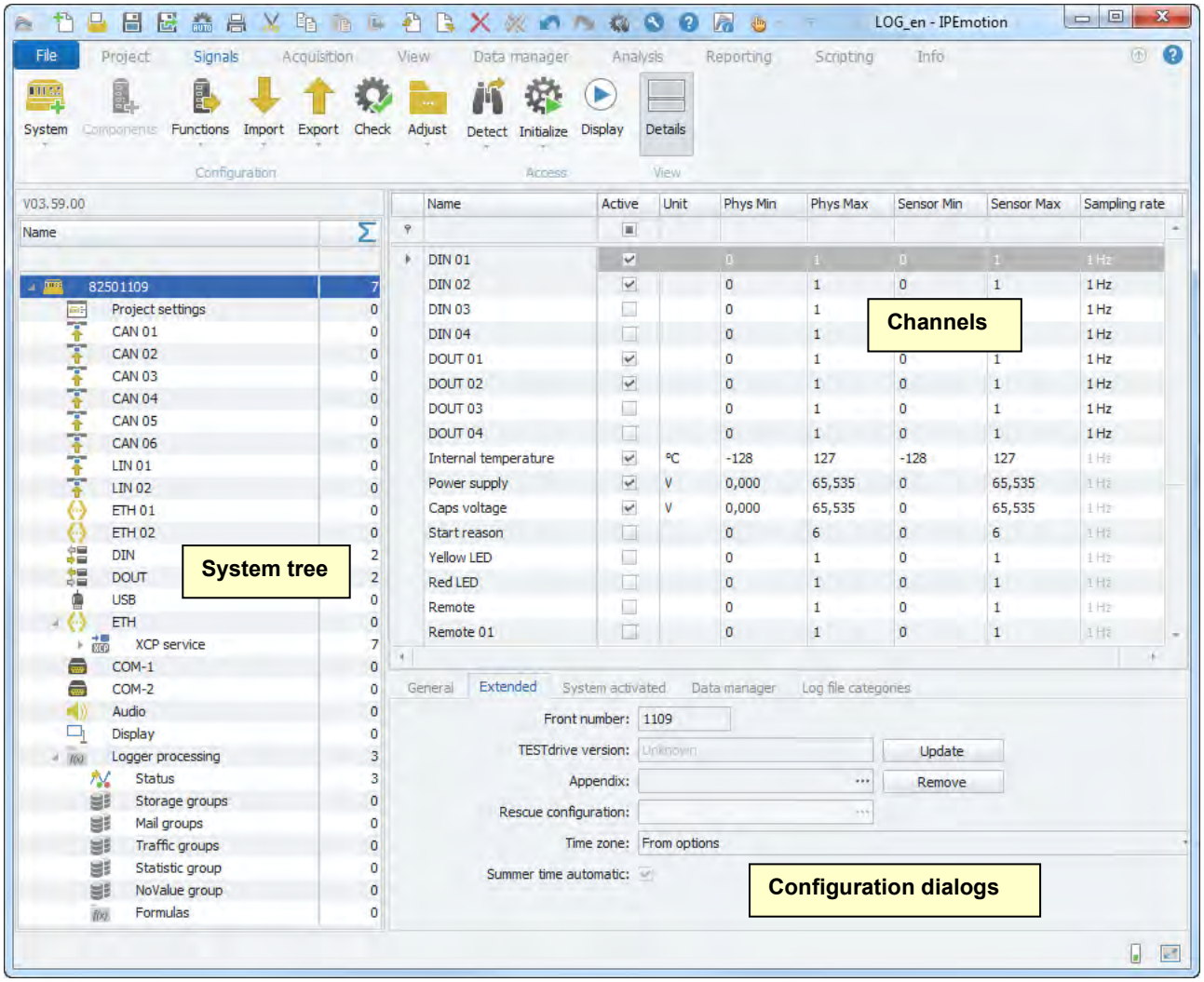
| Project | Signals | Acquisition | View | Data manager | Analysis | Reporting | Scripting | Info |
|---|---|---|---|---|---|---|---|---|

| | |
|---|---|
| **Project** | Define your general user defined project data. |
| **Signals** | Configure the connected acquisition systems and modules. |
| **Acquisition** | Configure the desired storage groups and channels. |
| **View** | Take a measurement defined by the connected hardware modules and the set configurations. |
| **Data manager** | Manage your stored acquired data in all the supported formats. |
| **Analysis** | Visualize your channels with diagrams. |
| **Reporting** | Create reports and project documentations. |
| **Scripting** | Automate your acquisition sequences. |
| **Info** | Get a basic overview and general support. |

#### Project



## Signals




## 5.1.7 Info

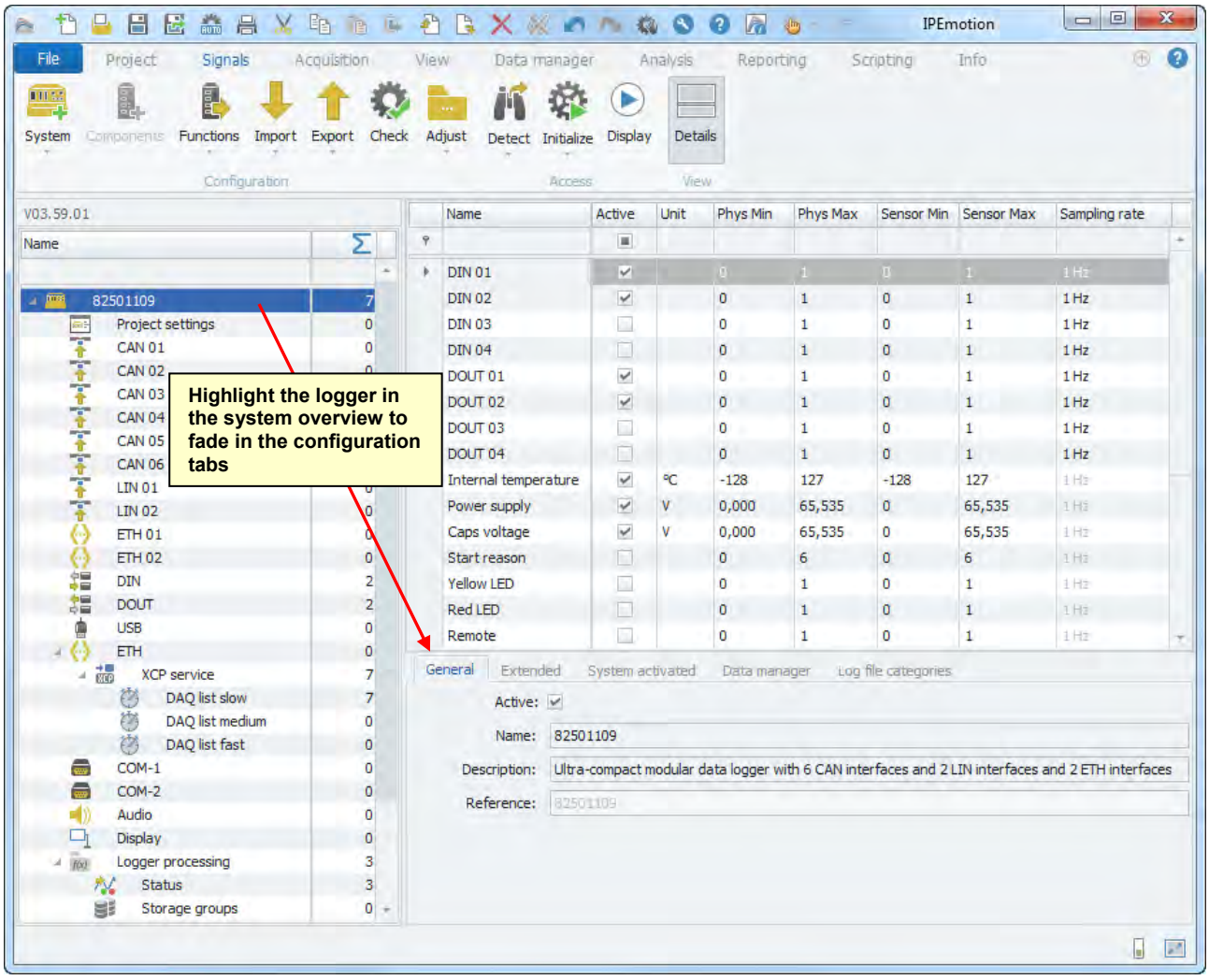
The chapter offers a basic overview of the IPEmotion software. In addition, it shows useful advices and tips and tricks on how to use IPEmotion.

The view Info is divided into the following menu points:

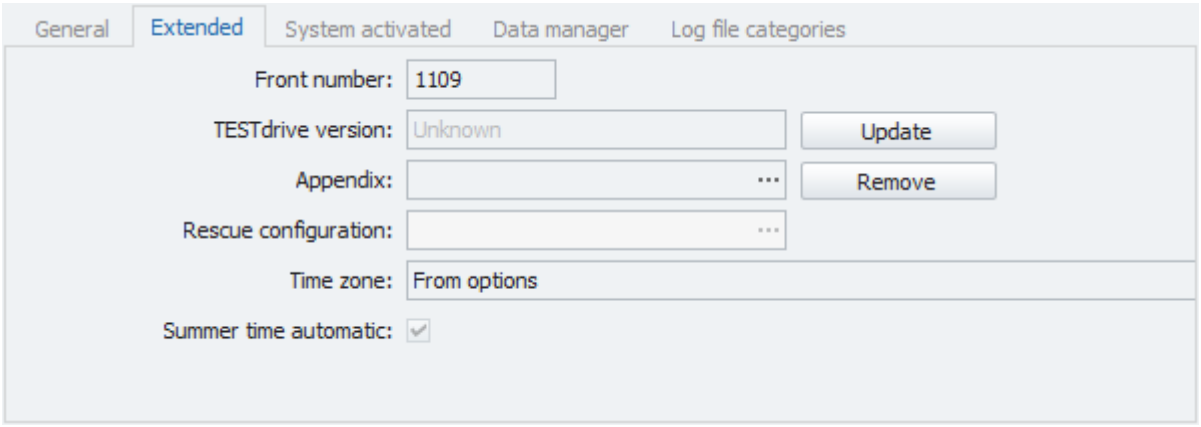
- Welcome
- Release Notes (only in English)
- Red thread
- Tips and tricks
- Keyboard handling
- Documentations
- Contact and support

## 5.2 Logger settings

### General




### Extended



**Front number:** The last four digits of the logger serial number.

**TESTdrive version:** Version of the TESTdrive application running on the logger. With a hardware detection through a LAN/WLAN connection, the current version is indicated. A mouse click to **Update** will upload the latest version.

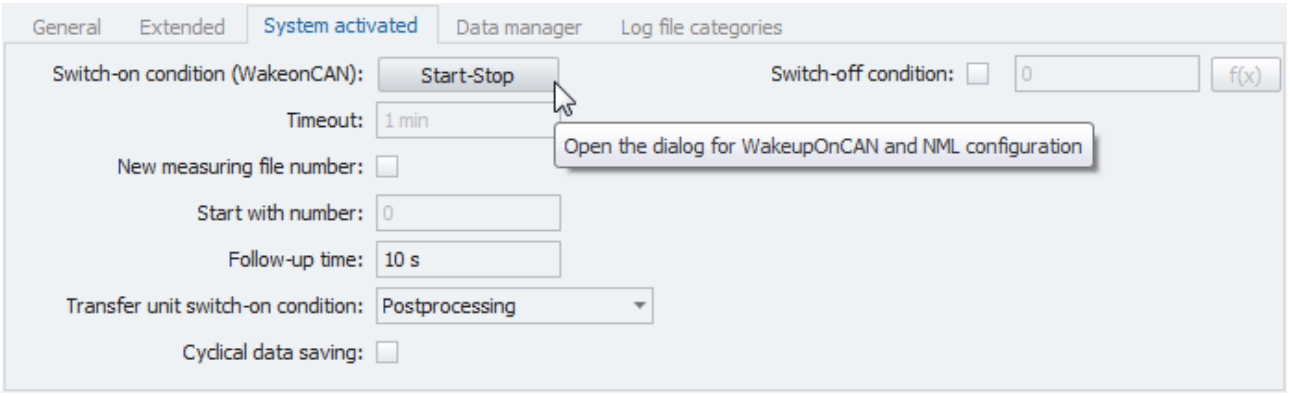
**Appendix:** Transfer a user specific file to the logger.

**Rescue configuration:** Store rescue configuration on the logger (fall back)

**Time zone:** Applying the PlugIn settings defined under **Options > PlugIns > PlugIn-specific settings > Time zone**

**Summer tine automatic:** Automatic switch-over winter time <> summer time

## System activated




**Switch-on condition (WakeonCAN):** Selecting the CAN bus(es) to be used for boot up the logger by the WakeonCAN feature.

**Timeout:** The logger waits this period for the remote signal resp. bus activity.

**Switch-off condition:** User defined condition (trigger, formula) to be used to switch-off the logger. In case a remote signal (REM = ignition line 15) is still high, the logger will shut down and power up again.

**New measuring file number:** Consecutively numbering of measurement data files beginning with the predefined start number.

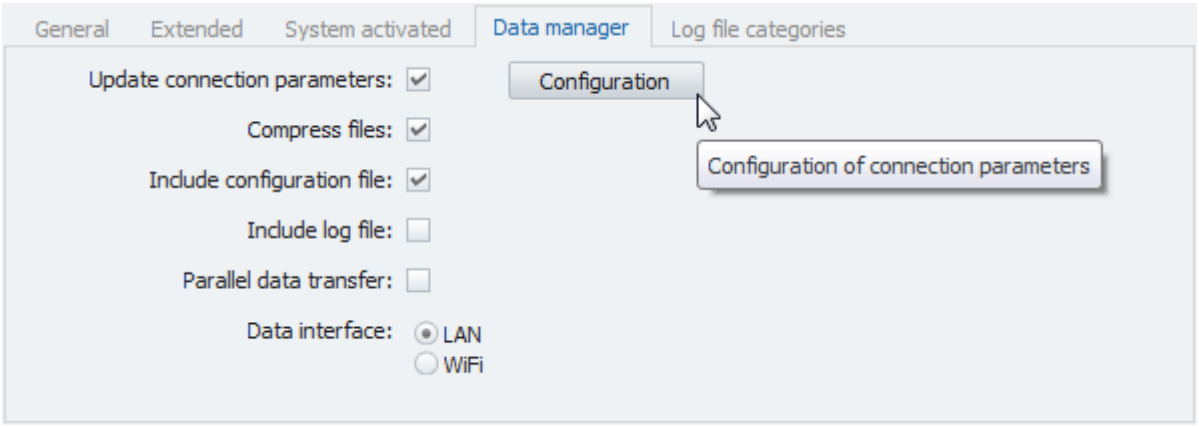
**Start with number:** Start number for the consecutive measurement file numbering

**Follow-up time:** When the switch-off signal has been identified, the logger waits this period (and keeps on operating) before shut down.

**Transfer unit switch-on condition:** Switch on the WiFi client respective the modem for wireless data transfer by the logger.

**Cyclical data saving:** Acquired data is continuously stored to the measurement data file (not only by stopping the measurement).

## Data manager




**Update connection parameters:** Enables the access to connection parameters settings by use of the **Configuration** button.

**Compress files:** Activated the compressing of measurement data files

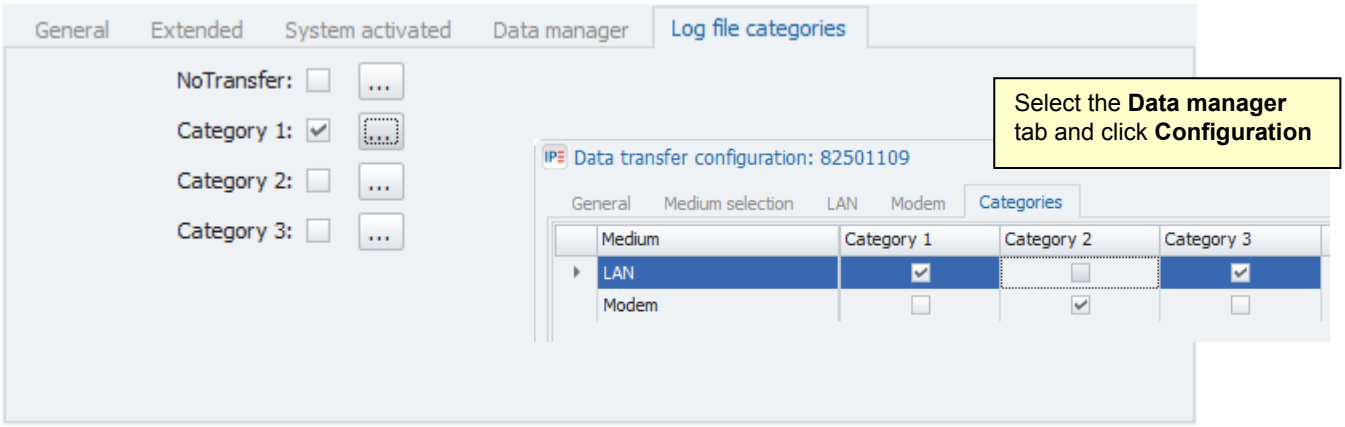
**Include configuration file:** Includes the DAQ configuration (**CFG_xxx.isf**) in the measurement data zip file **xxx_MEA_xxx.zip**.

**Include log file:** Includes the DAQ log file (**MEA_xxx.log**) in the measurement data zip file **xxx_MEA_xxx.zip**.

**Parallel data transfer:** Enables data transfer simultaneous to a running data acquisition

**Data interface:** Selection of the communication interface (LAN, WLAN)

## Log file categories




**NoTransfer:** Exclude selected data from transfer.

**Category 1:** Assign storage groups, log file, ... to the data transfer category #1

**Category 2** Assign storage groups, log file, ... to the data transfer category #2.

**Category 3:** Assign storage groups, log file, ... to the data transfer category #3

*Data groups can be linked with a single or multiple data transfer categories.*

*Although a single or multiple data transfer categories can be assigned to a data transfer medium (LAN, modem), a specific category can be assigned to a single medium only.*

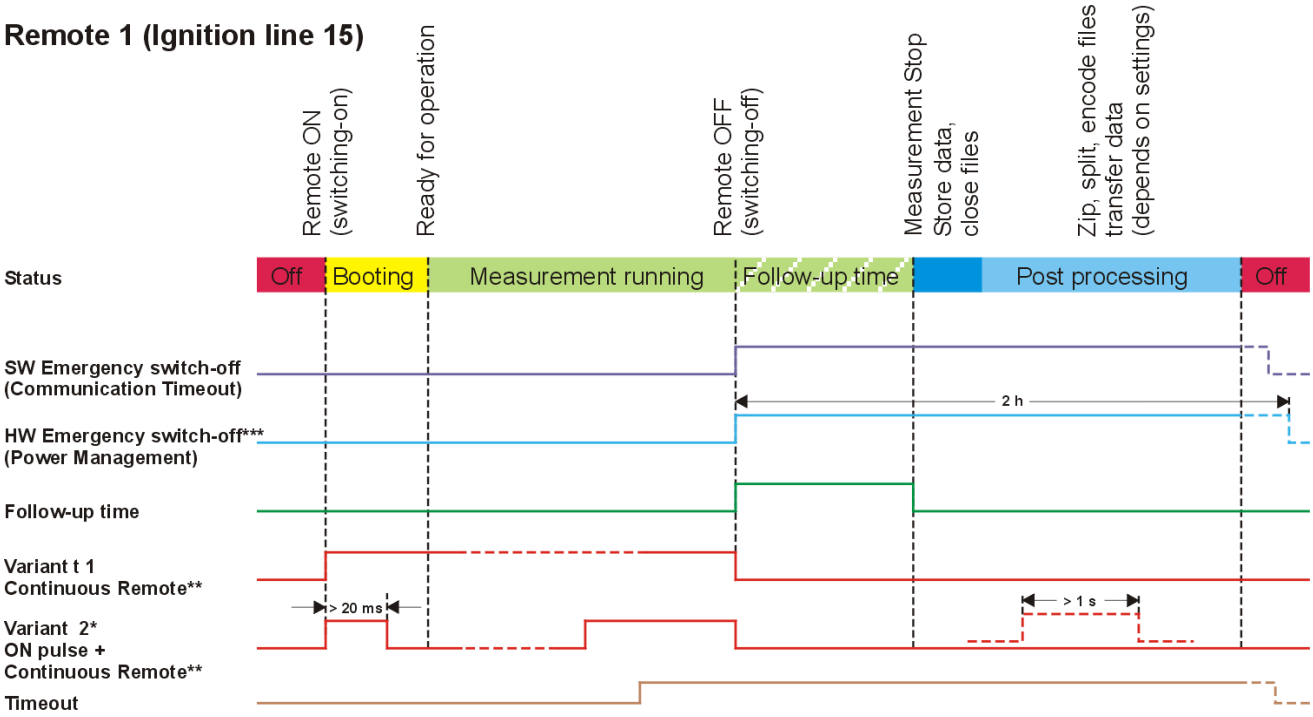
# 6 Basic functions

## 6.1 Switching-on / switching-off

Description of the supported switching-on / switchting-off modes

### 6.1.1 Ignition line 15

**Remote 1 (Ignition line 15)**



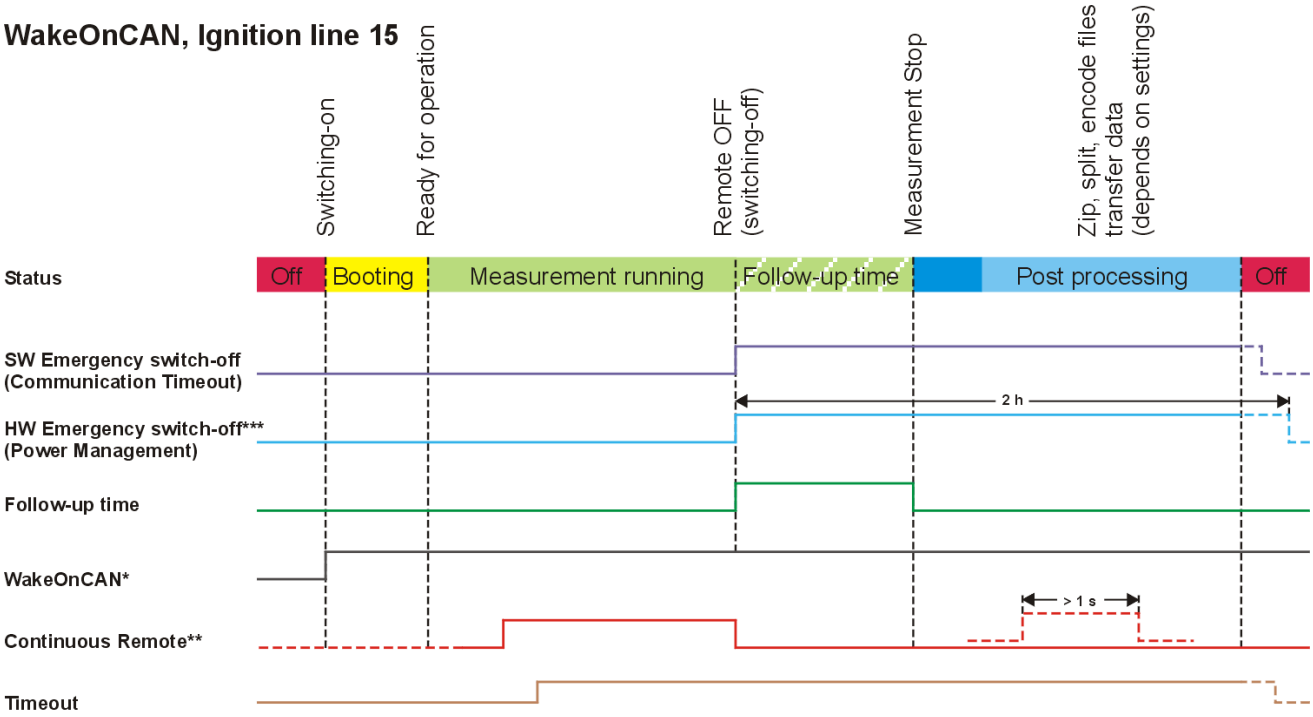

***Variant 2:** Switch-on condition (WakeOnCAN) enabled, continuous remote signal required before timeout expires

****Conitnuous Remote:** If a remote signal (duration > 1 s) is identified during post-processing, the logger will stop this task and start a new acquisition.

*****HW Emergency switch-off:** The logger will be absolutely switched-off by the power management 2 h after the previous falling edge of the remote signal.

## 6.1.2 WakeOnCAN

### WakeOnCAN, Ignition line 15

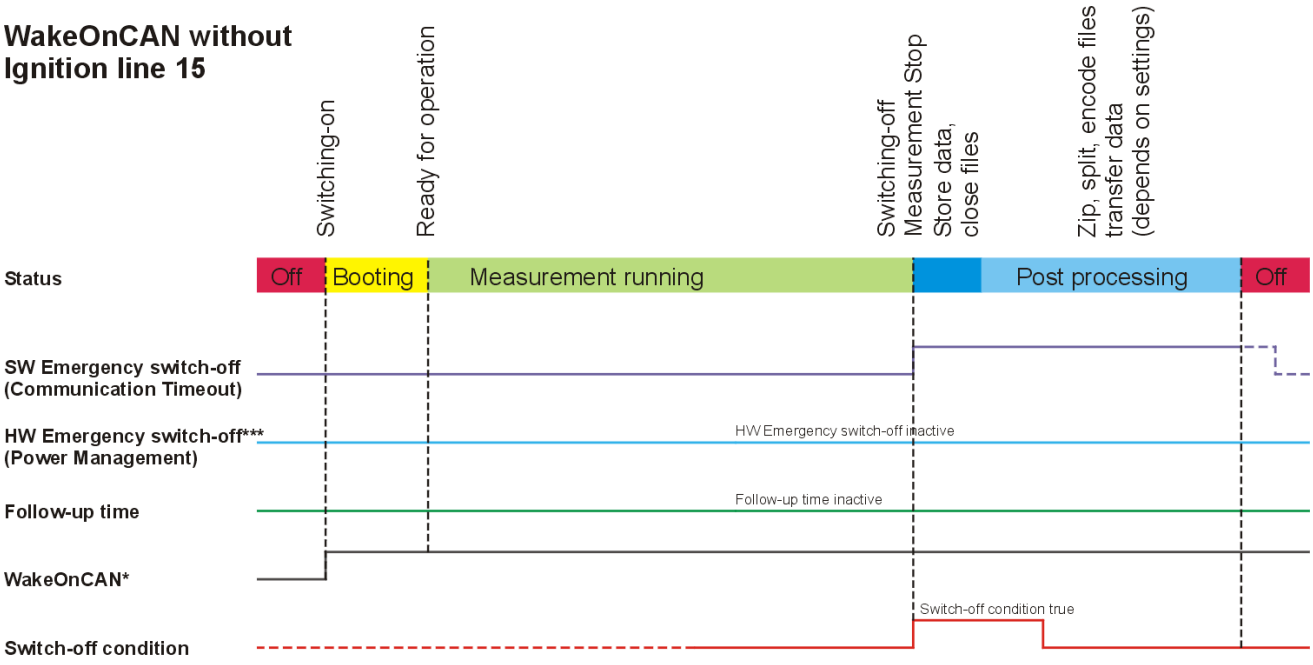



***WakeOnCAN Ign. 15:** Switch-on condition (WakeOnCAN) enabled, continuous remote signal required before timeout expires
jumper Pin 3-6 and Pin 8-9 at respective CAN input to indicate bus traffic (WakeOnCAN)!

****Conitnuous Remote:** If a remote signal (duration > 1 s) is identified during post-processing, the logger will stop this task and start a new acquisition.

*****HW Emergency switch-off:** The logger will be absolutely switched-off by the power management 2 h after the previous falling edge

### WakeOnCAN without Ignition line 15




***WakeOnCAN without Ign. 15:** Switch-on condition (WakeOnCAN) enabled,, Switch-off condition configured,,
HW Emergency switch-off and Follow-up time out of operation, Timeout inactive
Jumper Pin 3-6 and Pin 8-9 at respective CAN input to indicate bus traffic (WakeOnCAN)!

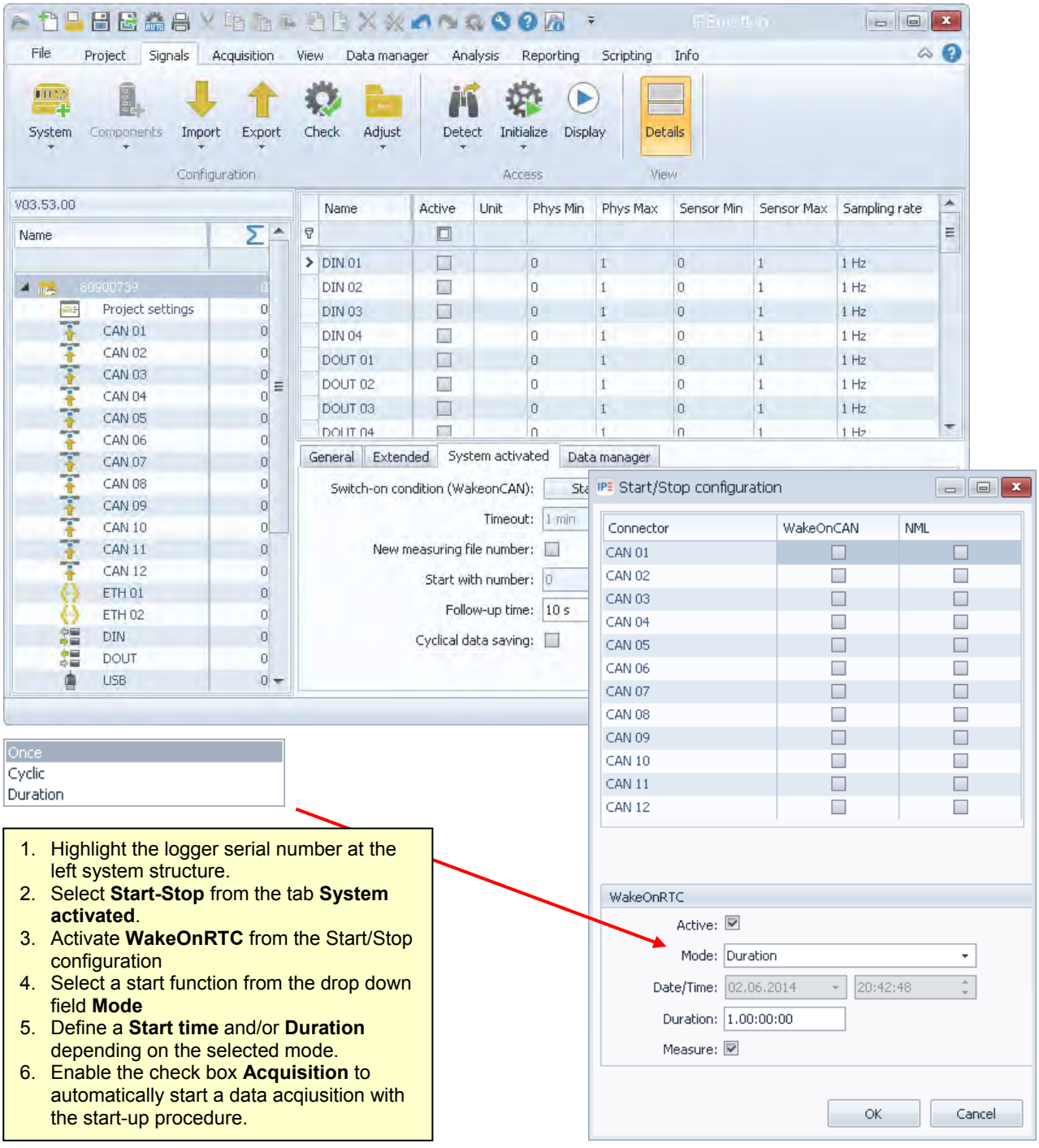
### 6.1.3 WakeOnRTC (IPElog/IPElog2 only)

With TESTdrive 3.52 and higher versions, IPElog supports switching on by its internal real time clock (RTC).

Supported events:

- Once start only one single time.
- Cyclic absolute cyclic start e.g. every hour (duration --> 1h), date/time stamp defines initial start up.
- Duration start e.g. every hour without predefined start-up.

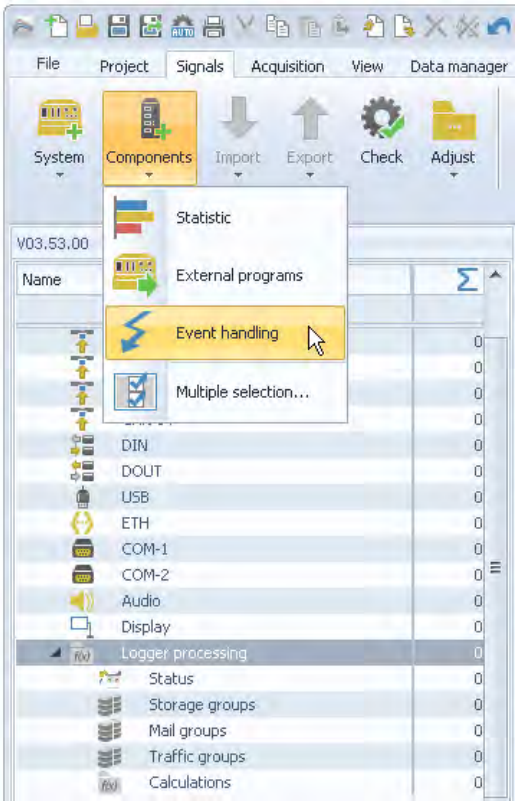
*If WakeOnCAN, WakeOnRTC or WakeOnSMS is configured with IPElog, the Ethernet status LEDs (LINK, ETH1, ETH2) will light green, even if the logger is in standby mode.*

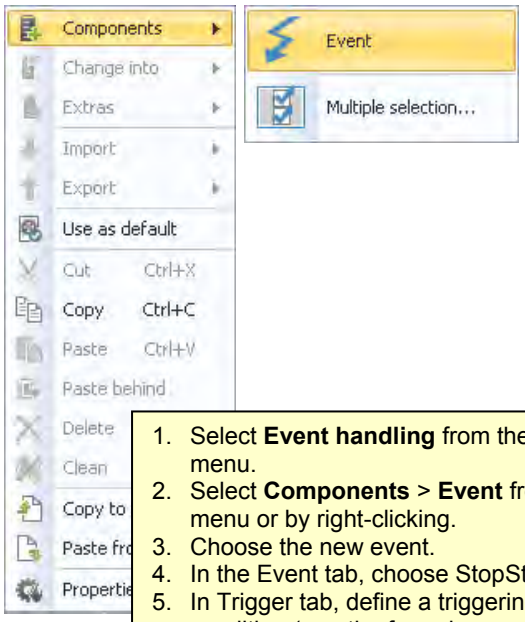



1. Highlight the logger serial number at the left system structure.
2. Select **Start-Stop** from the tab **System activated**.
3. Activate **WakeOnRTC** from the Start/Stop configuration
4. Select a start function from the drop down field **Mode**
5. Define a **Start time** and/or **Duration** depending on the selected mode.
6. Enable the check box **Acquisition** to automatically start a data acqiusition with the start-up procedure.

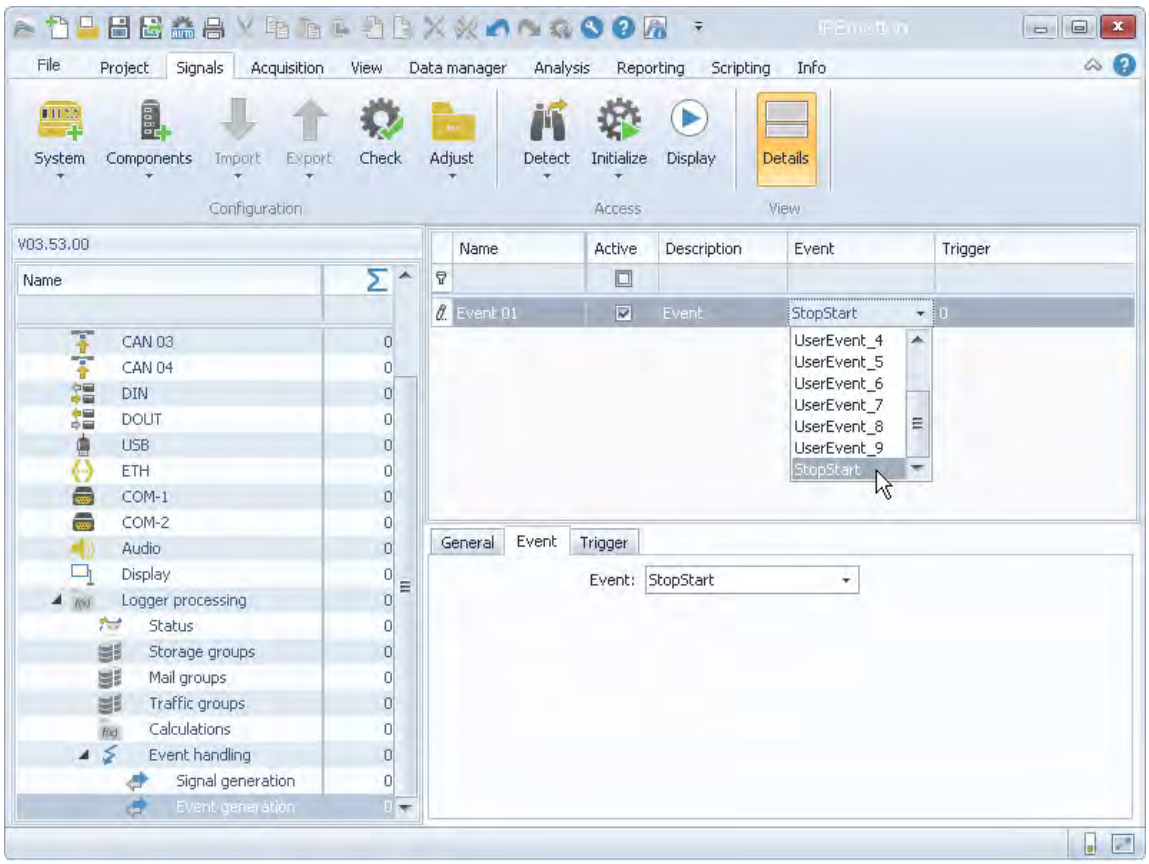
## 6.1.4 StopStart event (Logger processing)

The StopStart function closes the measurement file and after that immediately starts a new measurement. This function is triggered by a user-defined event and starts without switching off the logger.




1. Select **Event handling** from the left-hand menu.
2. Select **Components** > **Event** from the tab menu or by right-clicking.
3. Choose the new event.
4. In the Event tab, choose StopStart.
5. In Trigger tab, define a triggering condition (e.g. the formula: "DIN 01" AND "DIN 02").



## 6.1.5 Use Cases

### WakeOnCAN with ignition on/off

#### Application

Test run with continuous measurement via signal “Ignition on / ignition off“ (remote); The logger is being started using bus traffic (WakeOnCAN) in order to acquire the entire starting process. After test run, CAN activity is to be acquired using the condition „Ignition off“.

#### Requirements

- Data logger supporting WakeOnCAN feature
- Recognition communication stop on vehicle buses:
  > IPElog or MLOG with 4CANQS cards and FPGA as of 1.08.01, TESTdrive as of 3.52.00

#### Sequence

Logger turned off
↓
Remote control Central locking
↓
Vehicle buses wake up
↓
Logger boots
↓
TESTdrive is started
↓
Ignition on
↓
Drive begins
↓
Test run
↓
Drive ends
↓
Ignition off
↓
ECUs still communicate
↓
Bus is idle
↓
Follow-up time (if configured)
↓
Data post-processing
(Zipping, copying,… if configured)
↓
Data transfer (if configured)
↓
Logger turns off

## WakeOnCAN (bus activity)

### Application

Monitoring of vehicle component on CAN, i.e. measurement as soon as/as long as the component is active, independent from ignition state.

### Requirements

- Data logger supporting WakeOnCAN feature
- Recognition communication stop on vehicle buses:
  > IPElog or MLOG with 4CANQS cards and FPGA as of 1.08.01, TESTdrive as of 3.52.00
- Alternatively for other logger types / firmware versions:
  > Use of switch-off condition or WakeOnCAN timeout

### Sequence

Logger turned off
↓
Vehicle buses wake up
↓
Logger boots
↓
TESTdrive is started
↓
Bus is idle
↓
Follow-up time (if configured)
↓
Data post-processing
(Zipping, copying,… if configured)
↓
Data transfer (if configured)
↓
Logger turns off

## WakeOnRTC

### Application

Independent from ignition state and bus traffic, the logger should start at 04:00 am every night, acquire data via connected analog modules for 2 minutes and switch back to idle state.

### Requirements

- IPElog with TESTdrive as of 3.52.00

### Sequence

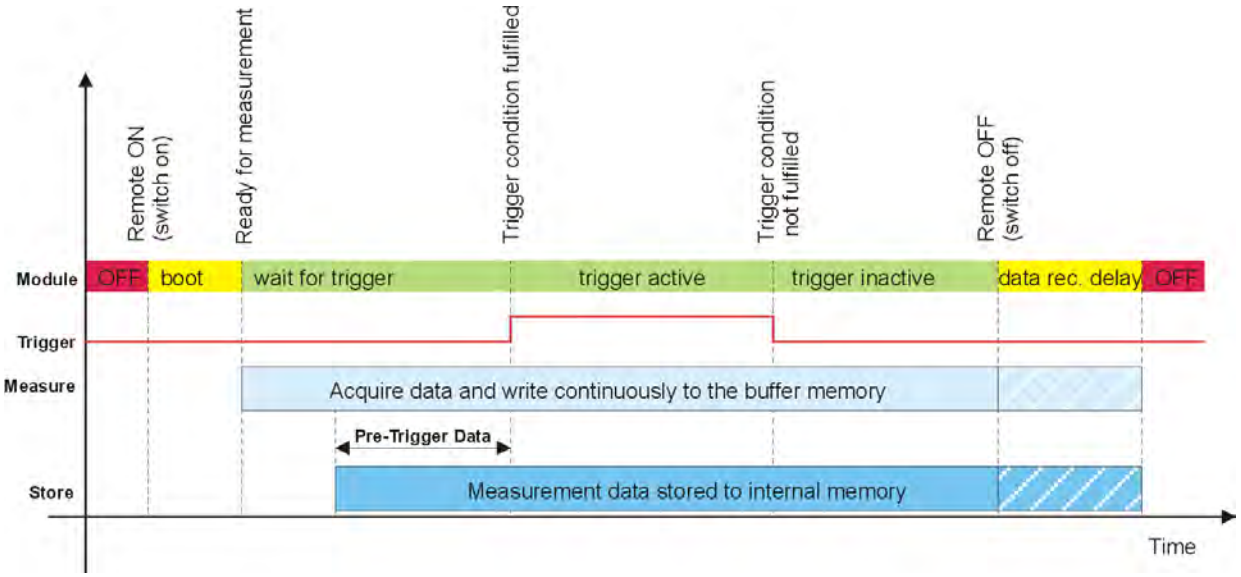
Logger turned off
↓
WakeOnRTC at 04:00 am
↓
Logger boots
↓
TESTdrive is started
↓
Configured turn-off condition becomes TRUE
↓
Data post-processing
(Zipping, copying,… if configured)
↓
Data transfer (if configured)
↓
Logger turns off

## 6.2 Triggering

The logger acquisition program offers 4 trigger conditions for every storage group to control the data acquisition. The trigger conditions can be deduced from acquired signals, as well as, from calculated channels. All data is written into the memory (RAM). If a storage condition is met, the data is asynchronously written from the cache into the open measuring file on the flash card.

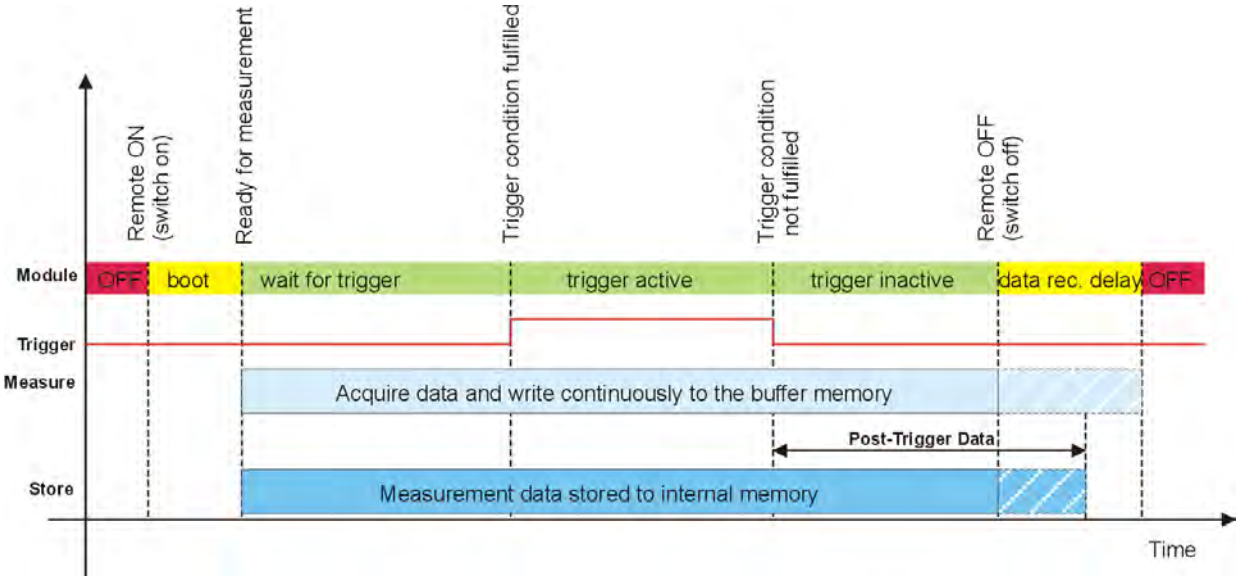
### 6.2.1 Start-trigger

Start of data storage if trigger condition (impulse) is met. Stop of storage with (correct) logger shut-down (Power down). A defined follow-up time extends the data acquisition for x seconds.




### 6.2.2 Stop-trigger

Begin of data storage with logger switch-on (Power up). Stop of data storage if trigger condition (impulse) is met. A defined follow-up time extends the data acquisition for x seconds. If no trigger event is set, data is recorded until “Remote OFF” or end of the follow-up time.

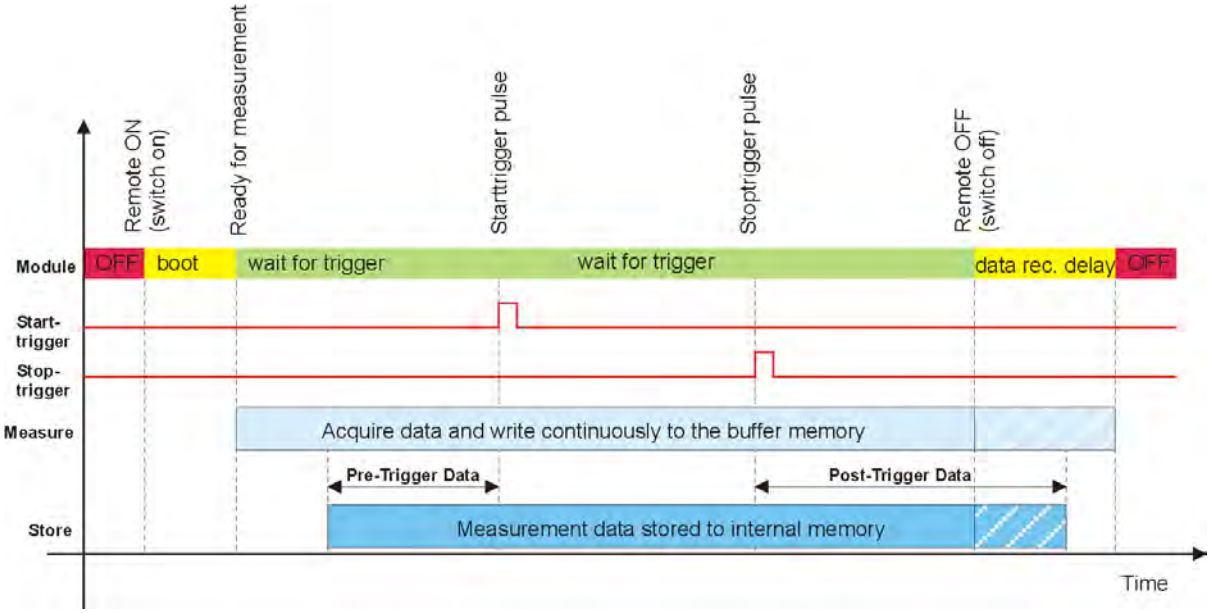



*We recommend to use the function VALID(x ; y) in order to avoid irregular NoValues which may influence a correct triggering.*
*Example:*
*Start-trigger: VALID(„Channel01“ > 15; 0)* *Storage as soon as Channel01 > 15*
*Stop-trigger: VALID(„Channel01“ > 15, 1)* *Storage as long as Channel01 > 15*

### 6.2.3 Start- and Stop-trigger

Begin of data storage if start-trigger condition (impulse) is met. Stop of data storage if stop-trigger condition (impulse) is met.

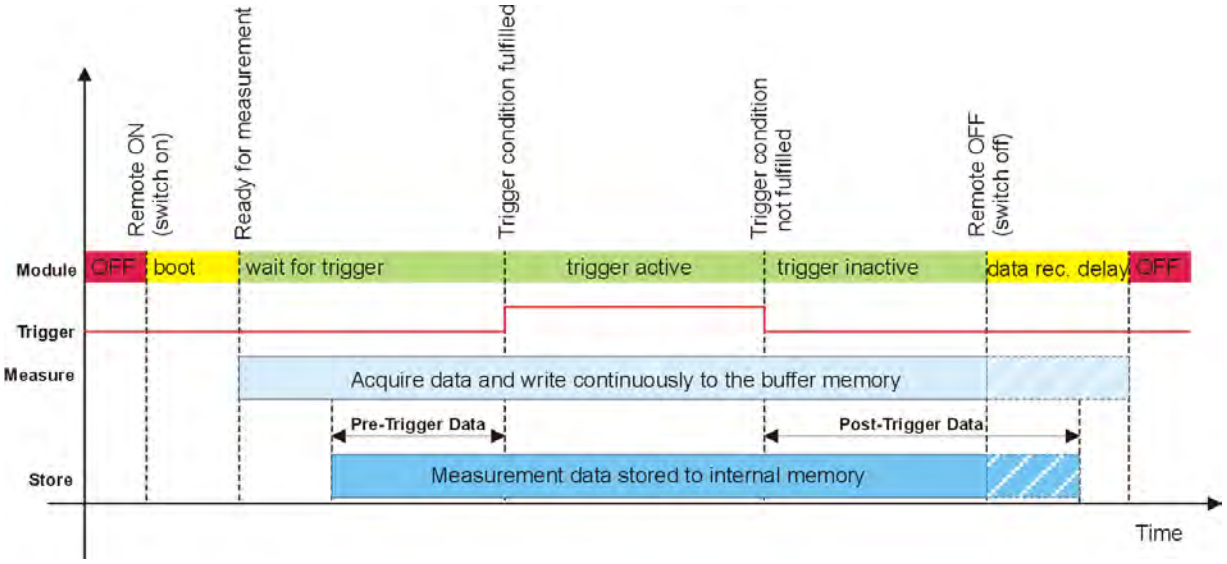
A defined follow-up time extends the data acquisition for x seconds. If no stop-trigger event is set, data is recorded until “Remote OFF” or end of the follow-up time.




### 6.2.4 Stop is inverted start

Data storage for the time of meeting trigger condition (status).

A defined follow-up time extends the data acquisition for x seconds. If trigger status does not change after successful trigger condition, data is recorded until “Remote OFF” or end of the follow-up time.




*The **Follow-up time** may not be mistaken for the **Post-processing time**. Both settings must be coordinated. The maximum post-processing time is set with **Logger** > **Settings** > **Data transfer timeout**. This setting limites the switch-on duration after successful “Remote OFF” signal. The logger is regularly shut down after this time even if the data post-processing (zipping, splitting, sending) is not yet completed. The post-processing value must be at least 5 min greater than the currently set follow-up time!*

### 6.2.5 Save trigger channel

If **Save trigger channel** is activated, trigger status and some additional information are stored in an implicit channel (Word data format) in every storage group.

| Bit coded information in trigger status channel | | |
|---|---|---|
| **Bit No.** | **Description** | **Description (if bit value = 1)** |
| **0** | Pre-Trigger | Pre-trigger time running |
| **1** | Between start and stop | Trigger signal status<br>This bit is set during the whole acquisition in **Continuous acquisition** mode. |
| **2** | Post-trigger | Post-trigger time running |
| **3** | Trigger Event | This bit is set for the time of one signal at every Low > High of the trigger signal. It is set once at acquisition start in **Continuous acquisition** mode. |
| **4** | Maneuver | Maneuver recording running (No NoValues available anymore) |
| **5** | Res | Currently not used!! |
| **6** | Res | Currently not used! |
| **7** | Res | Currently not used! |
| **8** | Res | Currently not used! |
| **9** | Res | Currently not used! |
| **10** | Res | Currently not used! |
| **11** | Res | Currently not used! |
| **12** | WakeOnCAN | WakeOnCAN ist active |
| **13** | Power Bad | The logger is disconnected from the power supply. |
| **14** | Power Good | Buffer capacitors loaded |
| **15** | KL. 15 | Debounced remote signall (terminal 15) |

## 6.3 Data groups (storage, e-mail, traffic, statistic, novalue)

TESTdrive 3,55 supports 5 different types of data groups:

- Storage groups — time related data storage on the logger
- Mail groups — send data snap shot at trigger event per e-mail
- Traffic groups — event related (time stamp) data storage on the logger (trace data), license required
- Statistic group — STG file (Statistic Group) containing min, max, mean values of the respective signals created at the end of each measurement
- NoValue group — NoValue monitoring of defined signals

TESTdrive supports various data groups to merge signals (direct signals, as well as, calculated channels) for further data processing. Storage groups can have an own storage rates – independent from the signal rate. This storage rate can be valid for all signals in the group, i.e. the signal is detected with the set sampling rate and recorded with the storage rate or the individual signal sample rate is used for storage.

The maximum storage rate of a group is defined by the signal with the highest sampling rate. The frequency for online calculations corresponds to the highest sampling rate in the calculation.

Each data group has its own trigger condition (see **6.2 Triggering**) which can be individually defined by the user. This offers the possibility to process (storage, sending) selected signals dependent on specific events occurring, which may reduces the amount stored data and provides an easier offline data analysis.

| **Overview** | | | | |
|---|---|---|---|---|
| **Data group** | **Settings** | **Trigger** | **Category** | **Remark** |
| Storage groups | storage rates, time stamp and trigger channel, NoValues, enable external storage medium , storage group name = mea prefix | different trigger modes available, formula to define trigger condition | Yes | time related cyclic data acquisition |
| Traffic groups | enable external storage medium , storage group name = mea prefix | different trigger modes available, formula to define trigger condition | Yes | time stamp related bus trace acquisition |
| Mail groups | recipients, subject line | formula to define trigger condition | No | signal snap shot at trigger event as status information |
| Statistic group | none | none | Yes | min/max /average values at measurement stop |
| Novalue group | none | formula to define trigger condition | No | log file reporting |

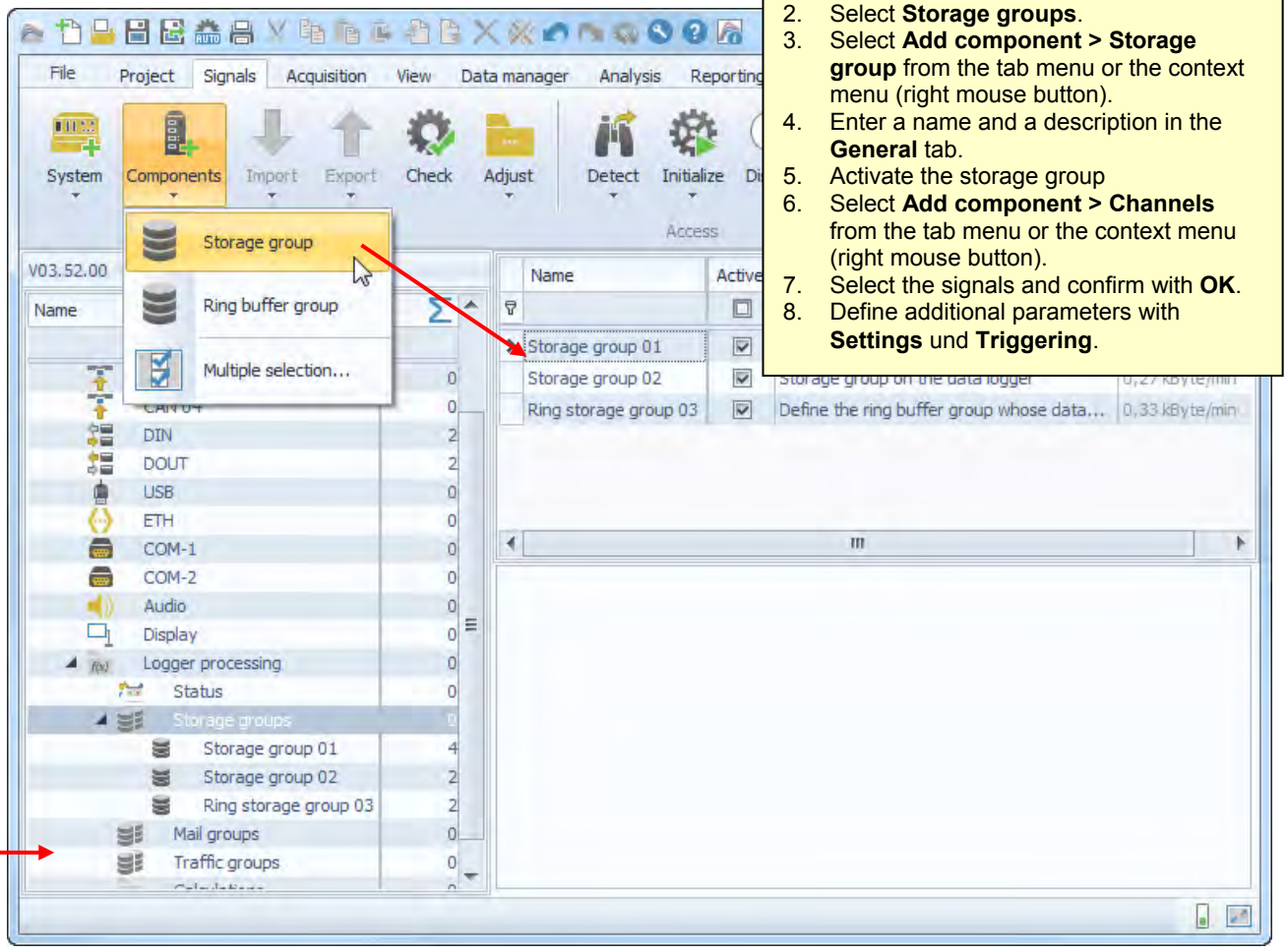
### Categories (Data transfer category)

Data groups that create measurement data files can be assigned to data transfer categories 1, 2, 3 or they can be excluded from data transfer. Each category can be linked with a single data transfer medium, whereas all three categories can be assigned to the same data transfer medium.

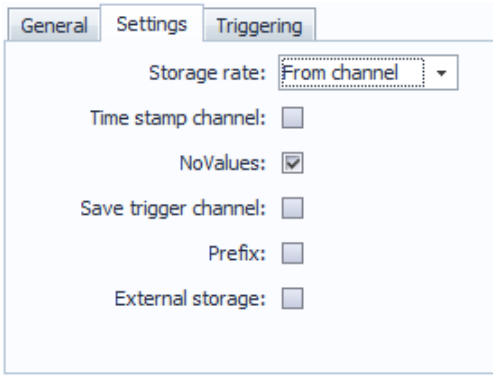
Refer to **11.2.3 Data transfer & communication**

## 6.3.1 Storage groups

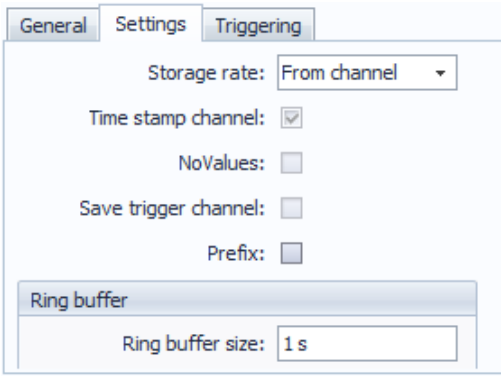
1. Select **Signals** navigation tab.
2. Select **Storage groups**.
3. Select **Add component > Storage group** from the tab menu or the context menu (right mouse button).
4. Enter a name and a description in the **General** tab.
5. Activate the storage group
6. Select **Add component > Channels** from the tab menu or the context menu (right mouse button).
7. Select the signals and confirm with **OK**.
8. Define additional parameters with **Settings** und **Triggering**.




**Storage group**




**Ring buffer group**




**Storage rate** — Selection of a fixed storage rate or a channel related storage rate (From channel). Using the setting **From channel** means, .the channel with the highest sample rate within the storage group will determine the storage rate.

**Time stamp channel** — The measuring data contain the absolute time channel (logger time).

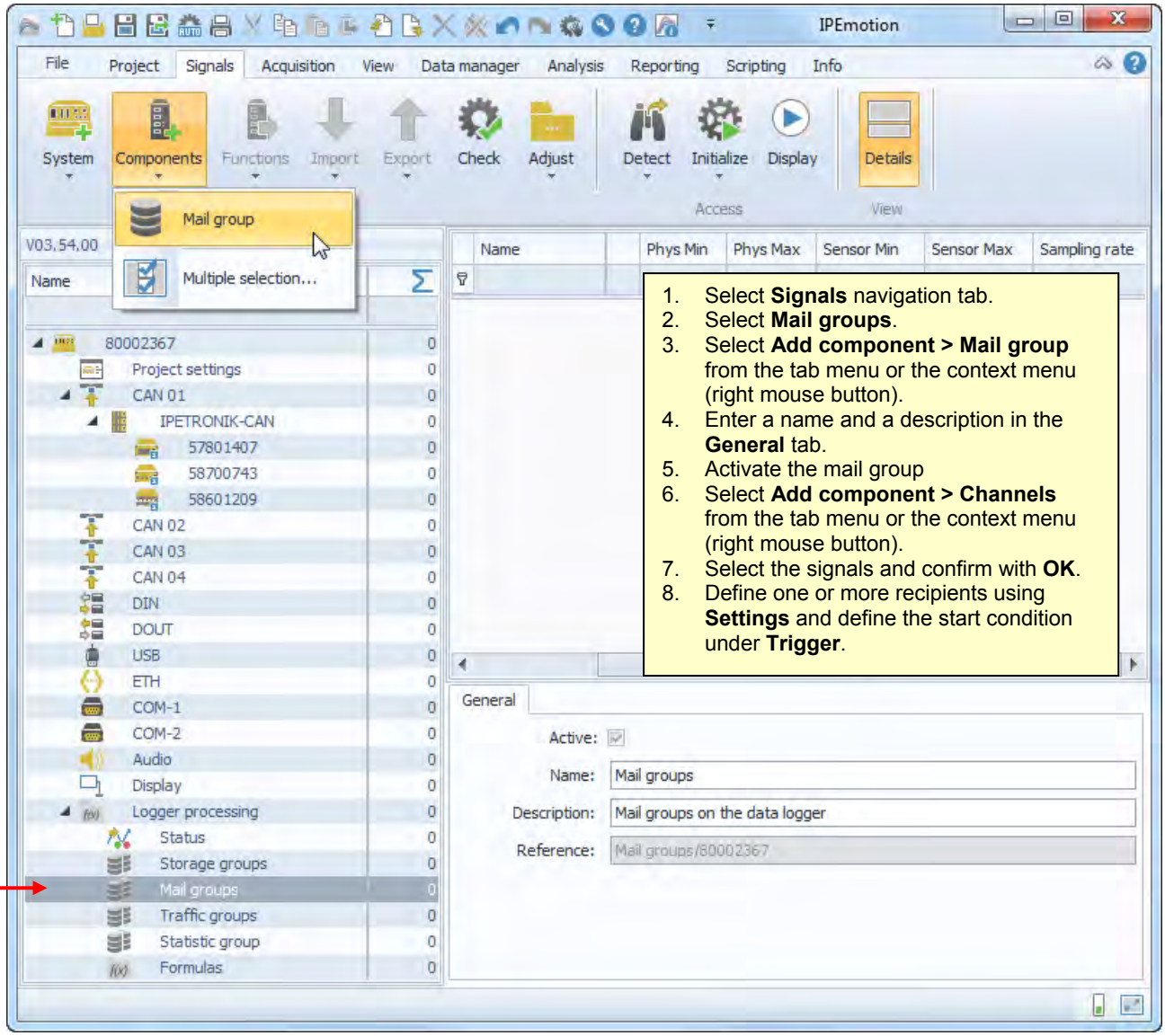
**NoValue** — The NoValue between the trigger events, which is defined in the respective channel, is added with **Format**.

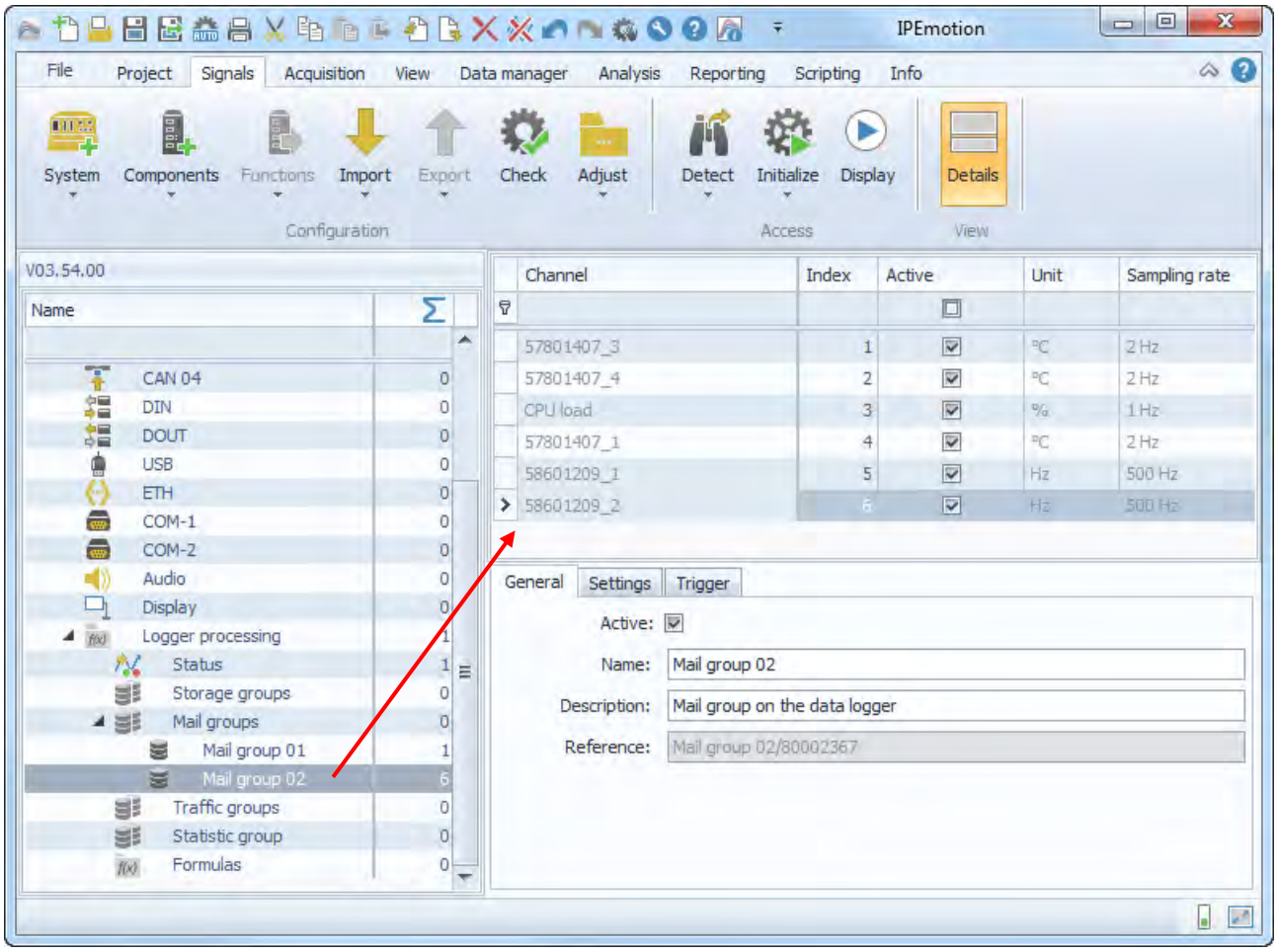
| | |
|---|---|
| **Save trigger channel** | The trigger status channel is added to the storage group and recorded. This channel contains additional information, which is saved with the single bits in the data set. |
| **Prefix** | The storage group name ist used for the measurement data file name. |
| **External storage** | Data storage on external USB storage medium.<br>Refer to ***6.5 USB medium for external storage*** for detailed information. |
| **Ring buffer size** | Defines the size of the ring buffer for measuring. If the max ring buffer size is reached, the memory is overwritten beginning with the oldest data. |

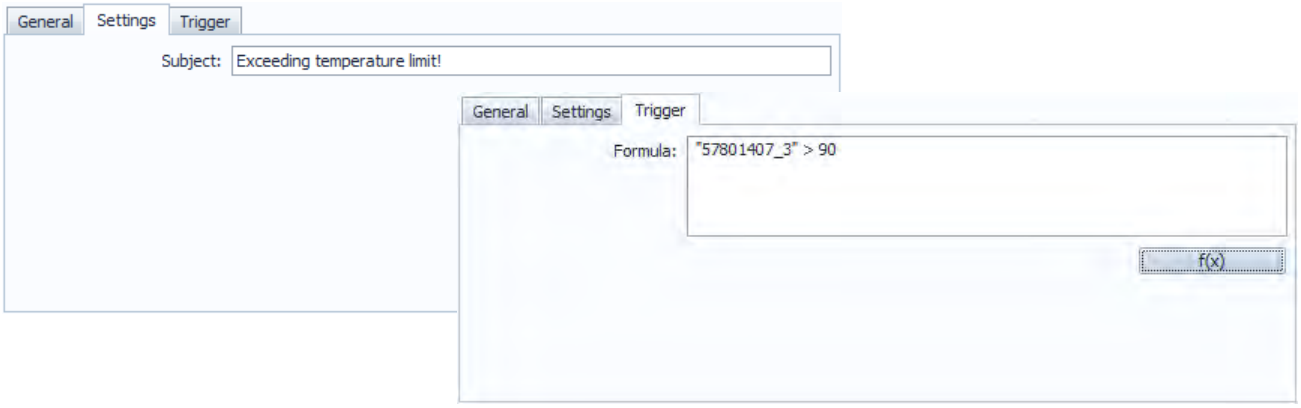
*Use the ring buffer group to record data that are only needed in case an unknown event happens, e.g. in the event of a fault. The ring buffer group contains all selected signals of a defined period before end of measurement.*

*Choosing ring buffer size, consider the number of channels and defined data rates*

## 6.3.2 Mail groups



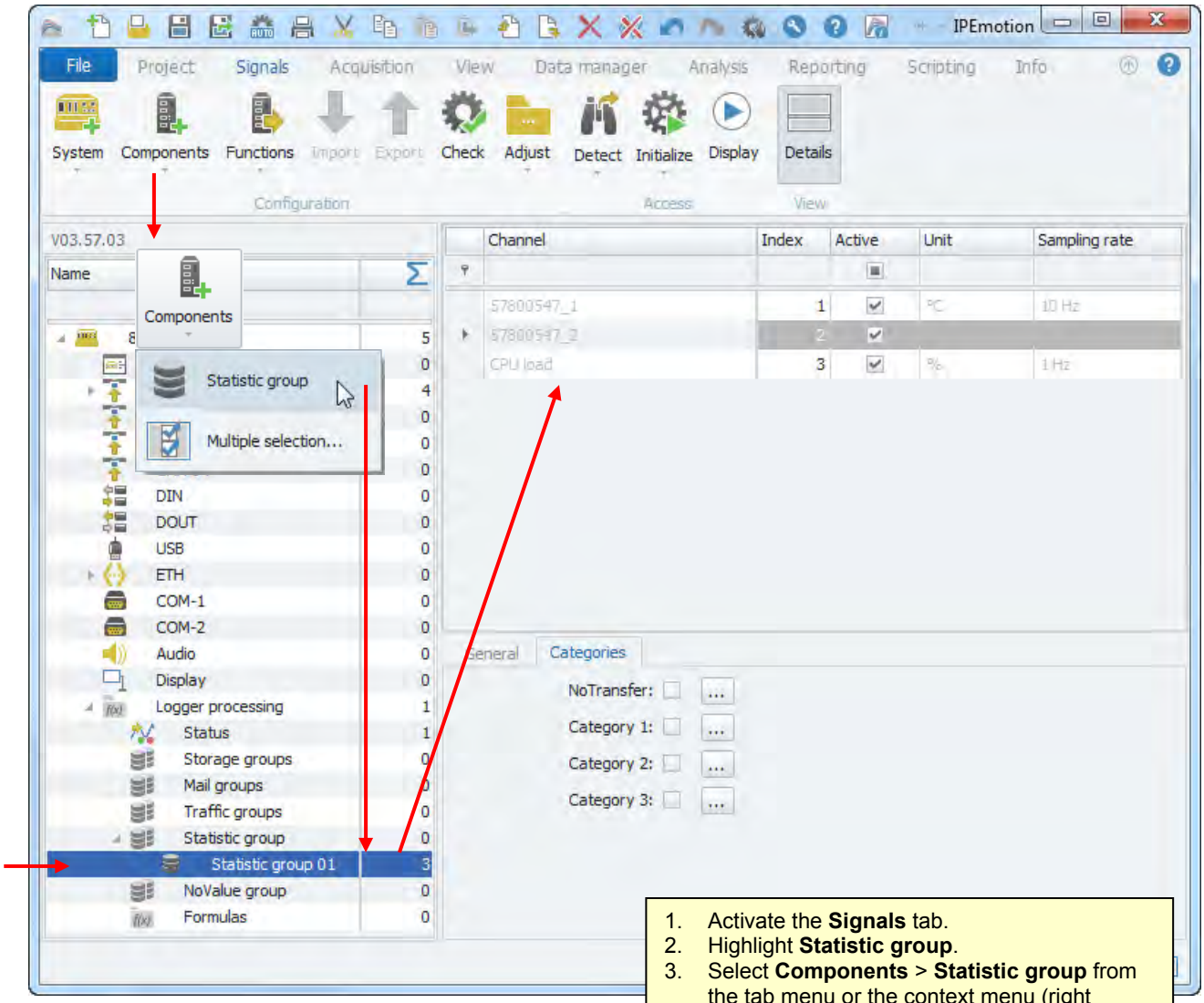



*Enter the logger settings **Data manager** > **Configuration** > **E-mail** to set the e-mail recipient and the mail server.*

*Creating mail groups is independent from a status e-mail.*
*The e-mail from the mail group configuration is send, as soon as the trigger event occurs. In case of activated e-mail delivery using the data transfer configuration, a status e-mail is send at the end of each measurement.*

## 6.3.3 Statistic group




1. Activate the **Signals** tab.
2. Highlight **Statistic group**.
3. Select **Components** > **Statistic group** from the tab menu or the context menu (right mouse button).
4. Enter a name and a description in the **General** tab.
5. Activate the **Statistic group**.
6. Select **Add component > Channels** from the tab menu or the context menu (right mouse button).
7. Select the signals and confirm with **OK**.
8. Definie a data transfer categorie to be used under **Categories**.

Statisic groups contain min, max and mean values of the respective signals stored in the STG file, which is created at the end of each measurement.

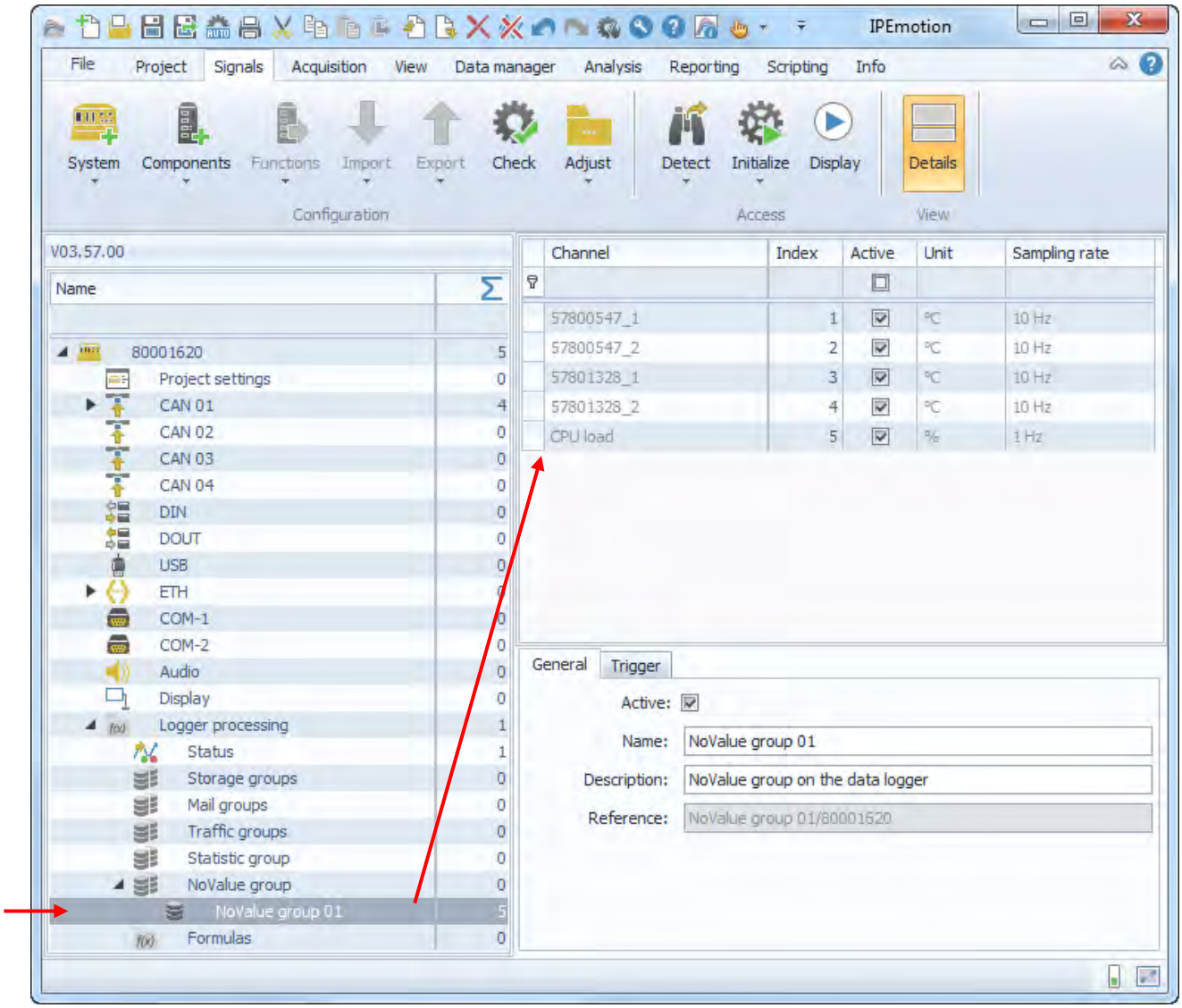
Please also refer to

**12.5.1 Data types > Min-Max list (STG file)**

You can now add your statistic information easily into a new created statistic group. You can create several statistic groups which can be transferred through different means.

Using the “Components” button, you can open the channel selection window to select the channel you would like to include in the Min-Max statistic calculation.

## 6.3.4 NoValue group




A NoValue group serves for monitoring invalid values occuring during data acquisition. Each NoValue event (means that a sequence of valid signal values is interrupted by one or more invalid values) creates a log file entry. You can use specific triggers to activate the NoValue group (for example: “CPU load“ > 30 %).

Refer to **7.1.3 NoValue and timeout settings**

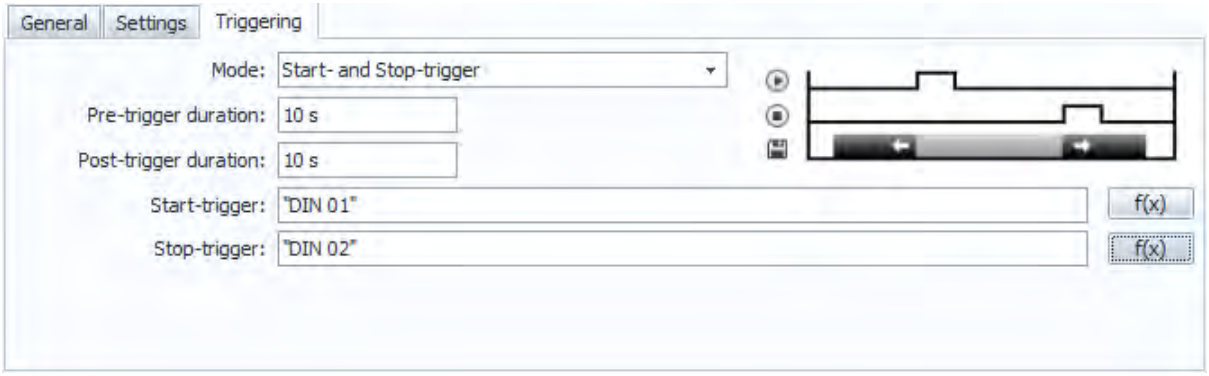
## 6.3.5 Traffic groups

Time stamp related storage of CAN bus and/or LIN bus messages.

Refer to **8.2.3 Traffic acquisition**

For explanations on cyclic and event-controlled data acquisition refer to **7.8 Event controlled measurement**

## 6.3.6 Trigger settings

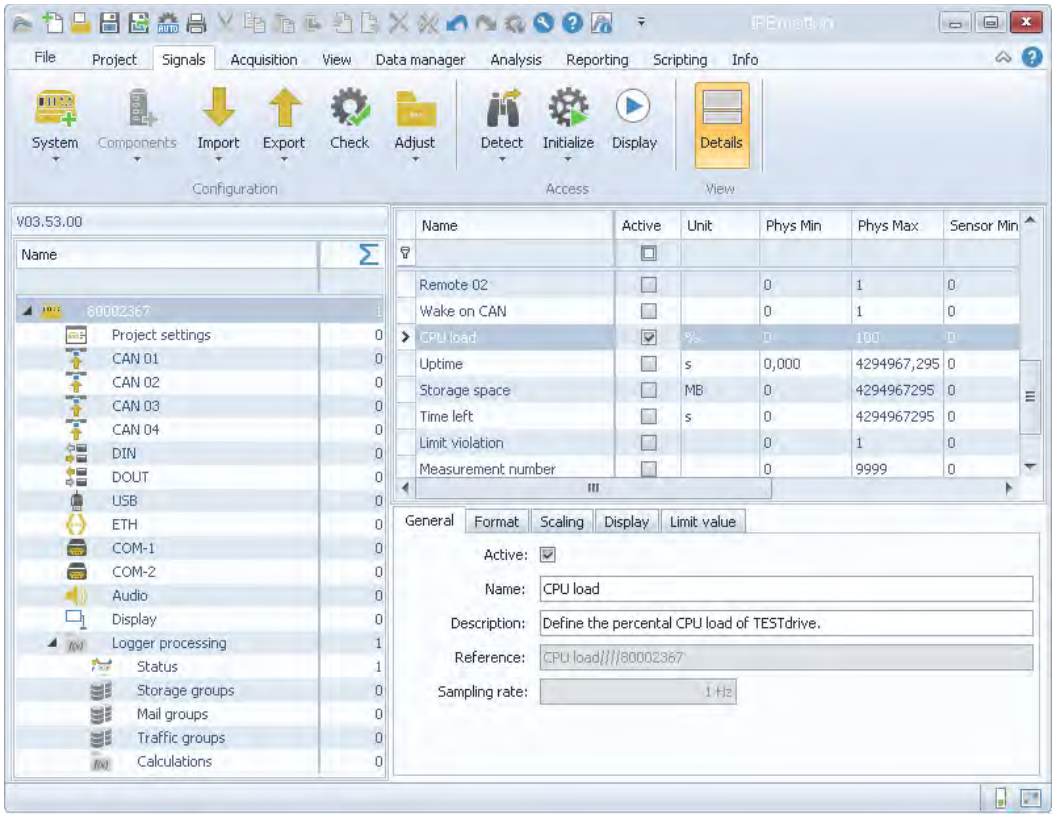



| | |
|---|---|
| **Mode** | Select the trigger mode from Continuous acquisition (no trigger), Start-trigger, Stop-trigger, Start- and Stop-trigger, Stop as inverted start (see Triggering) |
| **Pre-trigger duration** | Data recorded before the trigger event |
| **Post-trigger duration** | Data recorded after the trigger event |
| **Start-trigger** | Value, which starts the data storage of the respective group. |
| **Stop-trigger** | Value, which stops the data storage of the respective group. |
| **Scaling** | Triggering the physical value or the raw value of the numerical value calculated with the formula. |

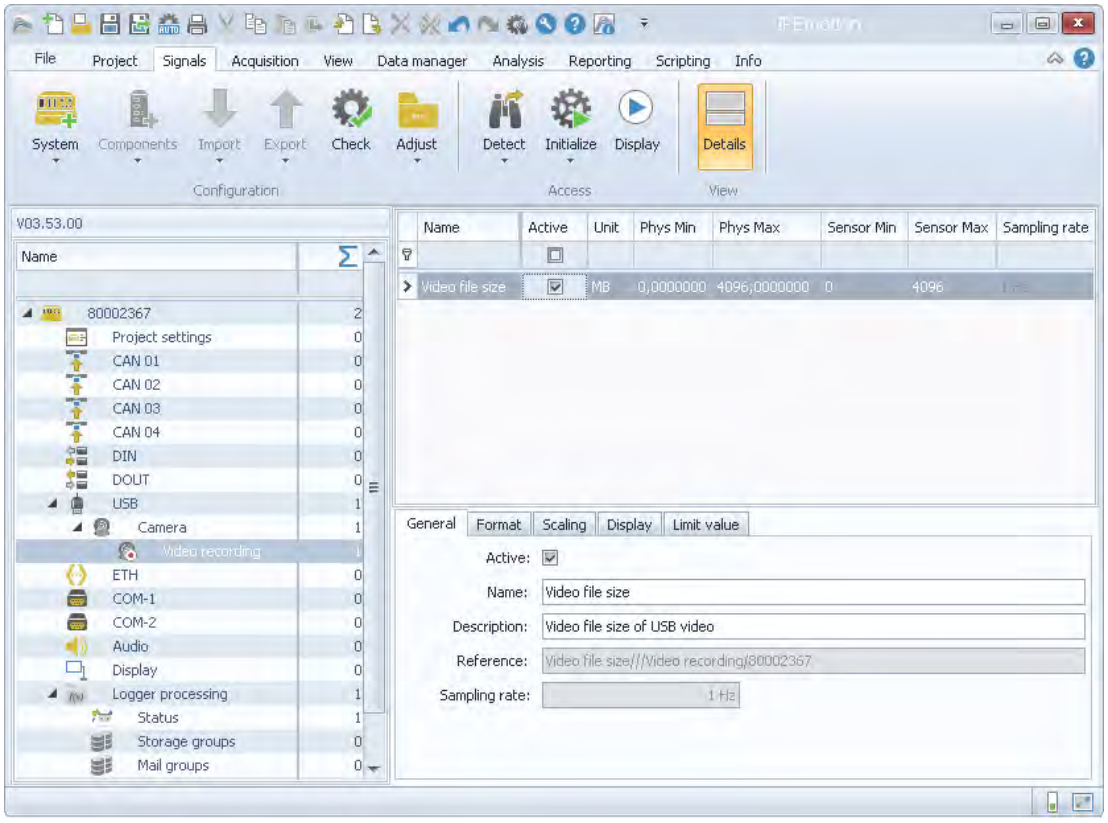
*Define the triggering conditions as formula from the system signals and calculated signals. You can optimize your data recording with a clever configuration of storage groups and triggering conditions to record signals at a high storage rate only if required. This method reduces the data volume, saves memory, and facilitates the final analysis!*

## 6.4 Status channels

### 6.4.1 Logger, Logger processing




### 6.4.2 Video recording



## 6.5 USB medium for external storage

TESTdrive supports USB sticks and USB hard disks for storage on external drives in two different modes:

- External USB medium as alternative storage (solely saving on internal drive OR external drive).
- External USB medium as supplemental storage (depending on data type settings saving on internal drive OR external drive).

### 6.5.1 External medium is the data drive of the logger

The USB medium is the data drive of the logger. Data is stored on this storage type exclusively. TESTdrive creates the standard folder structure on the USB drive. Depending on the current settings, data post processing and shutdown will be executed in different ways.

***Because of data loss, do not disconnect the USB medium during measurement!***

#### Behaviour with FullPostprocessing

Since TESTdrive V03.23.00 this is the default setting when using an USB data drive.

##### Requirement

The non-bootable USB device has to be insert before starting the logger (refer to **6.5.2 External USB medium as supplemental data drive** for preparing the device). There should be no TESTdriveCmd.xml available on the USB medium.

##### Utilization

During boot-up, the logger identifies the USB medium and reports “USB drive is data drive”. Measurement data is stored to the subdirectory named with the logger’s serial number on this storage medium. The FullPostprocessing procedure starts with the RemoteOff condition.

#### Behaviour without FullPostprocessing

This is the former procedure used for USB data storage until TESTdrive V03.22.01.

##### Requirement

The non-bootable USB device has to be insert before starting the logger (refer to **6.5.2 External USB medium as supplemental data drive** for preparing the device). There should be no TESTdriveCmd.xml available on the USB medium. With TESTdrive version V03.23.00 and subsequent releases the following command line has to be added to the dev_ctrl.xml in order to acitvate the usage:

<fastUsbDatatDriveChange>1</fastUsbDatatDriveChange>

##### Utilization

During boot-up, the logger identifies the USB medium and reports “USB drive is data drive”. Measurement data is stored to the subdirectory named with the logger’s serial number on this storage medium. The RemoteOff condition starts the following jobs:

- Checkdisk
- Preparing the directory structure on the USB medium
- Deleting temporary files and directories
- Power down

*Independent from any data transfer settings, measurement and configuration data will not be transferred for data drives without postprocessing.*

## 6.5.2 External USB medium as supplemental data drive

The USB storage medium is used in addition to the internal data drive of the logger. Depending on the current settings, data is stored either to the internal drive or to the external medium. TESTdrive will create the directory structure in any case.

The following data types are supported on the external USB medium:

- as of **TESTdrive V03.52** storage and traffic groups
- as of **TESTdrive V03.58** storage, traffic groups and USB video records

### Requirement

*The non-bootable USB storage medium has to be formatted correctly!*

The Windows console software **diskpart** can be used to prepare a non-bootable USB drive for external storage.

***Please take care that you select the correct drive of your PC when using diskpart!***

***Formatting an incorrect (internal) PC drive may cause serious problems or damage the computer!***

**DiskPart commands and meaning**

| | |
|---|---|
| **DISKPART> list disk** | → Listing available drives |
| **DISKPART> select disk 8** | → Selecting the respectiv drive (here disk 8) |
| **DISKPART> clean** | → Cleaning the selected drive |
| **DISKPART> create partition primary** | → Creating primary partition using the full drive size |
| **DISKPART> format fs=exfat label=ExtStorage quick** | → Fast formatting with the exFAT format and USB stick name ExtStorage (label is an optional parameter) |
| **DISKPART> exit** | → Closing DiskPart |

**Sample screenshot showing list commands**

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

F:\>diskpart

Microsoft DiskPart-Version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
Auf Computer: IT02-0277

DISKPART> list disk

  Datenträger ###  Status         Größe     Frei      Dyn  GPT
  ---------------  -------------  --------  --------  ---  ---
  Datenträger 0    Online          465 GB      0 B        *
  Datenträger 1    Online          698 GB      0 B
  Datenträger 2    Online          167 GB      0 B
  Datenträger 3    Online          465 GB      0 B
  Datenträger 4    Kein Medium        0 B      0 B
  Datenträger 5    Kein Medium        0 B      0 B
  Datenträger 6    Kein Medium        0 B      0 B
  Datenträger 7    Kein Medium        0 B      0 B
  Datenträger 8    Online         7640 MB      0 B
```

```
DISKPART> select disk 8

Datenträger 8 ist jetzt der gewählte Datenträger.

DISKPART> list partition

  Partition ###  Typ              Größe    Offset
  -------------  ---------------  -------  -------
  Partition 1    Primär           7636 MB  4032 KB

DISKPART> clean

Der Datenträger wurde bereinigt.

DISKPART> list partition

Auf diesem Datenträger sind keine Partitionen, die angezeigt werden können,
anden.

DISKPART> create partition primary

Die angegebene Partition wurde erfolgreich erstellt.

DISKPART> list partition

  Partition ###  Typ              Größe    Offset
  -------------  ---------------  -------  -------
* Partition 1    Primär           7639 MB  1024 KB

DISKPART> format fs=exfat label=ExtStorage quick

  100 Prozent bearbeitet

DiskPart hat das Volume erfolgreich formatiert.

DISKPART> exit

Datenträgerpartitionierung wird beendet...

F:\>_
```

*Upon completion of this preparation, the USB medium is ready for use with IPElog and M-LOG V3, but not with M-LOG, as the BIOS of the logger types are different!*

*For usage with M-LOG, the MBR (Master Boot Record) of the USB medium has to be deleted.*

***Once the MBR of the medium has been deleted, it can be no longer used with IPElog or M-LOG V3!***
***We recommend to prepare separate USB storage devices for the respective logger types.***

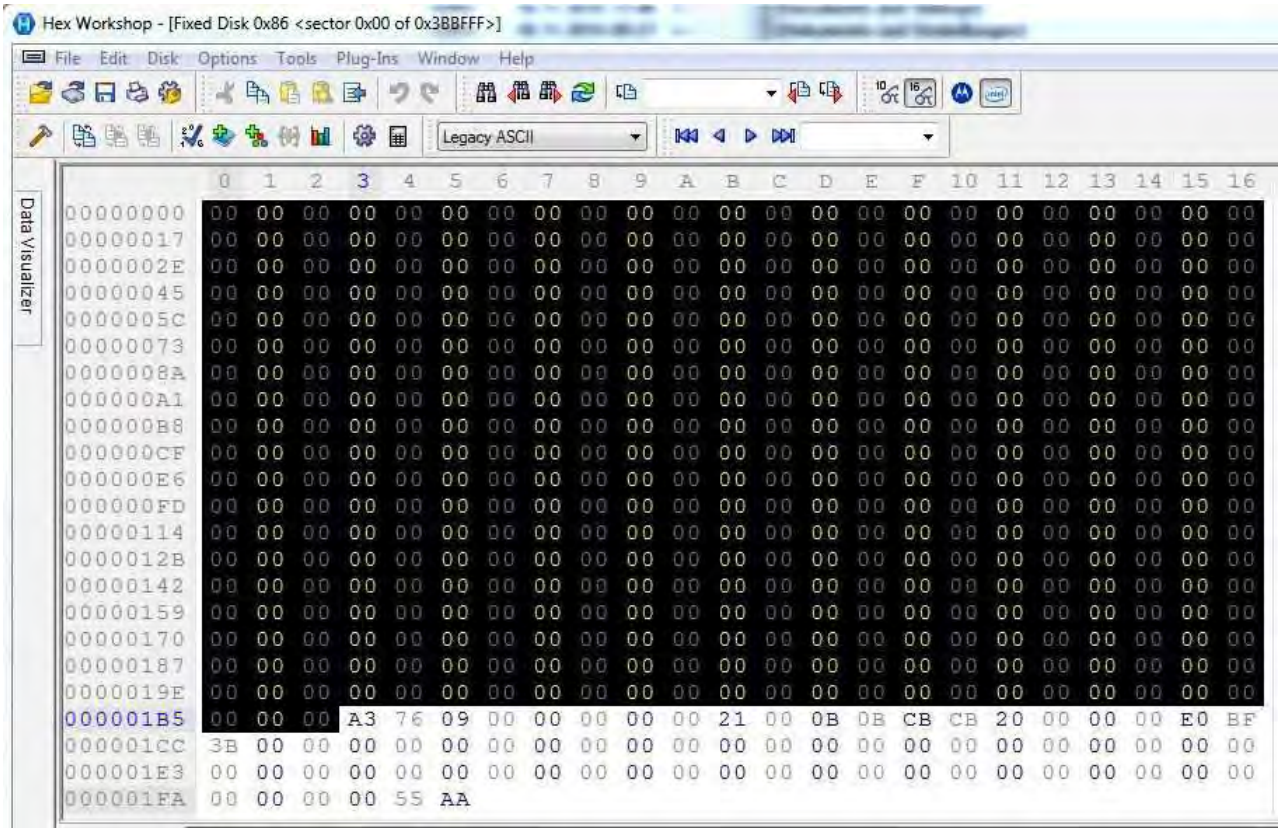
***Deleting the bootloader requires a determination of logical and physical drive!***

The tool **Hex Workshop** can be used for deleting the bootloader, as it supports access to the physical part of the drive.

The tool is available for download from:

http://www.hexworkshop.com/

The MBR (Master Boot Record) is the physical drive area. The bytes asigned to address range 0x000 to 0x01B7 (Bootloader) have to be deleted, this means set to 0.

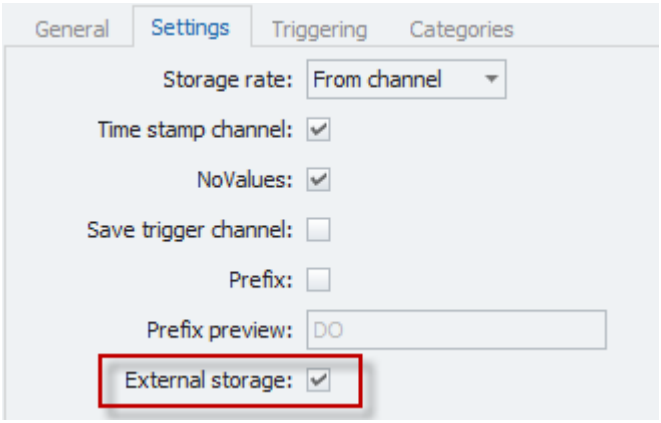



The TESTdriveCmd.xml with the contents listed below has to be copied to the USB medium:

```
<TESTdriveCmd>
	<Authentication>
		<user>guest</user>
		<password>ipe_guest</password>
	</Authentication>
	<OnConnect>
		<JobList>
			<Job>
				<Name>startExternalStorage</Name>
			</Job>
		</JobList>
	</OnConnect>
	<OnDisconnect>
		<JobList>
		</JobList>
	</OnDisconnect>
</TESTdriveCmd>
```

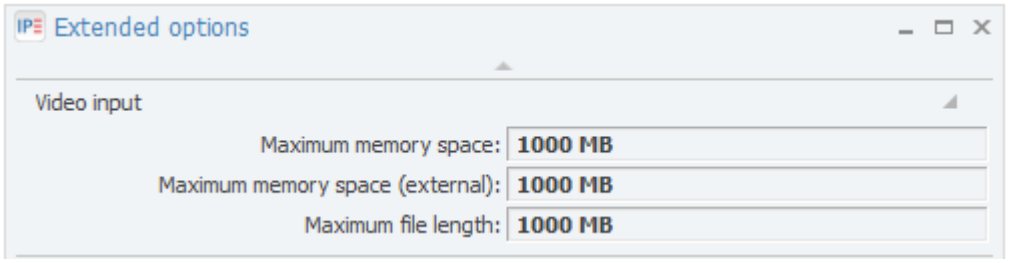
*Using a TESTdriveCmd.xml with other commands will execute these jobs and waiting for disconnecting the USB medium.*
*The storage to the external drive will not be started in this case.*

In addition to an appropriate TESTdriveCmd.xml, the correct settings for storage and/or traffic groups or USB video recording are required → Activate **External storage**.




With USB video recordings, the total memory space available on the USB device can be limited by PlugIn specific settings.

(Options → PlugIns → IPETRONIK LOG PlugIn-specific settings → Extended → Open system setup).




## Application

Storage on external USB device is supported for triggered measurements as well as continuous measurements (TESTdrive V03.54.00 or a newer release required).

The USB medium can be connected to the logger before boot up or when the measurement has been already started (TESTdrive ≥ V03.54.00). TESTdrive writes data directly to the external USB drive.

If the start trigger for data storage is active but no USB device available, the logger will report the respective warning:

"No USB-Stick with TESTdriveCmd. Storage group *[Name]* cannot be stored externally."

"No USB-Stick with TESTdriveCmd. Traffic group *[Name]* cannot be stored externally."

"Video(*[Nummer]*): No USB-Stick with TESTdriveCmd. Video files cannot be stored externally."

The external storage has to be stopped before unplugging the USB medium. We recommend to end the complete measurement (not only USB storage) as the operating system could respond unforeseeable on unplugging the USB medium. If this procedure is not possible, a stop trigger has to be configured to end the data storage regularly.

*Unplugging the USB medium while external storage groups are active causes data loss!*

*Configure stop trigger conditions or end the measurement (power down) to unplug the USB medium from the logger without data loss.*

# 7 Standard functions

## 7.1 Calculations

TESTdrive supports the online calculation of the signals acquired in the system. The desired calculation is defined with the corresponding software dialog – manually or with the formula editor.

IPEmotion Version 1.03 and PlugIn IPETRONIK-LOG ≥ V03.20 offer a shared formula editor, i.e. the following calculation functions are available in IPEmotion, as well as, in TESTdrive.

*Calculations, which are based on each other, must be run in the same cycle rate to get a correct result! If, e.g., the result of a calculation with low cycle rate is used in a calculation with higher cycle rate, a time offset results, which can influence the result according to a signal change (amplitude). In that case, the signal sampling rate of the first calculation must be raised.*

*The formula editor requires the use of the semicolon ";" instead of the comma "," as variable separator.*

*If a value within a calculation is "NoValue", the calculation result is "Novalue", too.*

### 7.1.1 Mathematic functions and operations

| 1 Basic operation | | | | |
|---|---|---|---|---|
| **Operator** | **Name** | **Syntax** | **Example** | **Result** |
| **+** | Addition | "Temp01" + "Temp02" | 15 + 10 | 25 |
| **–** | Subtraction | "Temp01" - "Temp02" | 15 - 10 | 5 |
| ***** | Multiplication | "Temp01" * "Temp02" | 15 * 10 | 150 |
| **/** | Division | "Temp01" / "Temp02" | 15 / 10 | 1.5 |
| **MOD** | Modulo, Division rest | "Temp01" MOD "Temp02" | 15 MOD 10 | 5 |
| **ABS( )** | Absolute value of a number | ABS("Temp01") | ABS(-15) | 15 |
| **SIGN( )** | Sign of a number | SIGN("Temp01") | SIGN(15)<br>SIGN(0)<br>SIGN(-15) | 1<br>0<br>-1 |
| **NEG( )** | Negation of a number | NEG("Temp01") | NEG(15)<br>NEG(-15) | -15<br>15 |
| **(** | Begin bracket term | ("Temp01" + "Temp02") * 2 | (15 + 10) * 2 | 50 |
| **)** | End bracket term | ("Temp01" - "Temp02") * 2 | (15 - 10) * 2 | 10 |

| 2 Power, Square root, Exponential and Logarithm functions | | | | |
|---|---|---|---|---|
| **Function** | **Name** | **Syntax** | **Example** | **Result** |
| **^** | Power | "Temp01" ^ 2 | 15 ^ 2 | 225 |
| **SQRT( )** | Square root | SQRT("Temp01") | SQRT(25) | 5 |
| **EXP( )** | Exponential function of basis e | EXP("Temp01") | EXP(5) | 148.41 |
| **LOG( )** | Logarithm of basis 10 | LOG("Temp01") | LOG(5) | 0,4771 |
| **LN( )** | Logarithm of basis e | LN("Temp01") | LN(5) | 1.0986 |

| 3 Trigonometric functions, Hyperbola functions | | |
|---|---|---|
| **Function** | **Name** | **Range of values in radiant** |
| **SIN( )** | Sine | +/-3.99 rad |
| **COS( )** | Cosine | +/-3.99 rad |
| **TAN( )** | Tangent | +/-3.99 rad |
| **ASIN( )** | Arc sine | +/-1.0 rad |
| **ACOS( )** | Arc cosine | +/-1.0 rad |
| **ATAN( )** | Arc tangent | +/-1.0 rad |
| **SINH( )** | Sine Hyperbolicus | +/-1.99 rad |
| **COSH( )** | Cosine Hyperbolicus | +/-1.99 rad |
| **TANH( )** | Tangent Hyperbolicus | +/-1.99 rad |

| 4 Comparative operations (comparison of variable values) | | | | |
|---|---|---|---|---|
| **Function** | **Name** | **Syntax** | **Example** | **Result** |
| **=** | Equal | “Temp01“ = “Temp02“ | 15 = 10<br>15 = 15 | 0<br>1 |
| **<>** | Unequal | “Temp01“ <> “Temp02“ | 15 <> 10<br>15 <> 15 | 1<br>0 |
| **<** | Less than | “Temp01“ < “Temp02“ | 10 < 15<br>15 < 15 | 1<br>0 |
| **>** | Greater than | “Temp01“ > “Temp02“ | 15 > 10<br>15 > 16 | 1<br>0 |
| **<=** | Less than or equal | “Temp01“ <= “Temp02“ | 10 <= 15<br>15 <= 15<br>20 <= 15 | 1<br>1<br>0 |
| **>=** | Greater than or equal | “Temp01“ >= “Temp02“ | 15 >= 10<br>15 >= 15<br>15 >= 20 | 1<br>1<br>0 |
| **IF( ; ; )** | If function | IF(“Temp01“ >= “Temp02“; x; y)<br><br>Query to a specific status.<br>If Occurrence > action 1, otherwise action 2<br>Example: IF(“Thermo_channel3“ > 30; 1; 0)<br>If Temperature > 30, Result: 1 otherwise 0<br>Action 1 and 2 can also be calculations. | x = 1; y = 0<br>“Temp01“ = 15<br>“Temp02“ = 10<br>“Temp01“ = 10<br>“Temp02“ = 15 | <br>1<br><br><br>0 |

| 5 Logic operations (comparison of signal states) | | | | |
|---|---|---|---|---|
| **Function** | **Name** | **Syntax** | **Example** | **Result** |
| **AND** | And | “Temp01“ > “Temp02“ AND “Temp01“ > 10 | 15 > 5<br>15 > 10<br>10 > 5 | 1<br>1<br>0 |
| **OR** | Or | “Temp01“ > “Temp02“ OR “Temp01“ > 10 | 15 > 5<br>10 > 5<br>10 > 10 | 1<br>1<br>0 |
| **XOR** | Exclusiv or | “Temp01“ > “Temp02“ XOR “Temp01“ > 10 | 15 > 5<br>10 > 5<br>15 > 15<br>10 > 10 | 0<br>1<br>1<br>0 |
| **NOT( )** | Not | NOT(“Temp01“ > “Temp02“)<br>(inverse state) | 15 > 5<br>15 > 15<br>5 > 15 | 0<br>1<br>1 |

| 6 Boolean operations (bitwise comparison of signal states) | | | | |
|---|---|---|---|---|
| **Function** | **Name** | **Description** | **Example** | **Result** |
| **ANDB** | And bitwise | Bits which are set in operand1 **and** in operand2 will be set in the result (bit = 1), all others will be not set (bit = 0) | 27 ANB 12<br>11011 ANDB 01100 | 8<br>01000 |
| **ORB** | Or bitwise | Bits which are set in operand1 **or** in operand2 will be set in the result (bit = 1), all others will be not set (bit = 0) | 26 ORB 8<br>11010 ORB 01000<br>27 ORB 13<br>11011 ORB 01101 | 26<br>11010<br>31<br>11111 |
| **XORB** | Exclusive or bitwise | Bits which are set only in operand1 **or** only in operand2 will be set in the result (bit = 1), all others will be not set (bit = 0) | 26 XORB 8<br>11010 XORB 01000<br>27 XORB 13<br>11011 XORB 01101 | 18<br>10010<br>22<br>10110 |
| **NOTB** | Not bitwise | Bits which are set in operand1 will be not set in the result (bit = 0), all others will be set (bit = 1) | NOTB 27<br>NOTB 11011 | 4<br>00100 |

| 7 Statistic functions | | | | |
|---|---|---|---|---|
| **Function** | **Name** | **Syntax , Description** | **Example** | **Result** |
| **MIN( )** | Minimum | MIN(“Temp01“) | 4 \| 12 \| 3 \| 25 \| 17 | 3 |
| **MAX( )** | Maximum | MAX(“Temp01“) | 4 \| 12 \| 3 \| 25 \| 17 | 25 |
| **MEAN( )** | Average | Average from all valid values | | |
| **MEAN( ; )** | Average from n values | Average from n valid values | | |
| **MINOR( ; )** | Less value | MINOR(“Temp01“; “Temp02) | 4 \| 12 | 4 |
| **MAJOR( ; )** | Greater value | MAJOR(“Temp01“; “Temp02) | 4 \| 12 | 12 |
| **FLOOR( )** | Round integer off | FLOOR(“Temp01“) | 13,72 | 13 |
| **CEIL( )** | Round integer up | CEIL(“Temp01“) | 13,41 | 14 |
| **ROUND( )** | Round integer | ROUND(“Temp01“) | 13,41<br>13,72 | 13<br>14 |
| **LIN( ; ; ; )** | Linearization | LIN(“Temp01“;<br>x node-1;y node-1;<br>x node-n; y node-n)<br>Runs a linearization with the defined nodes. Between n = 2 and n = 16 nodes can be defined. | | |

| 8 Other functions | |
|---|---|
| **Function** | **Description** |
| **EDGE_POS( )** | Detect positive edge<br>1 if current value is > 0 and the previous one <= 0 |
| **EDGE_NEG( )** | Detect negative edge<br>1 if current value is <= 0 and the previous one > 0 |

| Function | Description | Example | Result |
|---|---|---|---|
| **DIFF( )** | Runs a differentiation of an operand acc.to: (Opr1(t) – Op1(t-1)) * DeltaT | | |
| **INT( )** | Calculates the integral of an operand acc. to: "((Op1(t) + Op1(t-1))/2) * DeltaT" | | |
| **INT_ADD( )** | Calculates the integral of an operand acc. to: "((Op1(t) + Op1(t-1))/2) * DeltaT" and adds the result to the previous value | | |
| **INT_UP( )** | Calculates the upper integral of an operand acc. to: "Op1(t) * DeltaT" | | |
| **PREV( )** | Outputs the previous value. | PREV(“Temp01“) 4 \| 12 \| 3 \| 25 \| | NV \| 12 \| 3 |
| **(„Temp1“ + (PREV („Temp1“)) + (PREV (PREV(„Temp1“))) ) / 3** | Floating average from the current value and both previous values | | |
| **SHL( ; )** | Shift value bitwise to the left. | 1 SHL 2<br>001 SHL 2 | 4<br>100 |
| **SHR( ; )** | Shift value bitwise to the right. | 12 SHR 1<br>1100 SHR 2 | 6<br>0110 |
| **TESTBIT( ; )** | Checks the value (0, 1) of the defined bit. If the bit described by operand 2 is set, result = 1, otherwise result = 0<br>(Counting order starting right hand with 0) | TESTBIT(1101; 3)<br>TESTBIT(1101; 1) | 1<br>0 |
| **TESTMASKS( ; )** | Makes a comparison with a user definable bitmask. If at least one bit is set in operand 1 and also in operand 2, result = 1, otherwise result = 0 | TESTMASKS(27, 6)<br>TESTMASKS(27, 4) | 1<br>0 |
| **TIME( )** | TIME(“Temp01“)<br>A counter that adds the time intervalls of the corresponding sample rate continuously and outputs the sum as long as the operator's value is >= 0.5<br>In case the operator's value is < 0.5, the counter is reseted and 0 is output. | | |
| **TIMER( ; )** | If no new value is received from the channel defined as Parameter1 within the timeout (Parameter2 in seconds), the output value is '1'. | | |
| **VALID( )** | Check for validity<br>1 if value is unequal to NoValue, 0 if value is NoValue | | |
| **VALID(x; y)** | Avoid Novalues<br>x if value is unequal to NoValue, y if value is NoValue | | |
| **ISNOVALUE(x; y)** | Delay of a NoValue identification (x: signal value, y: optional time delay)<br>If y is not set resp. its value is 0, the result is true, as soon as x = NoValue<br>Is y ≠ 0 the result is true, as soon as x = NoValue for a minimum duration of y seconds.<br>Is x ≠ NoValue, the result is always false. | | |
| **RADIUS(y; z)** | Radius calculation polar coordinates (r = √(y^2+ z^2)<br>= SQRT( "phi_y" * "phi_y" + "phi_z" * "phi_z" ) | | |
| **ANGLE(y; z)** | Angle calculation in polar cordinates<br>= IF( "phi_z" >= 0; ACOS("phi_y" / "Radius" ) * 180/PI; -1 * ACOS("phi_y" / "Radius" ) * 180/PI + 360) | | |

### 7.1.2 Constants

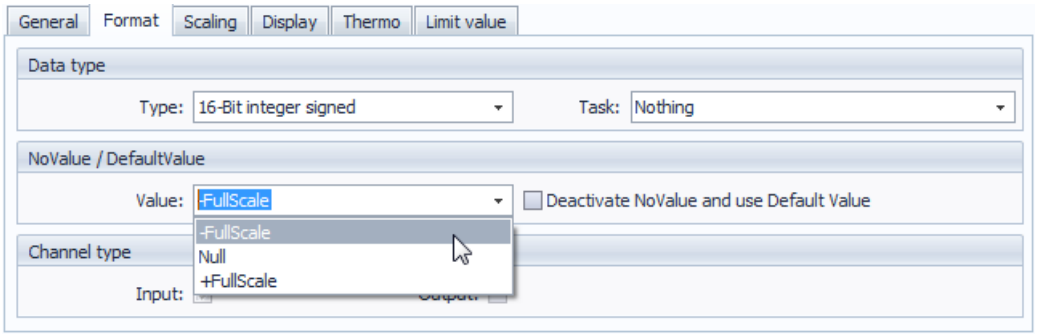
| 9 Constants | |
|---|---|
| **Constant** | **Description** |
| **PI** | Pi > π = 3.141592654 ... |
| **SYSTEMRATE** | TESTdrive internally works with a fix system rate. This rate depends on the configuration (Channel with highest sampling rate) and can be used for calculations. The system rate can be compared with the timer ticks of a PC clock and is set in Hz.<br>The system rate relates to the working frequency of the PC/Notebook (= Frequency of the High-Performance-Counter). |
| **SYSTEMTIME** | Reciprocal value of the system rate (=1/SYSTEMRATE) and is set in seconds.<br>Please note at using a system rate in the MHz range that the system time can only be correctly displayed if sufficient decimals have been defined and Automatic has been selected in Display > Formatting. |
| **SAMPLERATE** | Channel sample rate in Hz |
| **SAMPLETIME** | Channel acquisition intervall in 1/s |

### 7.1.3 NoValue and timeout settings

In order to differ invalid and valid measurement values, the user can select –FullScale, Null or +Fullscale (for analog inputs) to be identified as invalid value (NoValue). A signal value at the upper or lower limit range could be invalid, as it can not occur really. Another used case are missing messages at the CAN bus. An expired fimeout can be used to indicate invalid values.

#### Settings possibilities

##### Measurement input




The setting for the signal input (e.g. IPETRONIK CAN measurement module) or the CAN signal input defines the invalid value (NoValue) within the measurement range (selection -FullScale, Null, +FullScale).

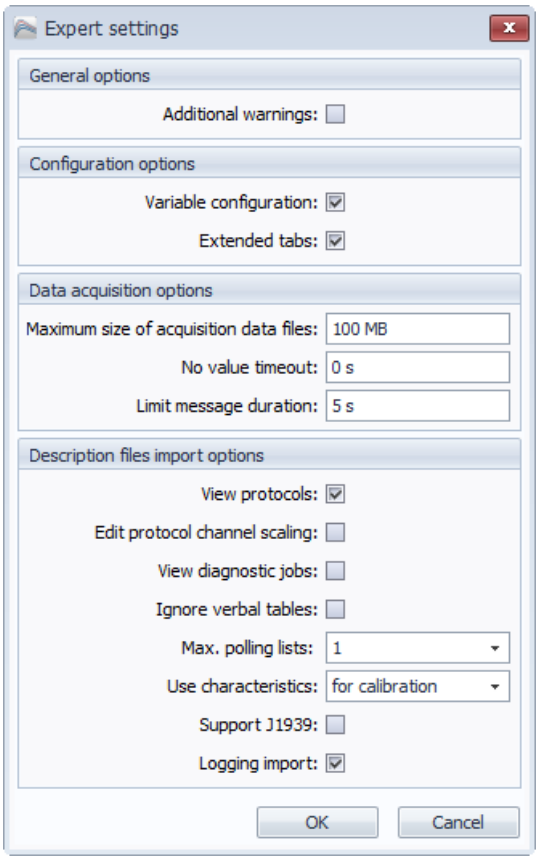
If this function is deactive, the measurement value is used (e.g. -60 °C with M-THERMO), NoValues are suppressed in this case.

A NoValue alarm can be activated for every signal within the **Format** tab.

## IPEmotion Options

The **IPEmotion Options Expert settings** (select **Basic settings** > **Expert mode**) provide a **No value timeout** for the online data acqusition. If values are missing with a cyclic data acquisition, subsequent samples will be output as NoValue if the timeout range has expired. The setting range for the No value timeout is 0 ... 5 s.

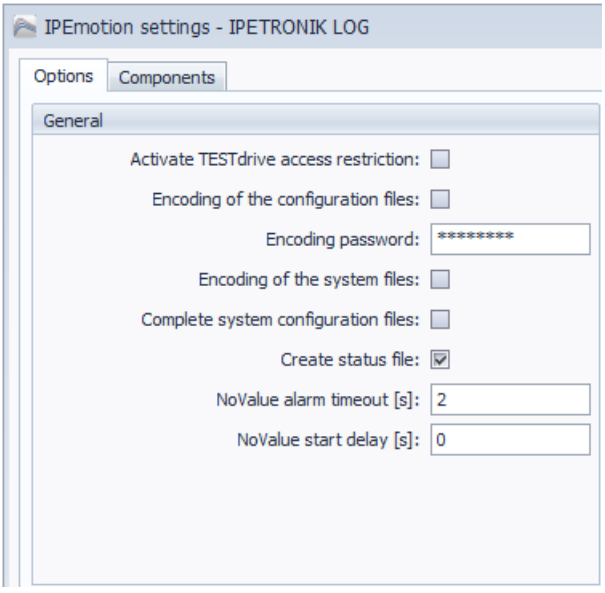
This setting has no effect on the data acquisition by the logger. Please use the PlugIn Options for this.




## PlugIn Options

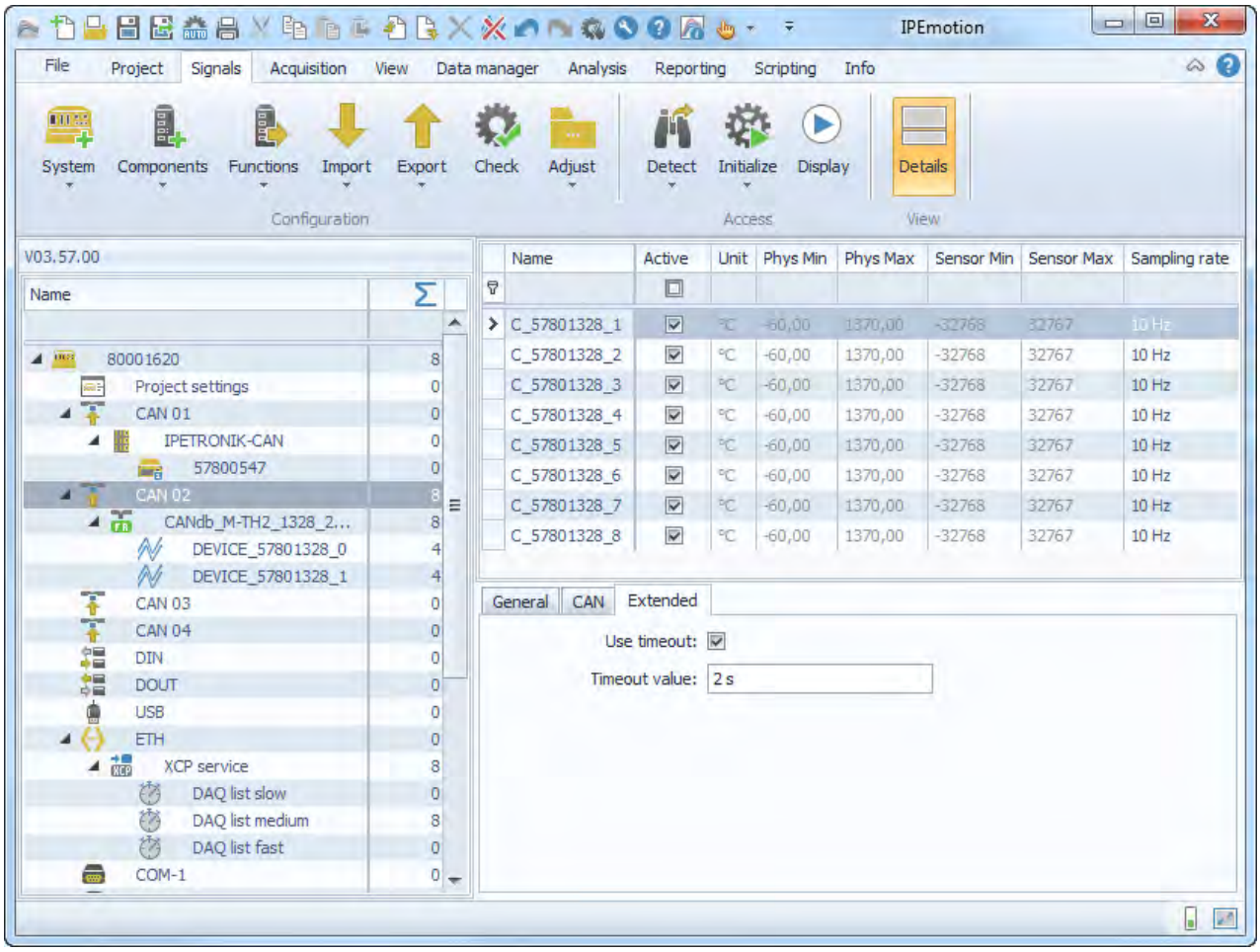
Entering the **IPEmotion Options**, the options tab of the PlugIn IPETRONIK LOG (select **current PlugIn** > **Options** > **NoValue-Alarm Timeout (s)**) provides the logger acquisition timeout. . If values are missing with a cyclic data acquisition, subsequent samples will be output as NoValue if the timeout range has expired.
The setting range for the No value timeout is 0 ... 120 s.

Conditioned by different delays during initialization (modules, logger) when measurement has been started, NoValues can occur during this phase.
Use the setting **NoValue start delay (s)** in order to avoid NoValues at measurement start.
The setting range for the No value start delay is 0 ... 600 s.



## CAN input

Each CAN input of the logger provides timeout settings with an adjustable delay time (select for example **CAN 02** > tab **Extended** > **Timeout value 2 s**). If the CAN input do not receive messages anymore and the timeout period has expired, the logger starts writing NoValues to the data record.




## Log file entries

If the NoValue monitoring is activated and NoValues are occuring, the system will record the series of NoValues itself (instead of the signal value) and the corresponding log file entry.

## Example

NoValue monitoring at CAN 01  Timeout value 20 s

Log file entry: *D CAN signal measurement timeout (no valid ID) on CAN 01 (t > 20000 ms)*

## 7.1.4 Calculation examples

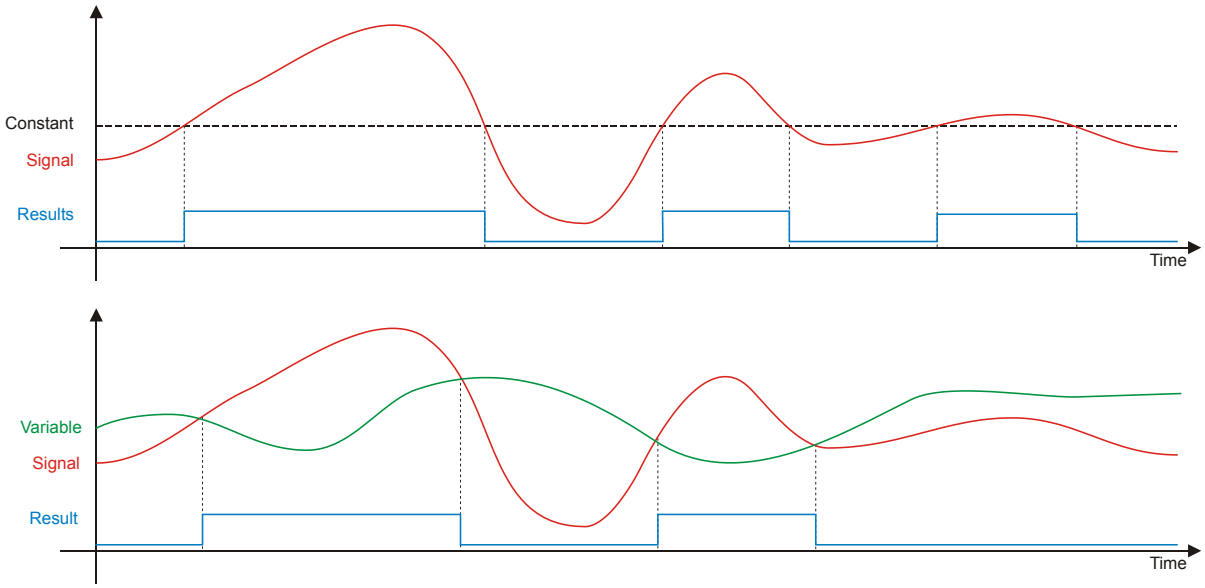
### Greater comparison „>“

**Syntax** **>**

**Description** Compares the current value (of a channel or a calculation) with a user defined constant or variable. The result is 1, as long as the comparison is true, otherwise the result is 0 (for a false comparison).

**Example** „Channel01“ > 2 *Comparison with a constant*

„Channel01“ > „Channel02“ *Comparison with a variable*



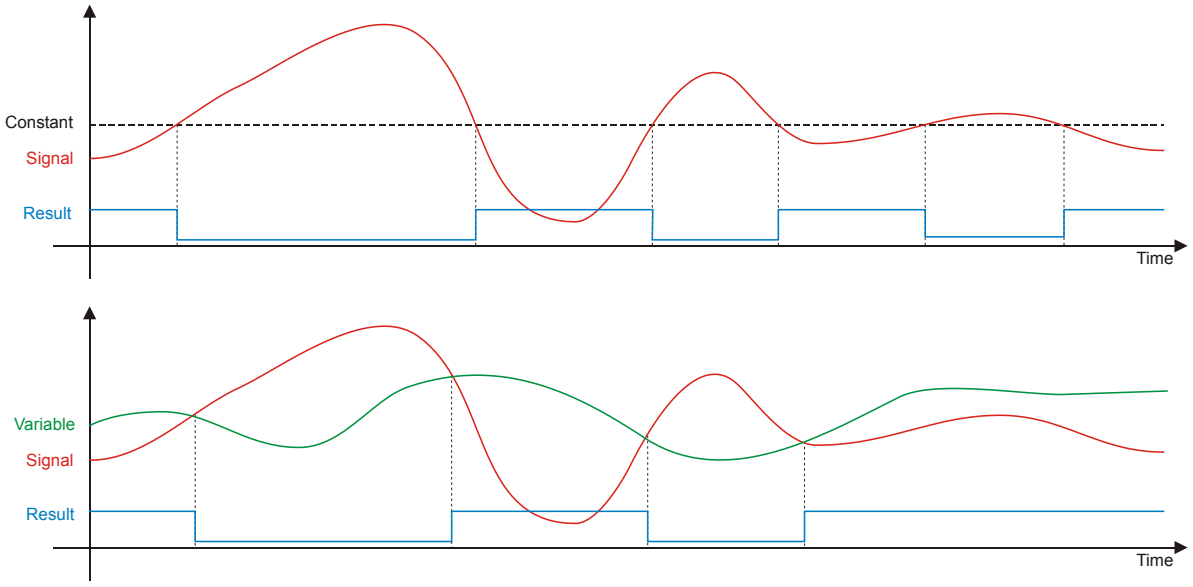

### Less comparison „<“

**Syntax** **<**

**Description** Compares the current value (of a channel or a calculation) with a user defined constant or variable. The result is 1, as long as the comparison is true, otherwise the result is 0 (for a false comparison).

**Example** „Channel01“ < 2 *Comparison with a constant*

„Channel01“ < „Channel02“ *Comparison with a variable*



## Counter (without reset)

**Syntax** **„Counter01“ + x**
**VALID(“Counter01”; 0) + x**

**Description** Counts continuously with the current sample rate, i.e. the counter value is increased by x with each sample. The rise of the counter slope depends on the currently configured sampling rate and the counter step. A measurement stop resets the counter to 0.

**Example**
„Counter01“ = „Counter01“ + 1 *Accumulates 1 with each sample*
„Counter01“ = „Counter01“ + 10 *Accumulates 10 with each sample*

**Tip** Used with TESTdrive (logger application) this recursive formula will result in valid values as the initial value for variables is set automatically. Used with IPEmotion (Acquisition > Calculations > Formulas) the respective formula has to be completed with the Valid function:
„Counter01“ = VALID(„Counter01“; 0) + 1
The Valid function sets the initial counter value to a defined value (here 0) in case the current value is invalid.

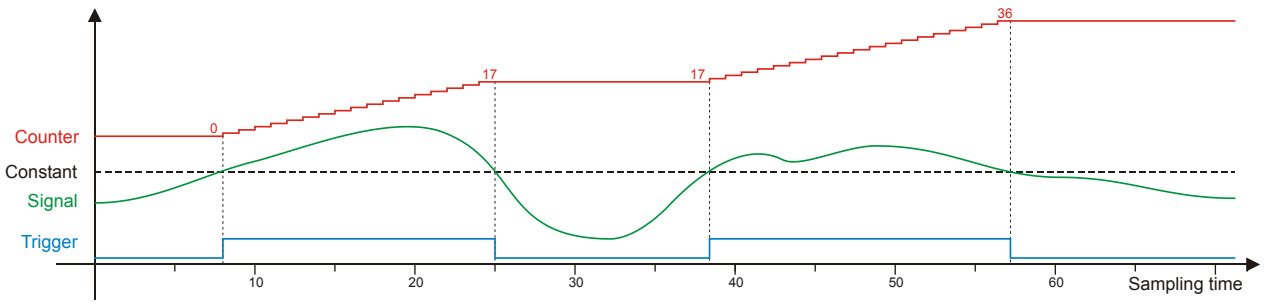
## Counter with counting condition

**Syntax** **IF(“Channel01” > x; „Counter01“ + y; „Counter01“)**
**IF(“Channel01” > x; VALID(“Counter01”; 0) + y; VALID(“Counter01”; 0))**

**Description** Counts continuously with the current sample rate, as long as, the current value of "Channel01" is greater than x. The counter value is increased by y with each sample, as long as "Channel01" fulfills the condition. Otherwise the counter remains unchanged. As son as the condition is fulfilled again, the counter continues with the previous value. The rise of the counter slope depends on the currently configured sampling rate and the counter step.
A measurement stop resets the counter to 0.

**Example**
„Counter01“ = IF(“Channel01” > 5; "Counter01“ + 1; „Counter01“)
*Accumulates 1 with each sample as long as "Channel01“ is greater than 5.*
*Stop counting and holding the latest value, as soon as „Channel01“ is less than 5.*

**Tip** Used with TESTdrive (logger application) this recursive formula will result in valid values as the initial value for variables is set automatically. Used with IPEmotion (Acquisition > Calculations > Formulas) the respective formula has to be completed with the Valid function.



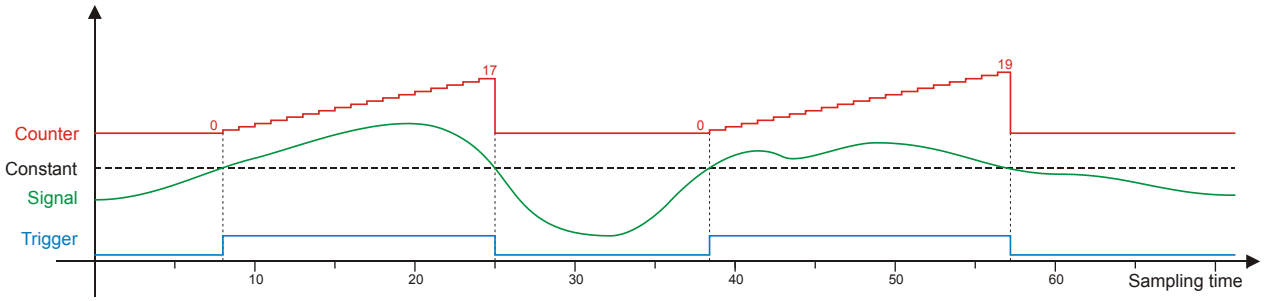
## Counter with counting condition and reset

**Syntax** **IF(“Channel01” > x; „Counter01“ + y; 0)**
**IF(“Channel01” > x; VALID(“Counter01”; 0) + y; 0)**

**Description** Counts continuously with the current sample rate, as long as, the current value of "Channel01" is greater than x. The counter value is increased by y with each sample, as long as "Channel01" fulfills the condition. Otherwise the counter will be reset to 0. As son as the condition is fulfilled again, the counter restarts with 0. The rise of the counter slope depends on the currently configured sampling rate and the counter step.
A measurement stop resets the counter to 0.

**Example** „Counter01“ = IF(“Channel01” > 5; „Counter01“ + 1; 0)
*Accumulates 1 with each sample as long as "Channel01“ is greater than 5.*
*Stop counting and reset to 0, as soon as „Channel01“ is less than 5.*

**Tip** Used with TESTdrive (logger application) this recursive formula will result in valid values as the initial value for variables is set automatically. Used with IPEmotion (Acquisition > Calculations > Formulas) the respective formula has to be completed with the Valid function.




## Validation check „VALID“

**Syntax** **VALID(x)**
**VALID(x;y)**

**Description** VALID checks the current value (of a channel or a calculation) for validity, i.e. if the variable's status is „invalid“ (NoValue).
With the VALID(x) function the result is1, as long as x is valid and changes to 0, as soon as the x has the status NoValue.
With the VALID(x;y) function the result is x as long as x is a valid value. As soon as x will be an invalid value (NoValue) the result changes to y. In case y is a variable (instead of a constant) the result of VALID(x,y) will be NoValue, as soon as x, as well as, y are an invalid value.

**Example** VALID(„Channel01“) *1 if „Channel01“ ≠ NoValue, 0 if „Channel01“ = NoValue*
VALID(„Channel01“;4) *„Channel01“ if „Channel01“ ≠ NoValue, 4 if „Channel01“ = NoValue*

**Tip** When using a measurement value within a recursive formula (x = x + y) we recommend to use the VALID(x; y) function in order to avoid invalid values (NoValue). Even if the measurement signal becomes valid values, the recursive formula will not be calculated correctly, once a NoValue occurred.
We recommend to use the VALID(x; y) function for any signal which serves as a trigger for a storage group. An invalid trigger signal could prevent the start of the data storage even if the trigger signal will become temporarily valid.

## Mean value (averaging) „MEAN“

**Syntax** **MEAN(x)**
**MEAN(x; n)**

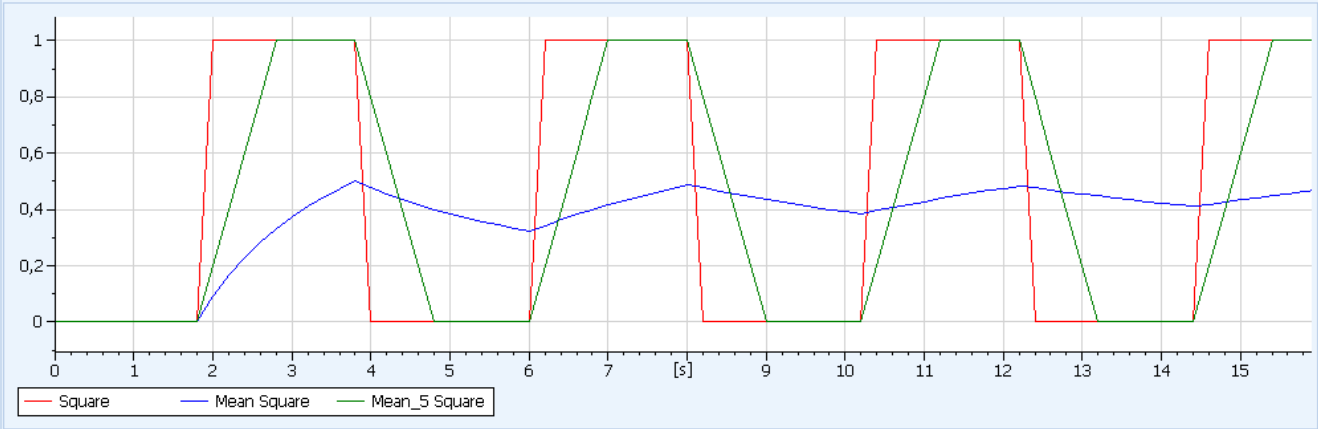
**Description** MEAN(x) continuously calculates the moving average from all valid measurement values.
MEAN(x; n) berechnet continuously calculates the moving average from n previous measurement values.

**Example**

| | |
|---|---|
| MEAN(„Channel01“) | moving average from all values of a measurement |
| MEAN(„Channel01“; 10) | moving average from 10 values at a time |

**Tip** The diagram shows the differences of the MEAN functions with a rectangle signal (clock 5 Hz)

| | | |
|---|---|---|
| Square | red | origin signal |
| Mean Square | blue | moving average |
| Mean_5 Square | green | moving average from 10 values at a time |



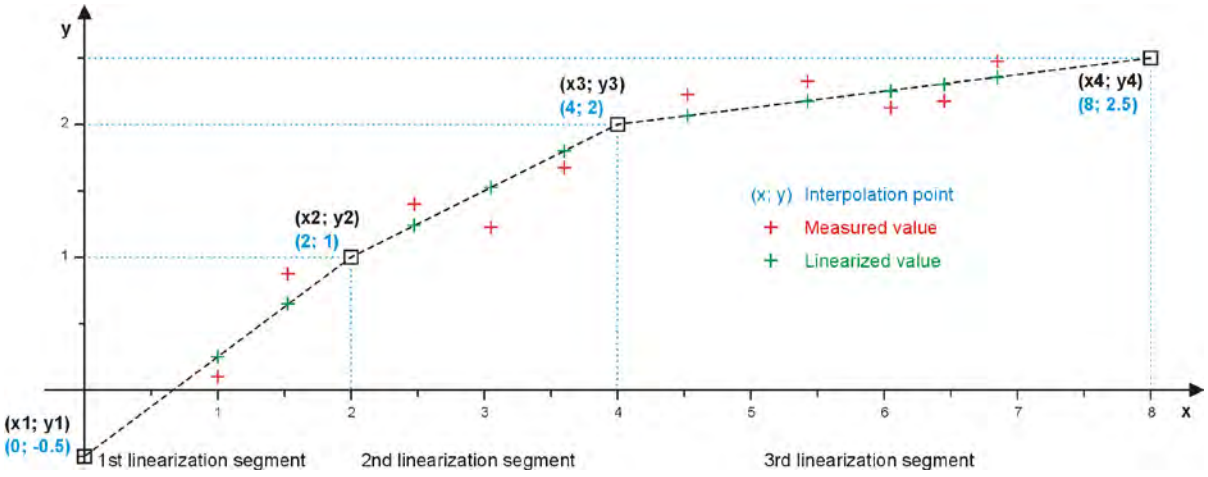

## Linearization „LIN“

**Syntax** **LIN(Operand; x-Node01; y-Node01; x-Node02; y-Node02)**

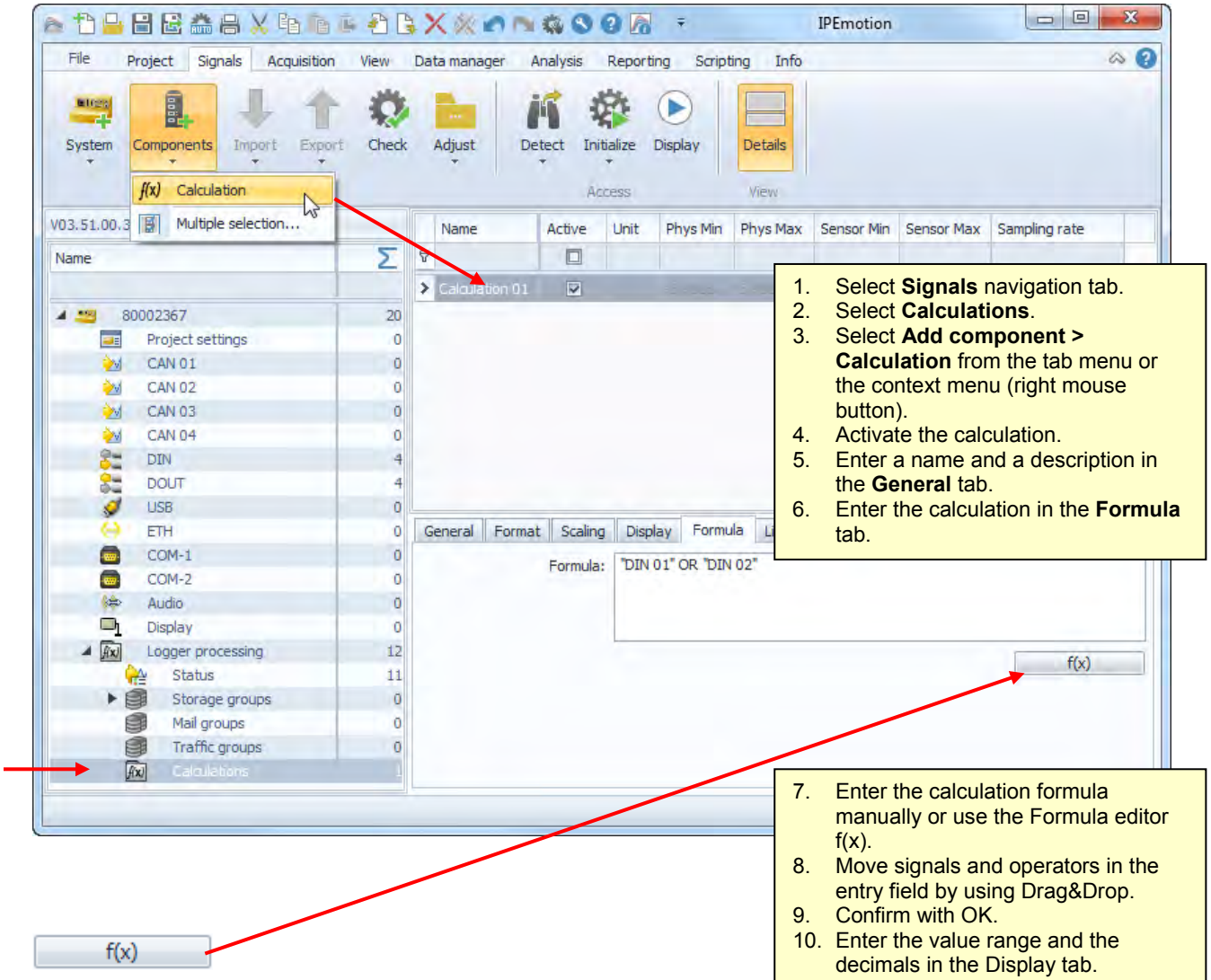
**Description** Runs a linearization with user defined nodes. Between n = 2 and n = 16 nodes can be defined. The scaled values (Y-axis) are calculated from the original measurement values (X-axis) using factor and offset of the partial linear slope between two nodes.

**Example** LIN(„Channel01“; 0;-0,5; 2;1; 4;2; 8;2,5)

**Tip** The linearization enables a non-linear scaling (multipart scaling) used for physical graphs without having a mathematical equation.
Using this multipart scaling an approximation of a mathematical equation is more precise as more nodes are used.



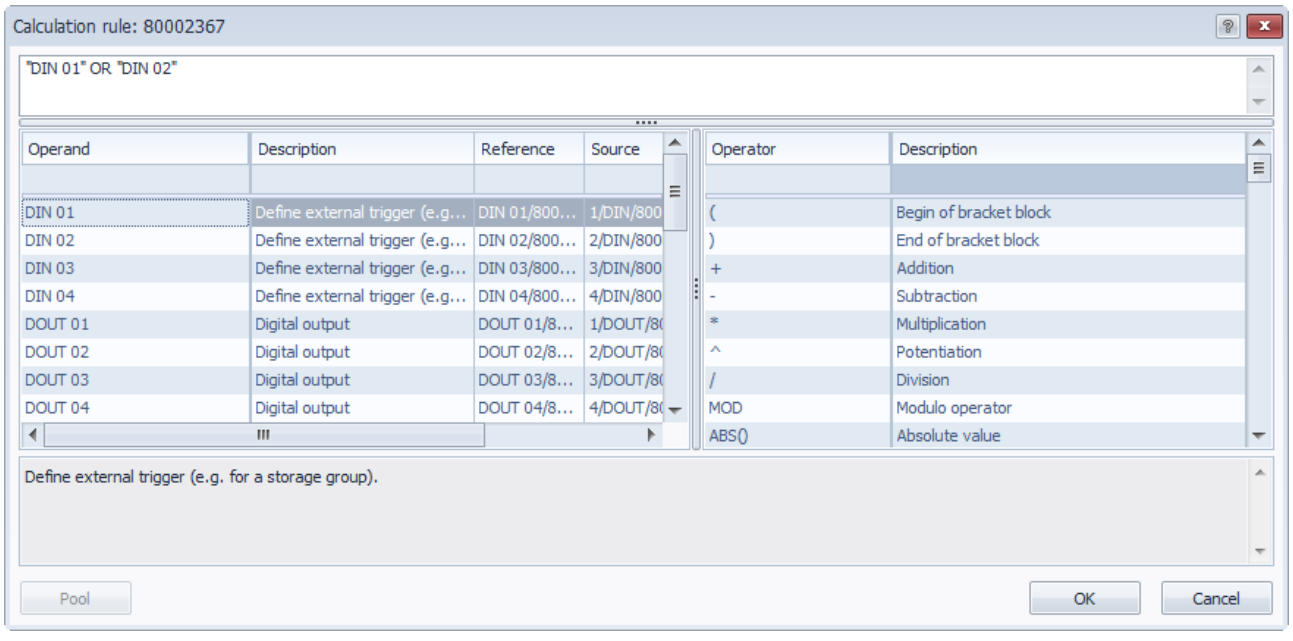
## 7.1.5 Calculation




1. Select **Signals** navigation tab.
2. Select **Calculations**.
3. Select **Add component > Calculation** from the tab menu or the context menu (right mouse button).
4. Activate the calculation.
5. Enter a name and a description in the **General** tab.
6. Enter the calculation in the **Formula** tab.

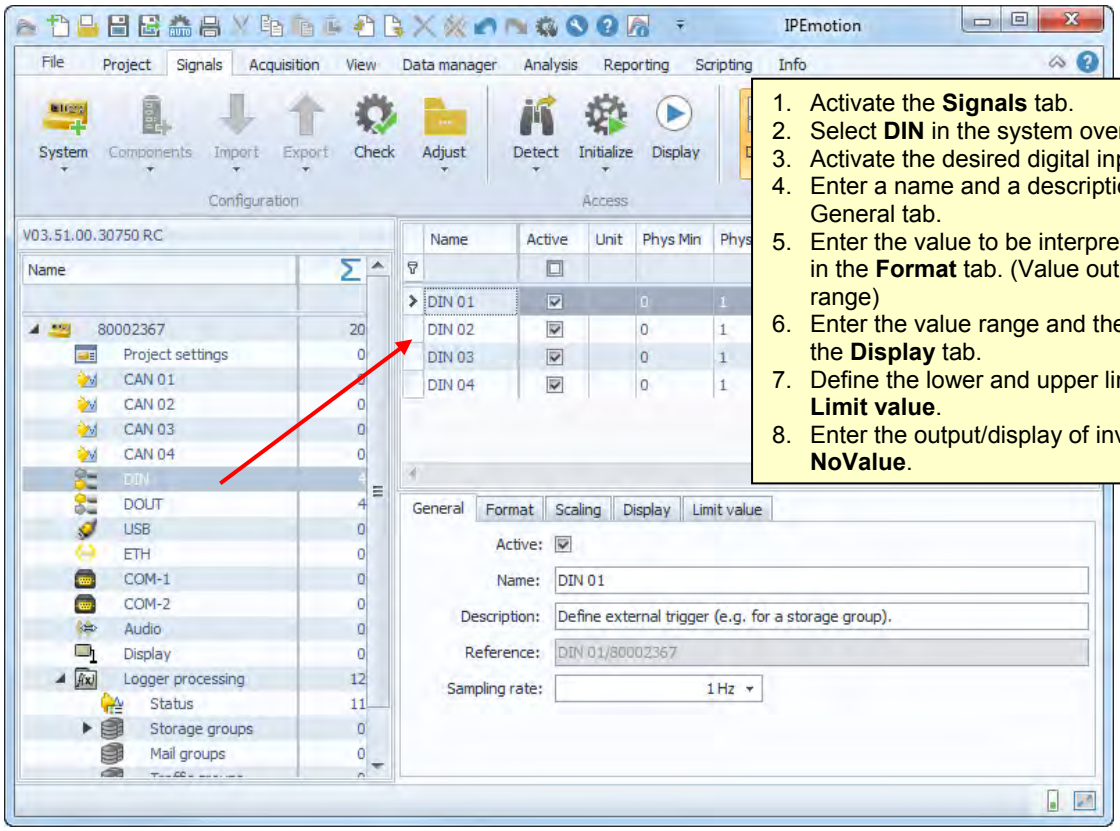
f(x)

7. Enter the calculation formula manually or use the Formula editor f(x).
8. Move signals and operators in the entry field by using Drag&Drop.
9. Confirm with OK.
10. Enter the value range and the decimals in the Display tab.
11. Define additional limit values and/or a NoValue.



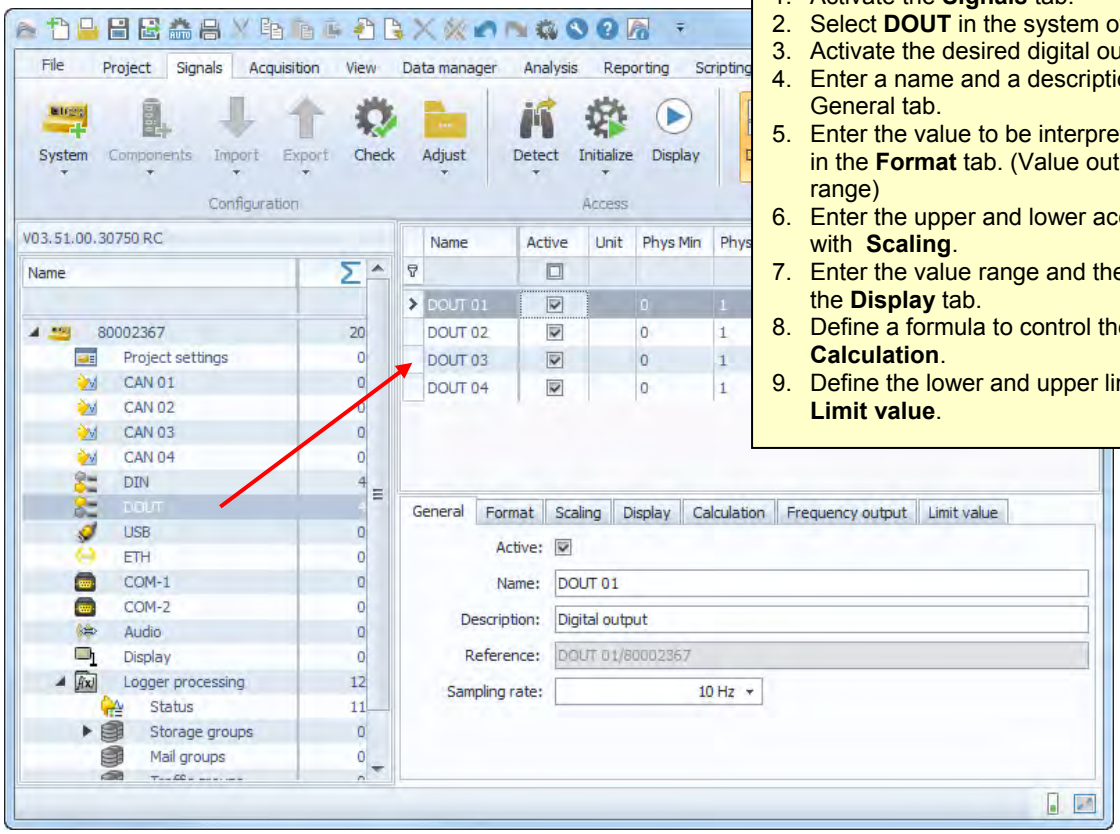
## 7.2 Digital inputs and outputs

### 7.2.1 Digital inputs




1. Activate the **Signals** tab.
2. Select **DIN** in the system overview.
3. Activate the desired digital inputs.
4. Enter a name and a description in the General tab.
5. Enter the value to be interpreted as NoValue in the **Format** tab. (Value out of the valid range)
6. Enter the value range and the decimals in the **Display** tab.
7. Define the lower and upper limit value with **Limit value**.
8. Enter the output/display of invalid value with **NoValue**.

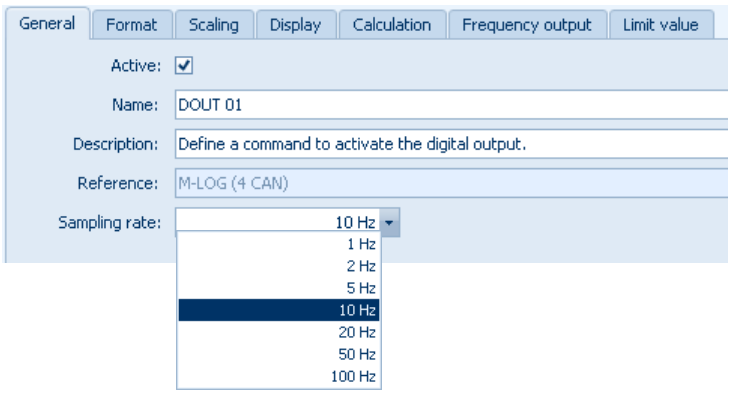
### 7.2.2 Digital outputs




1. Activate the **Signals** tab.
2. Select **DOUT** in the system overview.
3. Activate the desired digital outputs.
4. Enter a name and a description in the General tab.
5. Enter the value to be interpreted as NoValue in the **Format** tab. (Value out of the valid range)
6. Enter the upper and lower acquisition range with **Scaling**.
7. Enter the value range and the decimals in the **Display** tab.
8. Define a formula to control the output with **Calculation**.
9. Define the lower and upper limit value with **Limit value**.

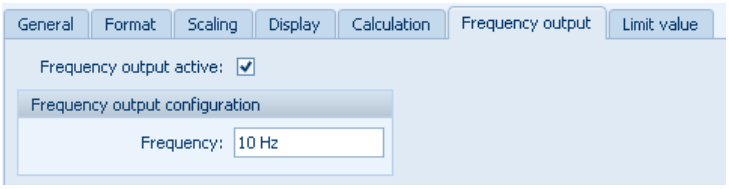
## Sampling rate DOUT

The logger's digital outputs (DOUT) support a definable sampling rate (cycle rate) of up to 100 Hz.




## Frequency output DOUT

If the frequency output is activated, a square wave voltage is sent. The signal frequency can be configured.



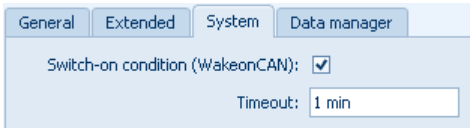
## 7.3 WakeOnCAN

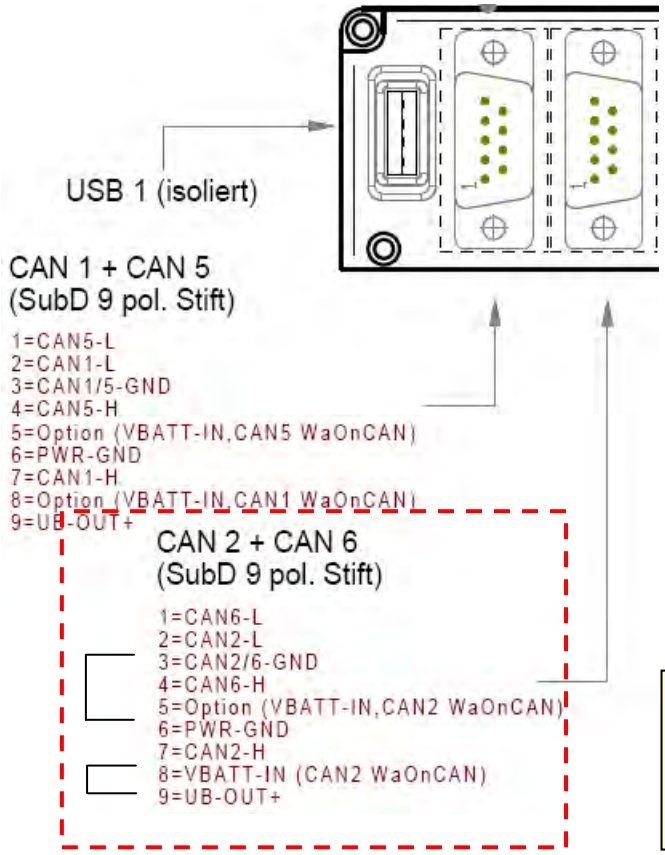
The WakeOnCAN function switches the logger on as soon as the respective CAN bus is active, i.e. when messages are transferred.

### 7.3.1 ON via WakeOnCAN, OFF via Ignition 15

If there is no **Switch-off time** configured, the remote signal (e.g. terminal 15) must be active within the set timeout to permanently switch-on the logger. The logger is turned off if the remote signal is not received after the timeout. The logger stays in operation mode as long as the remote signal is high. As soon as remote is low, the **Follow-up time** will start and the logger is switched-off when datat postprocessing is finished.

| | | | |
|---|---|---|---|
| Bridge | Pin 3 (CAN2/6-GND) | → | Pin 6 (PWR-GND) |
| Bridge | Pin 8 (WaOnCAN) | → | Pin 9 (UB-OUT+) |

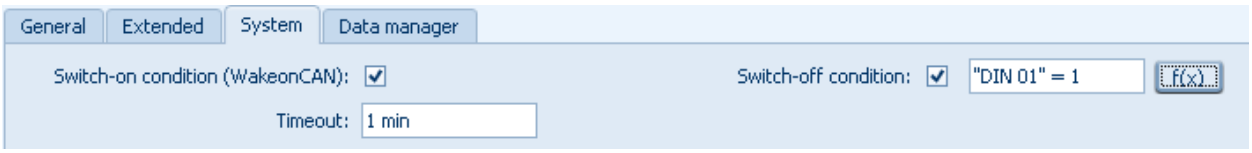



**Depending on the CAN card/Port replicator, the WakeOnCAN function supports**
- **only at CAN 2,**
- **at CAN 2 and CAN 3**
- **at all CAN inputs (special version)!**

### 7.3.2 ON via WakeOnCAN, OFF via Switch-off condition

With a configured Switch-off condition the Timeout setting has no influence. The logger stays in operation mode as long as the Switch-off condition is false.




*If a temporary remote signal (duration > 1 s) is identified in this mode, the Follow-up time starts with the falling signal edge and the logger is switched off (when postprocessing is finished), even with a false Switch-off condition.*

## 7.4 CAN-Send: Output signals to CAN bus

The CAN-Send feature outputs measured signals, calculated channels and status signals using a common data rate up to 100 Hz. You can use the automatic CAN ID assignment or you may enter individual settings for each channel. Save the current signal settings to a CANdb file (*.dbc) for a quick and easy exchange with the external application using the CANdb export feature.

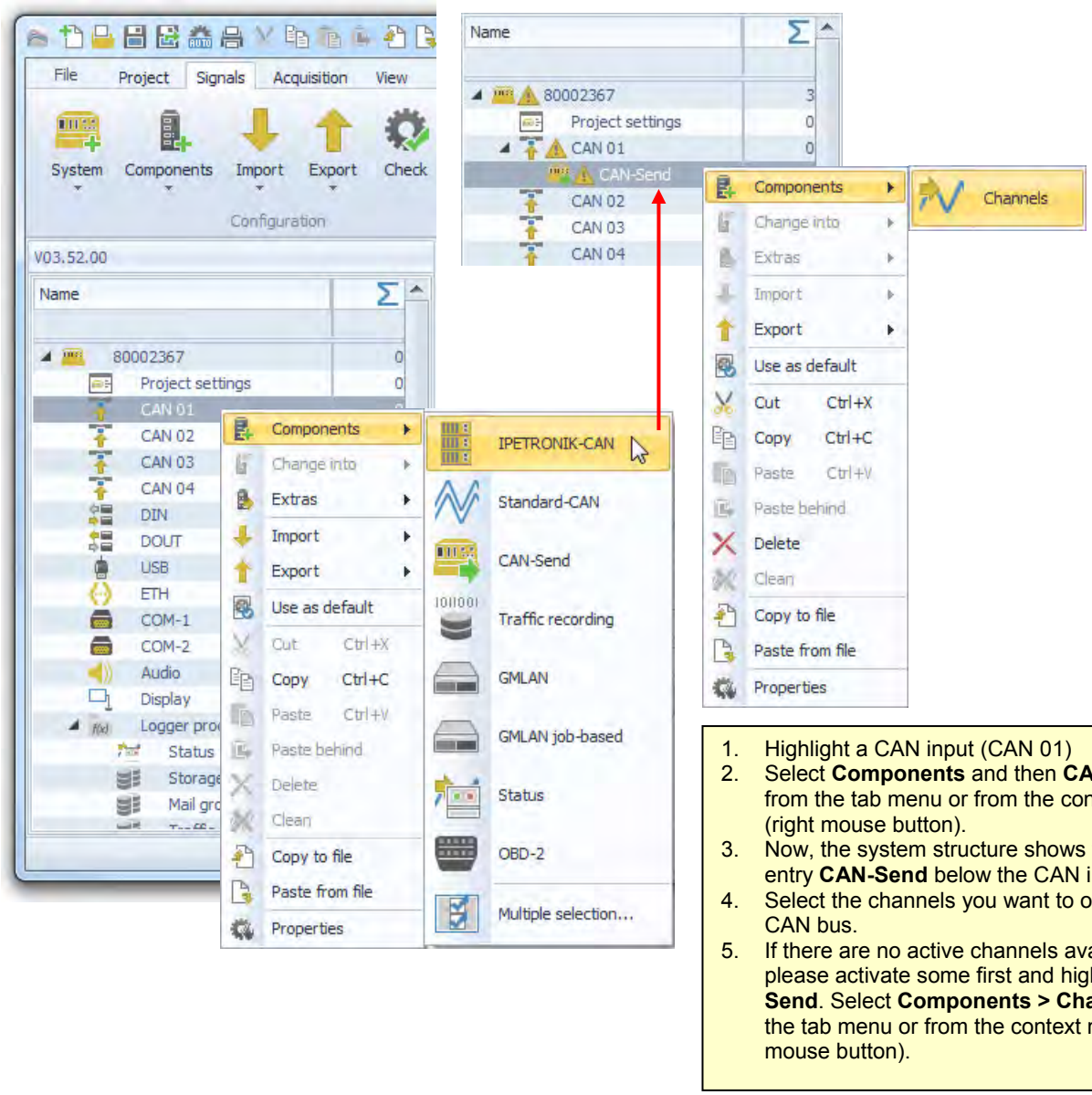
**Requirements:** **CAN board with FPGA version > 1.04.00**
**CAN-Send support for one CAN bus of the logger only**

At least one CAN participant must be connected to the CAN interface to transfer data to the CAN. If this is not the case, TESTdrive reports the following:

```
E Error sending CAN message in CANSendWorkStation. Counter = 1
```

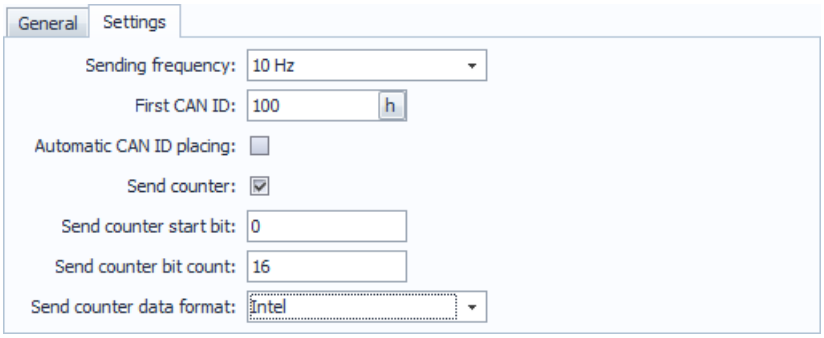
*Just after successfully connecting a CAN device data output to the CAN bus is startet, even when TESTdrive reports “Maximum number of error messages has been reached”.*

### Creating CAN-Send, Adding channels




1. Highlight a CAN input (CAN 01)
2. Select **Components** and then **CAN-Send** from the tab menu or from the context menu (right mouse button).
3. Now, the system structure shows the new entry **CAN-Send** below the CAN input.
4. Select the channels you want to output to CAN bus.
5. If there are no active channels available, please activate some first and highlight **CAN-Send**. Select **Components > Channels** from the tab menu or from the context menu (right mouse button).

## CAN settings




| | |
|---|---|
| Sending frequency | common output rate of all signals: 0.5/ 1/ 2/ 5/ 10/ 20/ 50/ 100 Hz |

If the data output rate is greater than or equal to the acquisition rate of source channel, this error report is written to the log file:

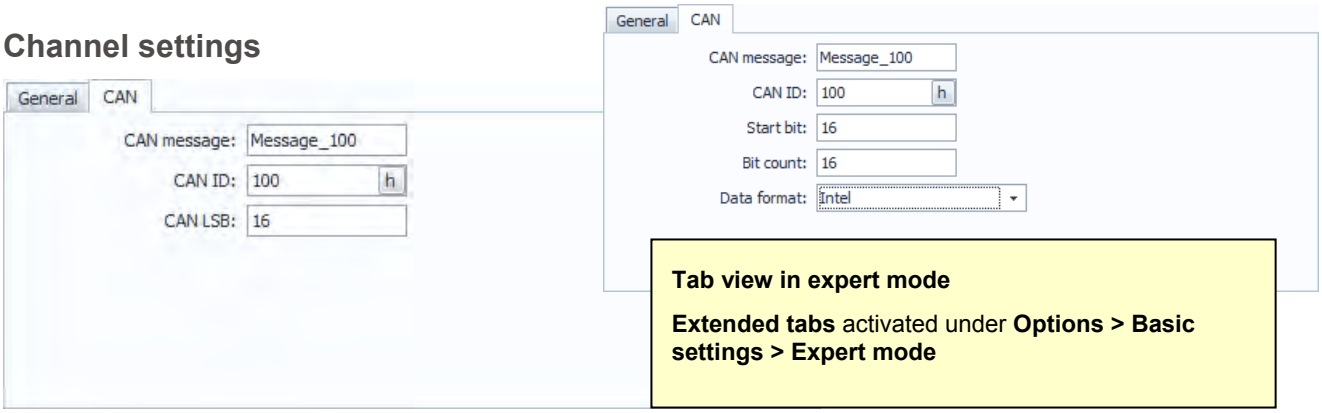
```
D ERROR in CCANSendWorkStation::Put() Fifo full!
```

| | |
|---|---|
| First CAN ID | Define the first CAN ID used for automatic CAN ID placing |
| Automatic CAN ID placing | Padding of CAN messages with selected signals and automatic CAN ID assignment to the respective message |
| Send counter | Counts up the sending tasks |
| Send counter start bit | Start bit used for the send counter value within the CAN message |
| Send counter bit count | Data length of the send counter value |
| Send counter data format | Data format of the send counter value (Intel, Motorola forward) |

*With automatic CAN ID placing, the first CAN message contains the value of the send counter (depending on data format byte 1 to 4).*

*Monitoring the send counter by the receiver is a simple way to verify a correct CAN communication.*

## Channel settings



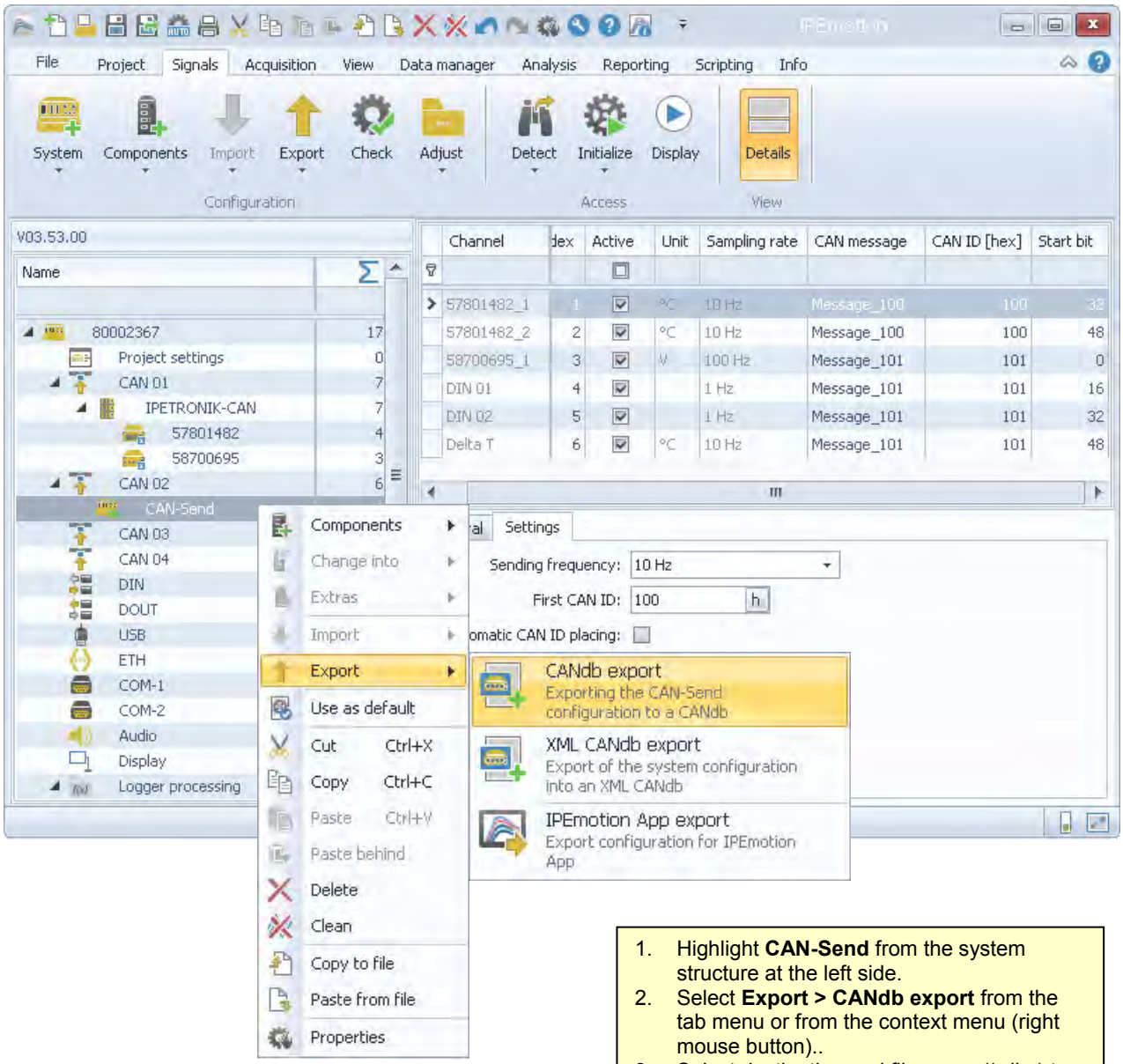

| | |
|---|---|
| CAN message | Name of the CAN message |
| CAN ID | Identifier of the CAN message (standard ID and extended ID supported) |
| CAN LSB | Start bit within the CAN message of the signal to be output |

**Extended settings (Expert mode only)**

| | |
|---|---|
| Start bit | Start bit within the CAN message of the signal to be output |
| Bit count | Number of bits per signal within the CAN message |
| Data format | Data format of the signal value (Intel, Motorola forward) |

## Exporting the settings to a CANdb

Export the current CAN send settings to a CANdb file after you finished the system configuration. This makes is easy to configure an external CAN device (notebook, or CAN data display) in order to acquire/display the signals streamed by the logger.



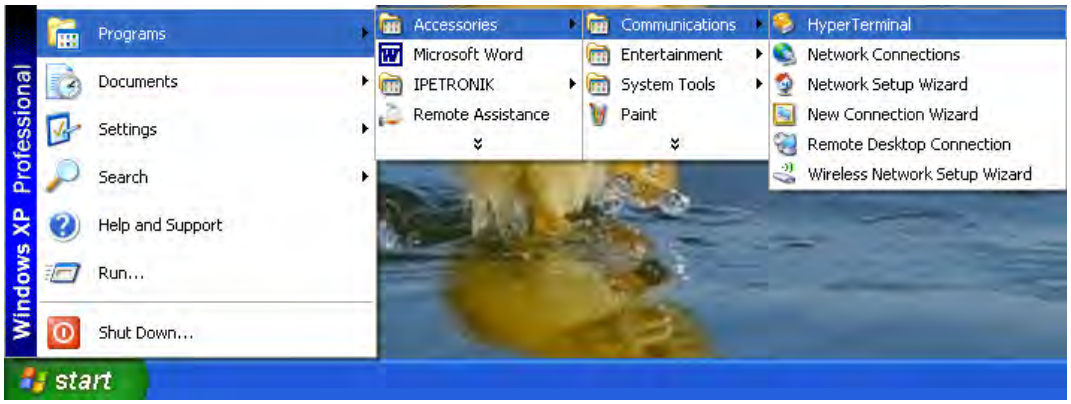

1. Highlight **CAN-Send** from the system structure at the left side.
2. Select **Export > CANdb export** from the tab menu or from the context menu (right mouse button)..
3. Select destination and file name (*.dbc) to save your export.
4. Confirm by clicking the **Save** button.

## 7.5 Log file output via Hyperterminal

The logger status messages can be displayed online with a RS232 PC or notebook connection. The MLOG.ht file is copied into the following directory:

C:\Documents and Settings\[user]\Startmenu\Programs\Accessories\Communikation\ HyperTerminal\....

Prior to this, a hyperterminal connection must be created.




Or create a new hyperterminal connection:



The following components are required for the connection of M-LOG and PC:

**for PCs with serial interface:**

1 x USB to RS232 converter

1 x Null modem cable

1 x Gender Changer

**for PCs without serial interface:**

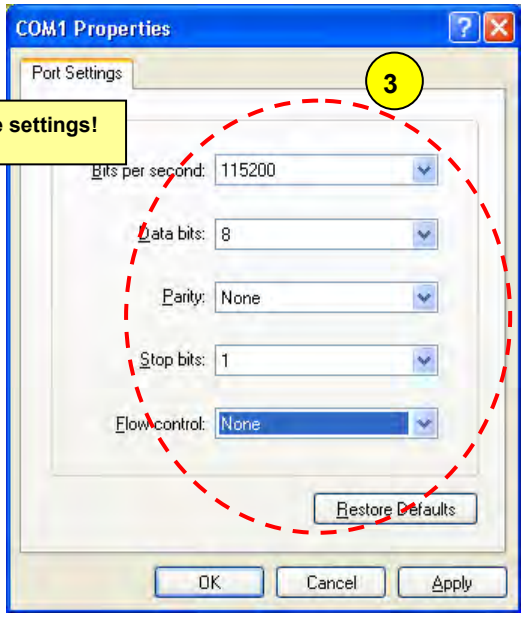
2 x USB to RS232 converter

1 x Null modem cable

1 x Gender Changer

The components for PCs without serial interface are available as an optional package (M-LOG-OPT-086).

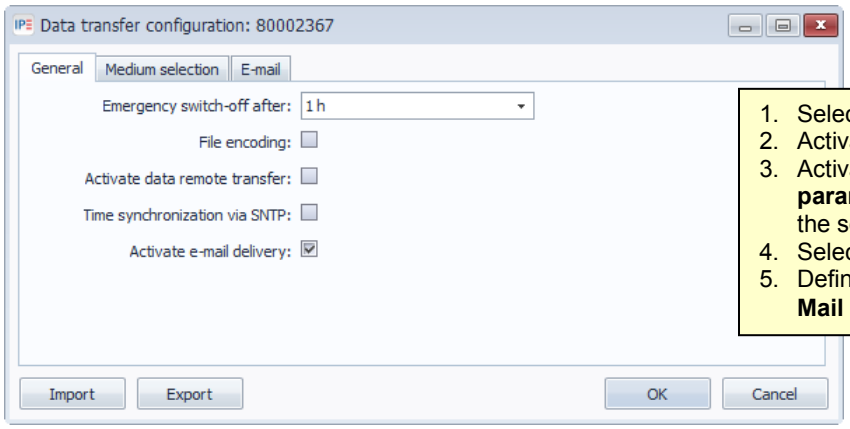
**Apply these settings!**



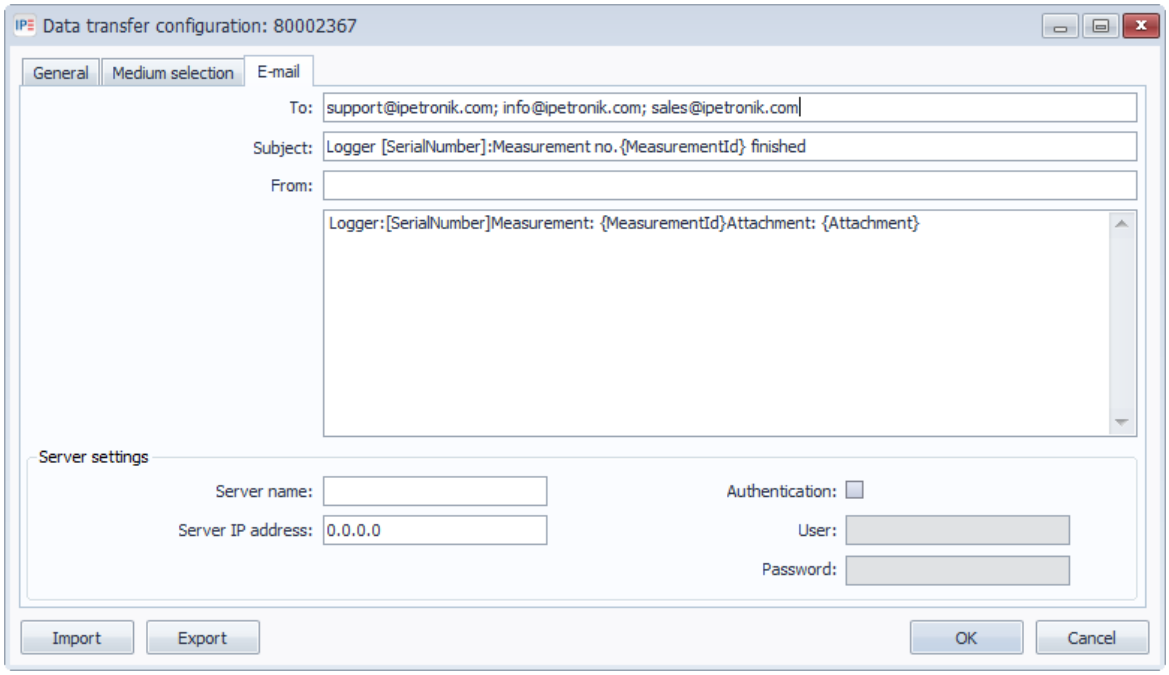
## 7.6 Send e-mail with status information

If this function is activated and the logger has an internet connection (LAN, WiFi, modem), it sends a status e-mail after acquisition stop with the following content:

- serial number and number of the current acquisition in subject line,
- attached log file
- attached measurement status file if activated
- attached STG file (Statistic Group, resp. Min-Max list) if activated




1. Select the logger in system overview.
2. Activate the **Data management** tab.
3. Activate the **Update connection parameters > Configuration** to change the settings for data transfer configuration.
4. Select **Activate e-mail delivery**.
5. Define the corresponding setting in the **E-Mail** tab.




| | |
|---|---|
| **To** | E-mail address of recipient |
| **Subject** | Subject line for serial number and mesaurement number file |
| **From** | E-mail address of sender (any text) |
| **Description field** | Description of serial number, measurement file number and attachement |
| **Server IP address** | IPE address of outgoing mail server (to e-mail account, e.g. smtp.mail.proivder.com) for sending |
| **Server name** | Alternative input of server name of outgoing mail server |
| **Authentication** | Access authorization to user e-mail account |
| User | User name |
| Password | Password |

## 7.7 Output messages to CAN / LIN

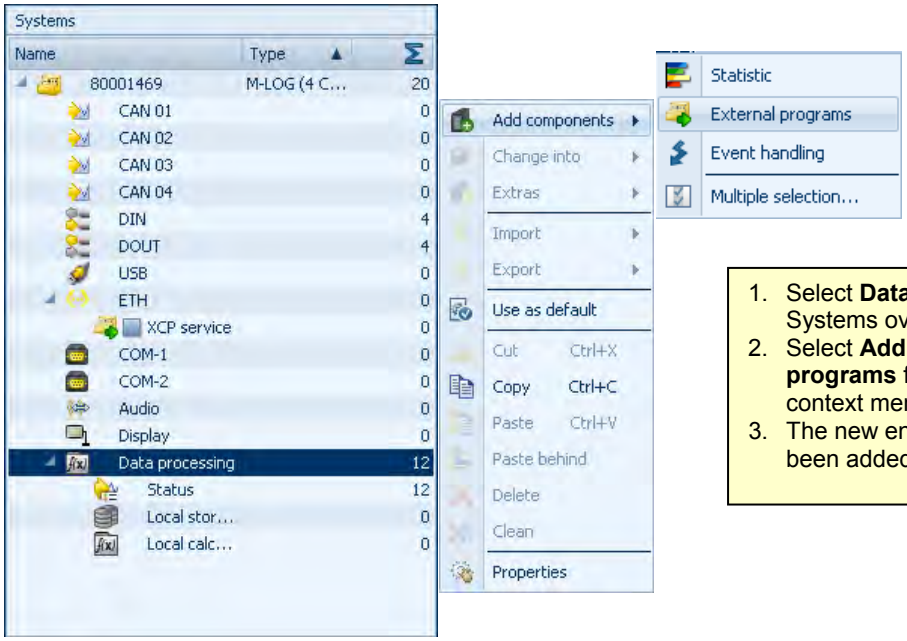
With the Logger PlugIn V03.21.00 / TESTdrive 3.21 and higher an output of user defined messages to CAN bus or LIN bus is supported.

The output timing can be selected from:
- one-time, at start measurement,
- one-time, at stop measurement,
- cyclical every x ms.

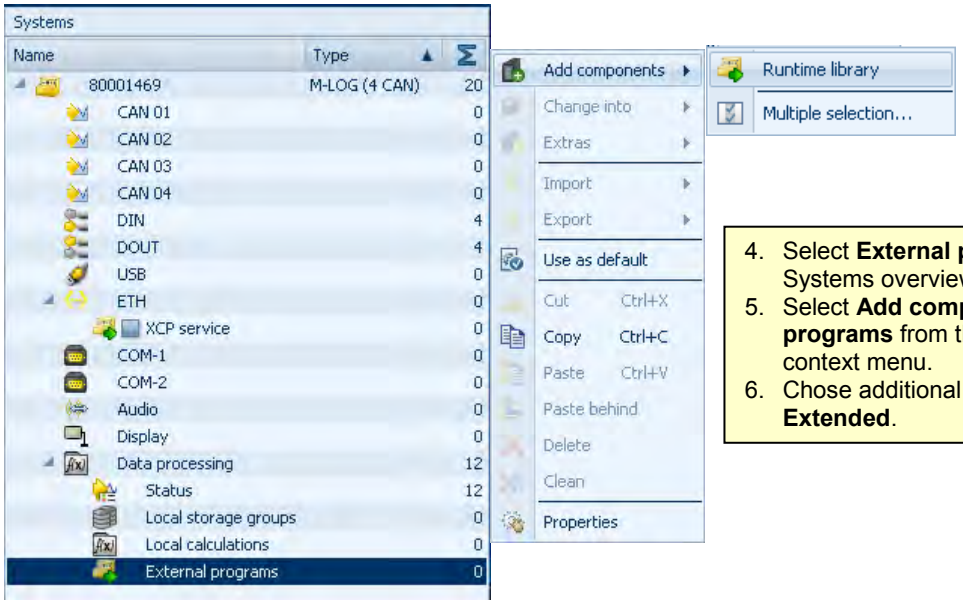
An external file with the *.DAT extension is used to configure the settings for the messages.
A sample file is available from:
**c:\Programs\IPETRONIK\IPEmotion PlugIn IPETRONIK LOG V03.xx.xx\Data\Channel.dat**

The header of this files contains detailed information regarding to the usage of the parameters.

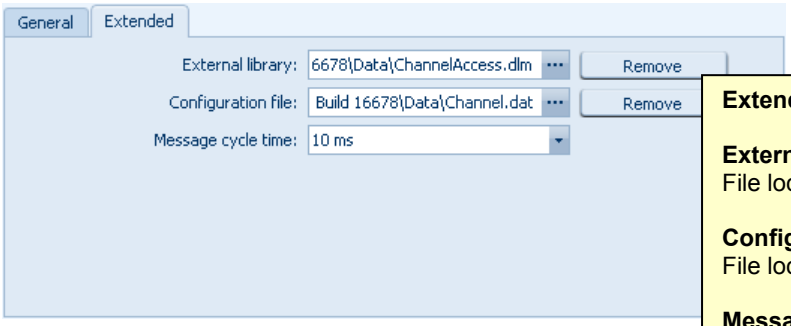
The screenshots below show the implementation of the external file in IPEmotion configuration.




1. Select **Data processing** in the left Systems overview.
2. Select **Add components** > **External programs** from the tab menu or from the context menu.
3. The new entry **External programs** has been added to **Data processing**.




4. Select **External programs** in the left Systems overview.
5. Select **Add components** > **External programs** from the tab menu or from the context menu.
6. Chose additional settings in the tab **Extended**.




**Extended (Runtime library)**

**External library**
File location of the external application (DLM)

**Configuration file**
File location of the configuration (DAT)

**Message cycle time**
Ouput rate of the message (repetition rate)

**Example Offset adjust for IPETRONIK modules connected to the logger's CAN input**

The message output feature offers the possibility to broadcast an offset adjust command to IPETRONIK CAN modules with adjust function.

The sample file **OffAddStart.dat** contains the adjust commands for task Manual and Group 1-4.
The sample shows a manual adjust on CAN 01 with a baud rate of 500 kBit/s, executed with a time delay of 2.5 seconds from start measurement.

Brief description of the parameters (see header of the sample Channel.dat for detailed settings):

// Hardware initialization **<---Basic setting of CAN/LIN input**

[Init]

| // Channel, | ChnType, | ChnIndex, | ChnMode, | Baudrate |
|---|---|---|---|---|
| CAN0, | 1, | 0, | 1, | 500000 |

**<------ChnType LIN=2 /CAN=1, ChnIndex Nr 0...11=Input , ChnMode 29Bit=2 11Bit=1 / Baud rate**

// List of messages (send/receive)
[Messages]
// Name, ID, Length Bytes **<------ ID CAN/LIN -ID; Length= Number of bytes; Bytes = Message**
//

| | | | |
|---|---|---|---|
| Manuell_1, | 0x0, 8, | 0x00, 0x91, 0x02, 0xCA, 0x3F, 0x00, 0x80, 0x80 | **<----Manual** |
| Manuell_2, | 0x0, 8, | 0x00, 0x91, 0x02, 0xCA, 0x3F, 0x00, 0xC0, 0x81 | |
| Manuell_3, | 0x0, 8, | 0x00, 0x91, 0x02, 0xCA, 0x3F, 0x00, 0x40, 0x8C | |
| Manuell_4, | 0x0, 8, | 0x00, 0x91, 0x02, 0xCA, 0x3F, 0x00, 0xC0, 0x8D | |
| Manuell_5, | 0x0, 8, | 0x00, 0x91, 0x02, 0xCA, 0x3F, 0x00, 0x00, 0x8E | |
| Group1_1, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC5, 0x3F, 0x00, 0x80, 0x80 | **<-----Group 1** |
| Group1_2, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC5, 0x3F, 0x00, 0xC0, 0x81 | |
| Group1_3, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC5, 0x3F, 0x00, 0x40, 0x8C | |
| Group1_4, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC5, 0x3F, 0x00, 0xC0, 0x8D | |
| Group1_5, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC5, 0x3F, 0x00, 0x00, 0x8E | |
| Group2_1, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC6, 0x3F, 0x00, 0x80, 0x80 | **<-----Group 2** |
| Group2_2, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC6, 0x3F, 0x00, 0xC0, 0x81 | |
| Group2_3, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC6, 0x3F, 0x00, 0x40, 0x8C | |
| Group2_4, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC6, 0x3F, 0x00, 0xC0, 0x8D | |
| Group2_5, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC6, 0x3F, 0x00, 0x00, 0x8E | |
| Group3_1, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC7, 0x3F, 0x00, 0x80, 0x80 | **<-----Group 3** |
| Group3_2, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC7, 0x3F, 0x00, 0xC0, 0x81 | |
| Group3_3, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC7, 0x3F, 0x00, 0x40, 0x8C | |
| Group3_4, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC7, 0x3F, 0x00, 0xC0, 0x8D | |
| Group3_5, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC7, 0x3F, 0x00, 0x00, 0x8E | |
| Group4_1, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC8, 0x3F, 0x00, 0x80, 0x80 | **<-----Group 4** |
| Group4_2, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC8, 0x3F, 0x00, 0xC0, 0x81 | |
| Group4_3, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC8, 0x3F, 0x00, 0x40, 0x8C | |
| Group4_4, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC8, 0x3F, 0x00, 0xC0, 0x8D | |
| Group4_5, | 0x0, 8, | 0x00, 0x91, 0x02, 0xC8, 0x3F, 0x00, 0x00, 0x8E | |

[Conditions]
// Name Condition

// Messages which have to be send at measurement start **<---- The output sequence is defined by the sequence of the list**
[OnStart]
// Time, Channel, Message

2500000, CAN0, Manuell_1
**<-- 2500000 µs after Start /CAN input/ To adjust a group, replace with Group1_1**

2500000, CAN0, Manuell_2
**<-- 2500000 µs after Start /CAN input/ To adjust a group, replace with Group1_2**

2500000, CAN0, Manuell_3
**<-- 2500000 µs after Start /CAN input/ To adjust a group, replace with Group1_3**

2500000, CAN0, Manuell_4
**<-- 2500000 µs after Start /CAN input/ To adjust a group, replace with Group1_4**

2500000, CAN0, Manuell_5
**<-- 2500000 µs after Start /CAN input/ To adjust a group, replace with Group1_5**

Refer to the header of the *.DAT file for further parameter settings.

## 7.8 Event controlled measurement

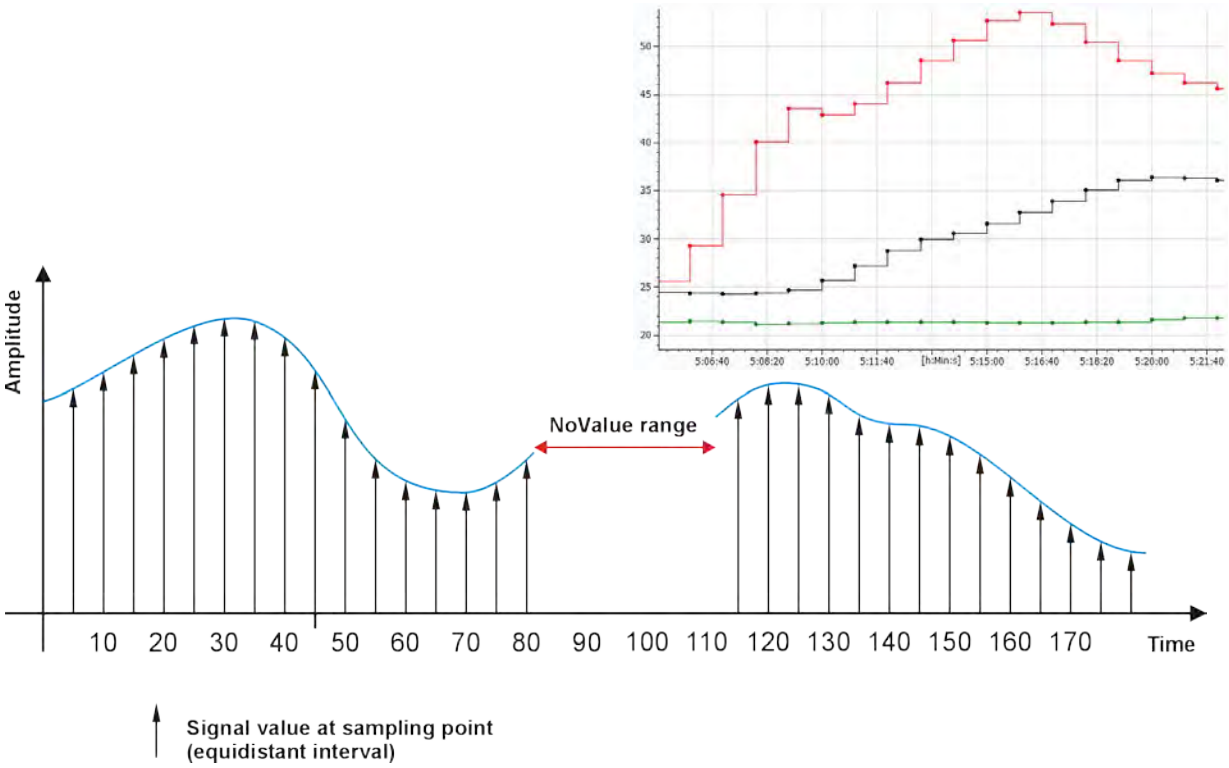
### 7.8.1 Possibilities for data acquisition

For measurement signal acquisition and storage in electronic systems, analog signals need to be digitalized, first. This is done by taking discrete measuring values from the continuous signal sequence (Sample & Hold) and recording them cyclically. Cyclic recording is also used for native digital signals, e.g. for measuring data packages from bus systems.

For some applications, it is useful to not record CAN bus data cyclically but event-controlled.

In the following, the basic features of both data acquisition types are described.

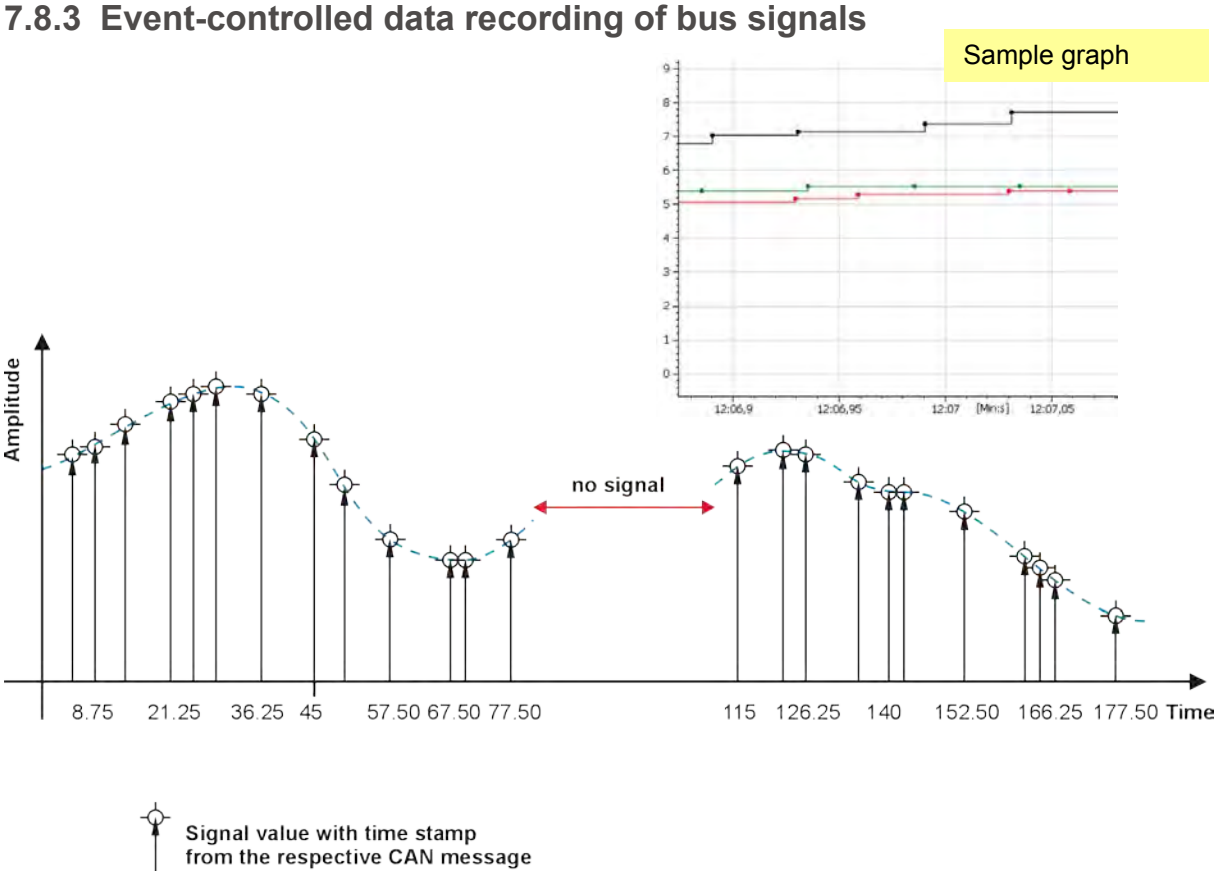
## 7.8.2 Cyclic data recording of continuous signals




### Features of cyclic data recording (PlugIn IPETRONIK-LOG)

- time-based recording in a fixed time grid, e.g. sampling rate 100 Hz
- different storage groups allow different data rates for recording
- individual time channel for each storage group in measurement data set
- continuous recording in equidistant intervals
- suitable for analog signals
- improved time accuracy due to increased sampling rate (oversampling)
- clear comparability of different signals using a synchronization clock (Master Sample Clock MSC)
- protocol measurement during data acquisition using bus systems is possible (CCP, XCP, FlexRay, ... )
- bus signals are allocated to the respective time grid (sampling rate)
- (time) differentiation of two signals is not possible in the time grid
- sampling points without real signal value are indicated with "NoValue" (invalid) in data set

## 7.8.3 Event-controlled data recording of bus signals




### Features of event-controlled data recording (PlugIn IPETRONIK-LOG)

- event-controlled recording without fixed time grid for bus signals, e.g. CAN bus
- all signals of a message are provided with an exact time stamp, as with traffic measurement
- individual time channel for each message in measurement data set
- discontinuous recording without defined time grid
- suitable for measurement of bus signal differences and of sporadic or single signals
- measuring values in different messages feature different time stamps > in the result graph, the time stamps do not lie on the same x-values any more
- no protocol measurement possible (CCP, XCP; FlexRay, ... )
- determination of exact time difference between signals (difference between time stamps)
- "NoValue" entries indicating the absence of signals are avoided

## 7.8.4 Setting up event-controlled data recording (PlugIn IPETRONIK-LOG)

### General

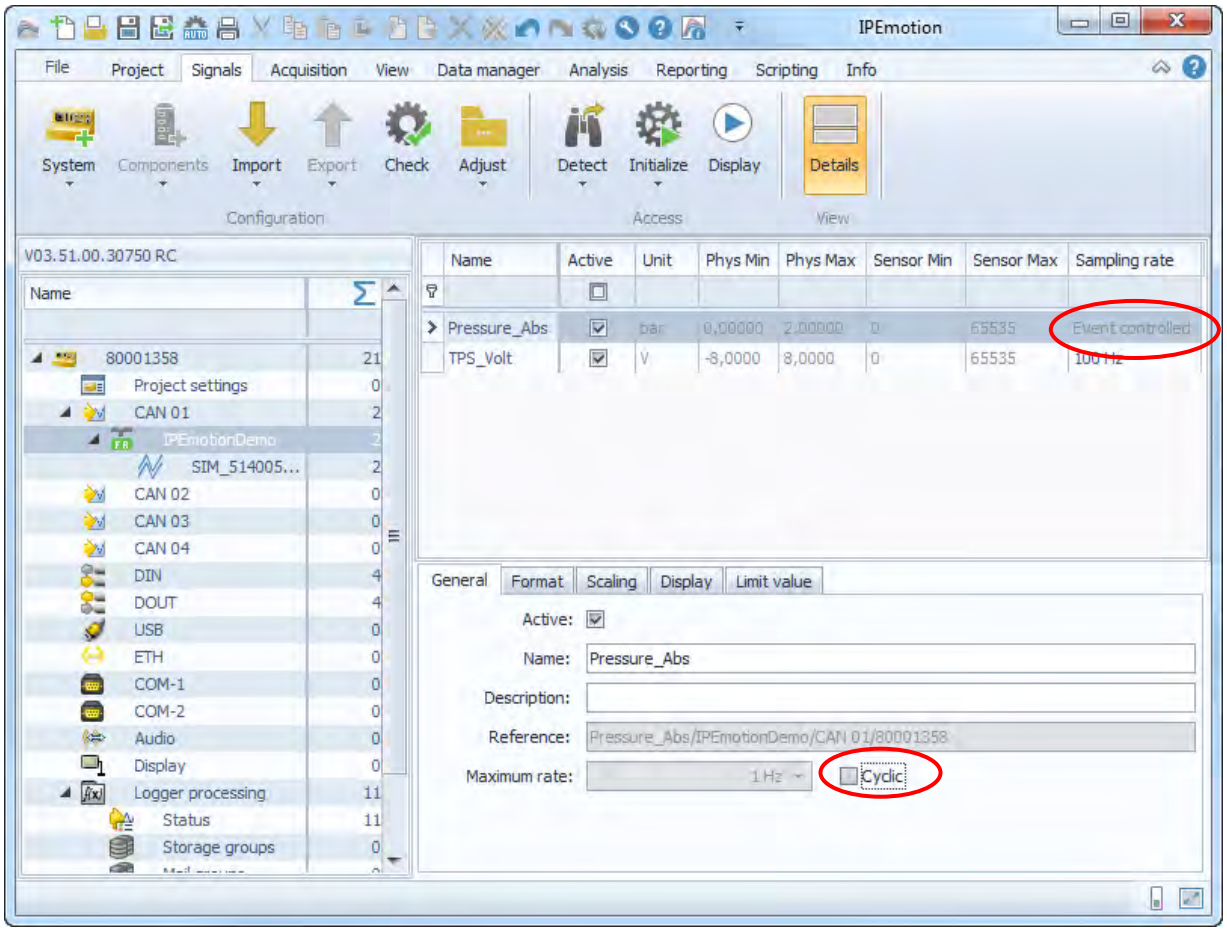
Event-controlled measurement, available from TESTdrive 3.51.00 onwards, features the benefits of signal measurement and traffic measurement:

Signal selection is already interpreted (name, scaling, unit,...). Yet, measurement is not performed cyclically but with "real" time stamp. Signals are only recorded if they really are present on the CAN bus. Using the exact time stamp.

Event-controlled measurement appears like a signal measurement on the configuration surface and in the measurement file. Loggers process the respective storage groups in traffic measurement mode.

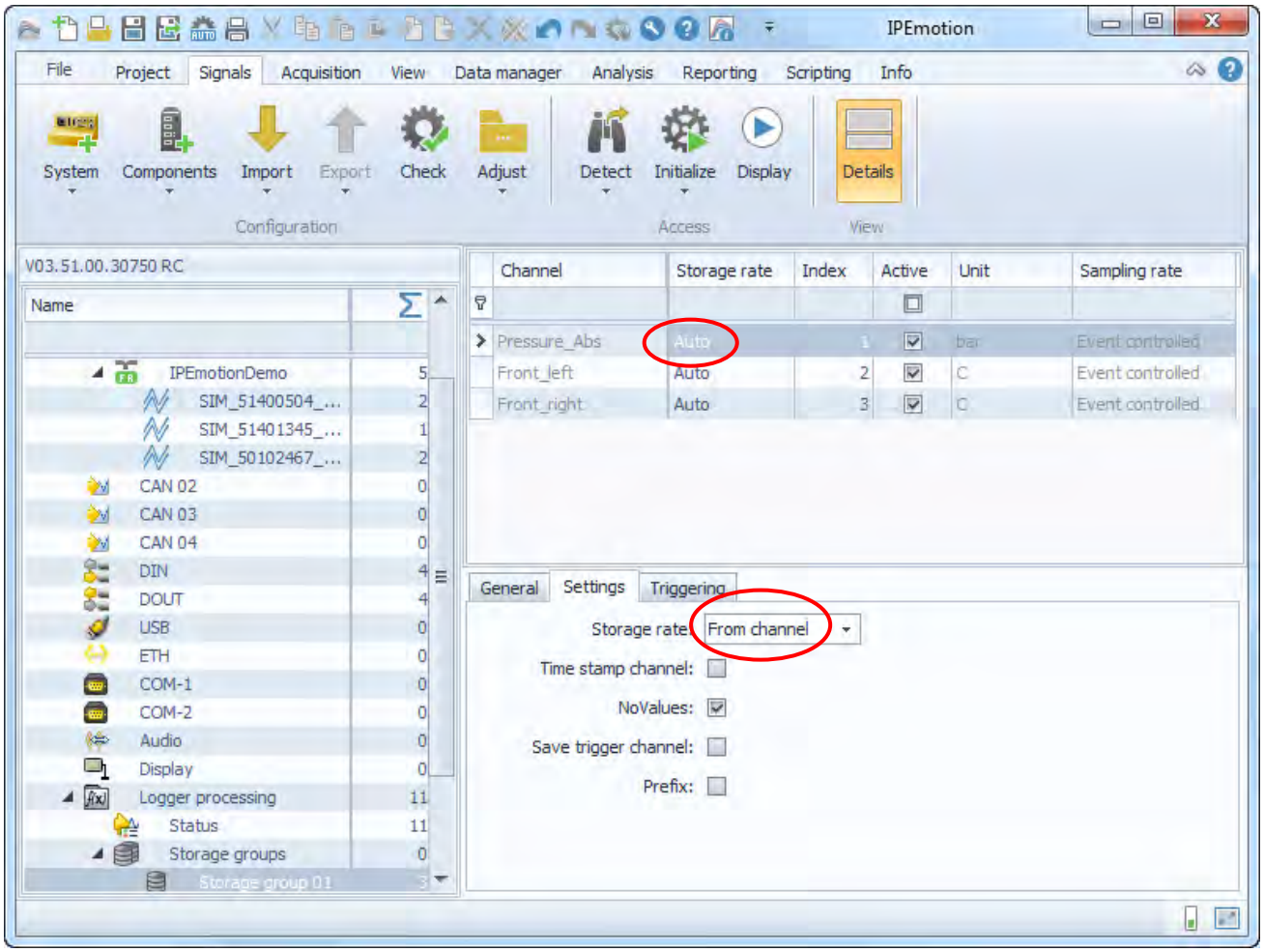
### Configuration settings

The respective signals must not be recorded cyclically.



In the storage group, the storage rate has to be set **From channel**. It should not be a fixed storage rate.

As a result, sampling rate is **Event controlled** and storage rate is **Auto**.

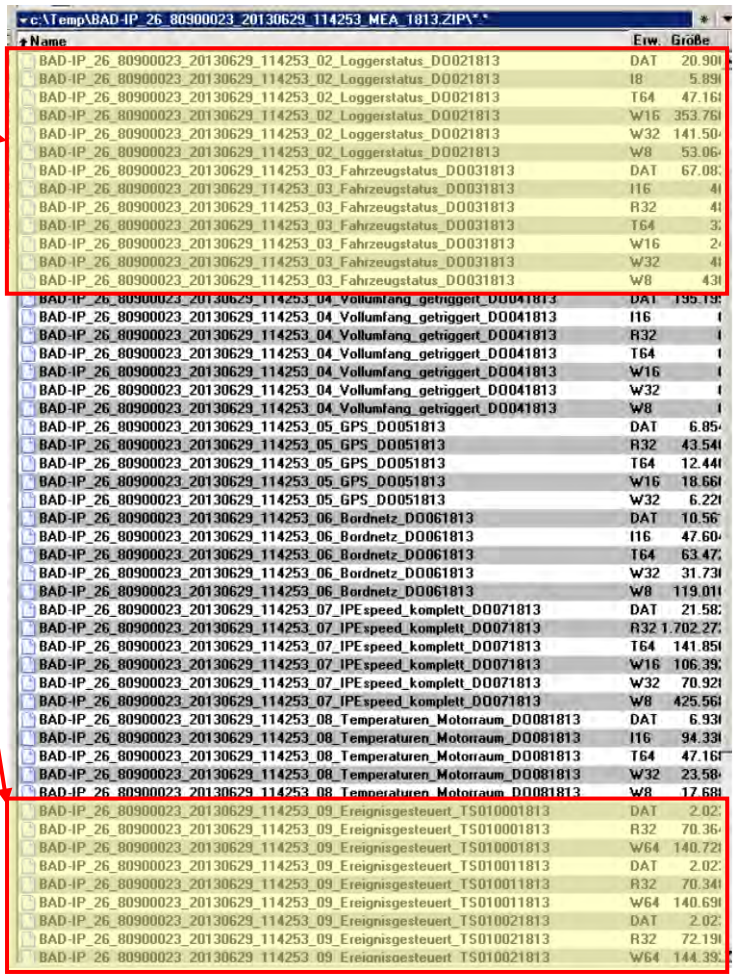


## Measurement file

Event-based signals are recorded in an event-based storage group TS_xxxxxx.DAT. A signal-based storage group DOxxxxxx.DAT with cyclic storage rate cannot contain event-based signals since event-controlled or cyclic recording types are properties of signals and not of storage groups.

### There are two types DAT files:

- DOxxxxxx.DAT for storage groups with cyclic storage rate and signals with fixed, cyclic sampling rate
- TSxxxxxx.DAT for storage groups with event-controlled, non-cyclic signals without fixed sampling rate




### Conclusions

- Each message that is recorded event-controlled receives an individual time channel
- In DAT format of the logger measurement each message creates an own internal storage group (TSxxxxxx.DAT)
- In order to ensure strong system performance, it is useful to check the number of event-controlled measuring signals according to the application (necessary number of storage groups).

## Measurement data set

In table view, it becomes obvious that time intervals are not equidistant any more and messages (signals) are no more identical. Each value is measured and saved at the time it actually occurs on the CAN bus.

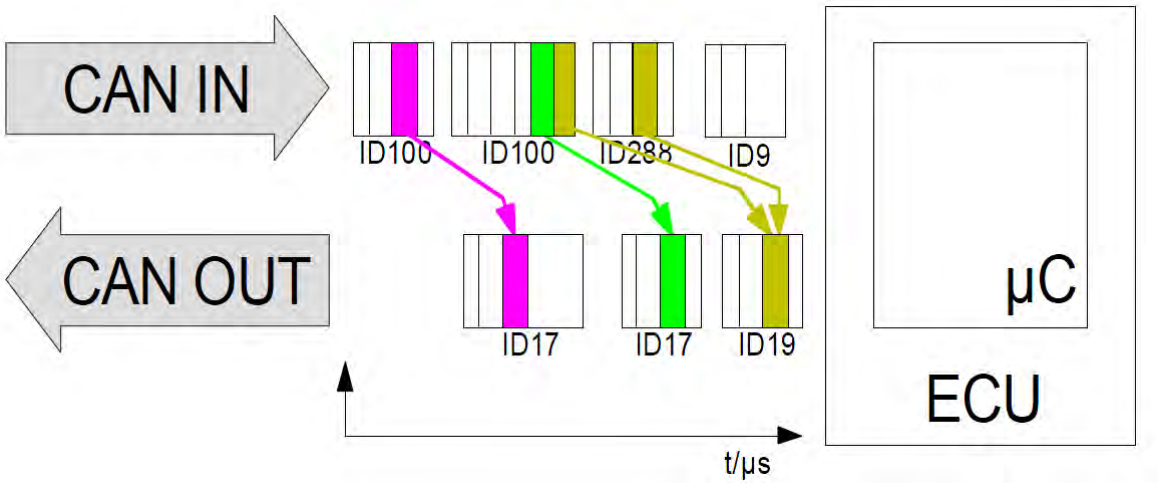
| Index | Time BAD-IP_26_8090... | Speed_mph_3 BAD-IP_26_80900023_... | Time BAD-IP_26_80... | Speed_mph_2 BAD-IP_26_809000... | Time BAD-IP_26_8... | Speed_mph_1 BAD-IP_26_809000... |
|---|---|---|---|---|---|---|
| 4934 | 240,898994 | 2,53171491622925 | 246,98334 | 2,76187086105347 | 247,638911 | 2,41663670539856 |
| 4935 | 240,97946 | 2,53171491622925 | 247,033392 | 2,76187086105347 | 247,688266 | 2,41663670539856 |
| 4936 | 240,999118 | 2,53171491622925 | 247,084021 | 2,76187086105347 | 247,738998 | 2,53171491622925 |
| 4937 | 241,079566 | 2,41663670539856 | 247,123429 | 2,76187086105347 | 247,788547 | 2,76187086105347 |
| 4938 | 241,098938 | 2,41663670539856 | 247,173719 | 2,64679265022278 | 247,848182 | 2,76187086105347 |
| 4939 | 241,179301 | 2,41663670539856 | 247,223519 | 2,64679265022278 | 247,88858 | 2,53171491622925 |
| 4940 | 241,260063 | 2,30155897140503 | 247,273564 | 2,64679265022278 | 247,948201 | 2,18648099899292 |
| 4941 | 241,289621 | 2,30155897140503 | 247,323497 | 2,64679265022278 | 247,998492 | 2,18648099899292 |
| 4942 | 241,318031 | 2,30155897140503 | 247,373686 | 2,64679265022278 | 248,058398 | 2,18648099899292 |
| 4943 | 241,389505 | 2,30155897140503 | 247,423454 | 2,64679265022278 | 248,088545 | 2,41663670539856 |
| 4944 | 241,41812 | 2,30155897140503 | 247,473635 | 2,64679265022278 | 248,148236 | 2,53171491622925 |
| 4945 | 241,489817 | 2,18648099899292 | 247,52342 | 2,64679265022278 | 248,188478 | 2,30155897140503 |
| 4946 | 241,508455 | 2,18648099899292 | 247,573539 | 2,53171491622925 | 248,248249 | 2,18648099899292 |
| 4947 | 241,579502 | 2,18648099899292 | 247,62349 | 2,53171491622925 | 248,288752 | 2,30155897140503 |
| 4948 | 241,608423 | 2,18648099899292 | 247,67366 | 2,53171491622925 | 248,348459 | 2,41663670539856 |
| 4949 | 241,679551 | 2,18648099899292 | 247,723447 | 2,53171491622925 | 248,388811 | 2,41663670539856 |
| 4950 | 241,708274 | 2,07140302658081 | 247,773624 | 2,53171491622925 | 248,448397 | 2,41663670539856 |
| 4951 | 241,779567 | 2,07140302658081 | 247,823407 | 2,53171491622925 | 248,48874 | 2,53171491622925 |
| 4952 | 241,808086 | 2,07140302658081 | 247,873659 | 2,53171491622925 | 248,548416 | 2,64679265022278 |
| 4953 | 241,879497 | 2,07140302658081 | 247,923452 | 2,41663670539856 | 248,588424 | 2,87694883346558 |
| 4954 | 241,908284 | 2,07140302658081 | 247,973633 | 2,41663670539856 | 248,638883 | 3,10710477828979 |
| 4955 | 241,979364 | 2,07140302658081 | 248,033872 | 2,41663670539856 | 248,688384 | 3,45233845710754 |

## 7.8.5 Practical example: Determination of the latency of two signals

### Task

An electronic system (ECU) receives messages via CAN bus (input signals) and sends them out again on the CAN bus, e.g. as forwarding or after calculation (output signals).

Processing time = response time of the system is to be determined.




### Realization

The response time results from the time difference between input and respective output signal. Since both incoming and outgoing CAN messages are provided with a time stamp, the exact time difference can be determined through event-controlled measurement.

Advantages:

- No event gets lost (not fixed to a sampling rate or to several signals within one sampling interval).
- Unique time stamps from TESTdrive are used.
- Event-controlled measurement is based on traffic measurement > traffic group.
- Each ID generates an event with a new signal value.

# 8 Options (license required)

## 8.1 Hardware options (internal)

M-LOG and S-LOG can be extended by 3 slots with different IPETRONIK cards. Max. 3 cards can be combined depending on the assignment to the extension slots and the port replicator. The active CAN cards support a highly accurate 1 µs time stamp. This time stamp is synchronous for all inputs within a card.

### 8.1.1 CAN cards

The following CAN cards with galvanically isolated high speed inputs acc. to ISO 11898-2 and low speed acc.to ISO 11992-1 are available:

- 2 x CAN High Speed, WakeOnCAN functionality, active data preprocessing and message buffer
- 4 x CAN High Speed, WakeOnCAN functionality, active data preprocessing and message buffer
- 3 x CAN High Speed, WakeOnCAN functionality, active data preprocessing and message buffer + 1 x CAN Low Speed (5 V, fault-olerant)
- 3 x CAN High Speed, WakeOnCAN functionality, active data preprocessing and message buffer + 1 x CAN Low Speed (24 V)

### 8.1.2 CAN / LIN cards

The following CAN LIN combination cards with galvanically isolated inputs are available:

- 2 x CAN High Speed, WakeOnCAN functionality, active data preprocessing and message buffer + 2 x LIN
- 2 x CAN Single Wire (GMW 3089 V2.1) + 2 x LIN
- 1 x CAN Single Wire (GMW 3089 V2.1) + 1 x CAN High Speed, WakeOnCAN functionality, active data preprocessing and message buffer + 2 x LIN

### 8.1.3 Ethernet cards

The following ETH card with galvanically isolated inputs is available:

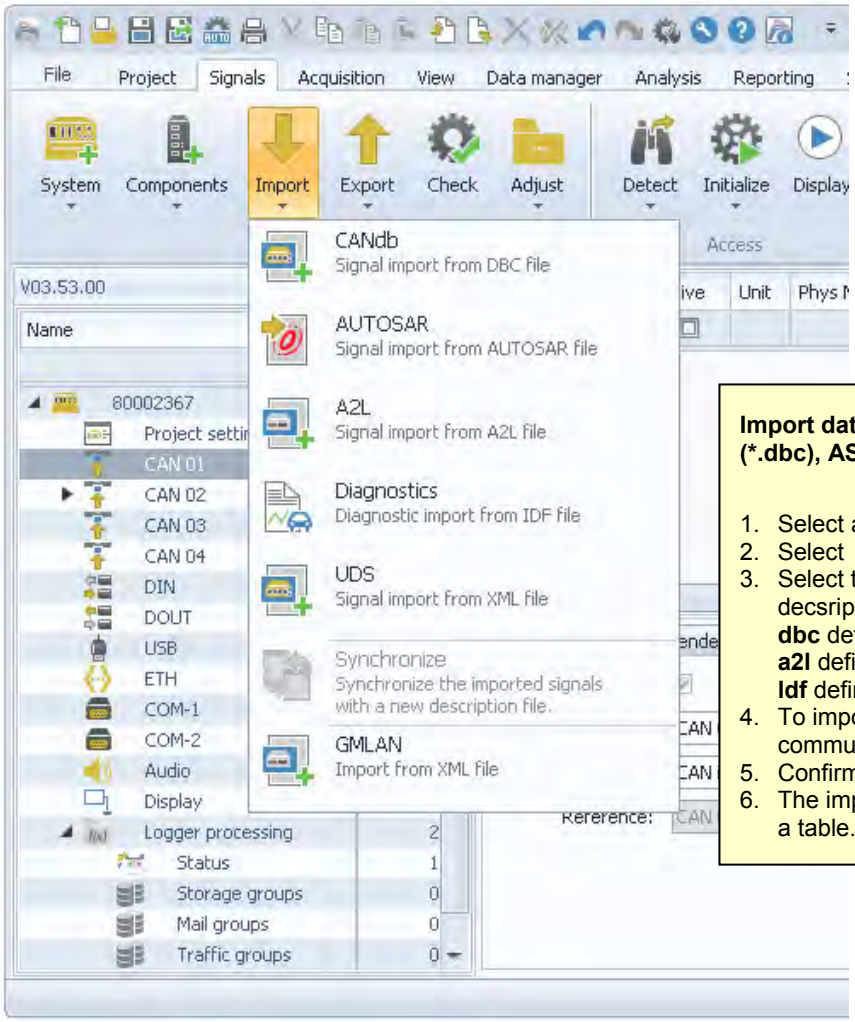
- 2 x ETH 10/100 MBit LAN, e.g. as input via XCPonEthernet or FlexRay-Ethernet converter

## 8.2 Software options

### 8.2.1 Import signal description files

IPEmotion supports the import and management of signal descriptions from CANdb (*.dbc), ASAP2 files (*.a2l), and diagnostic description files (*.idf).

The contents are read and managed by importing the original files into a database (Microsoft SQL-Server). The original files are therefore not longer required. Please note at updating the CAN system or the control unit that the current description file must be reloaded to update signal descriptions, if required. An export of the signal settings changed with IPEmotion into the original description file is not possible.




**Import data from a description file: CANdb (*.dbc), ASAP2 (.a2l), LINdb (*.ldf)**

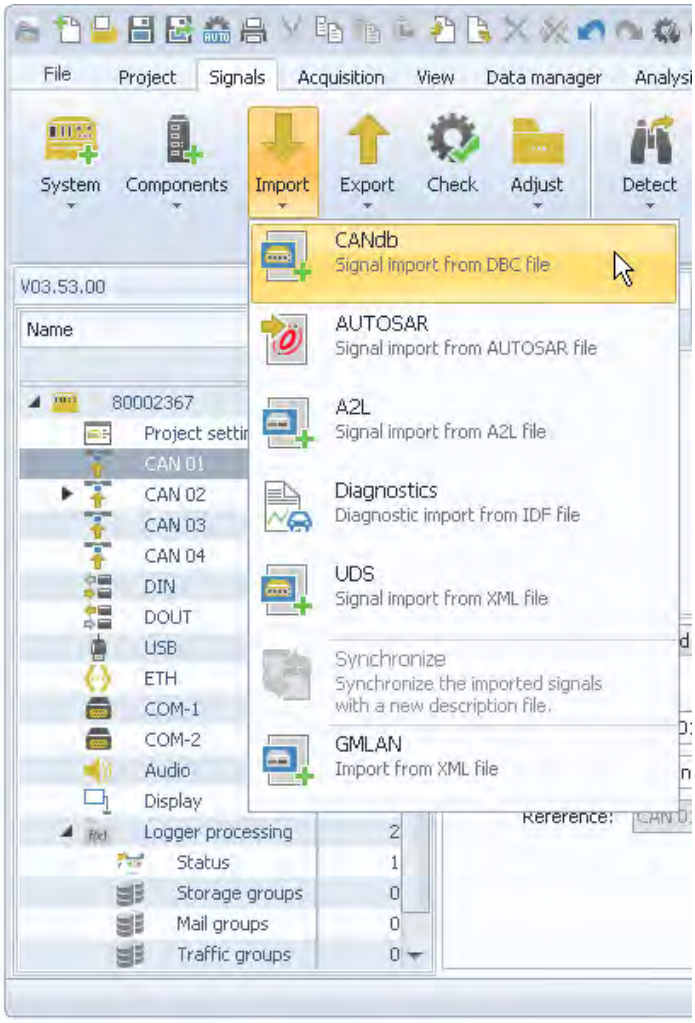
1. Select a CAN input (here CAN 01).
2. Select **Import** from the tool bar.
3. Select the directory and the type of the decsription file to open.
   **dbc** defines CANdb files
   **a2l** defines ASAP2 files
   **ldf** defines LINdb files (LIN Description File)
4. To import an A2L file, select the communication protocol first
5. Confirm with OK.
6. The imported signals (Pool) are displayed in a table.

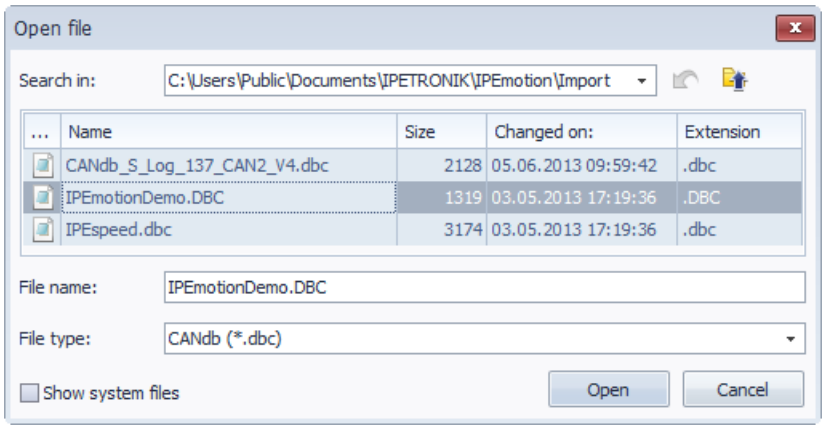
*General signals at CAN 01 cannot be configured if CAN 01 is already used for IPETRONIK devices!*

*Importing signal descriptions from ASAP2 or CANdb is the easiest and most secure method to configure signals.*

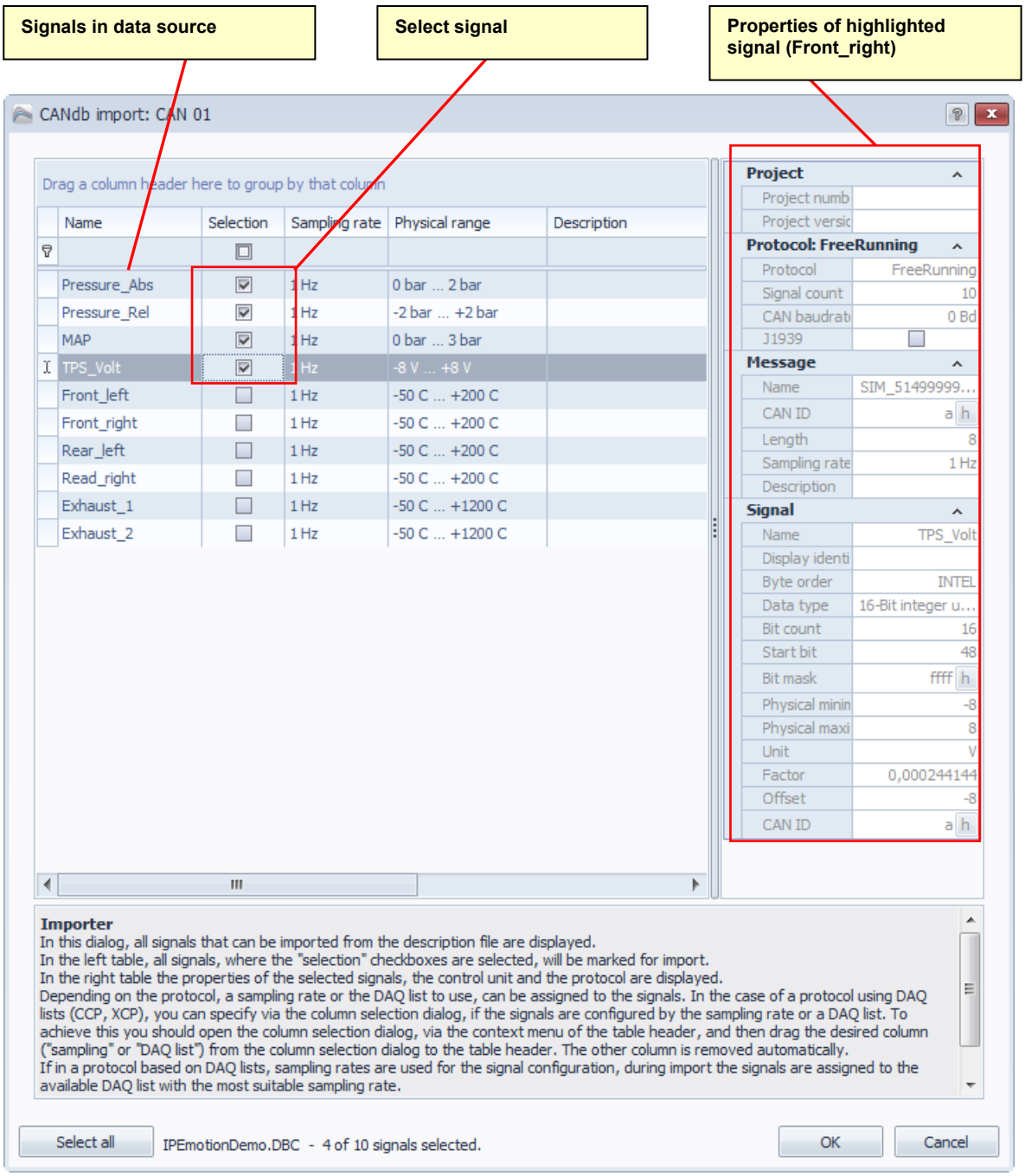
*Signals can also be manually created at an input (without description file). Select **Add components > Standard CAN** from the tool bar within the **Signals** navigation tab.*

## Import CANdb file

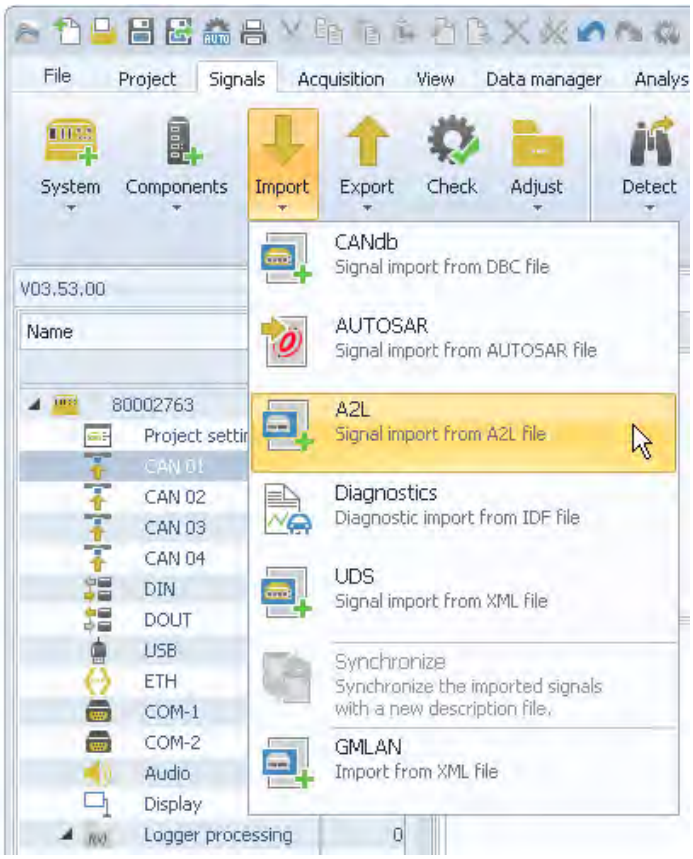


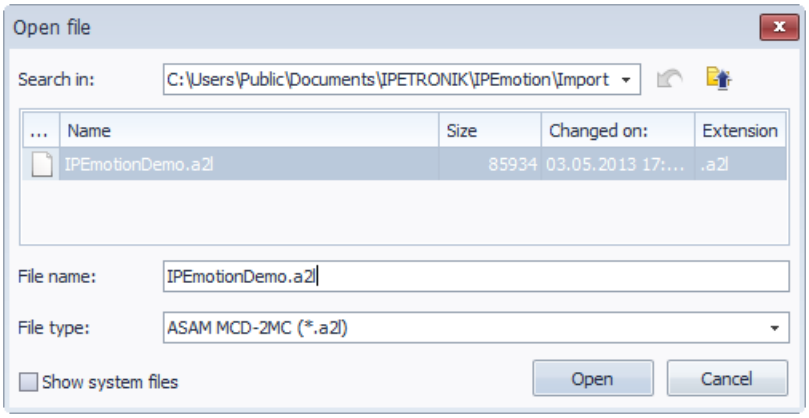



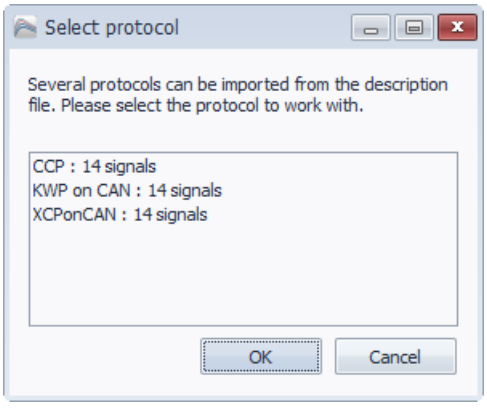
## Import signal descriptions from the CANdb



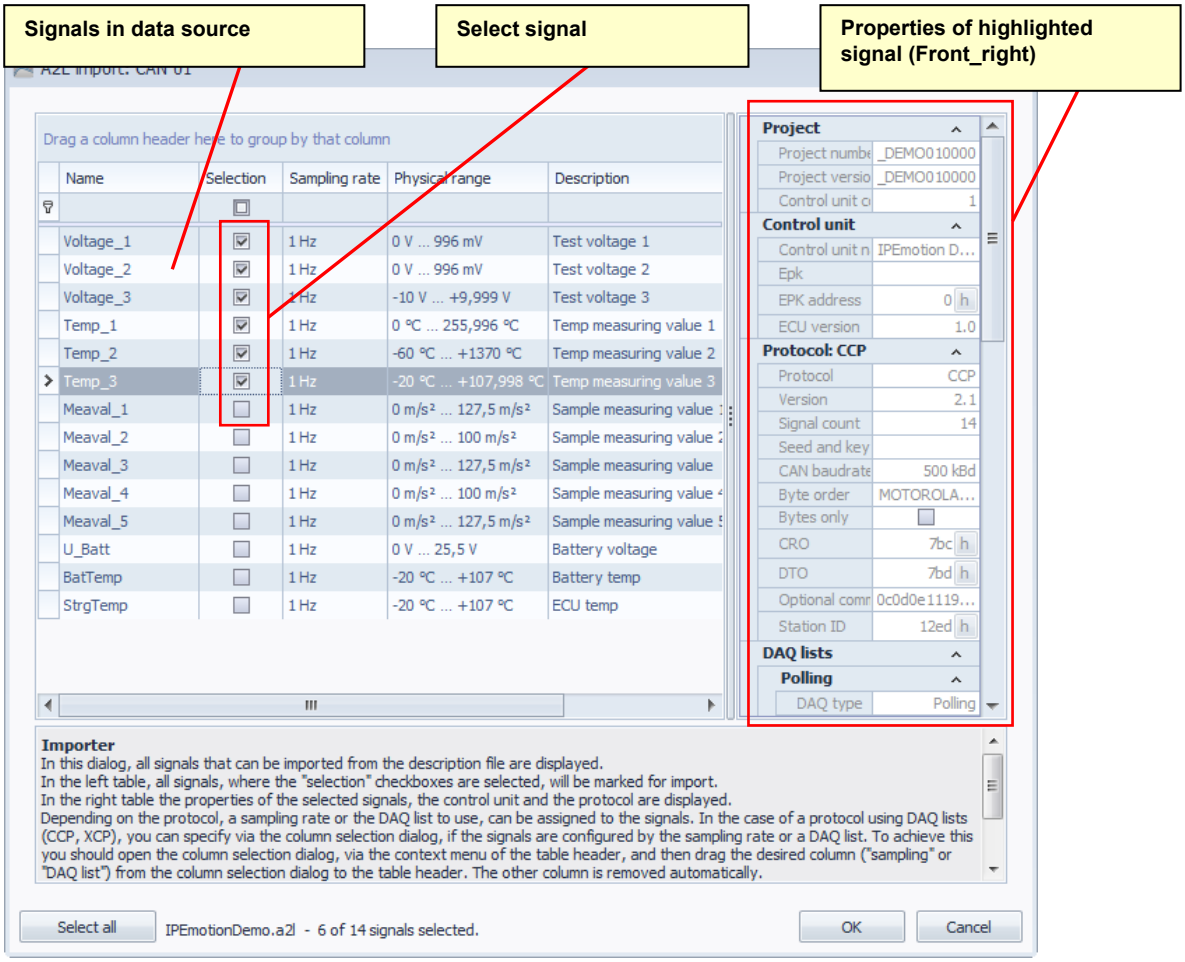
## Import ASAP file



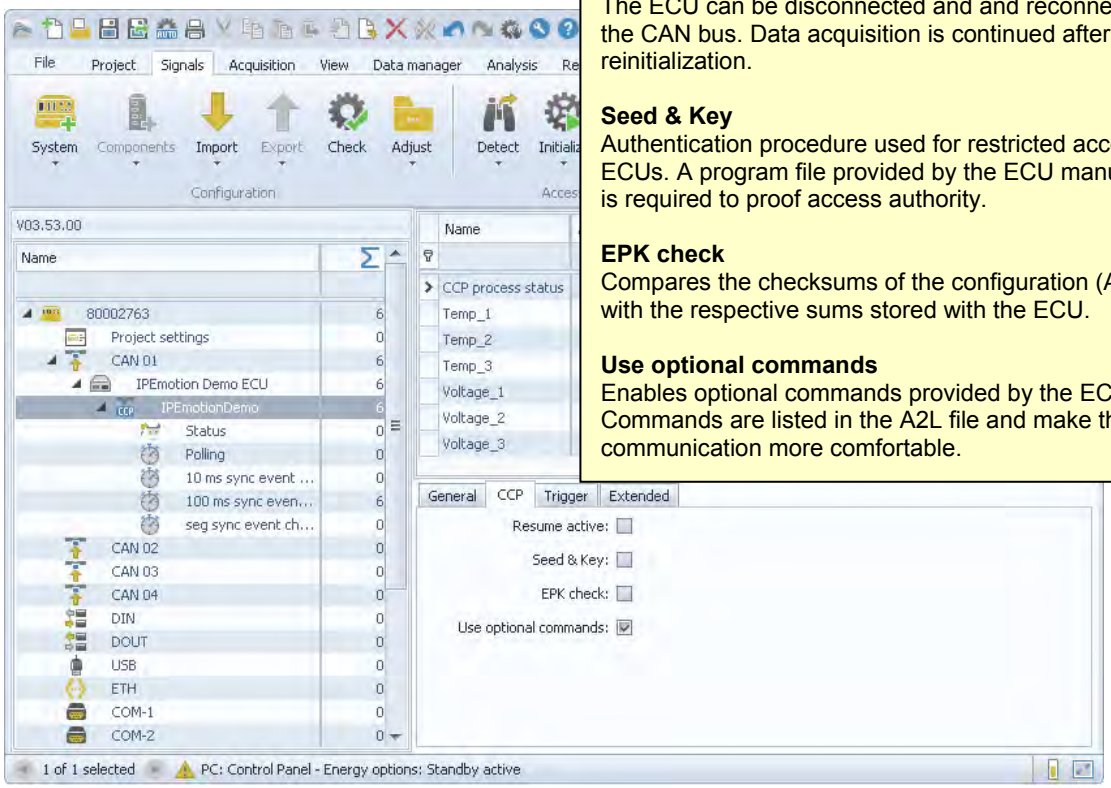



## Import signal descriptions from A2L




## CCP protocol settings




**Resume active**
The ECU can be disconnected and and reconnected to the CAN bus. Data acquisition is continued after reinitialization.

**Seed & Key**
Authentication procedure used for restricted access to ECUs. A program file provided by the ECU manufacturer is required to proof access authority.

**EPK check**
Compares the checksums of the configuration (A2L file) with the respective sums stored with the ECU.

**Use optional commands**
Enables optional commands provided by the ECU. Commands are listed in the A2L file and make the communication more comfortable.

*You need to copy the Seed&Key file which relates to the ECU to one of the these subdirectories. If the file is missing the authentication procedure fails.*

*C:\Users\Public\Documents\IPETRONIK\IPEmotion\Import*
*C:\Users\Public\Documents\IPETRONIK\IPEmotion\Custom\SeedAndKey*

### Import PDX file

In order to align the version of the A2L file with the contents stored in the ECU, several PDX files can be imported at a single CAN knot. Each ECU requires its corresponding PDX file.

The PDX file contains all necessary information to request program and data version from the ECU. After the import of this information, the logger reads the corresponding data from the ECU using the UDS protocol. A matching algorithm compares the contents of the relating *.a2l file with the data read from the ECU.

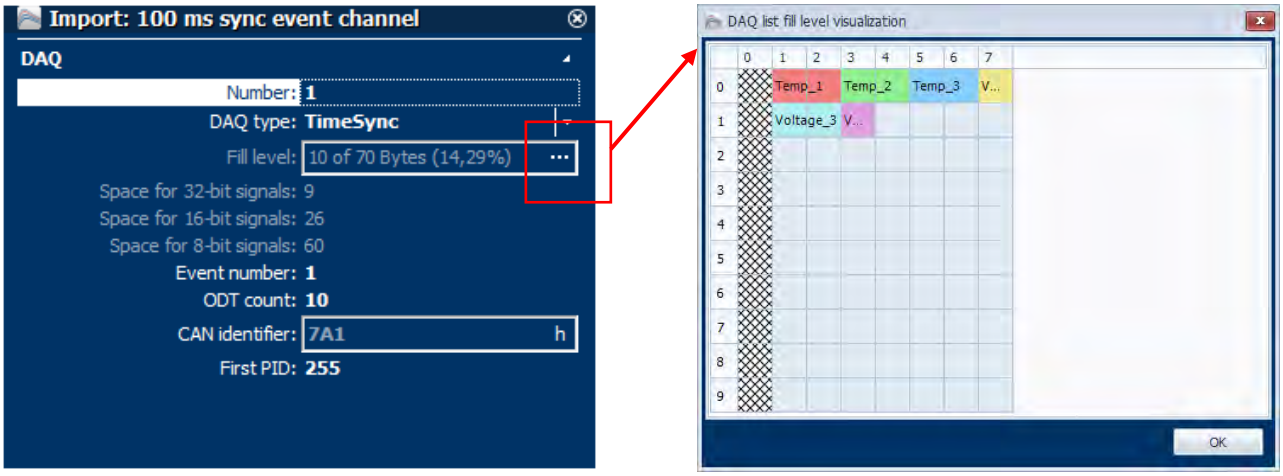
Depending on the result of the comparison, measurement will be resumed, stopped totally or partial, e.g. with protocol measurements with a CAN, FlexRay or Ethernet ECU.

Program an data version information imported from the ECU will be stored in the log file and the measurement status file. Tasks that require a Java script are not implemented.

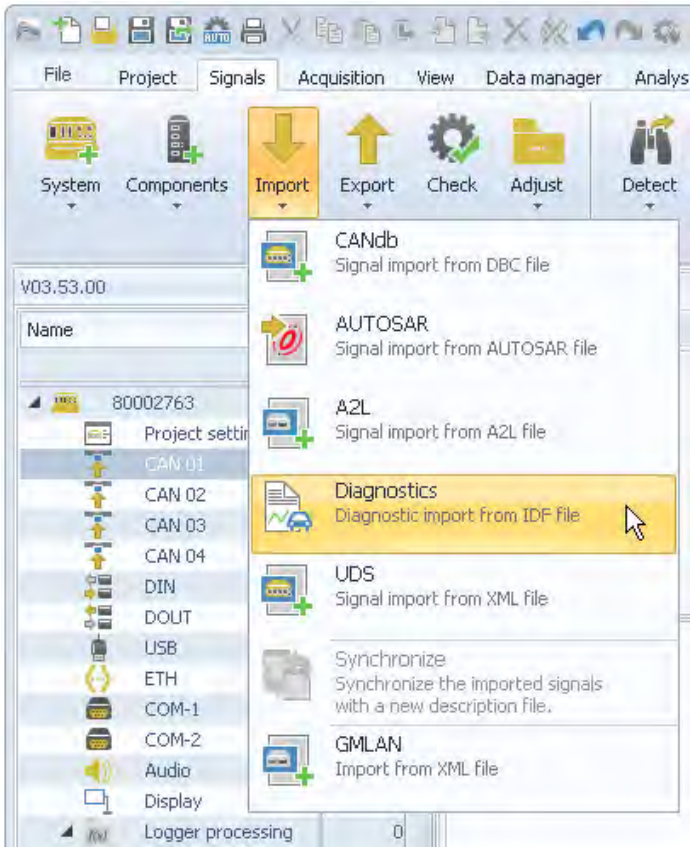
Supported physical connection are: Diagnostic CAN, OBD2 connector, standard CAN tapping point.

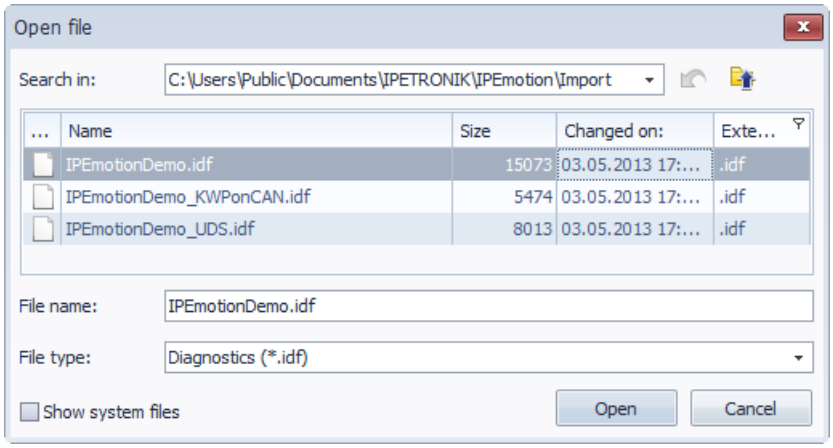
### Import properties – DAQ list fill level (numerical, graphical)

Highlight the respective DAQ list (e.g. 100 ms sync event channel) and right click with the mouse to open the context menu, select **Import properties**.



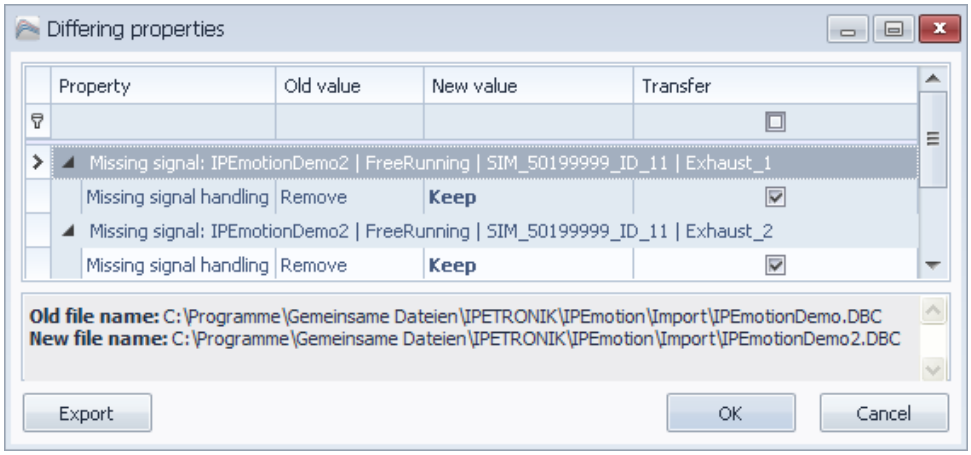
## Import Diagnostic file



## Description file synchronizing

The synchronization feature provides the possibility to compare existing signal descriptions (imported previously) with the respective entries of a modified description file.




**Possibility 1: Channels within a description file have been modified**

Previously a description file has been imported and signals have been assigned to the CAN measuring input. Now, a new description file containing modified scaling settings of several signals should be used.

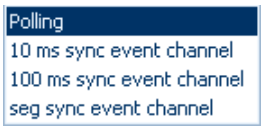
Highlight the description file at the CAN input in the left hand system structure and select **Import** > **Synchronize** from the main menu or from the context menu. Search for the modified description file on your computer and confirm your selection with **OK**. The new settings will we used for the current configuration.

**Possibility 2: Channels have been added to a description file**

Previously a description file has been imported and signals have been assigned to the CAN measuring input. Now, a new description file containing additional channel settings (signals have been added) should be used.

Highlight the description file at the CAN input in the left hand system structure and select **Import** > **Synchronize** from the main menu or from the context menu. Search for the modified description file on your computer and confirm your selection with **OK**. The CAN input will be synchronized with the new settings. Even if you open the import module again, new channels will be displayed in the left hand area. Select the channels to import the related settings into the current configuration.

## 8.2.2 Detection mode and cycle rate




### Polling

Polling functionality is supported by A2L for CCP and XCP, i.e. no DAQ lists are required.

IPEmotion offers three different speeds for data requests. The times are reference values because the values must be individually requested at the control unit and create a high bus load.

SLOW 1000 ms MIDDLE 100 ms FAST 10 ms

Example: If 5 values are detected in the SLOW polling mode, the 1st value is requested by the control unit a 2nd time after approx. 6 s.

***DAQ list or polling?***

*Communication via CCP protocol takes place by sending the data cyclically or triggered by an event after the first inquiry. The send rate is defined in the respective DAQ list, e.g. 10 ms, 100 ms or synchronous to an event. Not all control units support this mode or the DAQ lists are not included in the A2L file. In this case, the signals can be sampled individually with the polling mode. But this method causes a much higher bus load and therefore longer response times.*

## 8.2.3 Traffic acquisition

The CAN traffic acquisition (also CAN-Trace) allows high-capacitive recording of CAN and/or LIN messages (total bus traffic) with the CAN/LIN controller of each measuring input. Due to the fact that a huge amount of data can result from this method within a short time, data acquisition can be restricted with definable ID triggers (measuring input) and admission filters (traffic group).

Using the tab **Triggering** a trigger condition can be defined for each traffic group to control data recording individually.

### Filters and triggers

The **Traffic id trigger** at the respective CAN input of the system structure provides 6 individual ID triggers. As soon as at least one of the trigger conditions is fulfilled, traffic recording is started (logical OR function). Without any Traffic id trigger, traffic recording is started immediately after measurement start. Each id trigger can consist of a single identifier or an identifier range (First/Last CAN ID). Within the trigger definition, each of 8 bytes from the CAN message can be compared with a reference (supported operators: =, <, >, < =, > =, < >). Ony if all conditions defined by the tab ID-Trigger are fulfilled, the respective ID-Trigger becomes active (logical AND function).

Status channels of Traffic id triggers can be used in calculations, e.g. only if a specific id trigger indicates status 1, traffic recording is started. Without any use of the **Traffic trigger status**, activate the checkbox **Trigger direct** under the tab **ID-Trigger** to start the traffic acquisition properly.

### Data format and conversion

Traffic data is saved in a binary file with header (description) and the actual data, e.g. TD001234.bin. To generally use the data (e.g. import in CANalyzer), data is converted into the ASCII format with the data converter.

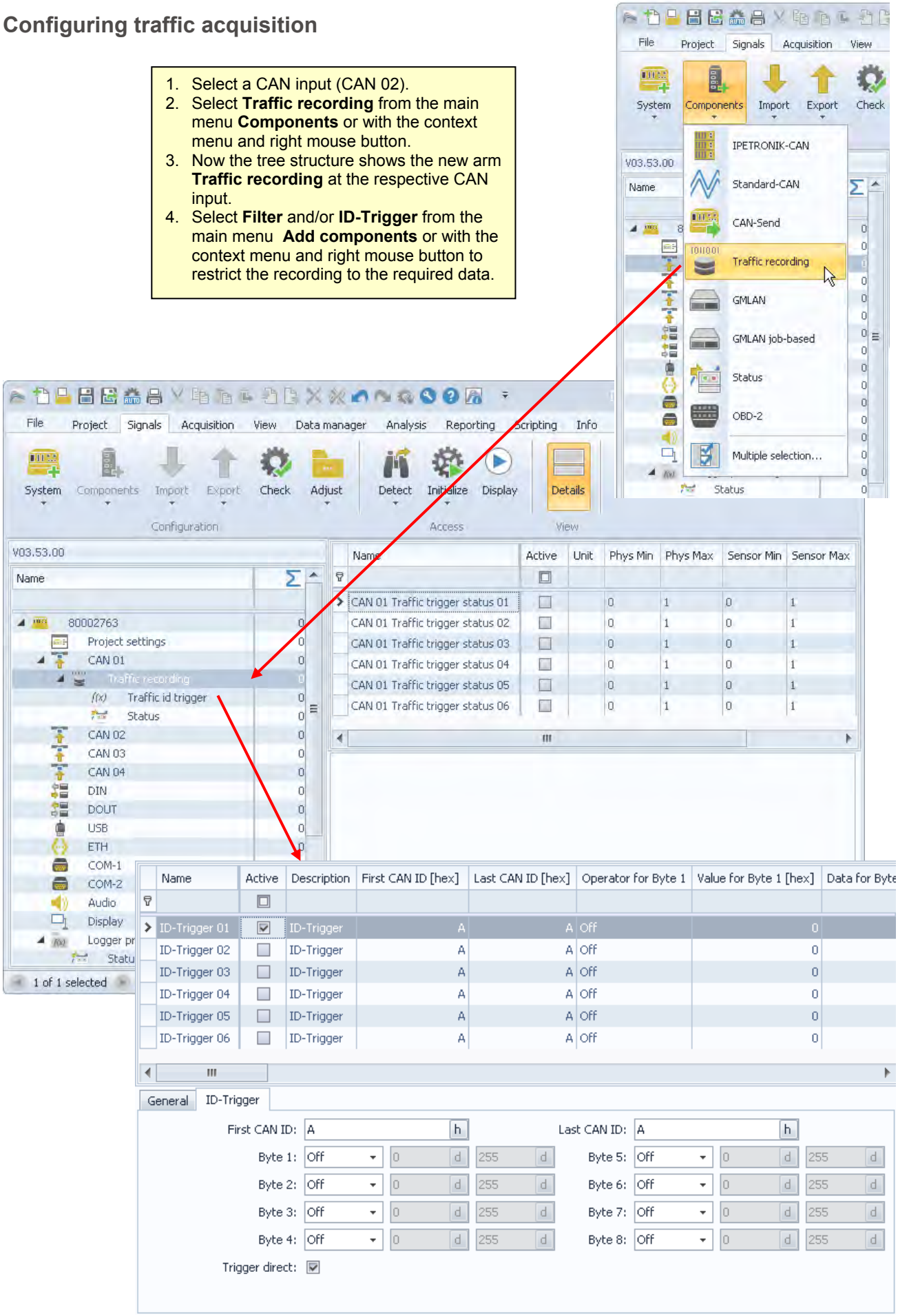
Use the IPETRONIK data converter version 2.xx IPEconverter WIN with graphical user interface (requires a license ) or the version with the command line IPEconverter CMD (no license required).

The software, as well as, the documentation are saved at the IPETRONIK CD:

...\IPETRONIK_SoftwareProducts\Tools\DataConverter\…

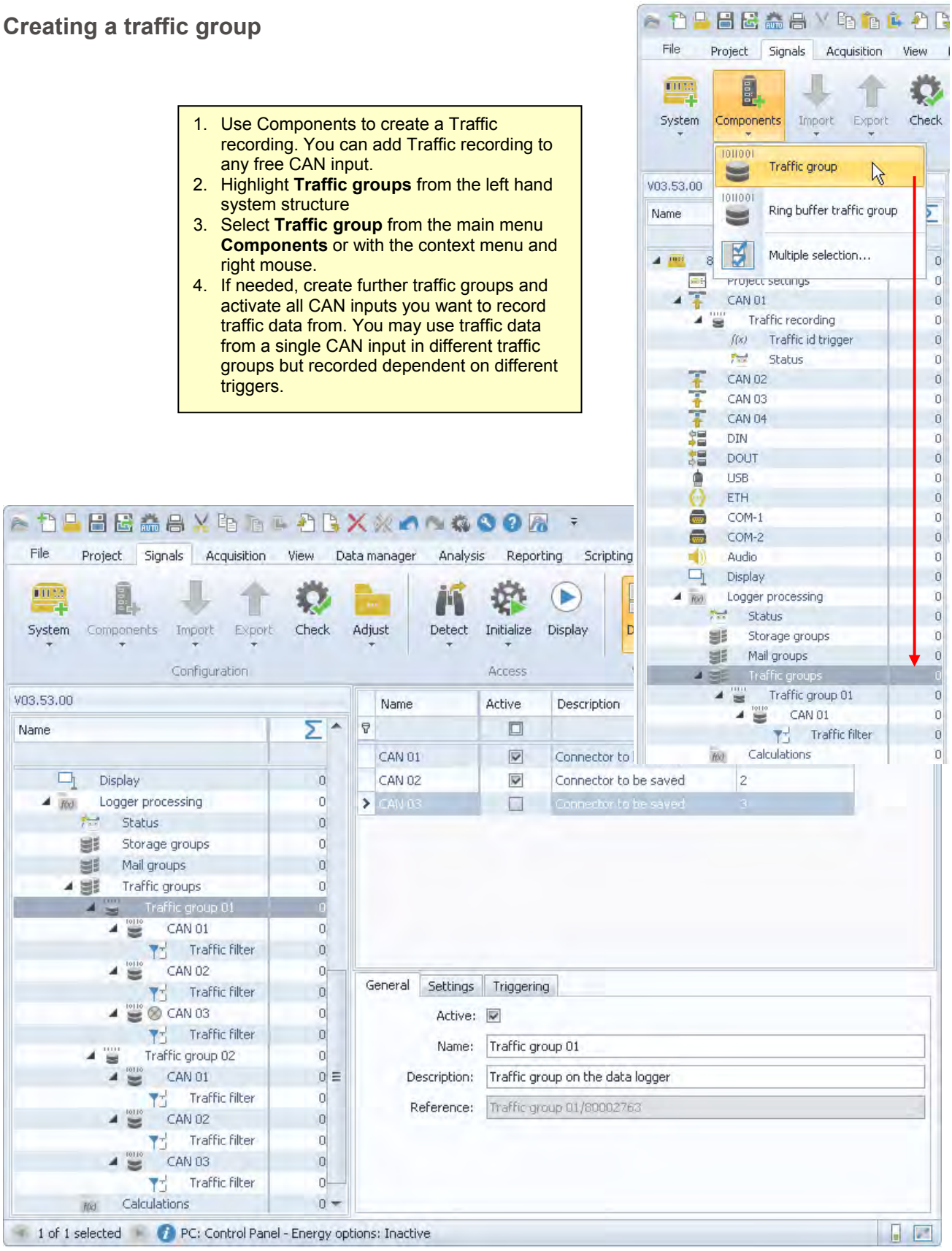
## Configuring traffic acquisition

1. Select a CAN input (CAN 02).
2. Select **Traffic recording** from the main menu **Components** or with the context menu and right mouse button.
3. Now the tree structure shows the new arm **Traffic recording** at the respective CAN input.
4. Select **Filter** and/or **ID-Trigger** from the main menu **Add components** or with the context menu and right mouse button to restrict the recording to the required data.



## Creating a traffic group

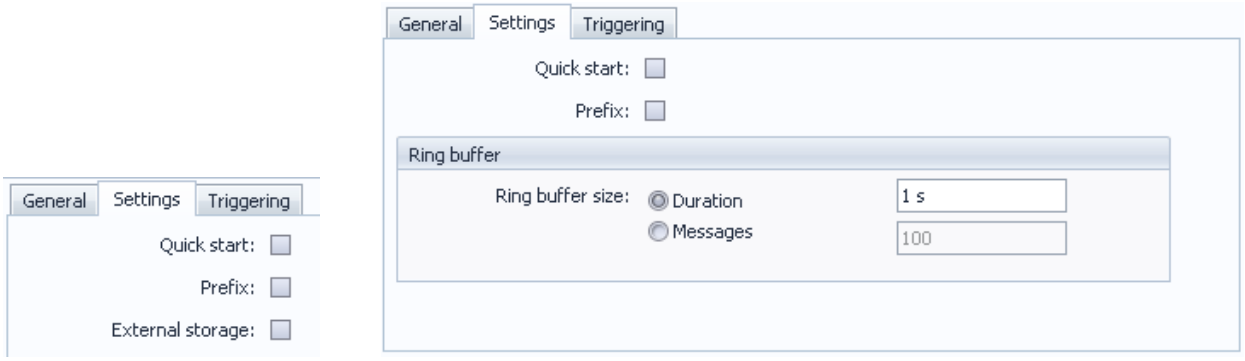
1. Use Components to create a Traffic recording. You can add Traffic recording to any free CAN input.
2. Highlight **Traffic groups** from the left hand system structure
3. Select **Traffic group** from the main menu **Components** or with the context menu and right mouse.
4. If needed, create further traffic groups and activate all CAN inputs you want to record traffic data from. You may use traffic data from a single CAN input in different traffic groups but recorded dependent on different triggers.



## Traffic group and ring buffer traffic group settings

**Traffic group**

**Ring buffer traffic group**



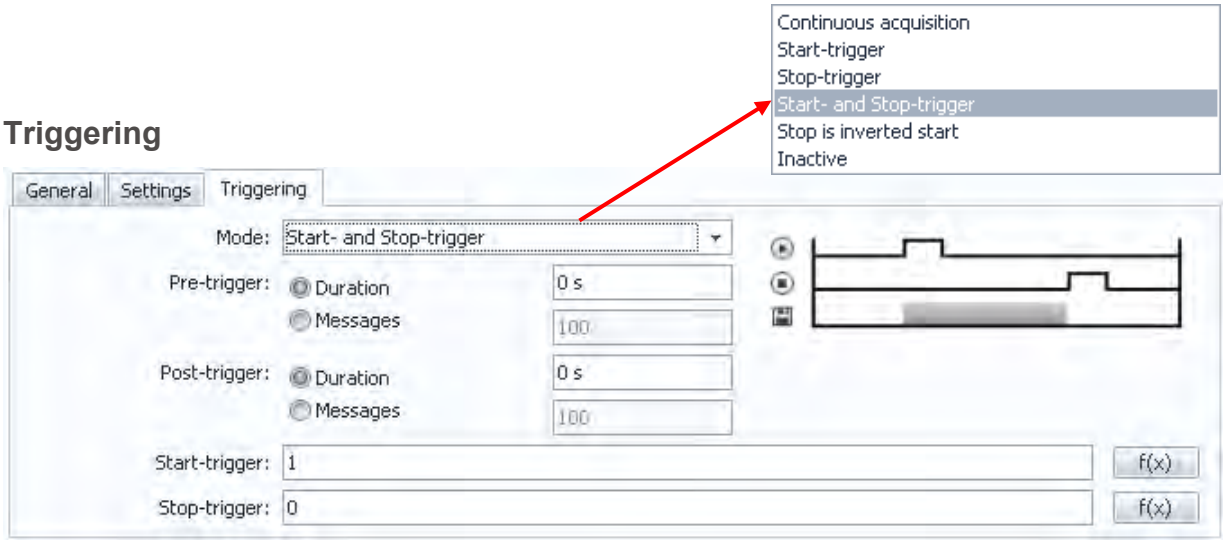

**Quick start** Quick start data (which is recorded during logger boot-up) is identified by a negative time stamp within the traffic data file.

**Prefix** The storage group name ist used for the measurement data file name.

**External storage** Data storage on external USB storage medium.
Refer to ***6.5 USB medium for external storage*** for detailed information.

**Ring buffer size** Defines the size of the ring buffer for measuring. If the max ring buffer size is reached, the memory is overwritten beginning with the oldest data.

## Triggering




**Mode** Select the trigger mode (depending on the traffic group) from:
Continuous acquisition (no trigger), Stop is inverted start
Traffic group only:
Start-trigger, Stop-trigger, Start- and Stop-trigger (see Triggering)

**Pre-trigger duration** Data recorded before the trigger event (duration = seconds or messages)

**Post-trigger duration** Data recorded after the trigger event (duration = seconds or messages)

**Start-trigger** Value, which starts the data storage of the respective group.

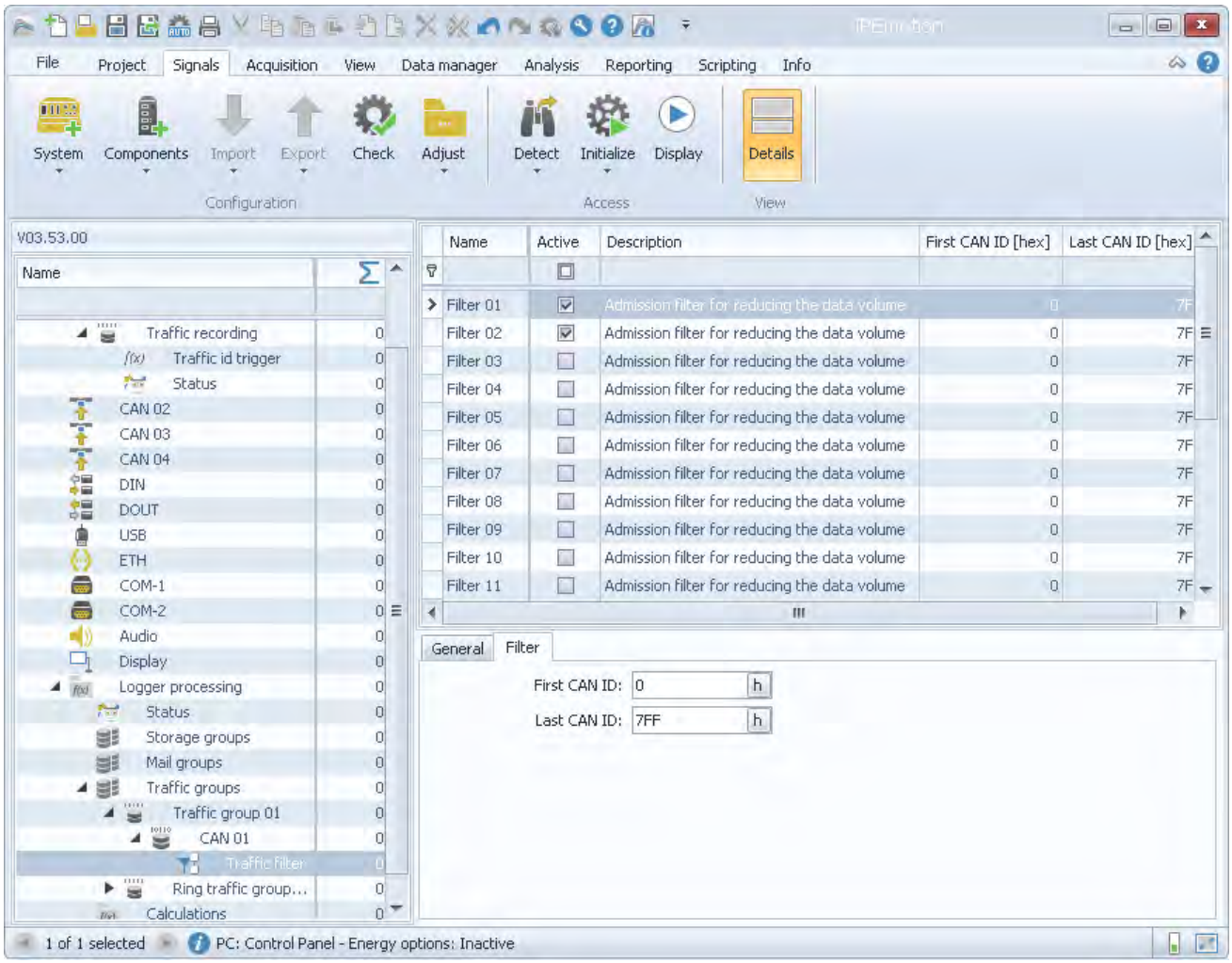
**Stop-trigger** Value, which stops the data storage of the respective group.

*Using the Start-Stop condition, storage data will not be saved if the stop condition is already fulfilled at measurement start!*
*We recommend to define suitable events used for start- and stop-trigger. To guarantee acquisition start (data storage) in any case, you can set „1“ as start-trigger condition.*

*To evaluate quick start data in relation to cyclic data of a storage group, you have to activate **Time stamp channel** (for the absolute time) from the settings tab of the respective storage group.*

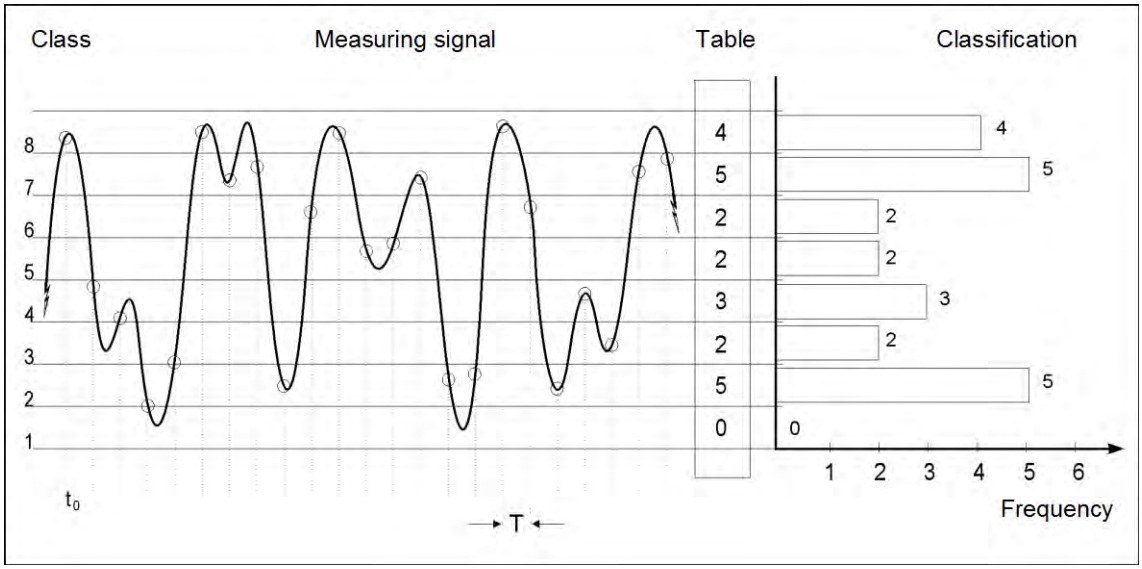
## Filtering

User definable pass through filters enable the filtering of defined areas out of the data stream. This provides an ID filtering like: Traffic group 01 using ID 100[hex] while Traffic group 02 uses other filters or no filters.



## 8.2.4 Statistics calculation

The frequency of signals is counted and evaluated with classifications. To do so, the acquisition range is divided into equal (equidistant) zones (classes). The current values is assigned to one class at every sampling and the frequency is counted (see figure).




| Advantages | Disadvantages |
|---|---|
| Much less storage required than with a time related acquisition | No original value |
| Very suitable for statistical analyses (e.g. life cycle tests) | No time reference of the signal |

### Classification methods

Different classification methods have been developed in the past whereof IPETRONIK supports the most common methods (acc. to DIN 45667, FVA sheet):

- Random sampling counting
- Level crossing counting
- From-To counting
- Edge counting
- Rainflow method (available upon request)

Please find further information in the Classification.pdf document (Classifying with KIM/KAR and DIS) on the IPETRONIK CD or at the FTP info server.

### Requirements

| | |
|---|---|
| **Hardware** | M-LOG, S-LOG, FLEETlog, IPElog |
| **Configuration** | IPEmotion |
| **Software** | TESTdrive V03.06 or V03.18 (IPEmotion) or V03.50 (IPElog) or higher |

*It is recommended to use the latest software version for guaranteeing a clean functionality of all components..*

## Configuring statistic / classification

**1st Step** **Creating configuration**

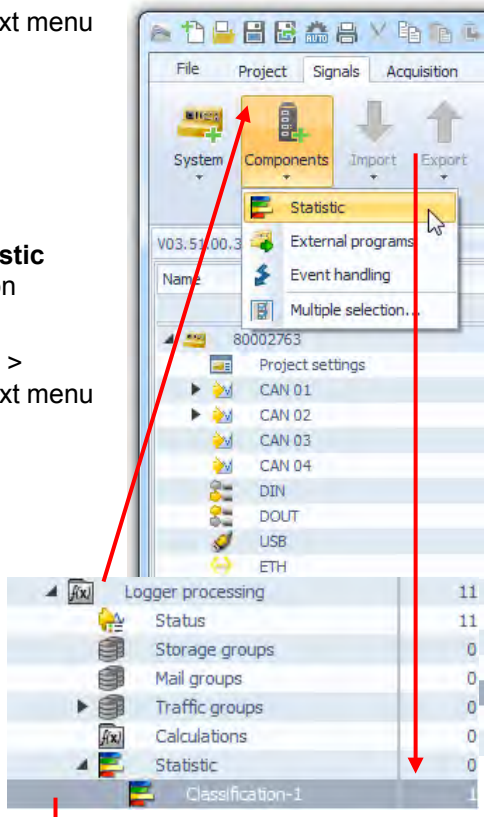
- Start IPEmotion and load an existing configuration or create a new configuration.
- Import the corresponding signal descriptions (CANdb or ASAP2) and/or configure additional IPETRONIK devices.
- Activate the desired inputs and run the required scaling.
- Select a sampling rate or accept the default sampling rate. Please note that the sampling rate, which is selectable in the classification, cannot be higher than the maximum sampling rate.

**2nd Step** **Defining storage group** **(if time related data is also required)**

- Create a new storage group with **Storage groups** and the context menu **Add components** to also record the time related signals.
- Assign the desired signals to the corresponding storage group.

**3rd Step** **Defining classification**

- Select **Data processing** in the system structure and select **Statistic** from the main menu **Add components** or with right mouse button > context menu.
- Select **Statistic** in the system structure and select **Components** > **Channel** from the main menu or with right mouse button > context menu to create channels.
- Define additional classification settings within the tabs **General**, **Settings**, and **Trigger**.




**Reset behaviour** Data is written into a new classification file at changing the configuration or at starting acquisition.

**Sampling rate** Data storage rate of classification

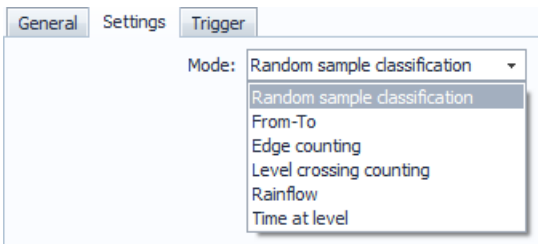
**Trigger (Statistik)** General trigger, defines start- and stop-trigger, value range 0 / 1

**Classification**

**Name** any name for classification

**Mode** selected classification method

**Trigger** defines start- and stop-trigger of the active channel (classification start and stop



### 8.2.5 Operating in FTP mode (terminal server)

TESTdrive version 3.09.00 allows starting M-LOG as FTP server. Data can easily be transmitted with a FTP software (e.g. Total Commander or WS_FTP). Depending on the user rights, data can additionally be deleted or written. A separate USB flash drive incl. TESTdriveCmd.xml file is required for this functionality. The running acquisition is stopped and the log file saved with connecting the USB.

If the TESTdriveCmd.xml file includes the "StartFTPServer" OnConnect job, there is no post processing. TESTdrive reads the corresponding parameters and starts the FTP server.

Server access requires the following user data:

| Version | User | Password | Access rights |
|---|---|---|---|
| V03.09.00 | guest | none | Read to TO directory |

A reboot is automatically running at connecting USB flash drive to correctly stop the service as FTP server.

#### Procedure:

1. Switch-on M-LOG
2. Connect USB flash drive with TESTdriveCmd.xml
3. Connect Ethernet cable between M-LOG and PC, e.g. 600-591.xxx (M-LOG PR05, S-LOG)
4. Configure network settings of PC, create additional "Alternative configuration"

   User defined

   IP address: 192.168.0.1 (Example)

   Total Commander settings: Server name: 192.168.0.2, Enter user name and password

### 8.2.6 Recording audio and video data

#### Recording video data

Data logger supports recording video or single shots with a camera. The camera is connected to the logger USB port.

Following settings are available:

| | | | |
|---|---|---|---|
| Resolution | Defines the image quality | | |
| "Low" | | 160 x 120 Pixel (B x H) | 30/20/10/5 images per second |
| "Standard" | | 320 x 240 Pixel (B x H) | 30/20/10/5 images per second |
| "High" | | 432 x 240 Pixel (B x H) | 30/20/10/5 images per second |
| "Ultra" | | 640 x 480 Pixel (B x H) | 30/20/10/5 images per second |
| Max. recording time | Defines the recording duration | | |
| Frame rate | Defines the number of images per second | | |

All three options directly influence the required memory.

| | |
|---|---|
| Trigger | A trigger condition must be defined to start recording. |
| Trigger mode | Triggering to raw data or scaled values |
| Raw | Triggering to raw data |
| Phys. | Triggering to physical values (acc. to scaling) |

## Recording audio data

Data logger (M-LOG, S-LOG) supports recording audio signals (sounds, speech) with a microphone at the audio input.

Following settings are available:

| | |
|---|---|
| Bit rate | Defines the audio signal quality<br>22050 Bit/s (FM radio), 11025 Bit/s (AM radio), 8000 Bit/s (Phone quality) |
| Max. recording time | Defines the recording duration |

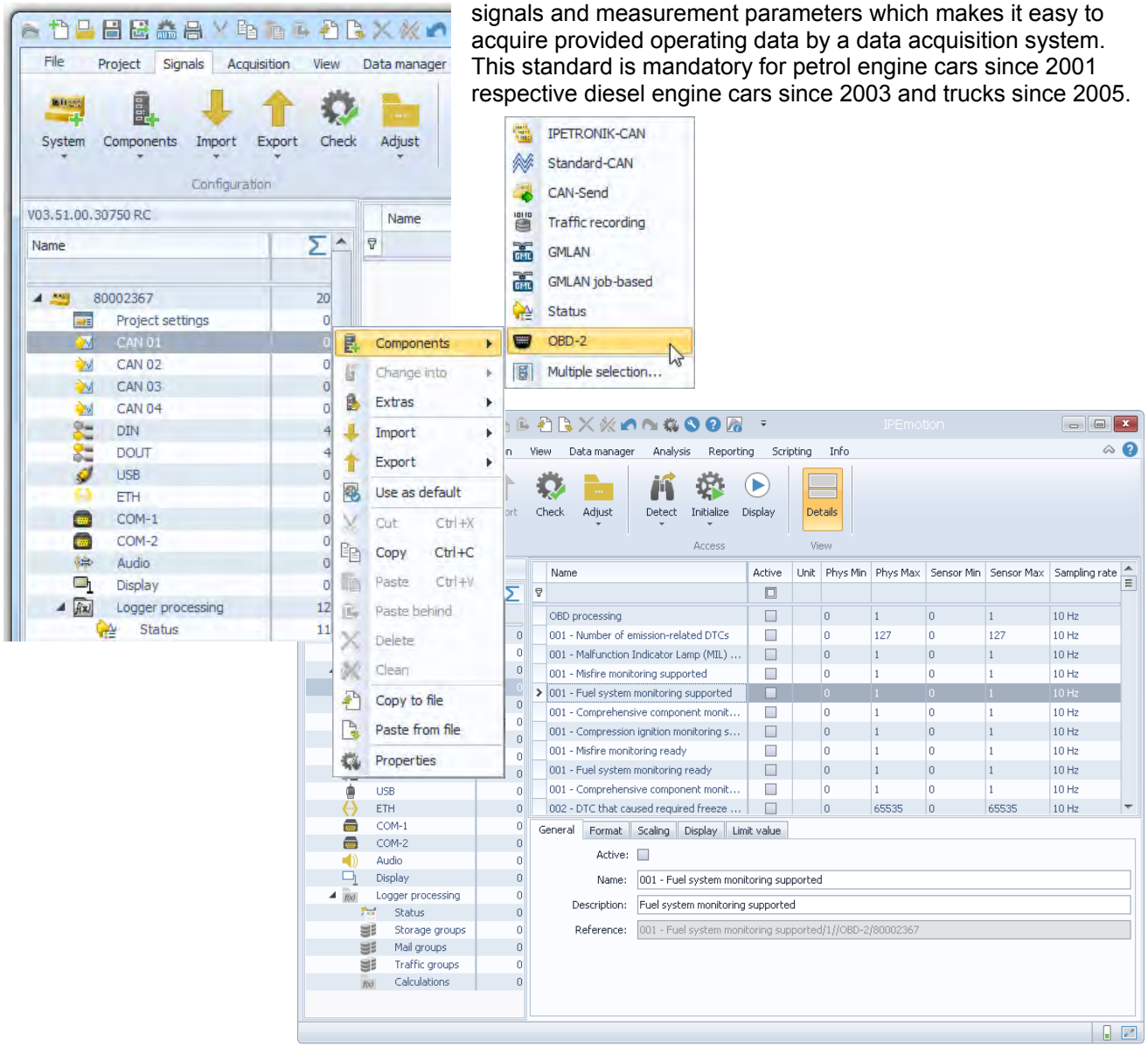
Both options directly influence the required memory.

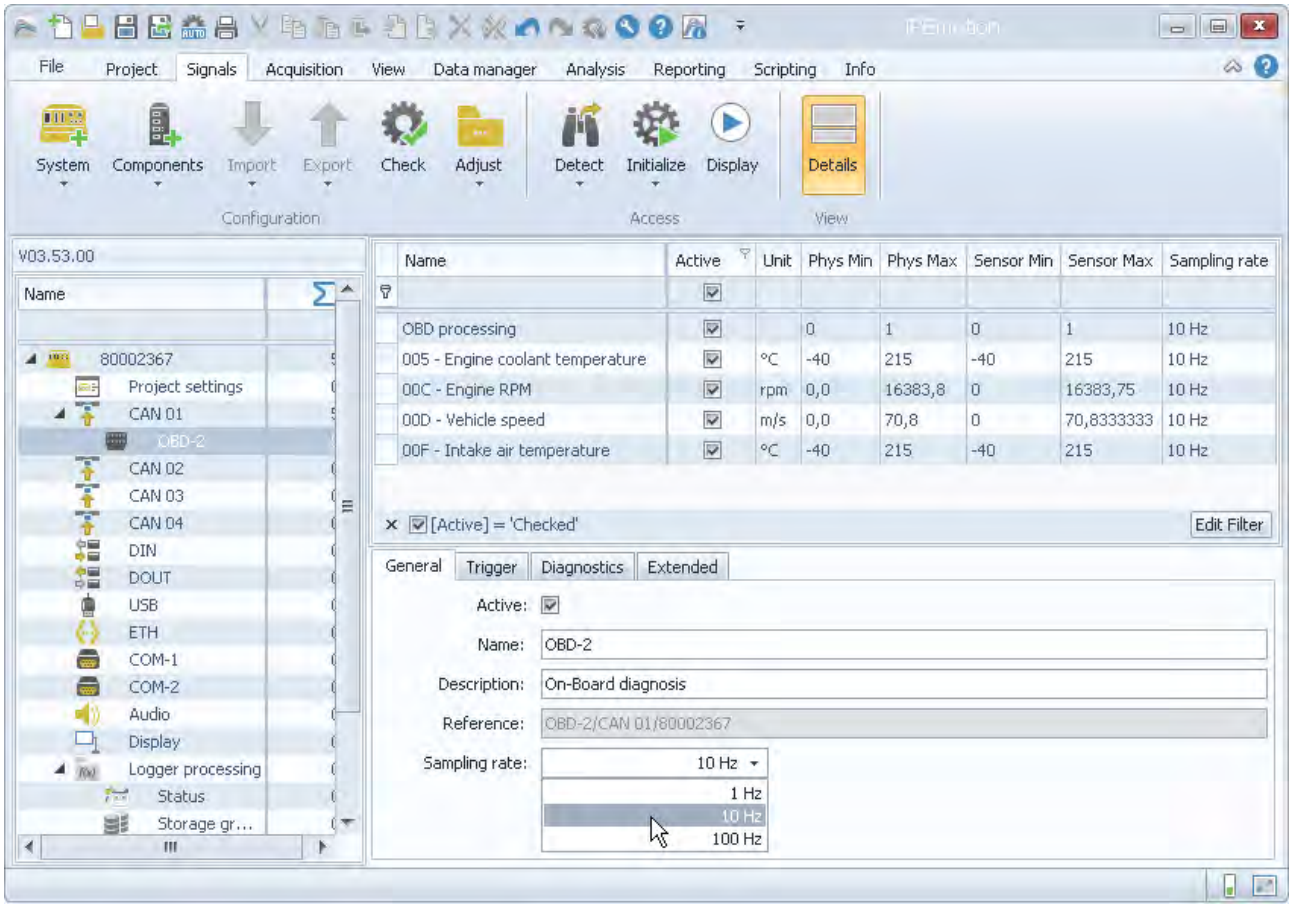
| | |
|---|---|
| Trigger | A trigger condition must be defined to start recording. |
| Trigger mode | Triggering to raw data (Raw) or scaled values (Phys) |

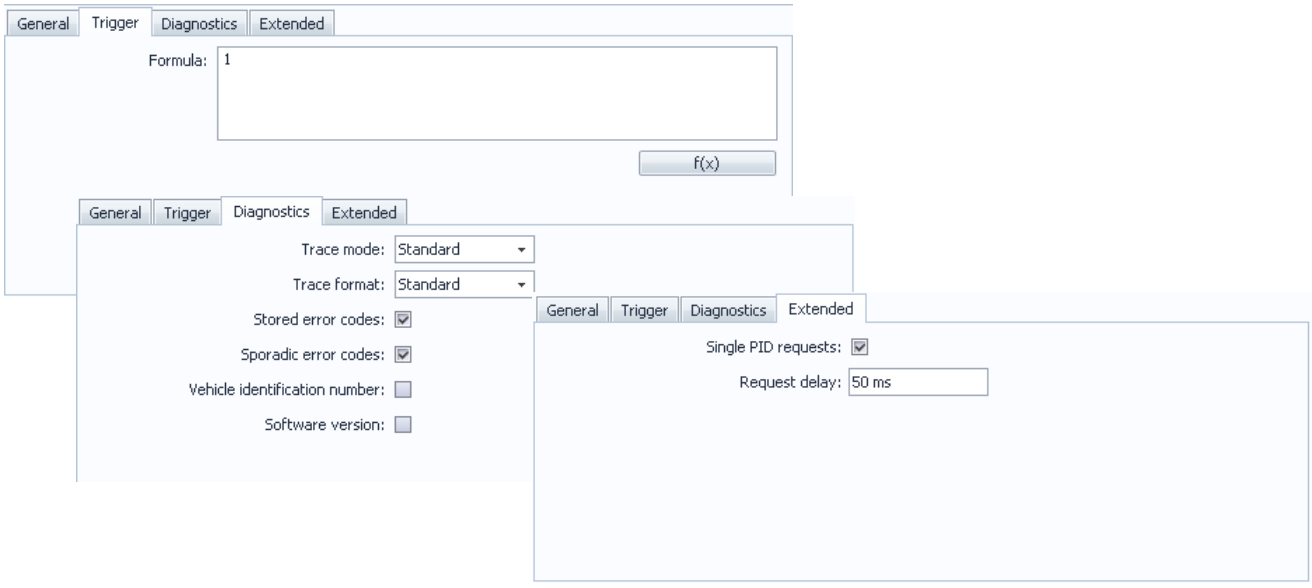
## 8.2.7 OBD-2 measurement

With releases of the PlugIn IPETRONIK-LOG / TESTdrive ≥ V03.22 data acquisition using the OBD2 standard (CAN bus) supported.

The OBD-II international standard details a list of 96 predefined signals and measurement parameters which makes it easy to acquire provided operating data by a data acquisition system. This standard is mandatory for petrol engine cars since 2001 respective diesel engine cars since 2003 and trucks since 2005.






## OBD-2 Extensions (TESTdrive V03.52)

- Selectable data rate 1/ 10/ 100 Hz (depends on the ECU)
- Measurement start by trigger event
- Support of Extended IDs (29 Bit)
- PID single request support (each request to the ECU with a PID)
- Request delay (delay time between data receipt and sending the following request)

## 8.2.8 UDS protocol (Unified Diagnostic Services)

UDS protocol combines KWP2000, GMLAN, and DiagnosticOnCan in one protocol. An advantage is the clear session handling (higher compatibility of different control units). Furthermore, UDS supports modern memory structures, which require a > 32 Bit addressing.

The corresponding description file has the ODX extension.

Control units of some manufacturers already support UDS, which will be used as standard diagnostic in near future.

### Jobs overview

TESTdrive V03.15 supports the following jobs, which can be read via UDS:

- FS_READ
- FS_READ_DETAIL
- IDENT
- READ_DATA_REFERENCE
- FG_READ
- DYNAMICALLY_DEFINE_LOCAL_ID

These jobs can be defined with an *.idf file.

### Storing data

The results are optionally stored as binary file (*.CSV and *.J**) or as trace and binary file (*.CSV, *.J** and *.T**).

The files are identified as follows:

| | |
|---|---|
| Single data detected via KWPonCAN:<br>(former description) | BDKxxxx.CSV bzw. BDKxxxx.Jxx<br>BDJxxxx.CSV bzw. BDSxxxx.Jxx |
| UDS data detected via trace mode: | BDUxxxx.txx |

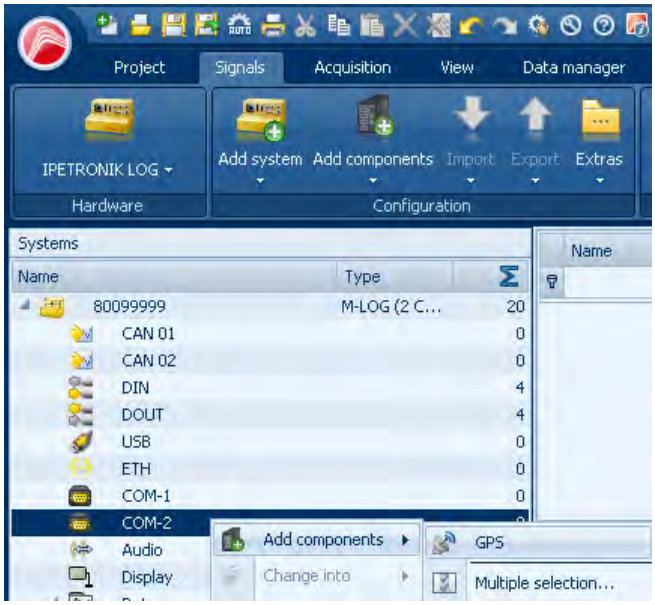
### Selecting protocol

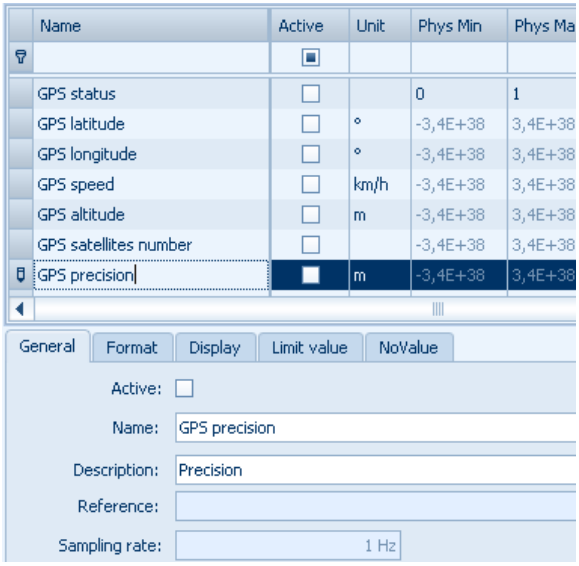
The UDS protocol is selected with the corresponding tab in the import module.

*Read more details on the UDS protocol and relating applications from this document:*
***Manual ECU Diagnostics.pdf***

## 8.3 Recording GPS-Data

The GPS receiver GPS 18 and the NMEA protocol option, M-LOG, S-LOG, as well as, FLEETlog WAN with integrated GPS receiver support the continuous recording of GPS data with the global satellite navigation system. This functionality allows positioning and logging of  test routes with a data rate of 1 Hz.




Predefined settings are available to configure the acquisition. The single channels are activated as required.

*The accuracy of the positioning data is considerably defined by the number of received satellites (12 satelites are in the geostationary orbit).*

*Due to physics, the accuracy of the height acquisition (Altitude) with this method is considerably lower than that of the length acquisition (Latitude = geographical width, Longitude = geographical length).*

*Different altitude signals with FLEETlog STD and FLEETlog WAN.*

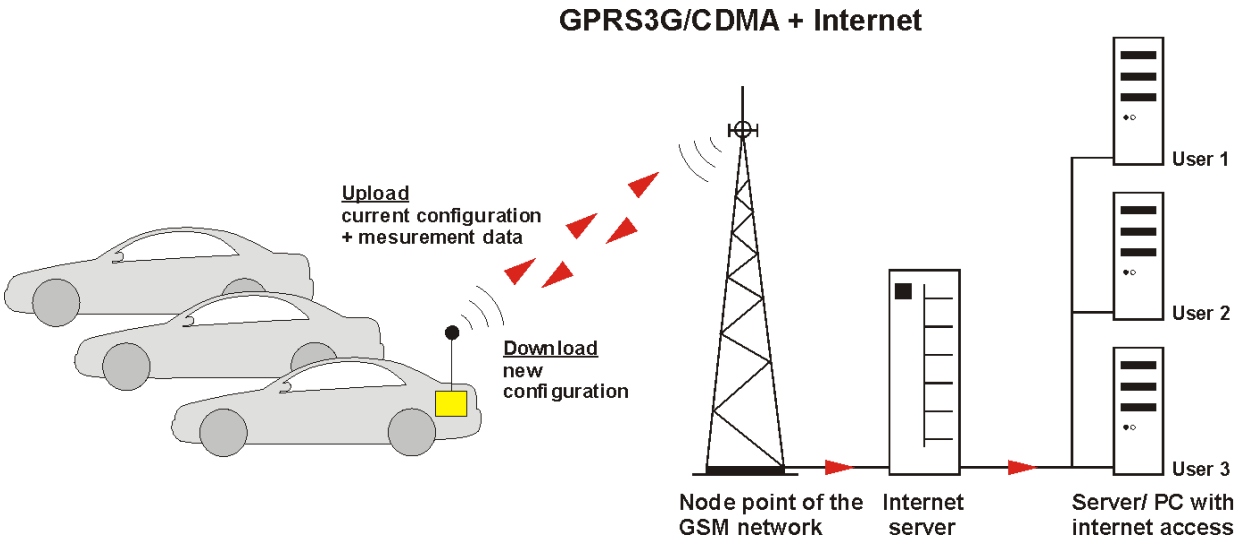
*Depending on the GPS hardware of FLEETlog, the logger acquires altitudes which may differ from the reference height (geographic NHN). This behaviour is caused by the use of the "ellipsoidal height" for the height reference.*

*Please execute a calibration measurement on a heigh which is known exactly to verifiy GPS data of your logger system.*

## 8.4 Remote data transfer

The logger offers the availability of wireless data transfer by corresponding options. Single vehicles, as well as, entire vehicle fleets can be managed from one or several bases. Due to the worldwide good to excellent GSM network coverage, regional and global test drives can be managed from any base.

### 8.4.1 Transferring data using GPRS and Internet to FTP server



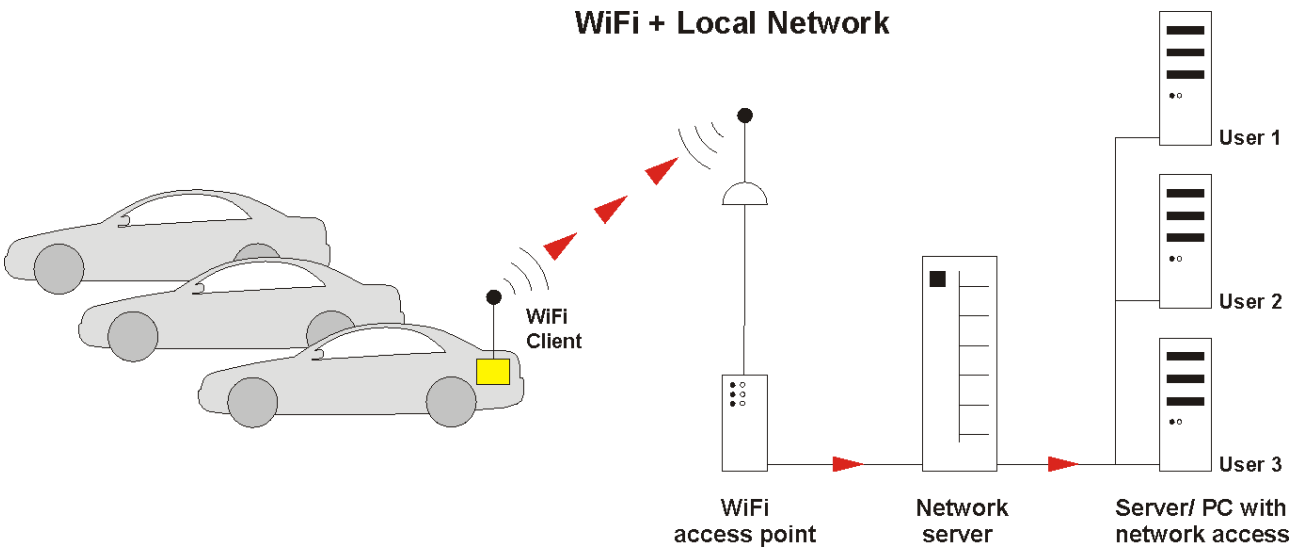

#### Components

- M-LOG/M-LOG V3, S-LOG, FLEETlog2, IPElog/IPElog2
- M-LOG, S-LOG with GPRS data transfer option with COMgate WAN, antenna, data transfer software or with modem, antenna, data transfer software
- Connecting cables
- SIM card for modem (depending on provider)

#### Functional principle

A logger in measuring mode continuously stores data as defined in the configuration. If the test series is completed (status of remote signal is inactive, e.g. terminal 15), data is packed and transferred via GPRS in GSM network to the next node point. This data is then transferred via internet to a FTP server for being available for download. All data transfer settings are defined in the IPETRONIK software.

## 8.4.2 Transferring data using Wireless LAN to netzwork server



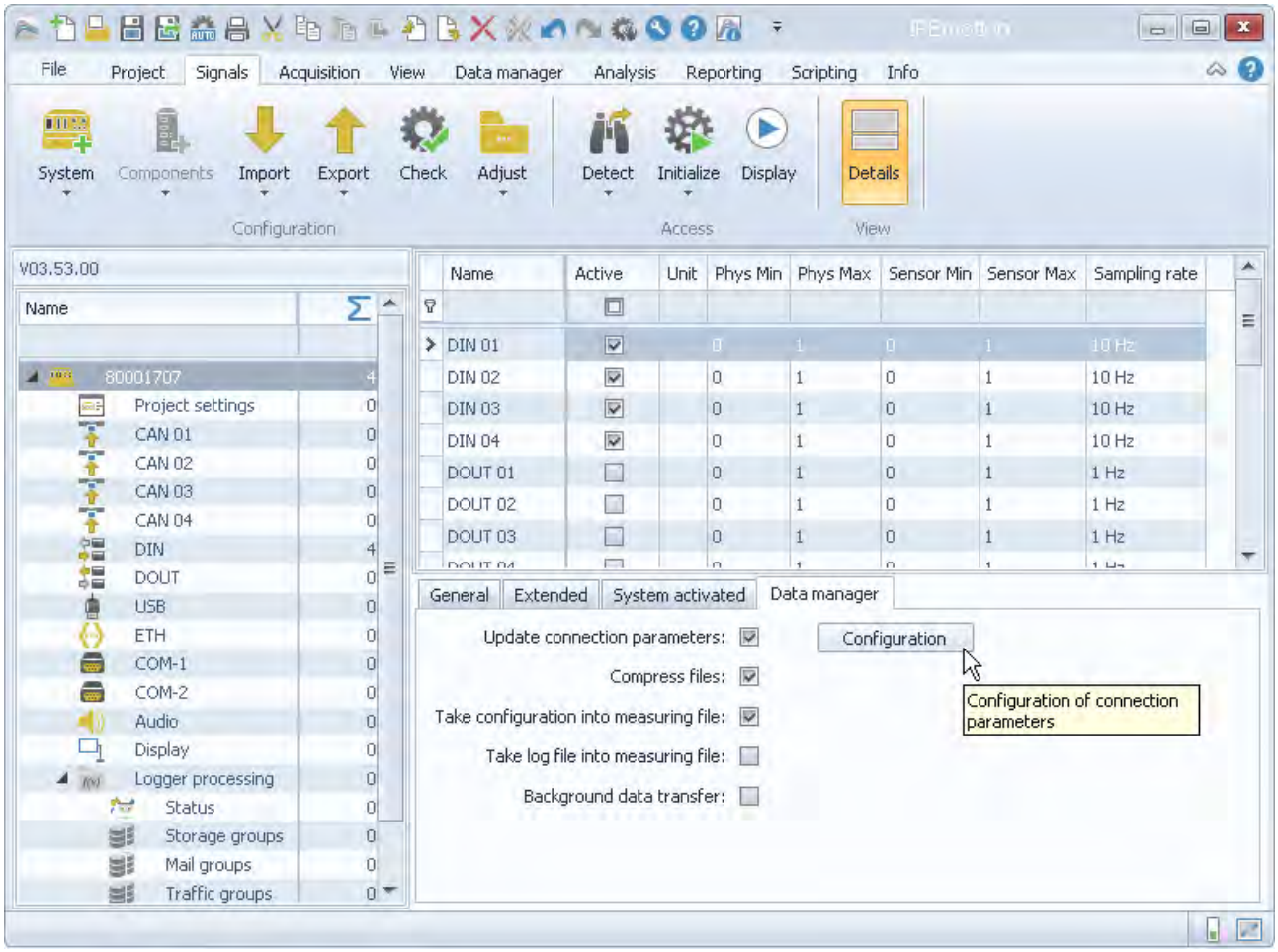

### Components

- M-LOG/M-LOG V3, S-LOG, FLEETlog2, IPElog/IPElog2
- M-LOG, S-LOG with WiFi data transfer option with COMgate, antenna, data transfer software or with client, antenna, WiFi software
- Connecting cables
- WiFi access point to connect with network (M-LOG, S-LOG)

### Funktional principle

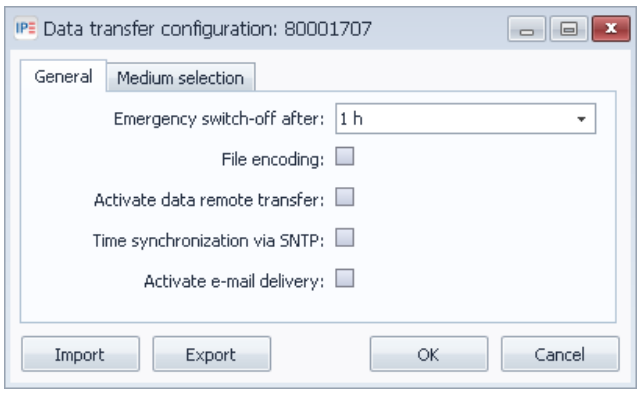
A logger in measuring mode continuously stores data as defined in the configuration. If the test series is completed (status of remote signal is inactive, e.g. terminal 15), data is packed. If the vehicle is within reach of an access point (up to 300 m outdoors), data is transferred via WiFi to the access point. This data is then transferred via local network to a server. If the local network is connected to the internet, data can also be transferred to a FTP server for being available for download. All data transfer settings are defined in the IPETRONIK software. Due to multiple encoding options (transmission protocol and user), a very high degree of security against unauthorized access is guaranteed.

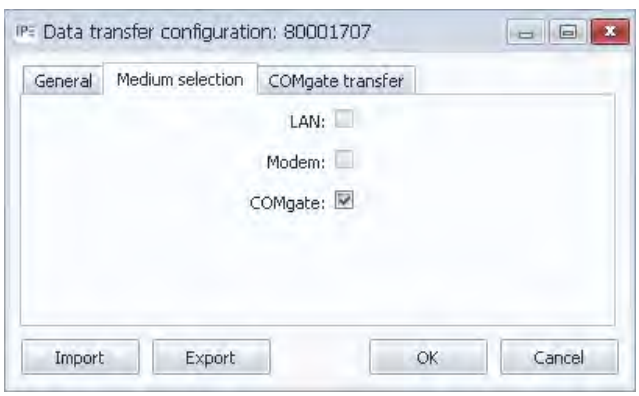
## 8.4.3 Data transfer configuration




*Activate **Update connection parameters** to enter the data transfer settings using the **Configuration** button.*

*To ensure an accurate data transfer after TESTdrive update to 3.52.01 or higher, you have to click this check box also, even if no modification has been made. (The current settings will be transferred to the logger in any case.)*



## 8.5 Configuring COMgate

**COMgate transfer**

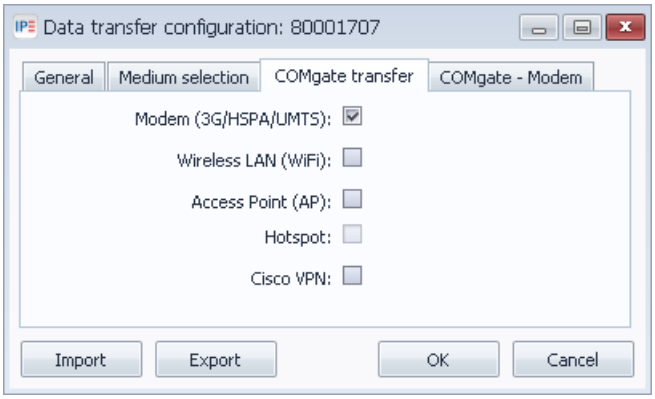
**Modem (3G/HSPA/UMTS)**
Data transfer by internal modem (COMgate WAN only).

**Wireless LAN (WiFi)**
Data transfer by internal WiFi client.

**Access Point (AP)**
COMgate operates as WiFi access point.

**Hotspot**
Hotspot (public WiFi access point, e.g. T-Mobile for Germany) to connect the internet.
(Only with Wireless LAN activated!)

**Cisco VPN**
Encrypted data transfer through VPN tunnel.




**COMgate - Modem**

**Predefined provider**
Select a predefined provider from the drop down list (use of provided standard settings).
Provider: T-Online, Vodafone, O2, E-Plus

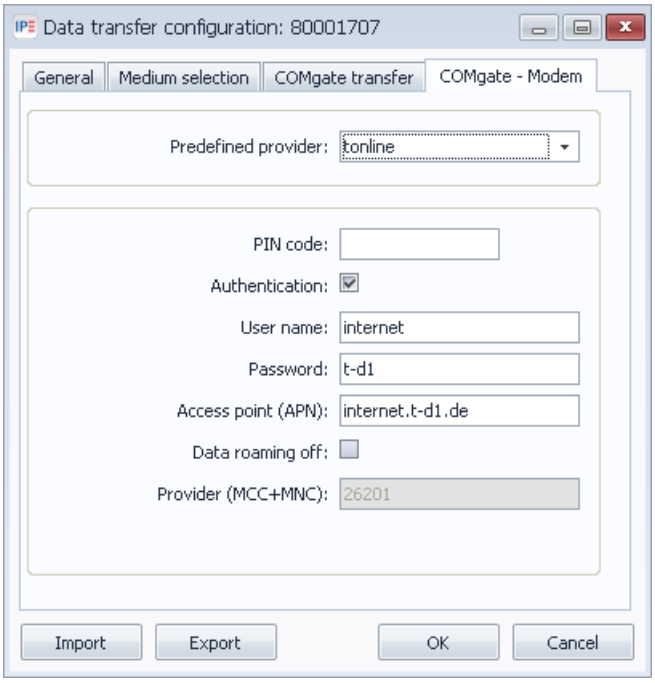
**PIN code**
Identification number of the SIM card

**Authentication**
Access authorisation by **User name** and **Password**.

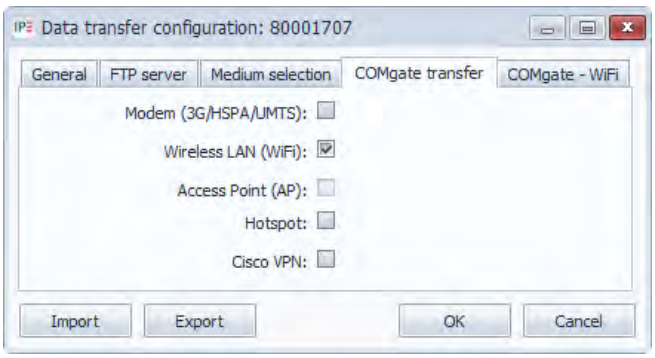
**Access point (APN)**
Name of the access point for the modem connection (APN = Access Point-Name).

**Data roaming off**
Data transfer enabled only for the defined provider (radio network availability provided)

**Provider (MCC+MNC)**
**M**obile **C**ountry **C**ode and **M**obile **N**etwork **C**ode are required identification numbers for manually setting of the radio network provider.



### COMgate transfer




**Modem (3G/HSPA/UMTS)**
Data transfer by internal modem (COMgate WAN only).

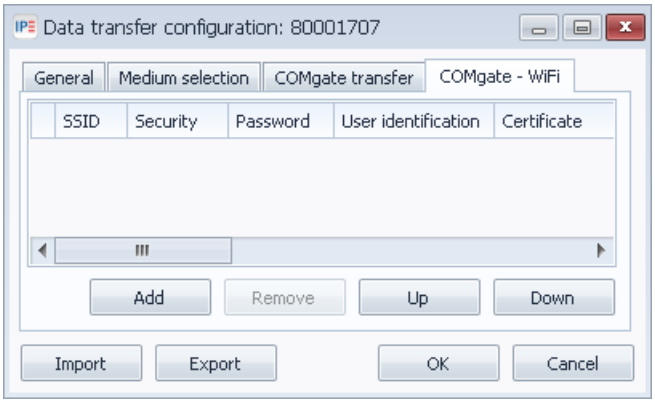
**Wireless LAN (WiFi)**
Data transfer by internal WiFi client.

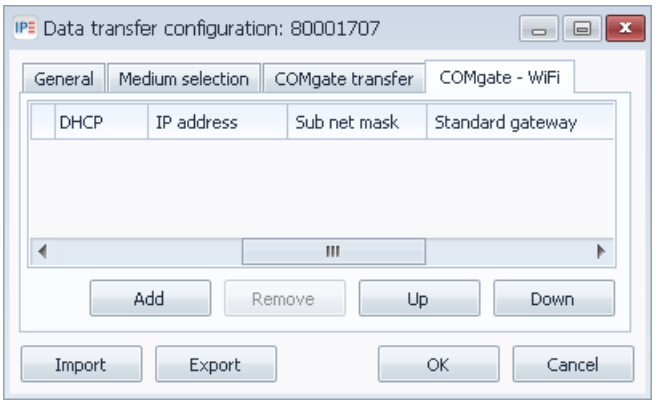
**Access Point (AP)**
COMgate operates as WiFi access point.

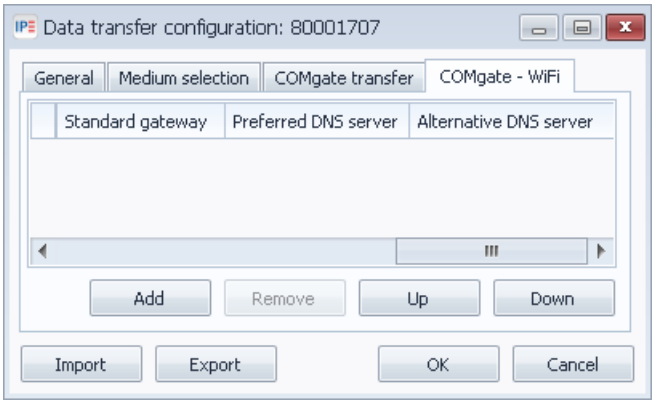
**Hotspot**
Hotspot (public WiFi access point, e.g. T-Mobile for Germany) to connect the internet.
(Only with Wireless LAN activated!)

**Cisco VPN**
Encrypted data transfer through VPN tunnel.

### COMgate - WiFi




**SSID**
Net work name of the related access point (Service Set IDentifier)

**Security**
WiFi transfer protocol WPA, WPA2, Radius, MSCHAPV2

**Password**
Password for user access.

**User identification**
User name for user access.

**Certificate**
File providing the certificate for the respective network connection.

**DHCP**
Automatic handling of IP addresses (IP assignment) in the network by a server supporting DHCP.

**IP address**
Manually assigned client IP address

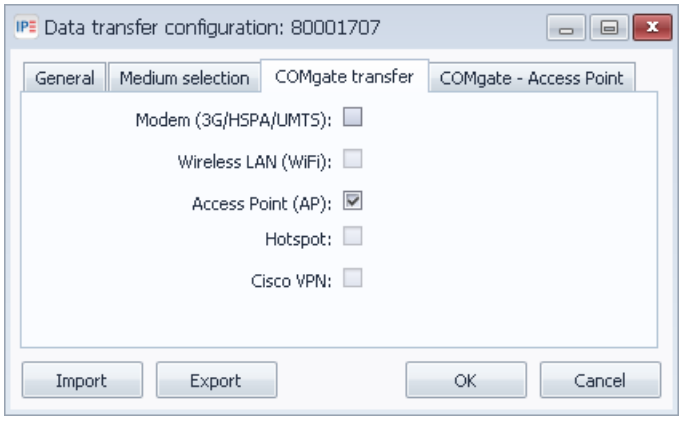
**Sub net mask**
IP address range of the sub network

**Standard gateway**
Network address of the standard gateway

**Preferred DNS server**
Address of the 1st name server (DNS = Domain Name System) to link the host name, in case that only the name of the target server has been entered.

**Alternative DNS server**
Address of the 2nd name server as fall back, in case the 1st DNS server is not available.

### COMgate transfer




**Modem (3G/HSPA/UMTS)**
Data transfer by internal modem (COMgate WAN only).

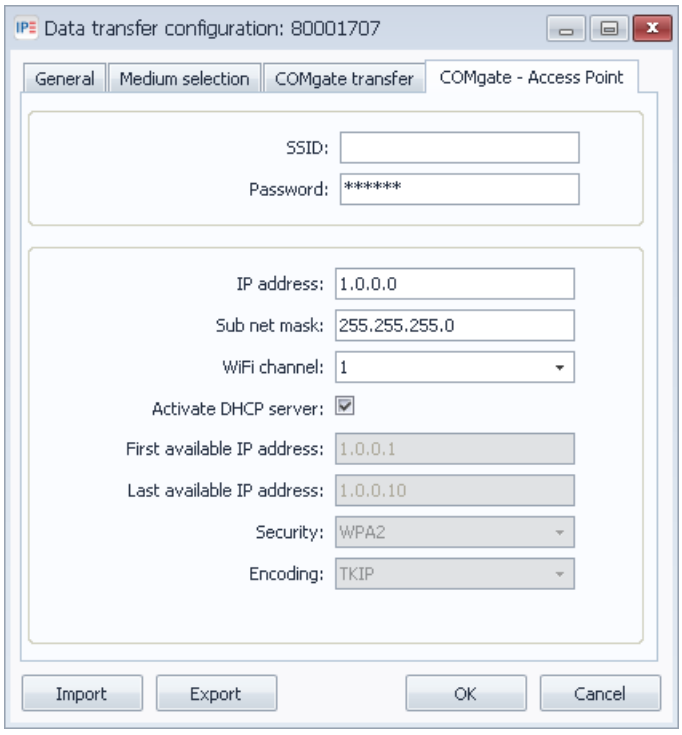
**Wireless LAN (WiFi)**
Data transfer by internal WiFi client.

**Access Point (AP)**
COMgate operates as WiFi access point.

**Hotspot**
Hotspot (public WiFi access point, e.g. T-Mobile for Germany) to connect the internet.
(Only with Wireless LAN activated!)

**Cisco VPN**
Encrypted data transfer through VPN tunnel.

### COMgate - Access Point




**SSID**
Net work name of the related access point (Service Set IDentifier)

**Password**
Net work password

**IP address**
COMgate IP address

**Sub net mask**
IP address range of the sub network

**WiFi channel**
Selection of the WiFi channel and the associated carrier frequency.

**Activate DHCP server**
COMgate acts as host with automatic IP address assignment for the clients.

**First available IP address**
Lower bound of the IP range, e.g. 198.164.0.101

**Last available IP address**
Upper bound of the IP range, e.g. 198.164.0.110

**Security**
WiFi transfer protocol WPA2

**Encoding**
Security protocol (Temporal Key Integrity Protocol)

*COMgate also provides XCP data transfer from the logger system to a mobile device (with Android operating system) by a WiFi connection. Please read more details regarding to the online data visualization (using XCP service) with the IPEmotion App in this manual:*

***IPEmotion-App-V02.xx.pdf*** .


**COMgate - Hotspot**

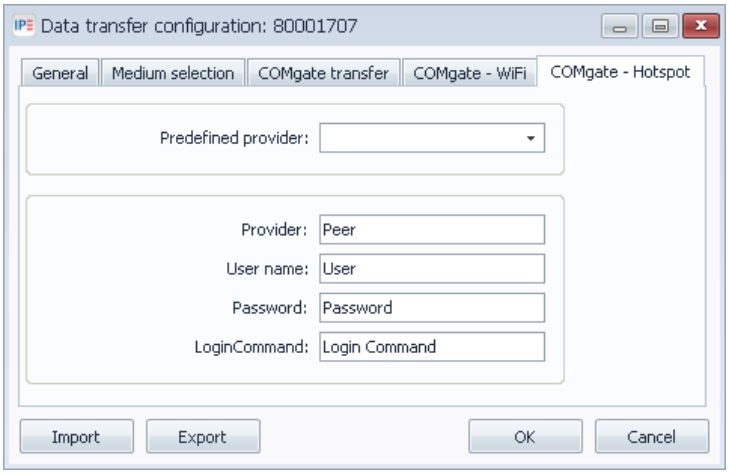
**Predefined provider**
Select a predefined provider from the drop down (use of provided standard settings).
Provider: T-Online

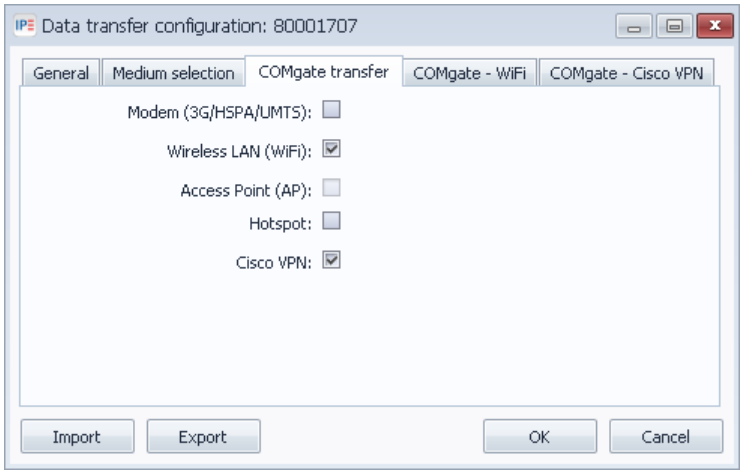
**Provider**
Name of the provider (entered manually)

**User name**
User name to access the account.

**Password**
Pass word to access the account.

**LoginCommand**
Login command to to access the account.




**COMgate - Cisco VPN**

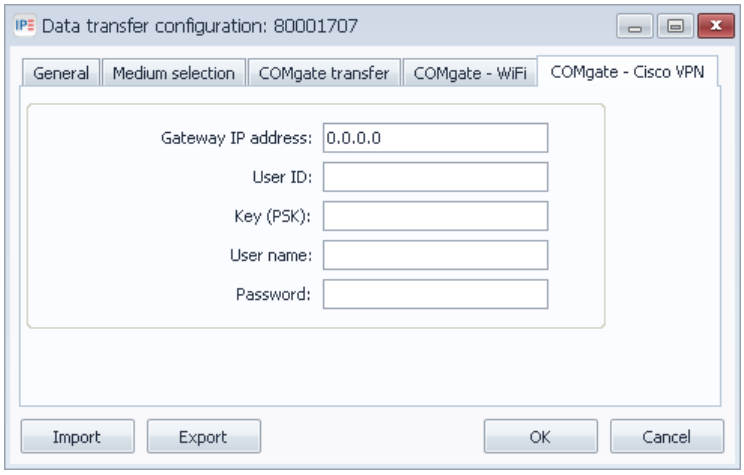
**Gateway IP address**
Network IP address of the gateway used for data transfer.

**User ID**
User identification

**Key (PSK)**
Pre-shared key used by two different peer systems identifying each other.

**User name**
User name to access the account.

**Password**
Password to access the account.



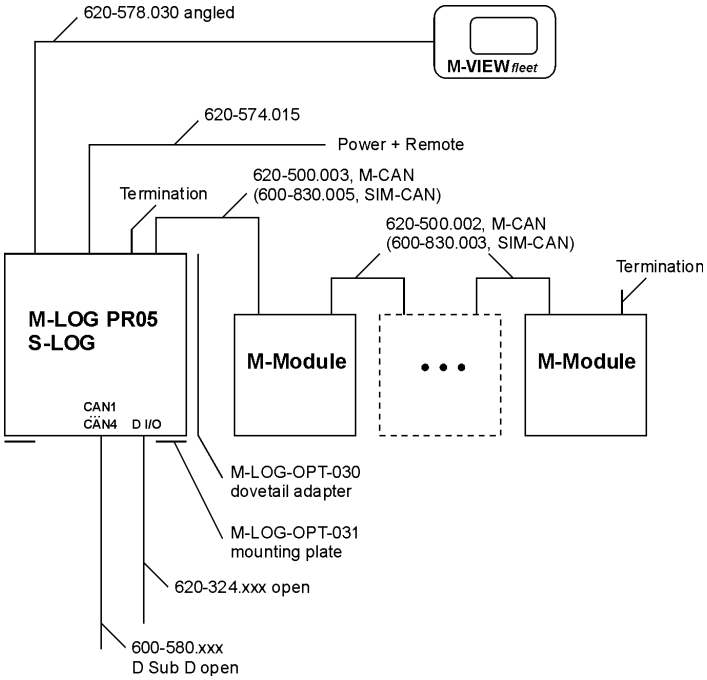
# 9 Display modules

## *9.1* M-VIEW*fleet*

M-VIEW*fleet* is a alphanumerical data display, which is connected to the USB of the logger. M-VIEW*fleet* provides different displaying modes besides four status LEDs and buttons. The number of displayed channels is only limited by the processor load.

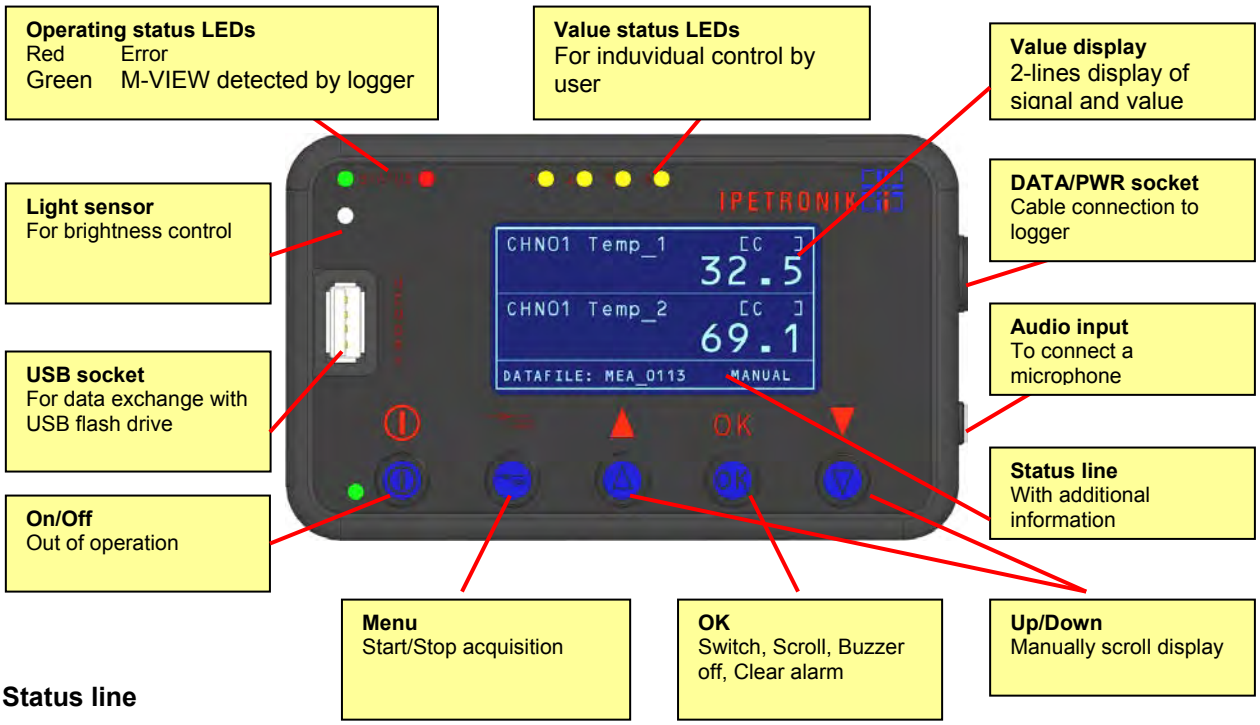


**Only connect** M-VIEW*fleet* to the logger **if** logger is **switched-off**. In case M-VIEW*fleet* is connected to logger during operation, the USB interface can be damaged.




M-VIEW*fleet* is connected to M-LOG with an USB 2.0 interface. A second USB port of the logger is available at the display. USB 2 is used for e.g. a program update or to exchange measurement or configuration data via USB flash drive. USB connections are designed for a maximum length of 5 m. Display settings are defined in the system configuration.

### 9.1.1 Keys and LEDs




**Status line**

The scroll mode ***MANUAL*** (with Up/Down keys) or ***AUTO*** (automatically) is displayed in the right zone of the staus line. The following informatiojn is alternately shown in the left zone:

| | |
|---|---|
| DATAFILE | Name of current data file |
| TIME LEFT | Available time in days (D) and hours (H) for data recording |
| DRIVER | Selected driver |
| SHIFT | Selected shift (track or road sections belonging together) |

**OK**

Function in standard display

| | | |
|---|---|---|
| 1. | Display Min/Max | Press longer than one second |
| 2. | Back to standard display | Press longer than one second |

Function in standard display in scroll mode AUTO:

(Changed from AUTO to MANUAL with Up/Down keys)

| | | |
|---|---|---|
| 1. | Back to auto scroll mode | Press longer than one second |

If the key is not pressed, the display changes after 30 s into auto scroll mode.

Switching from MANUAL to AUTO is only possible if auto scroll mode has been activated!

Function at configured alarm limit values:

| | | |
|---|---|---|
| 1. | Buzzer off | Shortly press to switch-off buzzer |
| 2. | Clear alarm | Press longer than one second, Back to standard view |

If the alarm has already been cleared 5 times, a final message appears to definitely delete the alarm.

**Up/Down ▲ ▼**

| | |
|---|---|
| One line up | ▲ |
| One line down | ▼ |
| Switch AUTO > MANUAL | ▲ or ▼ if AUTO (Auto scrolling) has been activated in configuration |

**Menu**

Stop acquisition -> Press longer than two seconds

Start acquisition -> Press longer than two seconds

*The option* **Allow start and stop of acquisition** *must be activated in the configuration. The measurement file number is increased by one at every acquisition start. If the option* **Track** *or* **Track/Driver** *(List type selection) is additionally selected, files can be merged to one file until the final end of data recording.*

*Devices connected to the logger* ***remain switched-off*** *until* ***acquisition stop****.*

**Operating status LEDs**

Red 1. Error

2. Logger is booting, initializing M-VIEW

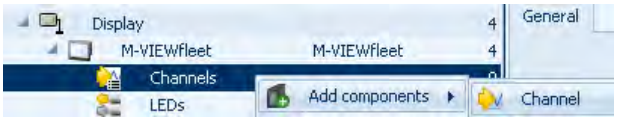
Green Operation, M-VIEW detected by logger

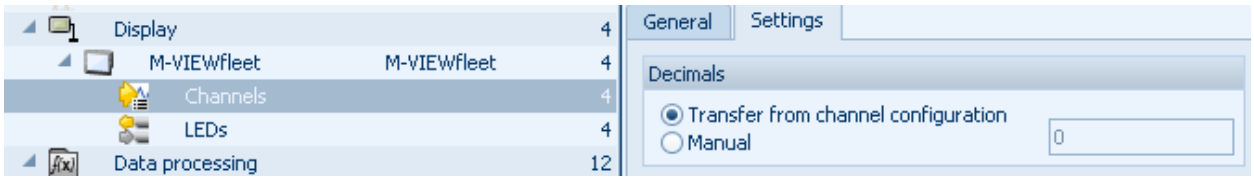
**Value status LEDs**

The 4 status LEDs can be controlled with calculations. Additional limit violations can therefore individually be displayed.

## 9.1.2 Configuring M-VIEW*fleet*

- Select Display in the tree structure.
- Select Add components.
- Select M-VIEWfleet.
- Select the desired signals.
- Confirm the desired signals with **OK**.




Define the alarm values for the upper and lower signal limit, if desired. If the current signal reaches the alarm limit, the display changes into the message window and the buzzer sounds. Clear the alarm with OK.

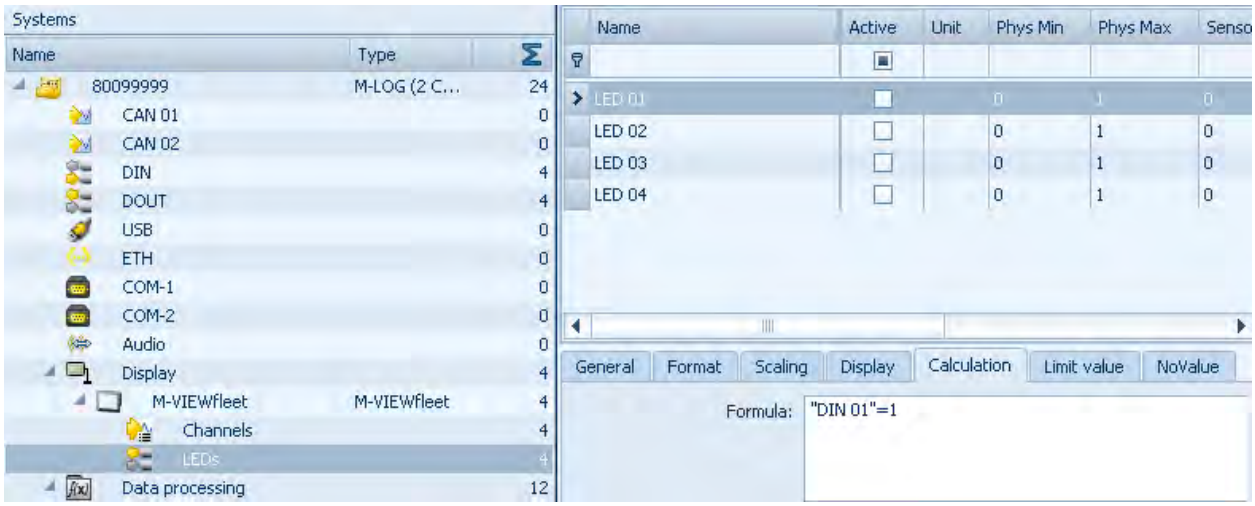
Activate the detection of the minimum and/or maximum values for the respective signal. With pressing OK, the display changes into showing the minima and maxima. If the detection has not been activated, **------.--** is shown. Pressing OK again changes back to standard display.

## Define status LEDs for limit display

The 4 status LEDs can be activated by separate and user-defined calculation formulas. Additional thresholds can therefore be defined and reaching these limits can optically be signalized. In addition, the LEDs can be used as status display for the 4 digital logger outputs.

To do so, the following options are available:

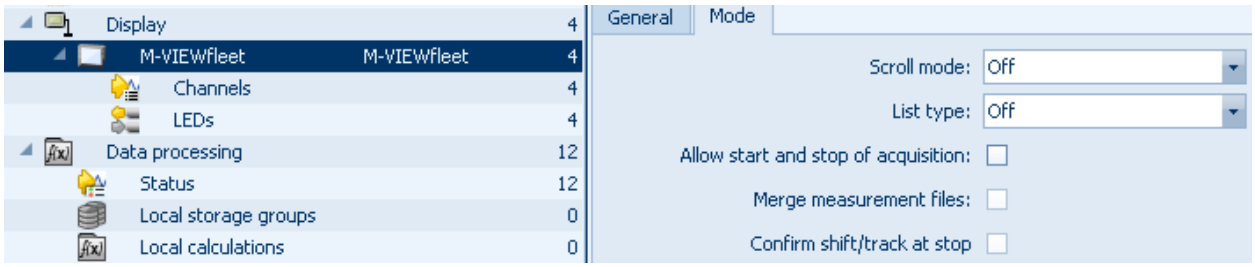
- Use the same formula like at the corresponding digital output
- Status query of digital output to 1 (LED ON, for the time digital output = 1)




## Setting display modes

Select ***Active*** to use M-VIEW*fleet* in the configuration.

*No further settings can be defined without activating M-VIEWfleet! If an existing M-VIEWfleet configuration is deactivated, the logger display has no function! This is signalized by the red LED. The red LED is also on if the USB port is damaged during operation by disconnecting the cable.*



**Scroll mode**

| | |
|---|---|
| Off | Manually switching lines with Up/Down |
| 5 s | Continuous line switching in interval of 5 s, display moves line by line from the bottom up in order of signal list |

**List type**

| | |
|---|---|
| Off | Neither track nor driver defined |
| Track | Track selection by driver is assigned in data |
| Track/driver | Track, as well as, driver are selected before start and set in data |

The text file of the track selection list is defined in:

...\IPETRONIK\\IPEmotion MAL-PlugIn IPETRONIK LOG V03.xx.xx\Data\MViewfleet\MVIEWfleetTracks.txt

The text file of the driver selection list is defined in:

...\IPETRONIK\\IPEmotion MAL-PlugIn IPETRONIK LOG V03.xx.xx\Data\MViewfleet\MVIEWfleetDrivers.txt

Changes of the entries can manually be defined in the respective text file.

**Allow start and stop of acquisition**

Pressing the menu key stops the data storage and the current measurement file is closed.

Pressing the menu key again starts the next data storage and the number of measurement file is increased by one.

**Merge measurement files**

This function requires a selected list type!

The end of every acquisition includes the ***End Shift xxx*** query. If it is confirmed with OK, all previous partial acquisitions of one file are merged and the number for the next measurement file is increased by one. If this function is not activated, an own file is written after every acquisition stop and a following acquisition is recorded with a new number.

**Confirm shift/track at stop**

A selected list type is required for this function and *Merge measurement files* must be activated!

The query runs after acquisition stop:

| | | |
|---|---|---|
| ***End Shift xxx!*** | ***Yes OK?*** | Track is stopped, partial acquisitions are merged in one measurement file |
| | ***No OK?*** | Track is continued at next start with same measurement file |

The query runs after acquisition start:

| | | |
|---|---|---|
| ***Shift xxx Good Trip!*** | ***OK?*** | Currently saved track is continued with new acquisition file |
| | ***Change OK?*** | Track and driver can be selected again |

*If nothing is entered after a query, an acoustic message sounds after 20 s and the saved settings are accepted for the next acquisition.*

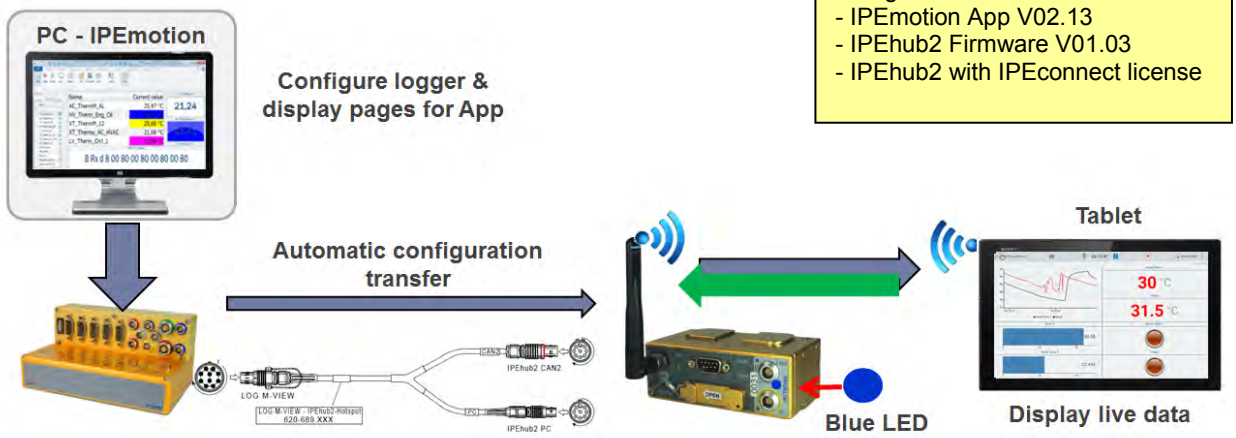
## 9.2 IPEconnect (Online display with Smartphone or tablet)

IPEconnect serves as an online display for IPETRONIK data loggers. The system consists of the components data logger (M-LOG, IPElog, FLEETlog), IPEhub2, corresponding connecting cables and the mobile device (Smartphone, tablet) with the IPEmotion App.

### 9.2.1 Overview

- Smartphone / tablet for online data visualization of the data acquisition running on the logger
- WiFi access point IPEhub2 as gateway from the logger to the Android display
- System configuration with IPEmotion
- Quick installation, easy and stable operation

**Minimum requirements**
- IPEmotion 2015 R3
- PlugIn IPETRONIK-LOG V03.56
- IPEmotion App V02.13
- IPEhub2 Firmware V01.03
- IPEhub2 with IPEconnect license

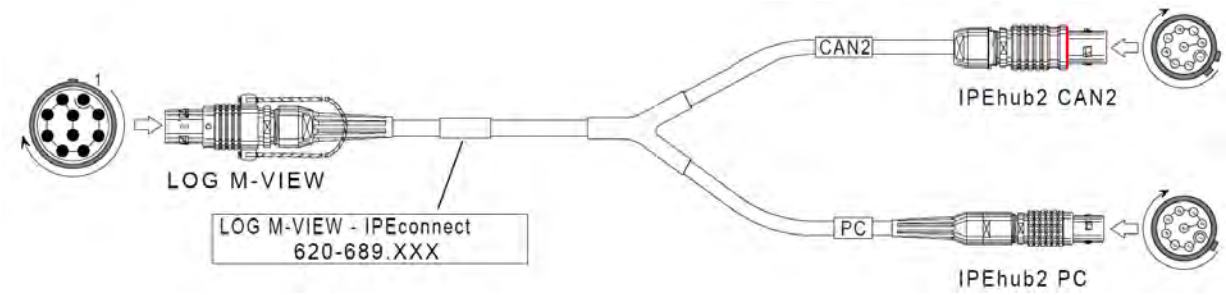



### 9.2.2 Features

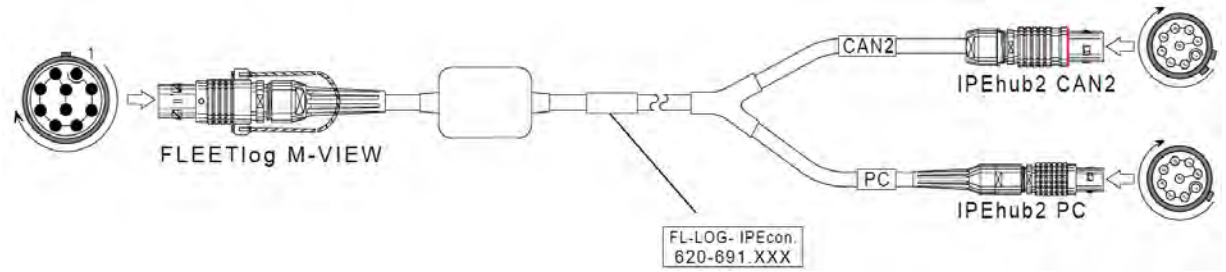
- Creating the data configuration (online data) based on the logger measurement configuration.
- The IPEmotion App imports a new configuration automatically.
- Creating the configuration for the online visualization with the mobile device.
- The blue LED at IPEhub2 indicates a successful WiFi connection.
  Name of the network / SSID: Logger_[Serial_number]
- The app configuration on the smartphone / tablet is saved on IPEhub2.
- Shortly after the app has been started, it shows all defined values.
- IPEhub2 can be connected and disconnected at any time (even during a running measurement).
- If a connection is disturbed or interrupted, IPEhub2 will re-establish the WiFi connection immediately.

## 9.2.3 Cables

### 620-689.xxx LOG-VIEW Cable IPEconnect (M-LOG, M-LOG V3, IPElog)



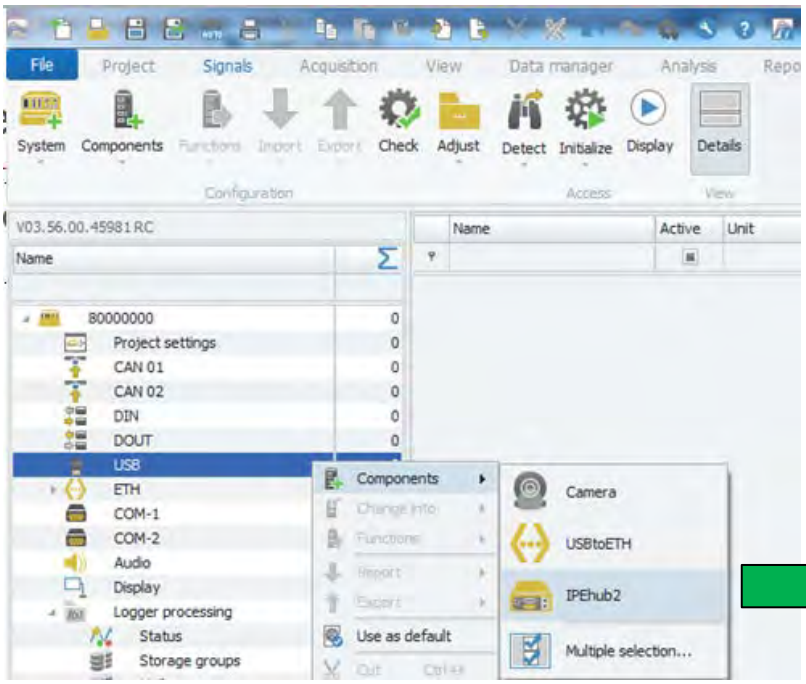

### 620-691.xxx FLEETlog-VIEW Cable IPEconnect (FLEETlog, FLEETlog2)

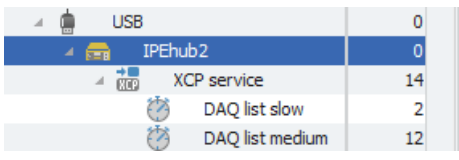


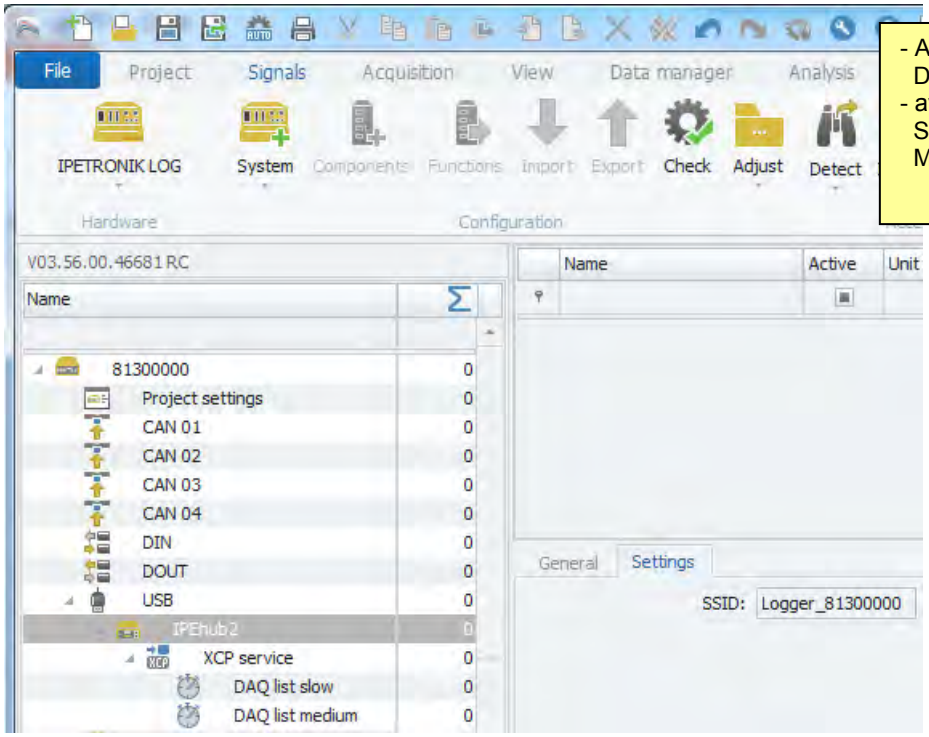

## 9.2.4 Settings

### Logger USB interface

- Setup IPEhub2 at the logger's USB interface

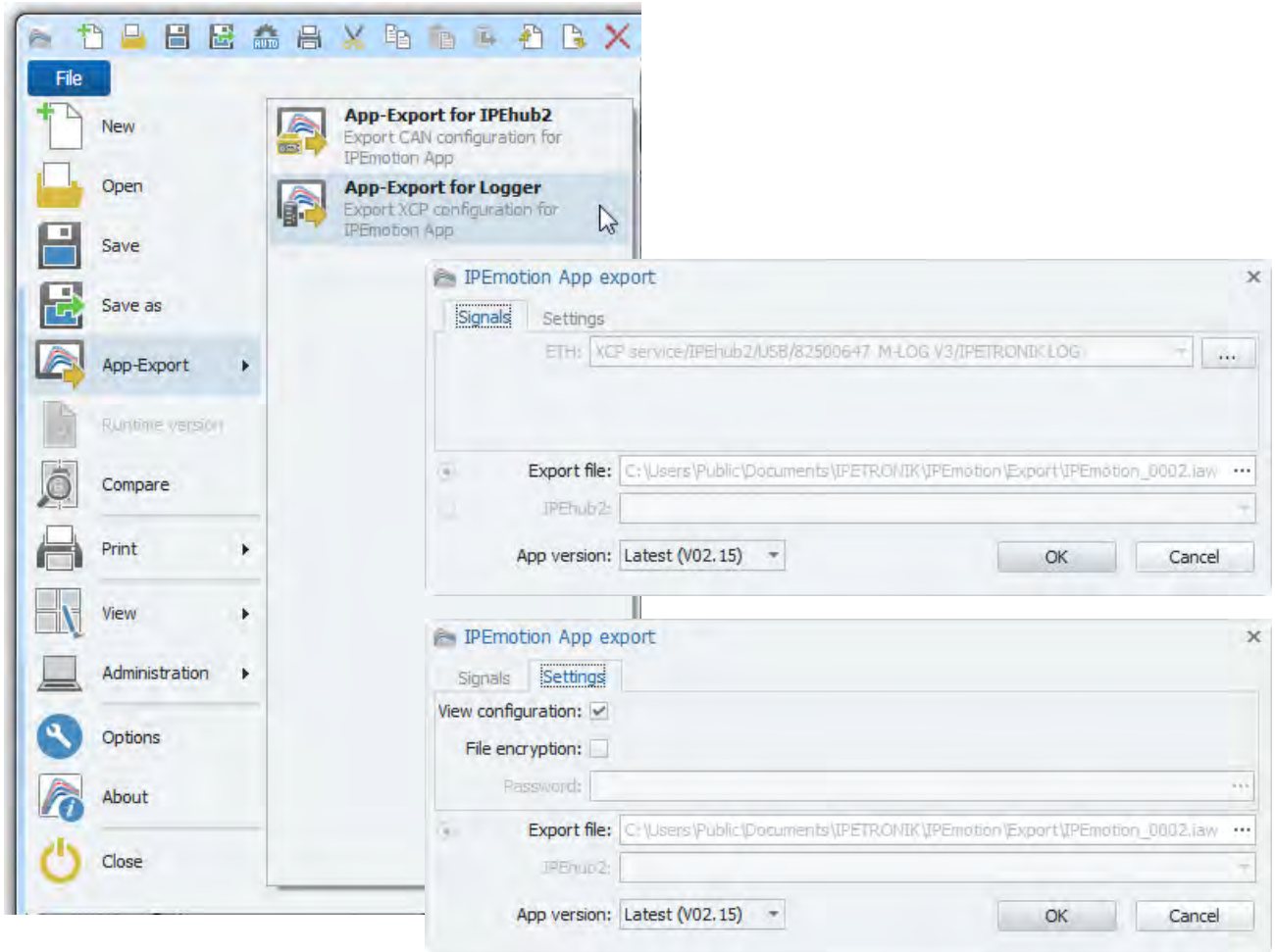




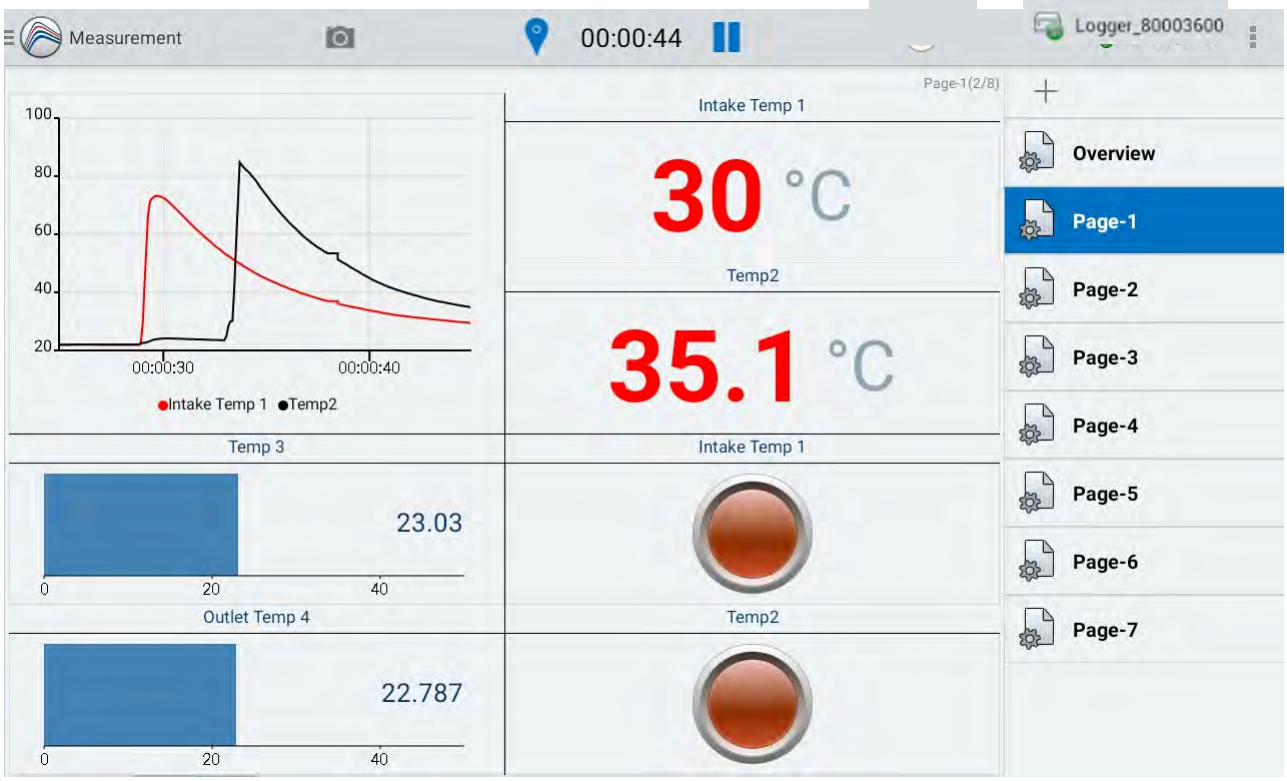

- Assign all required channel to the DAQ lists
- available DAQ lists:
  Slow > 1 Hz data rate
  Medium > 10 Hz data rate

## App export



### 9.2.5 App display

The app shows consecutively signal values as soon as the WiFi connection is established and the logger is operating in measuremend mode.

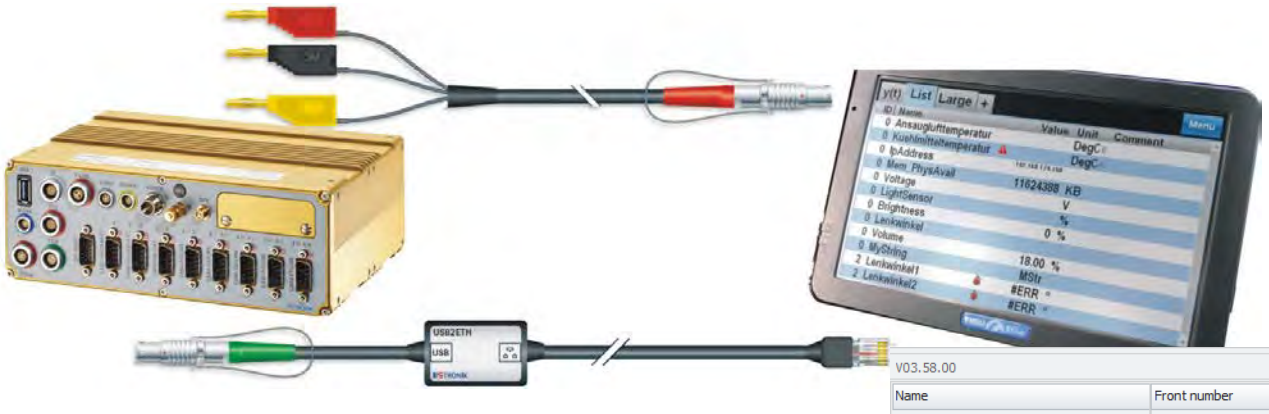


## 9.3 Connecting displays via openABK

With the implementation of the protocol openABK (short for open display and operating concept) in the PlugIn IPETRONIK-LOG V03.60 (and corresponding TESTdrive V03.60) standard displays that are supporting this protocol (e.g. the EMBU-Sys displays from the CANDICE series) are supported.

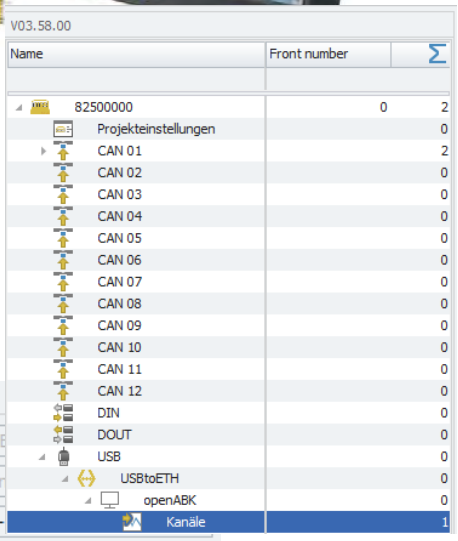
### 9.3.1 Cable connection

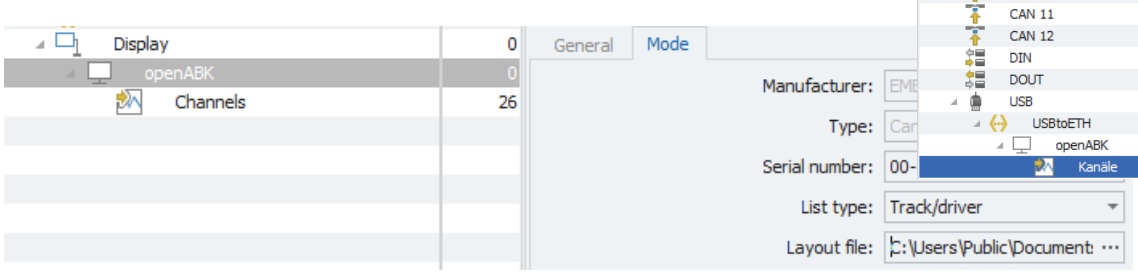
The display can be used for data loggers IPElog2, M-LOG V3 and FLEETlog2 with an USB2ETH cable connected to the VIEW socket of the logger.



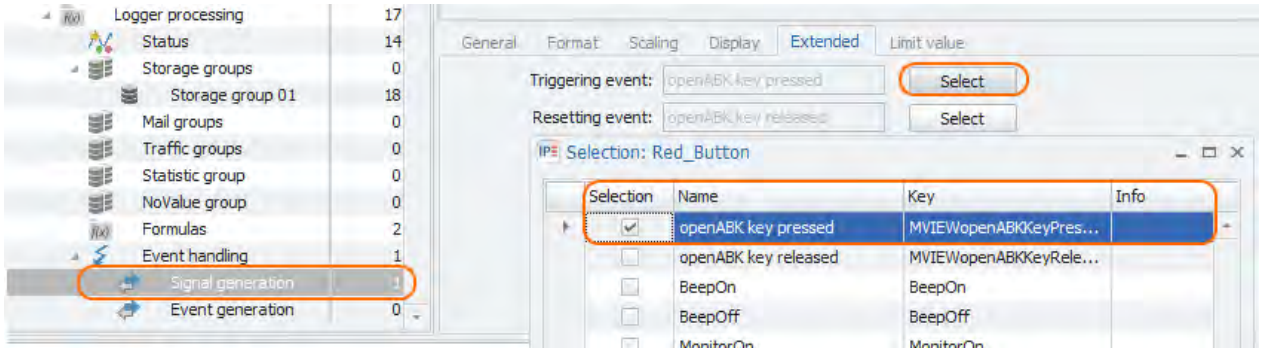
### 9.3.2 Display configuration

The configuration of the measurement data visualization is set by the software of the display manufacturer. For CANDICE the EMBU-Sketch software offers an individual configuration of display instruments and function buttons. Furthermore, the display has to be created within the logger configuration by adding an USB component.




### 9.3.3 Use display buttons for triggering



# 10 Accessories

## 10.1 Electrical accessories

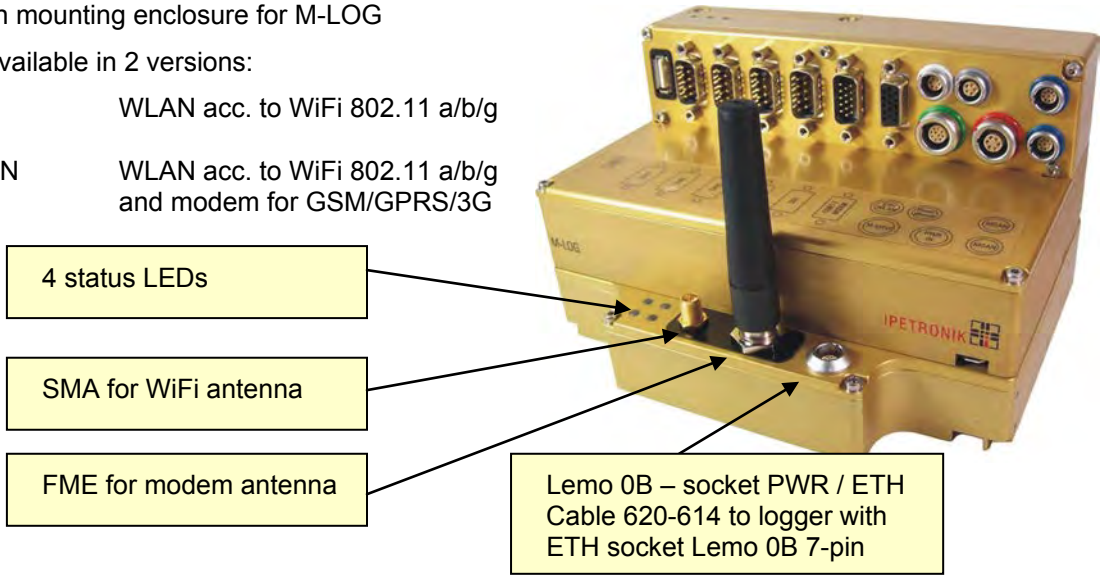
### 10.1.1 COMgate

COMgate is an intelligent extension device for logger and allows remote transmission of measurement and configuration data.

M-COMgate in mounting enclosure for M-LOG

COMgate is available in 2 versions:

| | |
|---|---|
| COMgate | WLAN acc. to WiFi 802.11 a/b/g |
| COMgate WAN | WLAN acc. to WiFi 802.11 a/b/g and modem for GSM/GPRS/3G |




### LED status display

| LED display | Status | Meaning |
|---|---|---|
| **GREEN** | PWR ON | Device is ready for operation<br>(operation: see LED yellow or orange) |
| | MODEM | Establishing connection to UMTS/GPRS network |
| | MODEM | Successfully registered to UMTS/GPRS network |
| **ORANGE** | MODEM | Steady connection |
| | WiFi | Establishing connection to WiFi network |
| | WiFi | Successfully registered to WiFi network |
| **YELLOW** | WiFi | Steady connection. |
| **RED** | ERROR | Interference, potential reasons are:<br>- transfer of new configuration<br>- wrong configuration loaded<br>- general operating interference |

*M-COMgate requires additional external cooling on M-LOG at operating in ambient temperatures > 70 °C (158 °F)!*

## 10.1.2 Extender

### CAN-Extender


Input with status LED (Bus traffic)

Operating status LEDs

ETH/PWR socket to M-LOG

M-LOG Extender is an extension device with 4 additional CAN inputs. M-LOG devices, which are already equipped with a LX800 processor board, as well as, the option Input 2x Ethernet, can be extended without modifying hardware. The extender is screwed to the bottom of the basis device with 4 screws and connected to the logger with the cable 620-406.002 (here PR08, or 620-404.002 for PR03 and 620-405.002 for PR04).

**Requirements**

- Data logger with LX800
- Option Input 2x Ethernet (with respective port replicator)
- free Ethernet input
- IPEmotion + LOG-PlugIn ≥ 03.19 (Creation and configuration see ETH 01/02 port)

**Advices**

- The bus inputs of the extender do not support WakeOnCAN and traffic acquisition.
- Data is directly stored on the logger.
- The maximum data rate is 100 Hz.
- The configuration is extended by the additional file *.ecf (Extender Configuration File).

## 10.1.3 IPEwifi

IPEwifi is as wireless extension for IPETRONIK data loggers (M-LOG, S-LOG) supporting radio data transfer using the WiFi standard 802.11 b/g.

Measurement data is transferred on command or automatically without any cable connection to a network access point. IPEwifi acts as a communication bridge and forwards measurement and configuration data. IPEwifi itself neither supports router functions nor subnetworks.

Status indication
PWR Ready for use
LINK LAN data transfer
WiFi Wireless LAN data transfer

LAN / PWR
LAN data transfer and module power supply


Antenna
SMA(R)-UFL connector

IPEwifi setup via web browser,
► refer to separate manual IPEwifi

## Setting up a LAN connection for the logger

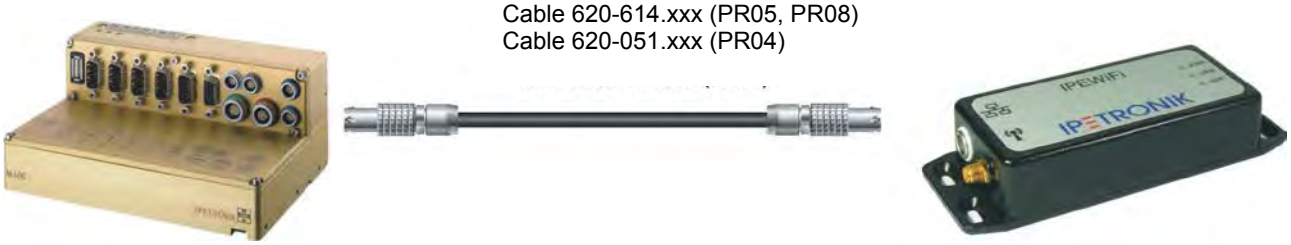
IPEwifi operates in bridging mode (means as LAN cable substitution), thus logger settings are similar to a direct LAN connection to the network connection point.

- Start IPEmotion
- Open an existing logger configuration or create a new one
- Highlight the logger in the left hand system structure
- Move to the configuration dialog (right hand lower area) with the **Data manager** tab, activate the check box **Update connection parameters** and click the **Configuration** button.
- Click to the **Medium selection** tab and activate the **LAN** check box.
- Activate the check box **Get IP address automatically**
- Confirm and close with the **OK** button.
- Transfer the current settings to the logger.

This settings enable the logger to receive its IP address through IPEwifi using DHCP services of the network server.

## Connecting IPEwifi and the logger with a LAN/PWR cable

- Connect IPEwifi with the logger as shown below.
- As soon as switched on, the green LED lights permanently.
- The yellow LED indicates a properly working LAN connection.
- The orange LED indicates a properly working WiFi connection.




*Depending on the logger settings under **COMgate switch-on condition** from the **System activated** tab, IPEwifi will be supplied with power by the logger*

| | |
|---|---|
| ***Acquisition*** | *during the measurement,* |
| ***Postprocessing*** | *during data post processing, as soon as measurement is stopped, (follow-up time is running),* |
| ***Always*** | *as soon as the logger has started.* |

*If the logger is not able to communicate with the network, although the settings are correct and the WiFi connection is steady (indicated by the orange LED), please check the entries of the log file (MEA_xxxx.log).*

*As former M-LOG devices with port replicator PR05 have a 6-pin Ethernet connector, a direct connection (single cable) using the cable 620-614.xxx is not supported. Only devices with 7-pin Ethernet connector provide the lines (Pin 6 and 7) for the IPEwifi power supply.*

## 10.1.4 GPS receiver


- GPS mouse for satellite positioning
- Connection to the serial logger interface
- Predefined settings of NMEA protocol configuration software

## 10.1.5 Bus isolator SAM-CAN-ISO

- High ohm connection of hardware to vehicle CAN
- Electrical isolation between vehicle bus and measurement system
- Connection with short stub
- Version “Hear only“ to avoid unintended influencing of vehicle CAN

**SAM-ISO011-23A0**

Has e1 approval (**V**ehicle **T**ype **A**pproval VCA) for directly connecting with the CAN bus of public vehicles without restricting the type approval.

## 10.1.6 iMIC

- compact multifunction device (1.57 * 1.44 * 0.98 in) (40 * 36,5 * 25 mm)
- Voice recording with audio input
- good voice quality
- illuminated trigger button
- 3 status LEDs (green, yellow, multicolor)
- integrated buzzer


### Button (illuminated)

Triggering of data recording and/or voice recording via digital input 1.

### Status LEDs

**LED Yellow** Indicates the status of digital output 2 (LED is on if output is active.)

**LED Red** Indicates the status of digital output 3 (LED is on if output is active.)

**LED Multicolor**

| | |
|---|---|
| green | Indicates the status of digital output 1 (green light) |
| blue | Indicates the satus of the yellow LED (blue light) |
| red | Indicates the completed boot process. The excitation of the connected devices (at M-CAN or SIM-CAN socket) is switched-on. |

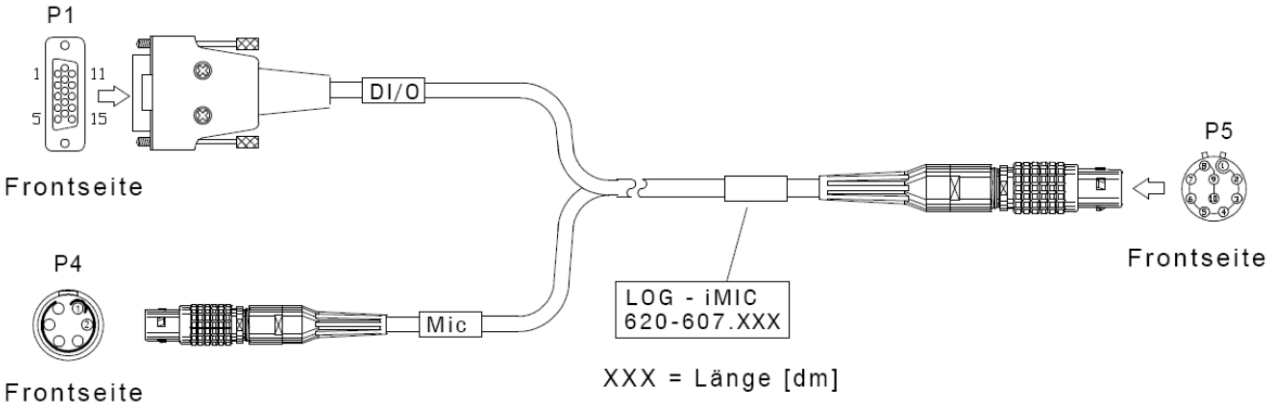
### Buzzer

Acoustically indicates the status of digital output 3 (Buzzer ON if output is active).

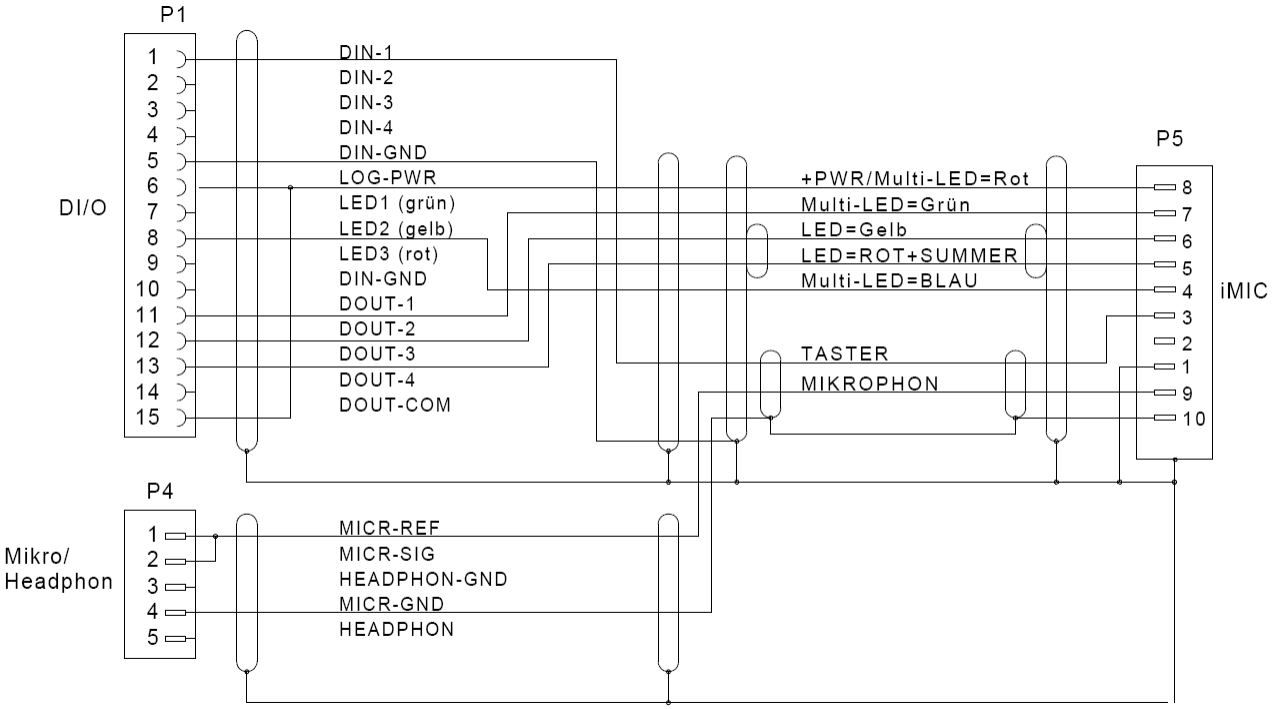
*The specific functions of the digital inputs and outputs are defined in the measurement configuration (see Standard functions calculations, trigger, Use of digital inputs and outputs).*

*The button functionality, as well as, the LEDs and buzzer functionality also depends on the wiring / PIN assignment. If another cable than 620-607.xxx is used, functionality can differ from the one described above.*

## Cable 620-607.xxx




## PIN assignment cable 620-607.xxx



## 10.2 Mechanical accessories

### 10.2.1 Module mounting

#### Dovetail adapter

Adapter plate for mounting to the right M-LOG housing for connecting M devices without tools.

#### Fastening elements

2 fastening strips for mounting at the device bottom to screw M-LOG to an even surface.

#### Snap-in fastener

2 snap-in holders + 2 fastening strips for mounting at the device bottom to fix M-LOG at an even plate without tools.

#### Snap-in adapter

Mounting plate to use a snap-in fastener to fix M-LOG to plate without tools.

#### Suction pad holder for M-VIEW*fleet* / M-VIEW*graph*

Suction pad holder with pump for fixing driver display on smooth surfaces e.g. at windshield.

Figures are subject to change!

*Due to safety reasons, the bottom of the display housing must rest on an underlay (e.g. instrument panel). Avoid a free mounting and do not use the suction pad holder alone for drive tests.*

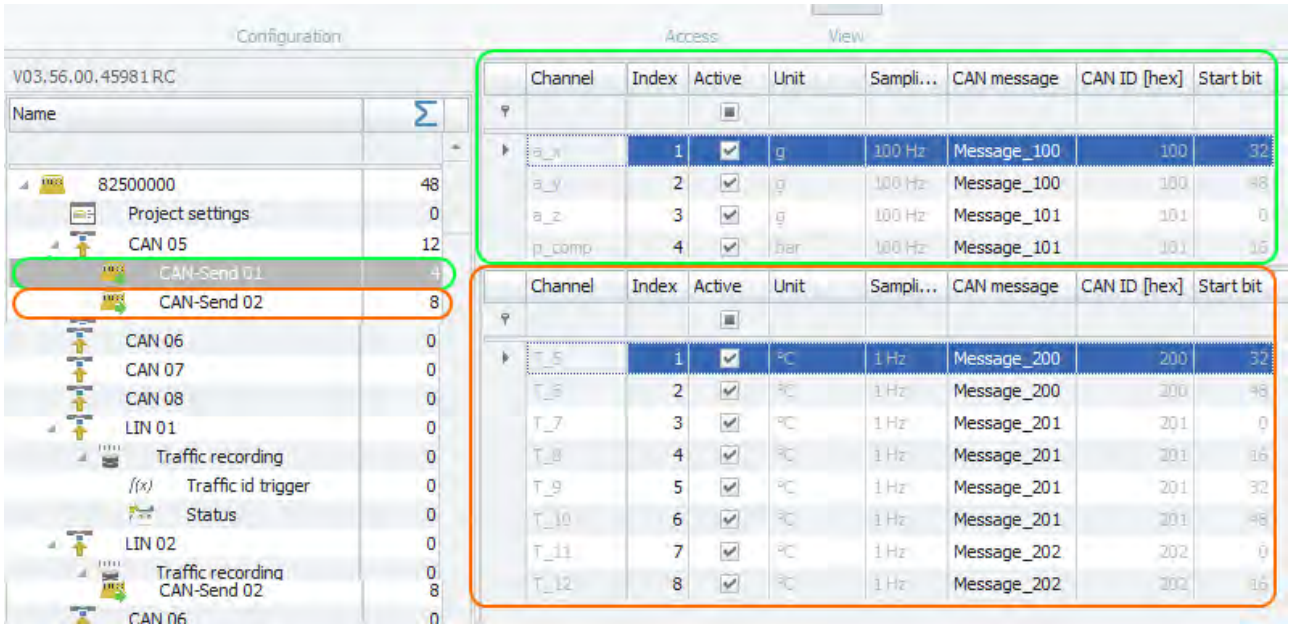
# 11 New features

## 11.1 PlugIn / TESTdrive V03.56

### 11.1.1 Measurement data processing

#### Using several CAN send blocks with a single CAN knot

Data of a CAN knot can be send with different data rates (with the send block) or you can send CAN messages using different start IDs. CANdb export is supported per CAN knot (a description file containing all CAN send blocks) or per CAN send block (a description file per CAN send block).



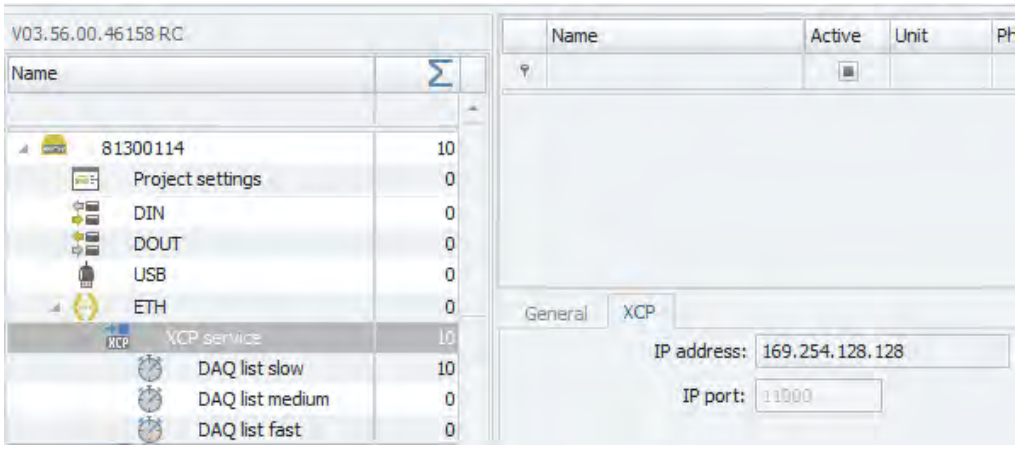

### 11.1.2 Status information (online)

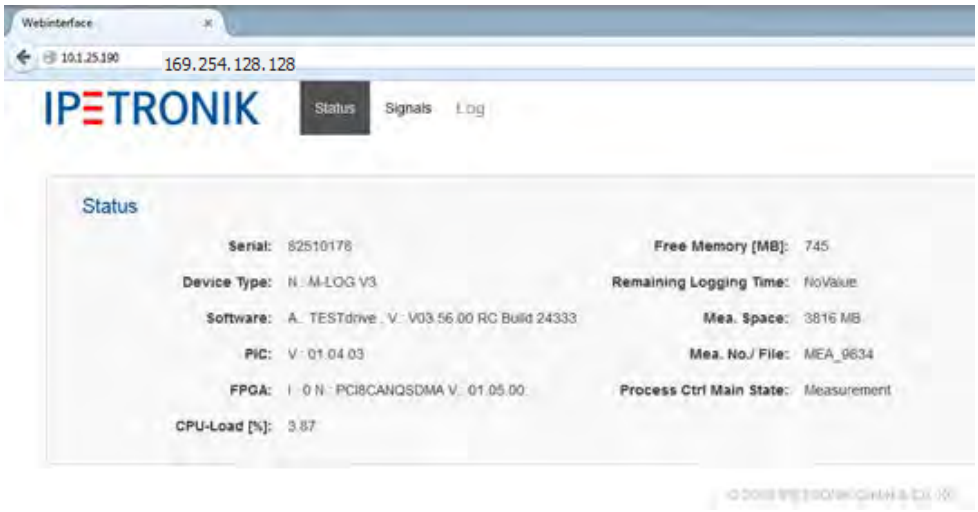
#### Web interface

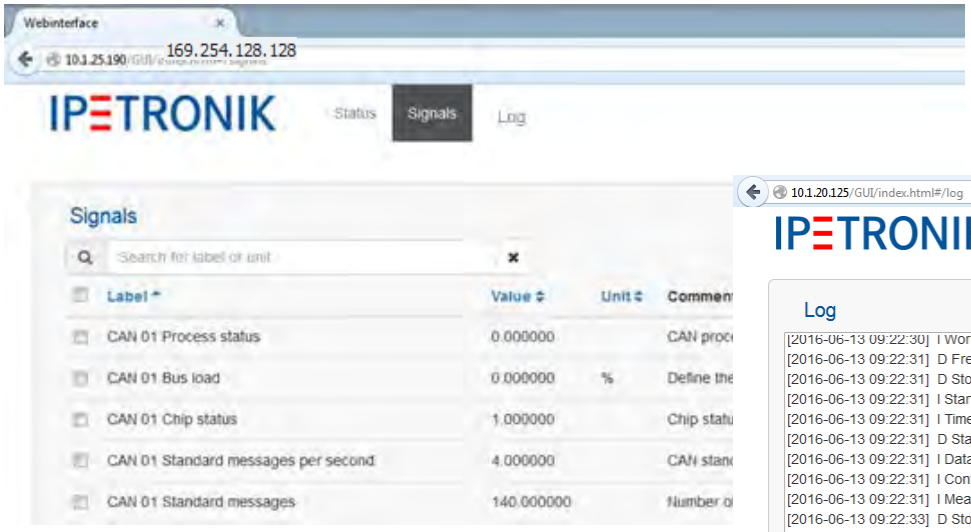
The logger´s service web site shows status information, log file contents and a listing of active signals.
A specific IPEmotion configuration is not required.

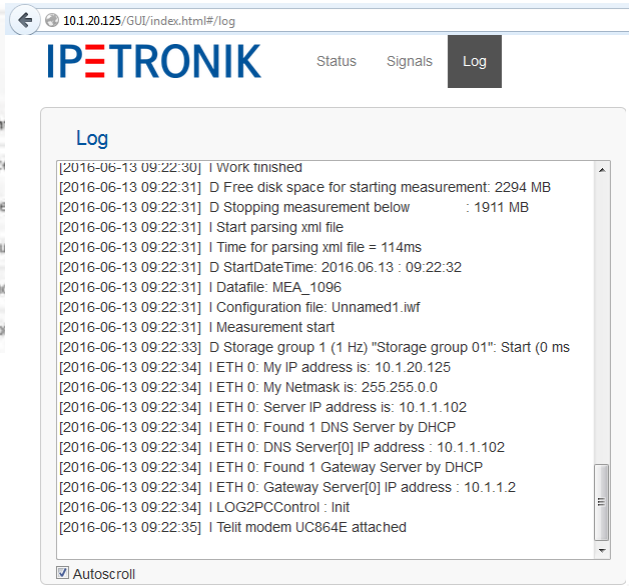
When using IPEhub2 (IPEconnect feature), the standard IP address **192.168.232.9** is used.

Alternatively the IP address, that has been assigned to the logger, can be used. Use the Ethernet branch in the system structure of the logger configuration to enter the IP settings.






***Sign of life and refresh rate***

*The status bar of the web interface shows the sign of life and the update cycle at the bottom left corner*:

| | |
|---|---|
| *Refresh rate* | *Web service polling rate (signal request from the logger) in Hz* |
| *Sign of live* | *Text output indicating the connection status (online or offline), The refresh cycle of the status indication is 10 s. During the status* ***offline*** *the refresh rate shows 0 Hz and signal values indicate* ***„Not available****“.* |

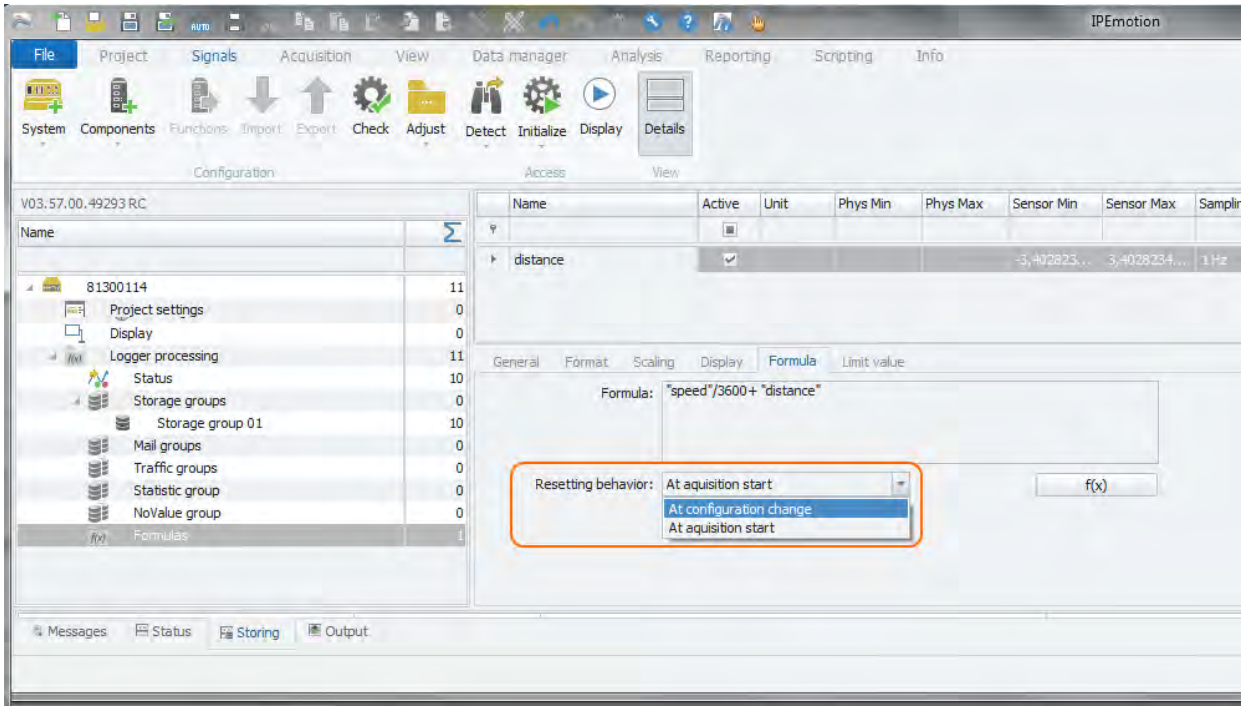
*The decimal places of the signal values listed in the the web interface are configured by the respective channel properties. Go to the IPEmotion* ***Signals*** *area, select the channel, open the* ***Display*** *tab and enter the drop down menu under* ***Formatting*** *to choose the* ***Decimal places*** *settings.*

# PlugIn / TESTdrive V03.57

## 11.2.1 Measurement data processing

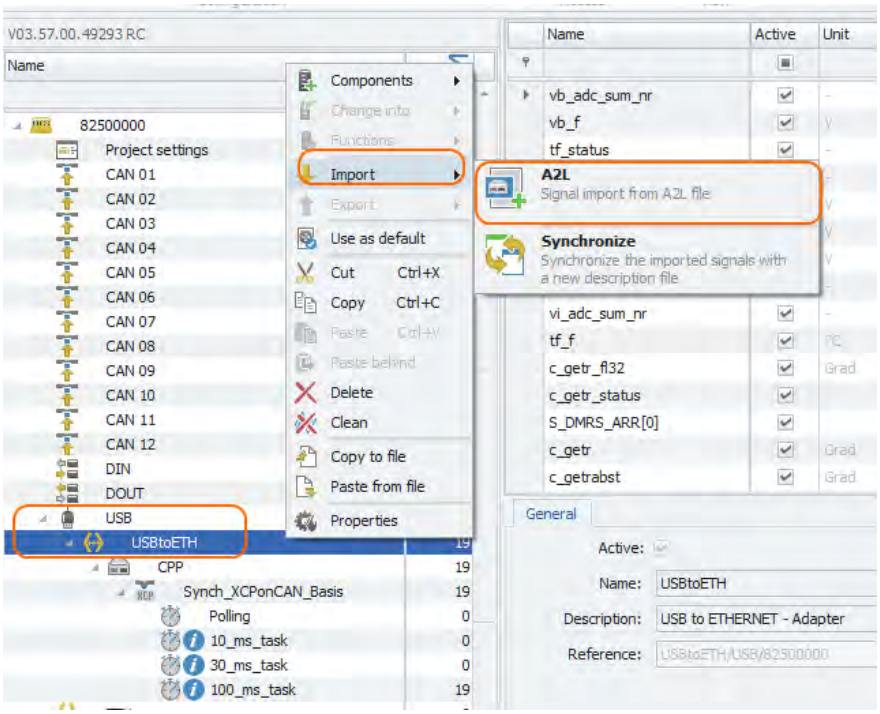
### Using last value from previous measurement

For calculations under **Logger processing > Formulas**, the user can define, if the last measurement value of the previous acquisition should be used as the first value of the current measurement. This feature is not supported for calculating trigger conditions (switch-on conditions, activating storage groups). The parameter **Resetting behavior** is set for each calculation. Default setting is **At acquisition start** and corresponds to the formerly behavior (no use of values from a previous measurement). Selecting **At configuration change** means, that the value is not recessed by starting a new measurement, but definitely with a configuration update.




### Release XCPonUDP import for USB2ETH adapter

The import of A2L description files is supported by an USB2ETH adapter.



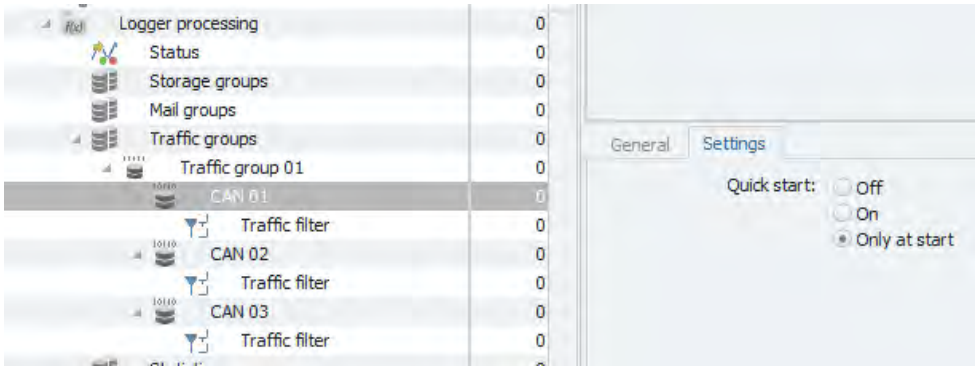
## 11.2.2 Data storage

### Quick start recording

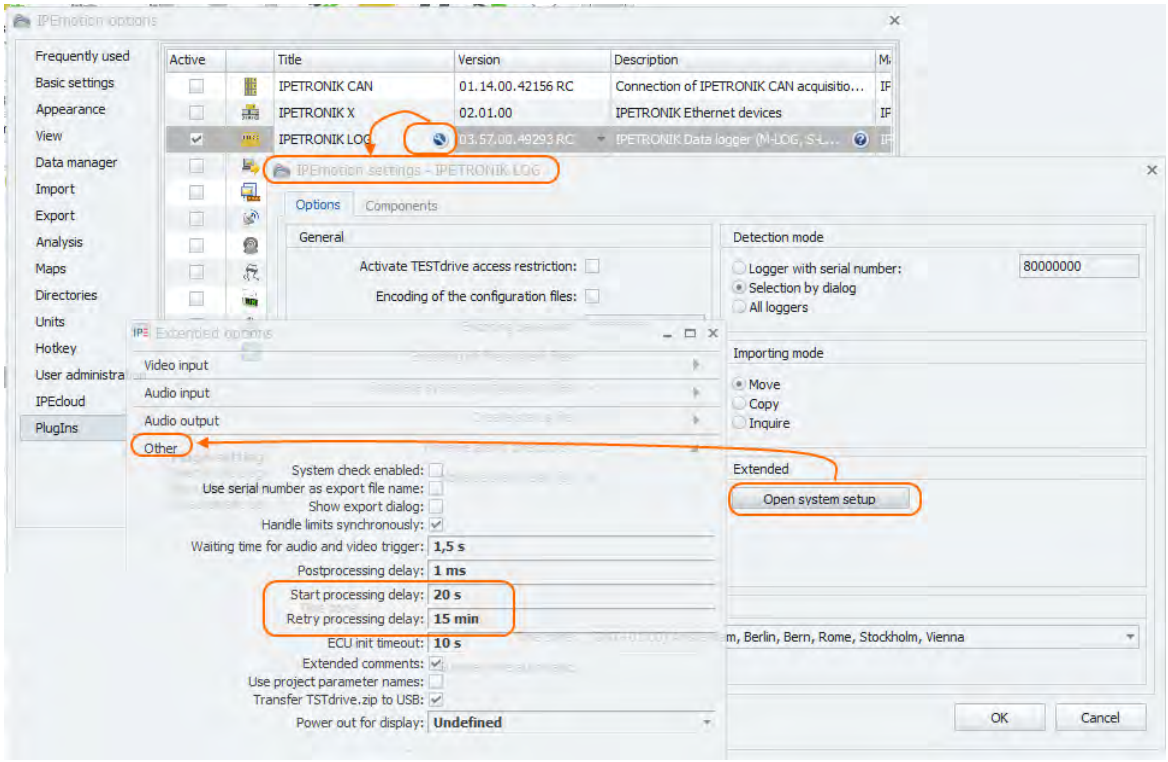
Depending on the traffic group settings, the Quick start feature records measurement data during the logger boot-up and between two consecutive acquisitions (previous acquisition is stopped but the following acquisition is not yet started). The modified function is now providing 3 options:

- **Off** No Quick start recording
- **ON** Quick start recording during boot-up and between measurements
- **Only at start** Quick start recording during boot-up only

The second option (**Quick start ON**) corresponds to the formerly behavior. Selecting **Only at start** will record data only during boot-up, not between stop and restart of an acquisition.




### Post processing delay times




If the Postprocessing has been activated, two specific option can be defined.

- **Start Delay** The postprocessing starts with a delay of x seconds past to measurement start.
  Value has to entered in seconds. Default setting = 20 s. Valid range: 10 s – 5 min.
- **Retry Delay** Restart of the data transfer in case of the unsuccessful attempt (sample cause: FTP server unavailable, insufficient data rate, ...):
  Value has to entered in seconds. Default setting = 900 s. Valid range: 30 s – 30 min.

To modifiy the current settings, enter the extended systems settings under

**Options > PlugIns > PlugIn IPETRONIK-LOG > Options > Open system setup**

## 11.2.3 Data transfer & communication

### WLAN status information

TESTdrive cyclically executes a WiFi SSID scan every 30 seconds in order to report all available WLAN networks and networks that are no longer accessible. The report only contains changes related to the previous scan. The log output of new connections is listed with SSID, alle channels and the signal strength (RSSI in dBm). Connections that are no longer available are listed with the SSID only. The WiFi scan feature is supported exclusively by data loggers with built-in WiFi device (no support for M-LOG/M-LOG V3 with COMgate/COMgate V3).

**Sample of a log output:**

**New contact to WLAN network(s) :**

SSID Channel(s) (RSSI [dBm])

IPE-TESTING ; 1 (-100); 5 (-65); 11 (-71);

Hotspot_Xdtfr ; 1 (-98); 5 (-65); 11 (-71);

Testbench_PT25432 ; 11 (-69);

**Lost contact to WLAN network(s):**

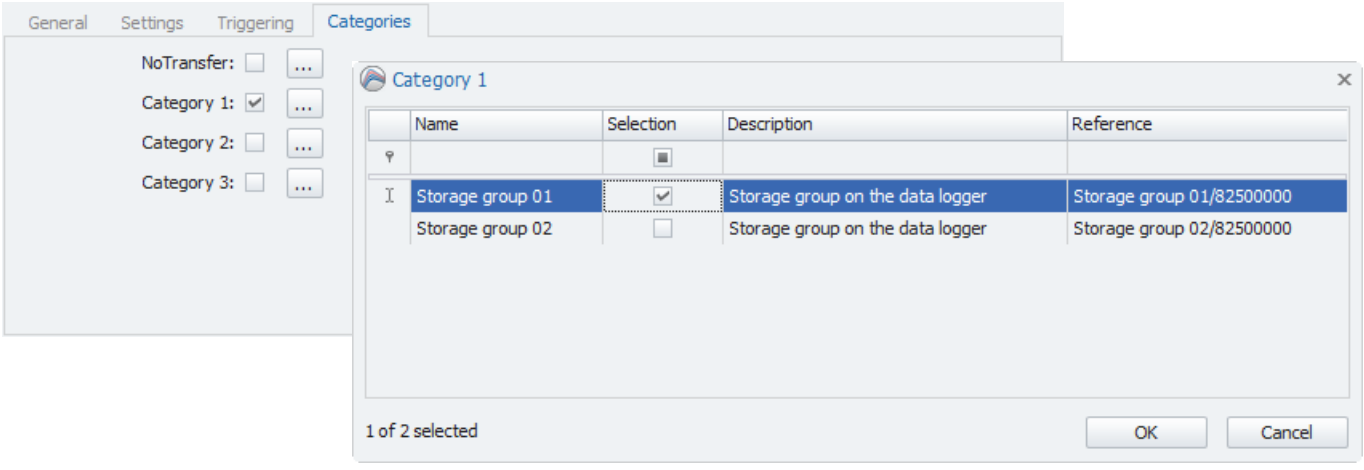
SSID: IPEhub2_03561

SSID: Testbench_PT25377

SSID: Hotspot_Xdtfr

### Category overview

Storage, traffic and statistic groups now provide the new tab **Categories**. Select a group and assign it to one or multiple transfer categories. The overview dialog shows all selected groups in the respective category.



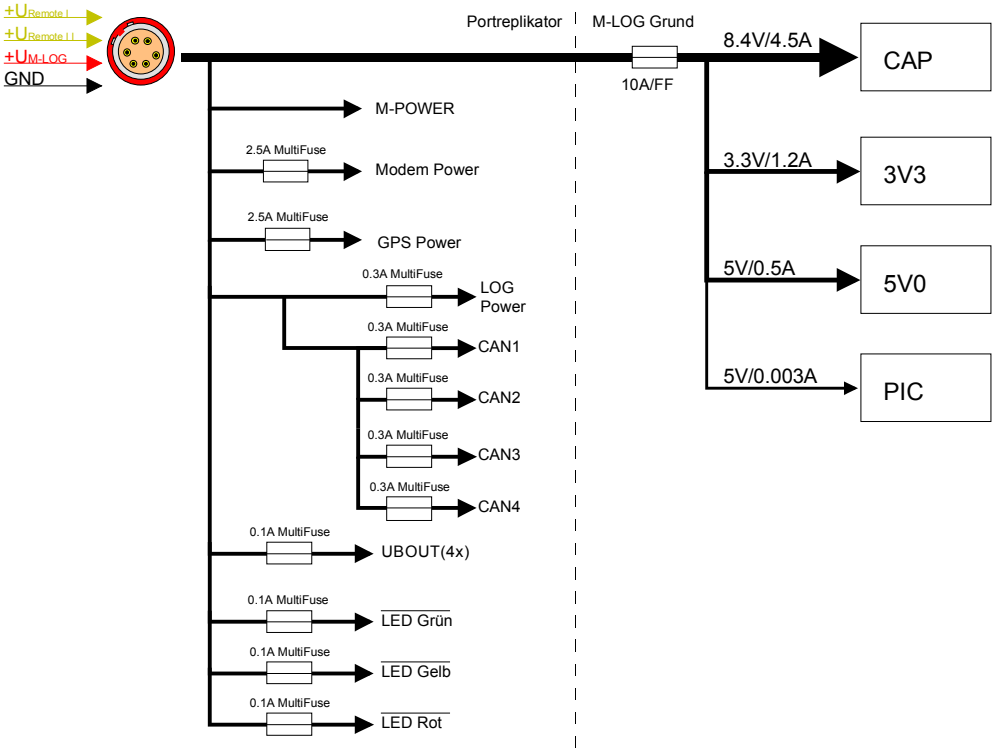
# 12 Appendix

## 12.1 Cable connection and Pin assignment

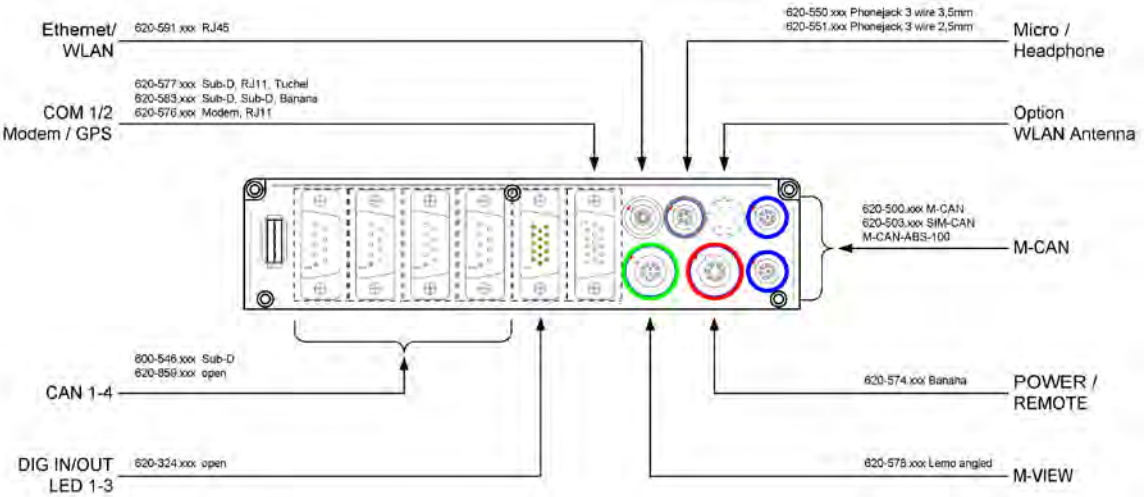
### 12.1.1 M-LOG port replicators

#### Port replicator PR05 (4x Sub D 9, PWR-IN/REM Lemo 1B 6 pin)

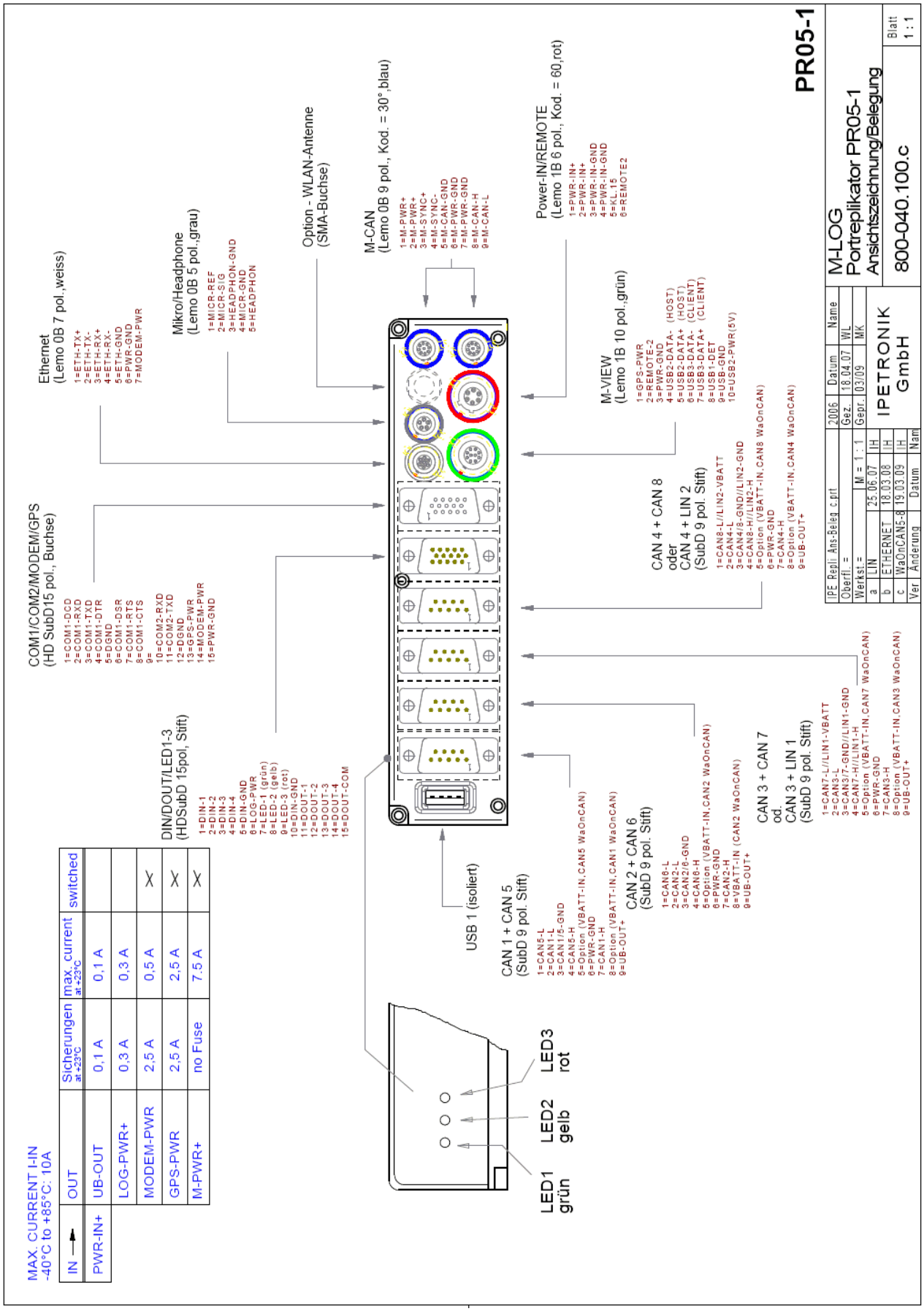
##### Internal circuits PR05




##### Cable reference PR05

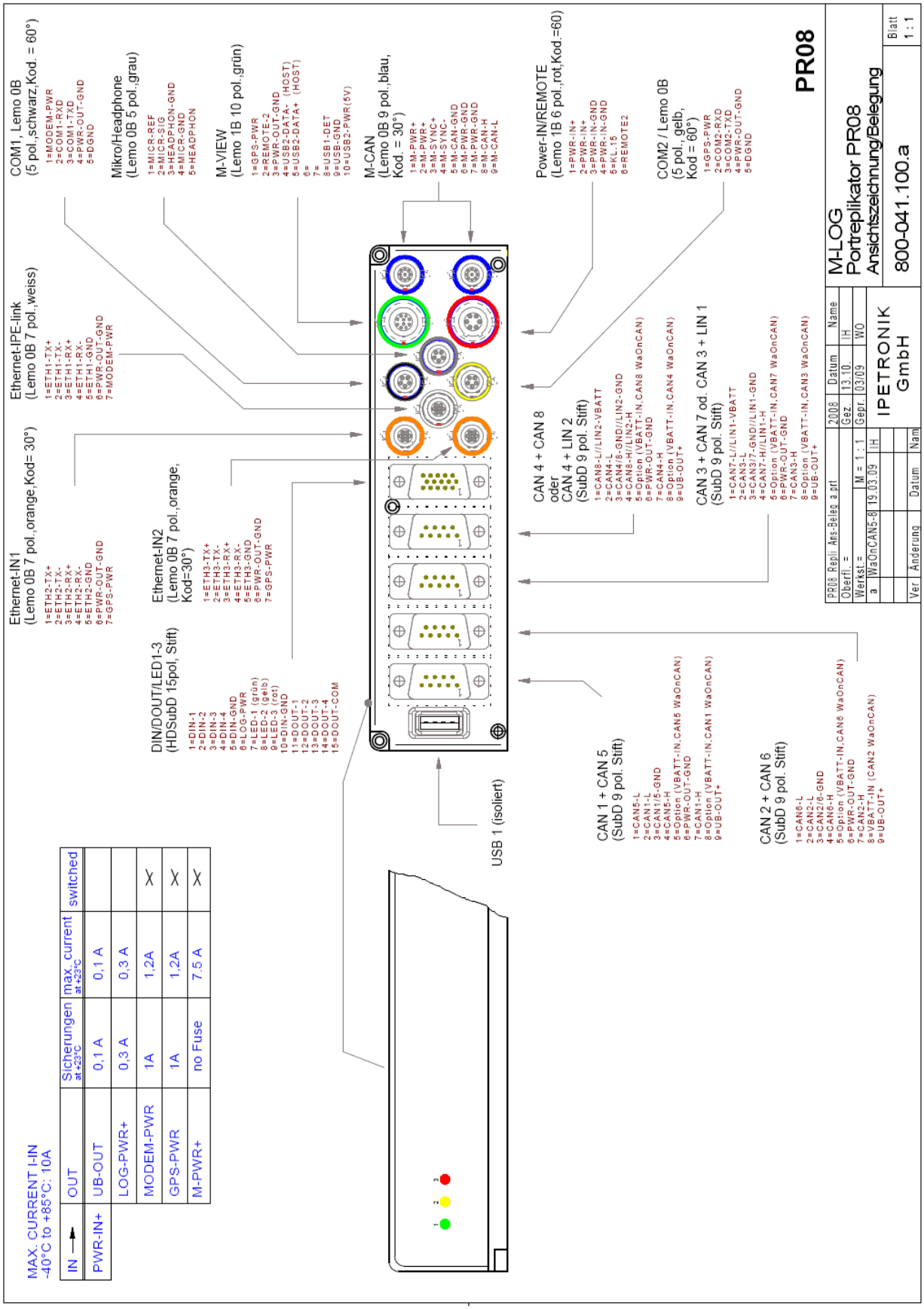


## PIN assignment PR05

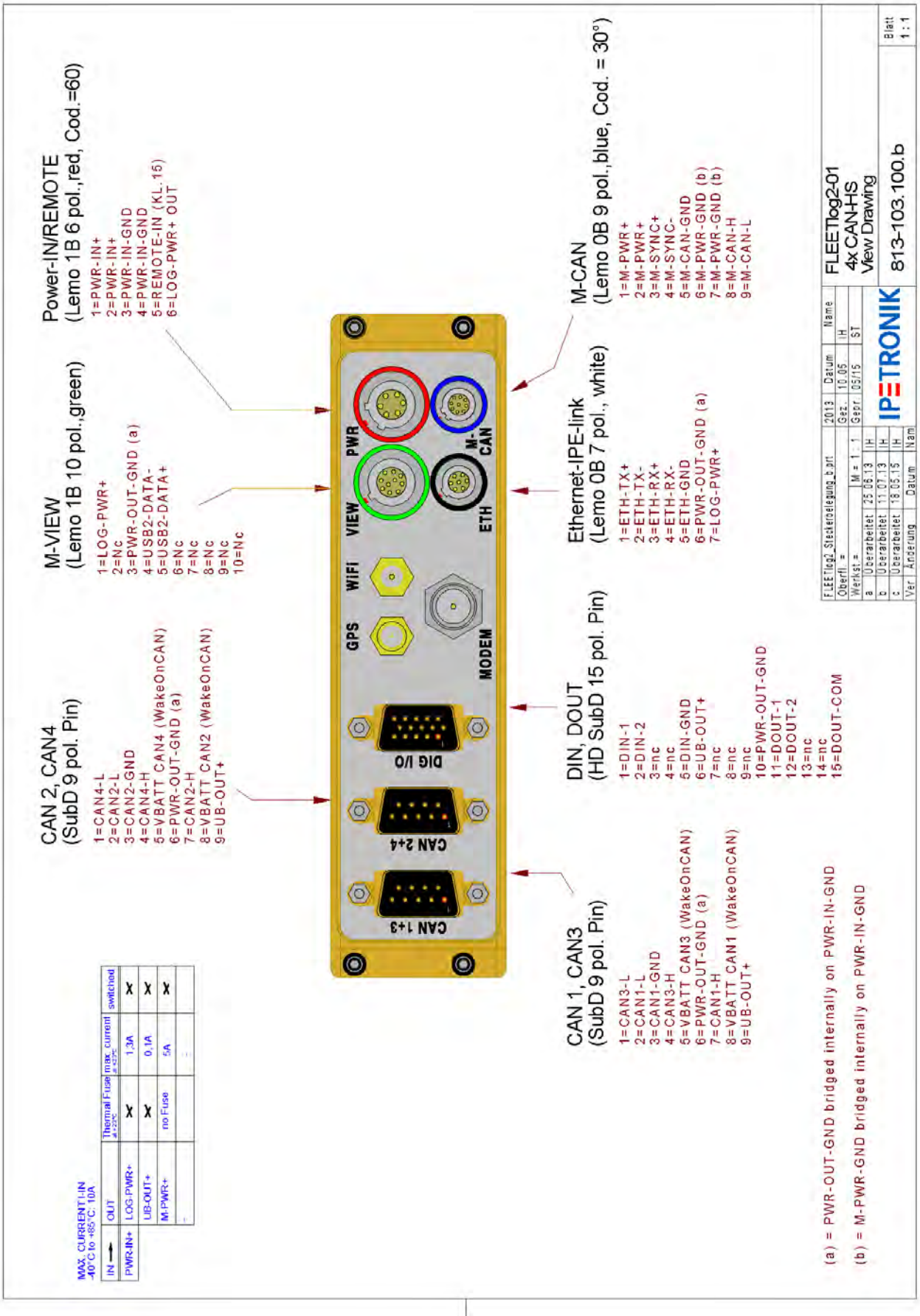


# Port replicator PR08 (4x Sub D 9, ETH, PWR-IN/REM Lemo 1B 6 pin)

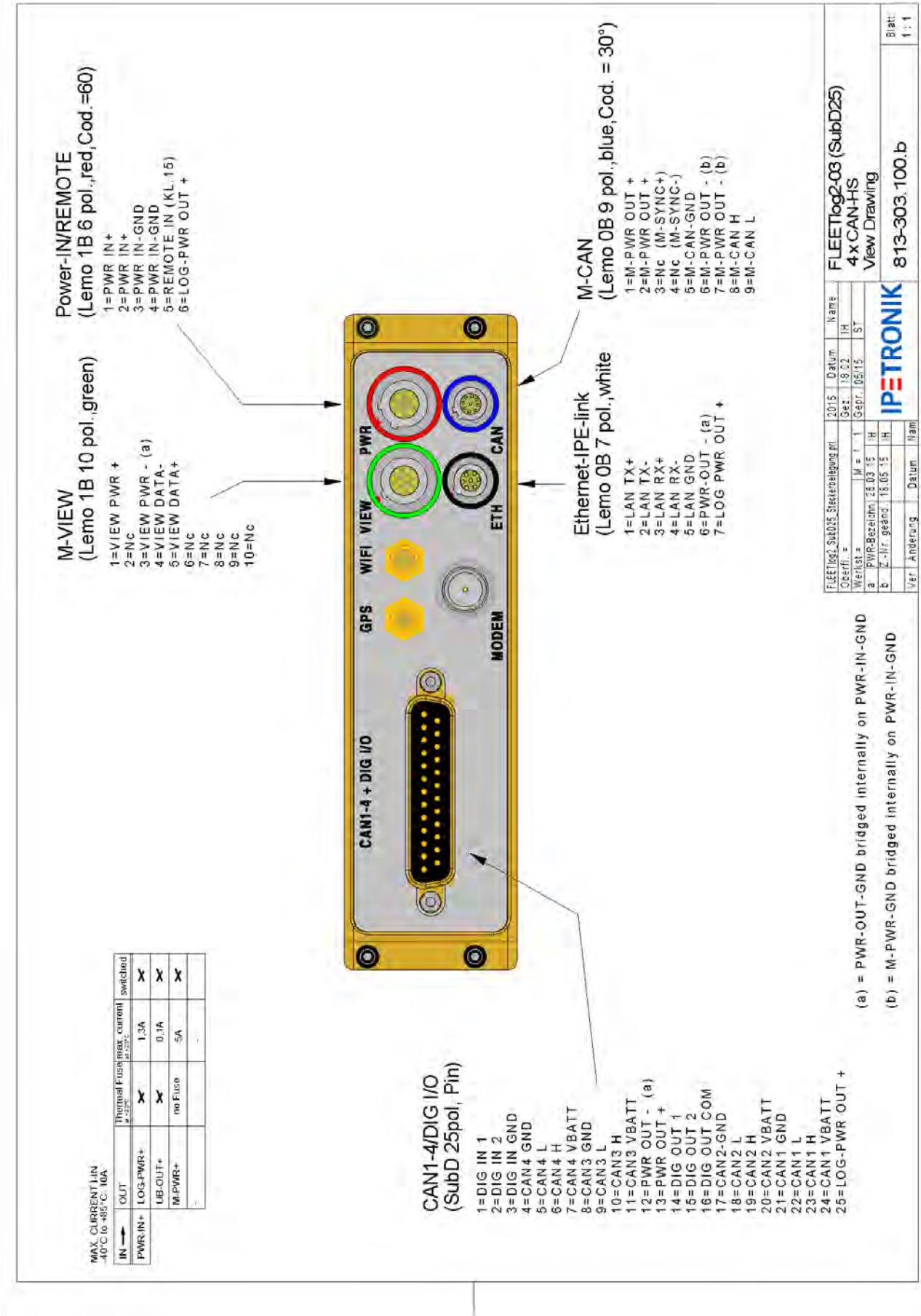
## PIN assignment PR08



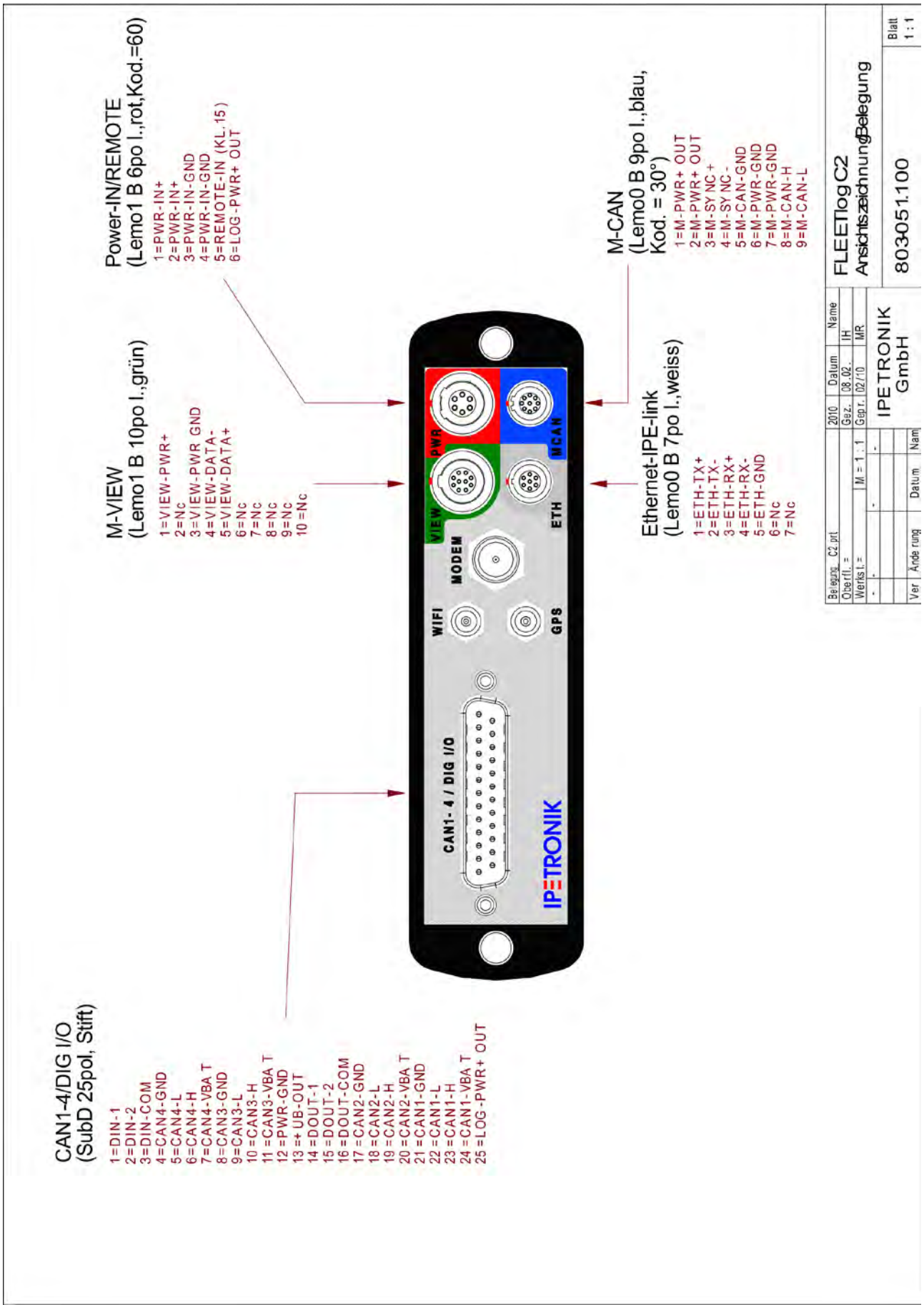
## 12.1.2 FLEETlog2-01



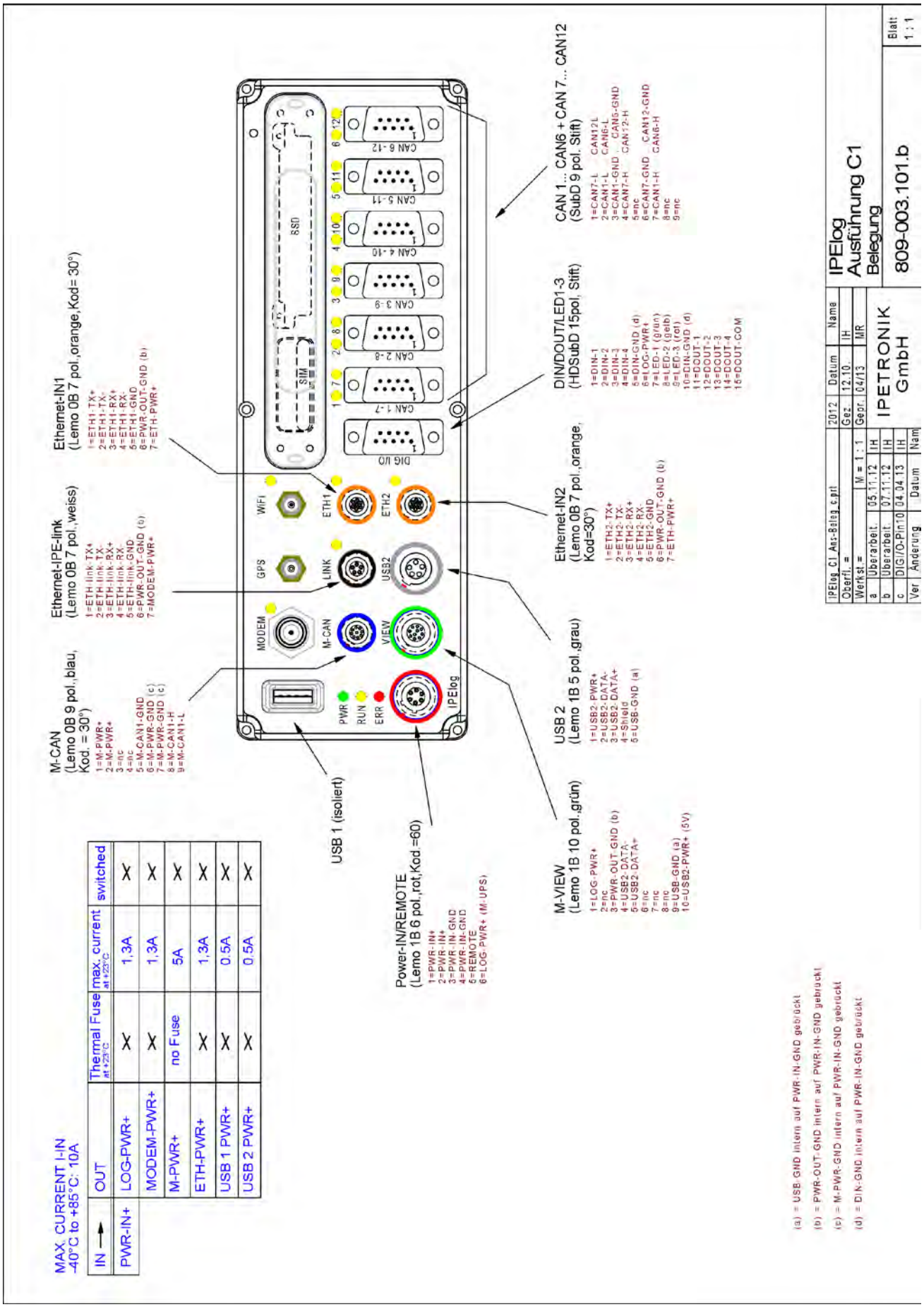
## 12.1.3 FLEETlog2-03



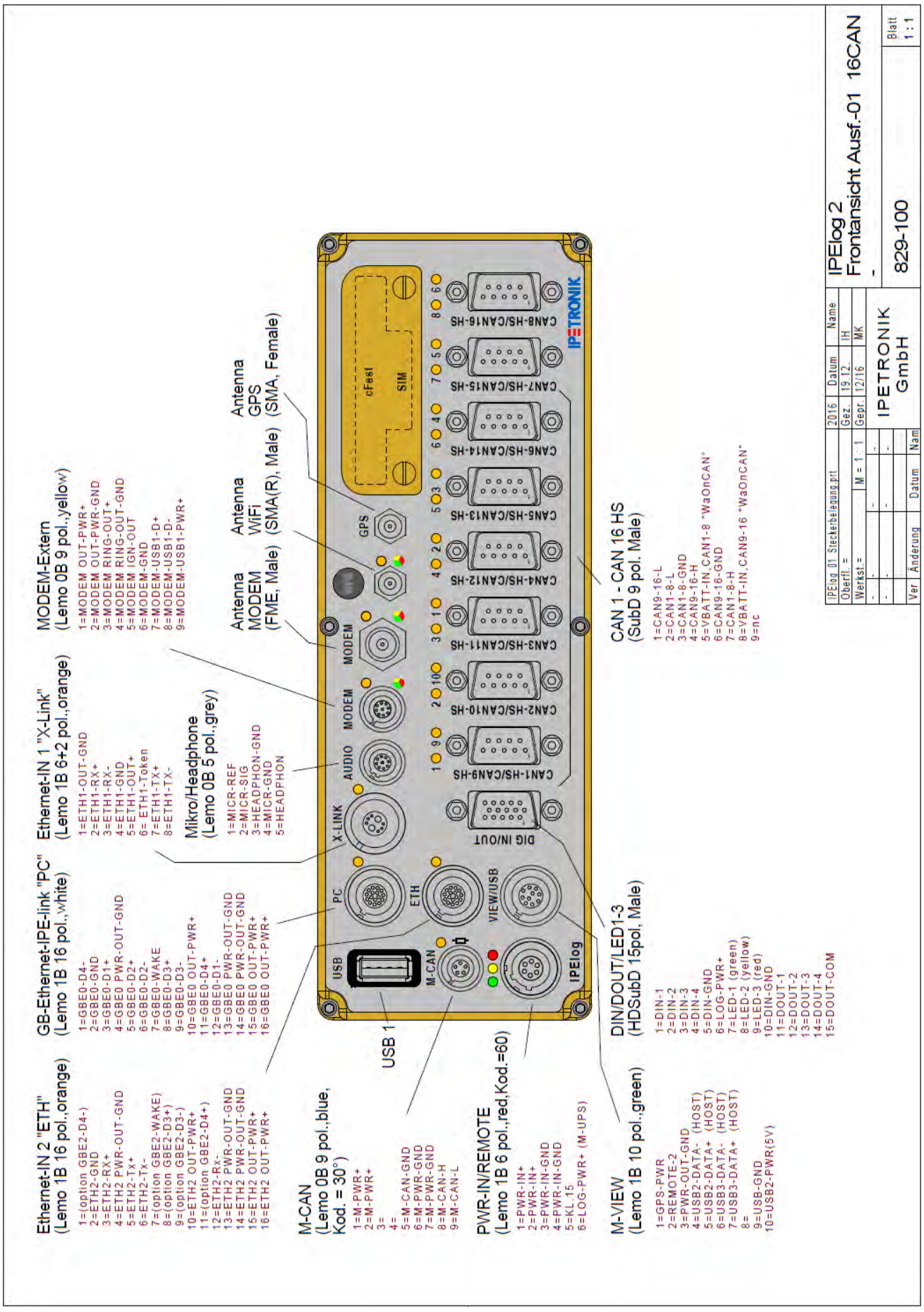
## 12.1.4 FLEETlog



## 12.1.5 IPElog

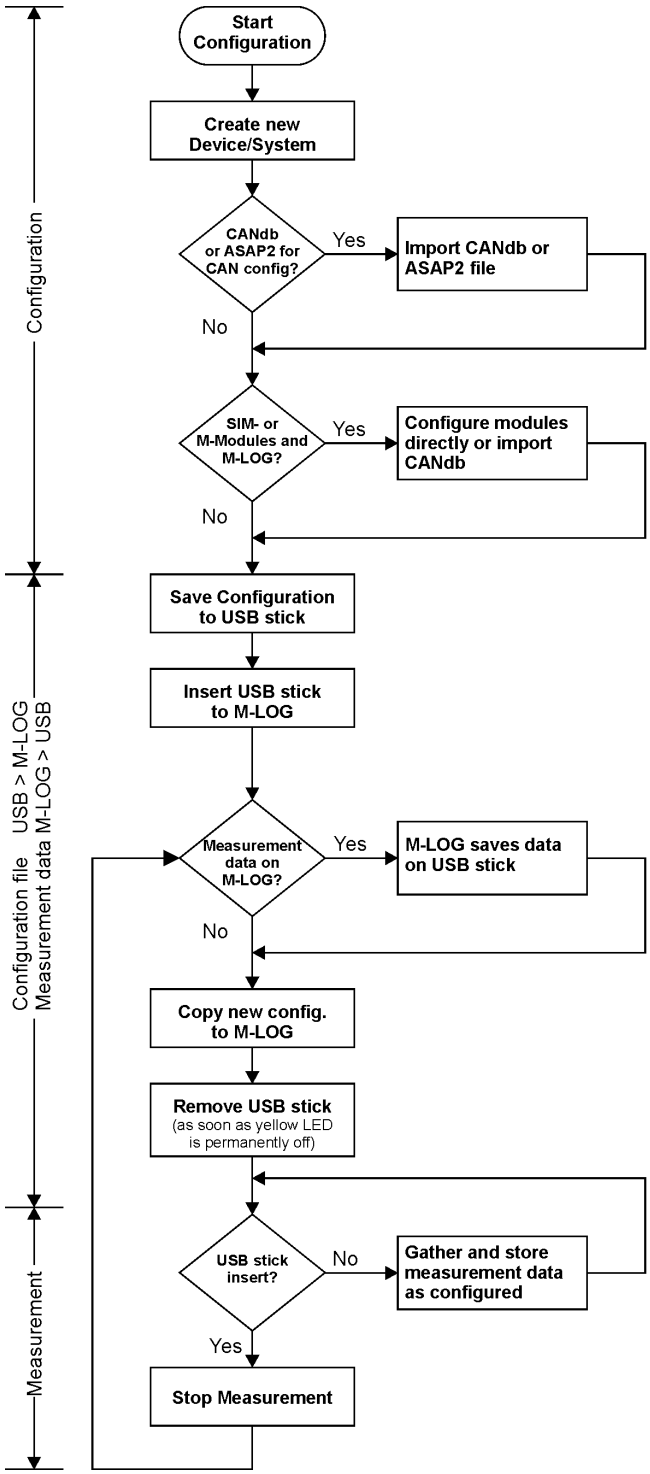

MAX. CURRENT I-IN
-40°C to +85°C: 10A
IN → | OUT | Thermal Fuse at +23°C | max. current at +23°C | switched
PWR-IN+ | LOG-PWR+ | X | 1.3A | X
MODEM-PWR+ | X | 1.3A | X
M-PWR+ | no Fuse | 5A | X
ETH-PWR+ | X | 1.3A | X
USB 1 PWR+ | X | 0.5A | X
USB 2 PWR+ | X | 0.5A | X
M-CAN
(Lemo 0B 9 pol.,blau, Kod. = 30°)
1=M-PWR+
2=M-PWR+
3=nc
4=nc
5=M-CAN1-GND
6=M-PWR-GND (c)
7=M-PWR-GND (c)
8=M-CAN1-H
9=M-CAN1-L
Ethernet-IPE-link
(Lemo 0B 7 pol.,weiss)
1=ETH-link-TX+
2=ETH-link-TX-
3=ETH-link-RX+
4=ETH-link-RX-
5=ETH-link-GND
6=PWR-OUT-GND (b)
7=MODEM-PWR+
Ethernet-IN1
(Lemo 0B 7 pol.,orange,Kod= 30°)
1=ETH1-TX+
2=ETH1-TX-
3=ETH1-RX+
4=ETH1-RX-
5=ETH1-GND
6=PWR-OUT-GND (b)
7=ETH-PWR+
USB 1 (isoliert)
Power-IN/REMOTE
(Lemo 1B 6 pol.,rot,Kod =60)
1=PWR-IN+
2=PWR-IN+
3=PWR-IN-GND
4=PWR-IN-GND
5=REMOTE
6=LOG-PWR+ (M-UPS)
USB 2
(Lemo 1B 5 pol.,grau)
1=USB2-PWR+
2=USB2-DATA-
3=USB2-DATA+
4=Shield
5=USB-GND (a)
M-VIEW
(Lemo 1B 10 pol.,grün)
1=LOG-PWR+
2=nc
3=PWR-OUT-GND (b)
4=USB2-DATA-
5=USB2-DATA+
6=nc
7=nc
8=nc
9=USB-GND (a)
10=USB2-PWR+ (5V)
Ethernet-IN2
(Lemo 0B 7 pol.,orange, Kod=30°)
1=ETH2-TX+
2=ETH2-TX-
3=ETH2-RX+
4=ETH2-RX-
5=ETH2-GND
6=PWR-OUT-GND (b)
7=ETH-PWR+
DIN/DOUT/LED1-3
(HDSubD 15pol, Stift)
1=DIN-1
2=DIN-2
3=DIN-3
4=DIN-4
5=DIN-GND (d)
6=LOG-PWR+
7=LED-1 (grün)
8=LED-2 (gelb)
9=LED-3 (rot)
10=DIN-GND (d)
11=DOUT-1
12=DOUT-2
13=DOUT-3
14=DOUT-4
15=DOUT-COM
CAN 1... CAN6 + CAN 7... CAN12
(SubD 9 pol. Stift)
1=CAN7-L CAN12L
2=CAN1-L CAN6-L
3=CAN1-GND CAN6-GND
4=CAN7-H CAN12-H
5=nc
6=CAN7-GND CAN12-GND
7=CAN1-H CAN6-H
8=nc
9=nc
PWR
RUN
ERR
MODEM
M-CAN
VIEW
IPElog
GPS
WiFi
LINK
USB2
ETH1
ETH2
SIM
SSD
DIG I/O
CAN 1-7
CAN 2-8
CAN 3-9
CAN 4-10
CAN 5-11
CAN 6-12
(a) = USB GND intern auf PWR-IN-GND gebrückt
(b) = PWR-OUT-GND intern auf PWR-IN-GND gebrückt
(c) = M-PWR-GND intern auf PWR-IN-GND gebrückt
(d) = DIN-GND intern auf PWR-IN-GND gebrückt
IPElog_C1_Anz-Belog_c.prt
Oberfl. =
Werkst.=
M = 1 : 1
a Überarbeit. 05.11.12 IH
b Überarbeit. 07.11.12 IH
c DIG I/O-Pin10 04.04.13 IH
Ver Änderung Datum Nam
2012 Datum Nama
Gez. 12.10. IH
Gepr. 04/13 MR
IPETRONIK GmbH
IPElog
Ausführung C1
Belegung
809-003.101.b
Blatt: 1 : 1

## 12.1.6 IPElog2



## 12.2 Starting up

### 12.2.1 Overview Configuration & Measurement




Configuration via IPEmotion (license required)

Create new or read and process existing configuration.

Configure bus measurement input by import of the respective description file

Configure via IPEmotion

Save configuration as an MCF file (*.mcf).

In case M-LOG is switched off, > switch on the module before. (Do not switch on M-LOG with the USB stick insert!)

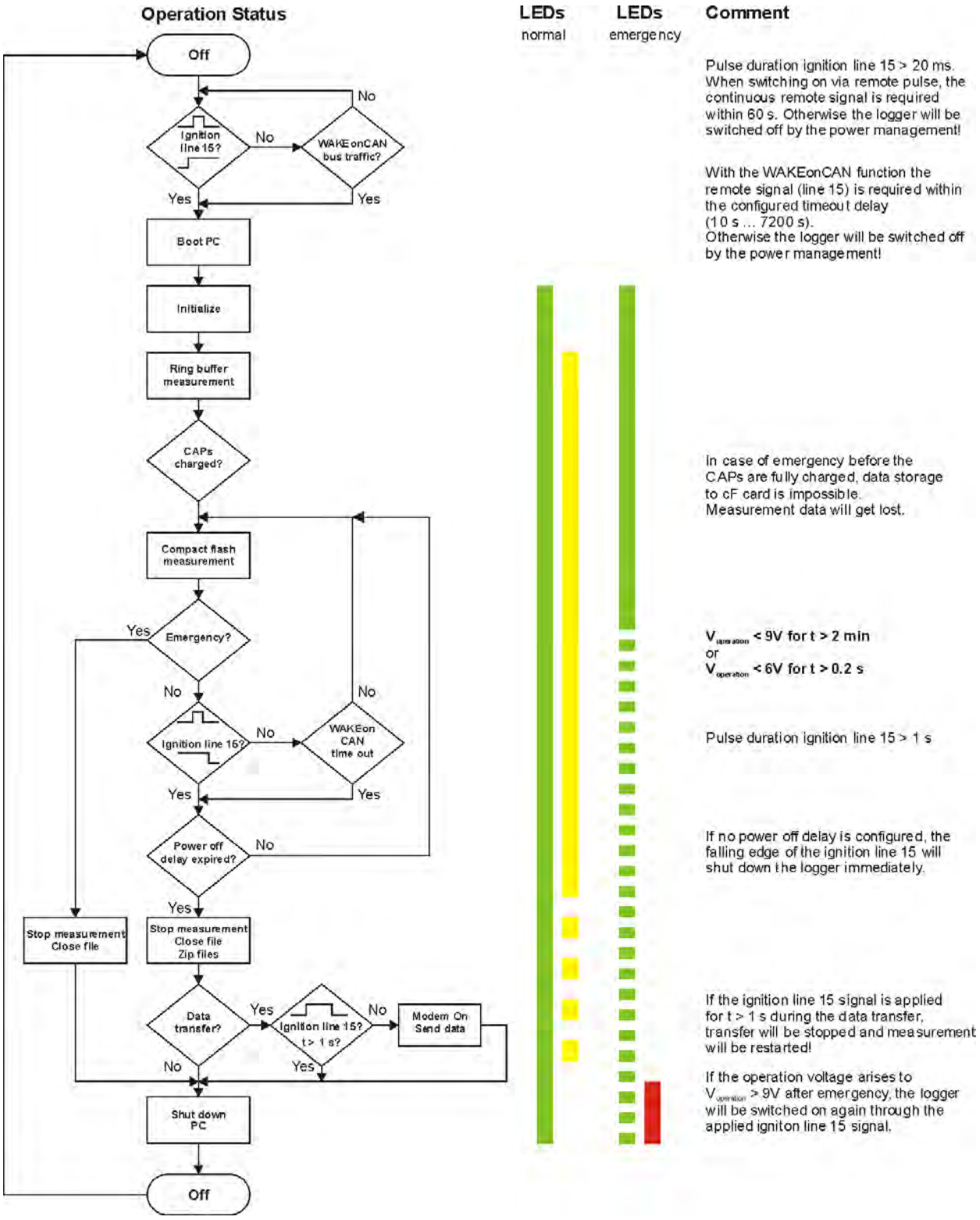
If not already existing, M-LOG creates an specific subdirectory and saves measurement data and the latest configuration to the USB stick. The name of the subdirectory consists of the serial number of M-LOG.

Transfer new configuration from the USB stick to M-LOG

M-LOG automatically starts the measurement application and reads the current configuration.

M-LOG will switch off as soon as the remote input becomes inactive.

## 12.2.2 Flow chart of the measuring process



## 12.3 Practical examples

### 12.3.1 Calculating memory footprint

The memory footprint per storage group results from the following data

- Memory footprint of acquisition channels +
- Memory footprint of time channel (relative and absolute time channel) +
- Memory footprint of header (description file)

**Memory footprint of acquisition channels**

Acqu. duration [s] x storage rate [1/s] x channel count x 2 Byte = memory footprint in Byte

Memory footprint in Byte / 1024 = memory footprint in kByte

**Memory footprint of time channel (IPE:Clock = relative time channel)**

Acqu. duration [s] x storage rate [1/s] x 4 Byte = memory footprint in Byte

Memory footprint in Byte / 1024 = memory footprint in kByte

Do not mix the time channel (relative) with the time stamp channel (absolute = Date, time). The time stamp channel is only recorded if activated in storage group dialog.

**Memory footprint of header file**

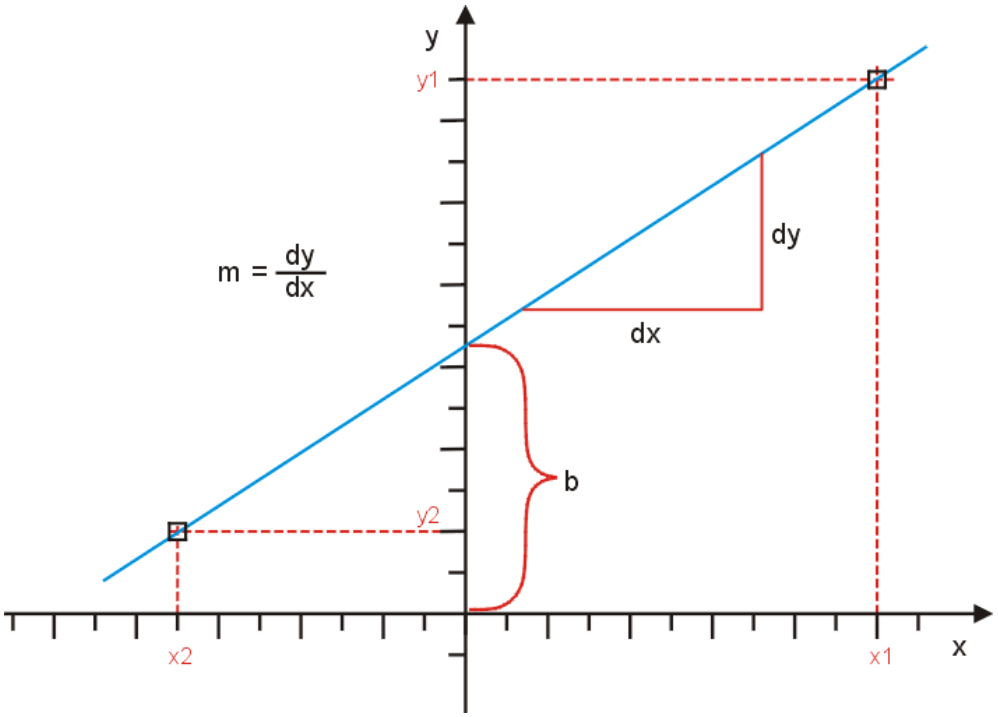
The memory footprint of the header file depends on the size of acquisition configuration (channel count, sampling rate, different data formats, ...) and cannot be defined by a general formula. The size of the header file is usually much smaller so that it can be neglected. Exceptations are short acquisitions or acquisitions with many channels at low storage rate (< 1 Hz).

## 12.3.2 Linear signal scaling

The conversion of a raw value (binary value, e.g. in a CAN message) into a physical value (value with unit) is effected with the scaling. IPETRONIK offers the scaling calculator and supports linear scaling with the line equation as factor/offset or 2 point scaling.

The scaling of a voltage or current signal (sensor output) into a corresponding physical or percentage value is effected in the same way. The following examples show the connections.

### Mathematical basics for the linear equation




**Point gradient form**

y = m * x + b
b = y - m * x

m = Gradient (gain)
b = constant (offset)

**Two points form**

m = dy / dx

m = (y1 – y2) / (x1 – x2)

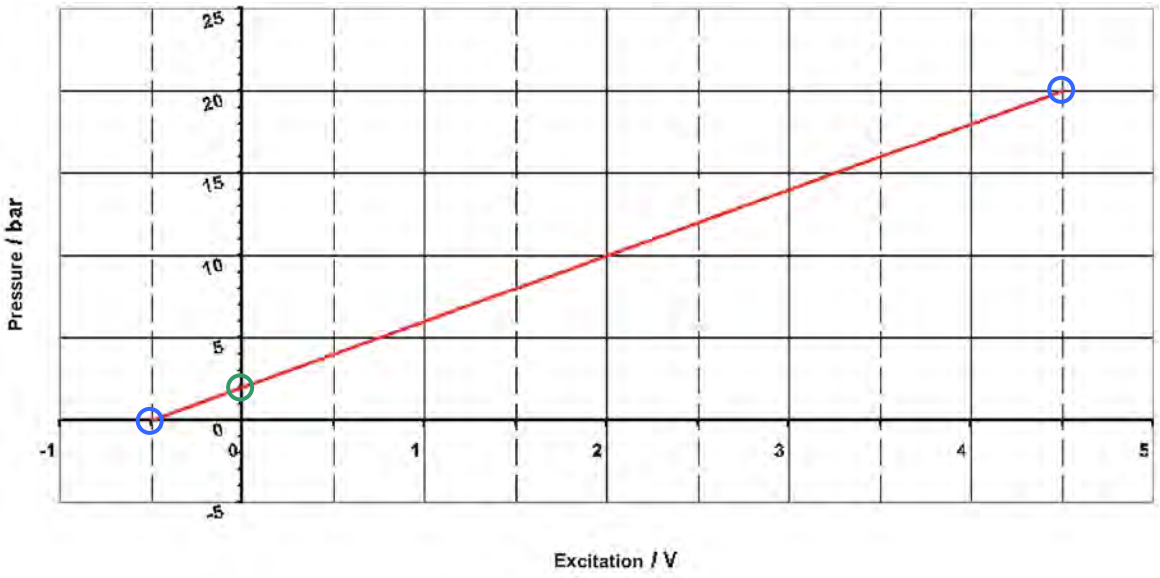
m = Gradient (gain)

**Calculation and explanation**

1. The linear equation **y = m * x + b** shows the mathematical connection.
2. Calculate the **m** gradient within any input range (signal) and the related output range (physical value).
3. Calculate the offset **b** by using the x and y values for a known point.
4. Calculate, if required, further y values by using the corresponding x values and the equation, e.g. for calculating the physical values for another input range (Channel min, max).

## Example pressure sensor

A pressure sensor has an output signal of -0.5 to 4.5 V in the acquisition range 0 … 20 bar.
The voltage signal is converted to the physical value with the linear scaling.

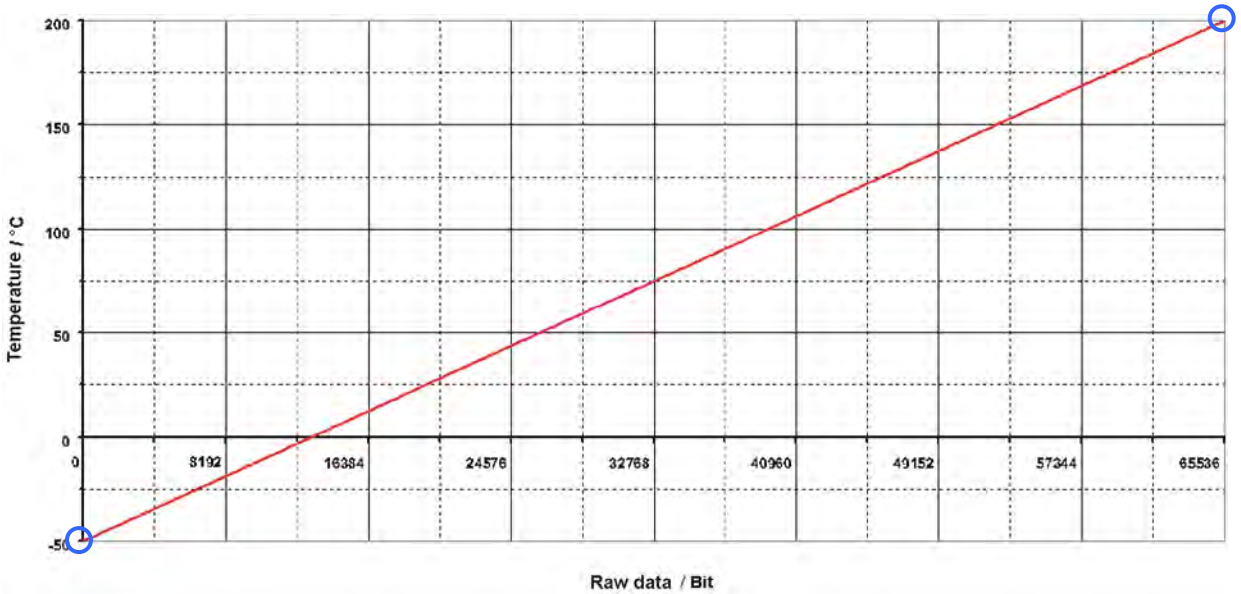



| x1 | x2 | y1 | y2 | m | b' | b |
|---|---|---|---|---|---|---|
| 4.5 | -0.5 | 20 | 0 | 4 | 2 | 2 |

## Example CAN raw value in the Word unsigned format als temperature

A temperature signal is a CAN message in the Word unsigned format. The value range of 0… 65535 (16 Bit) corresponds to a temperature range of -50 °C … +200 °C.

*Please note that the output range has an offset of –50 °C. This must be respected at calculating: (b' = Offset without output offset, b = Offset + output offset).*

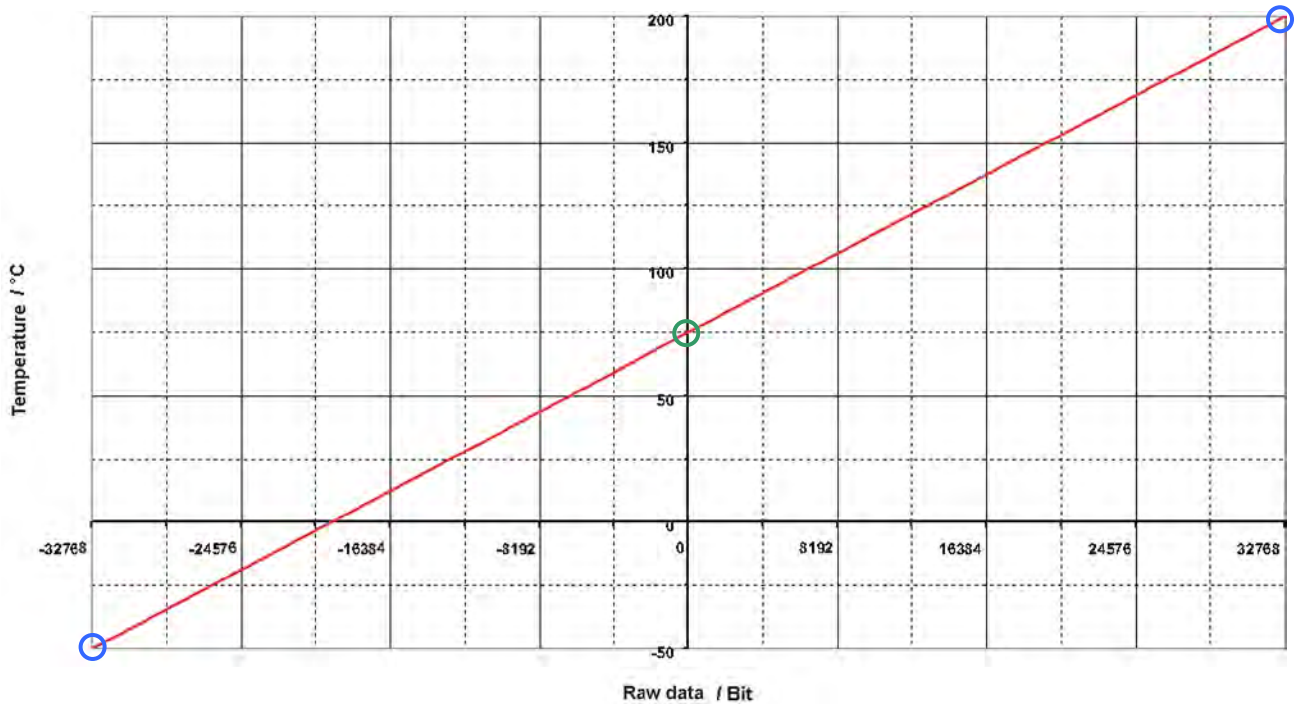



| x1 | x2 | y1 | y2 | m | b' | b |
|---|---|---|---|---|---|---|
| 65535 | 0 | 200 | -50 | 0.0038147 | 0 | -50 |

## Example CAN raw value in the Word signed format as temperature

A temperature signal is a CAN message in the Word signed format. The value range of -32768… 0 ... 32767 (16 Bit) corresponds to a temperature range of –50 °C … +200 °C.

*Please note that the output range has an offset of –50 °C. This must be respected at calculating: (b' = Offset without output offset, b = Offset + output offset).*




| x1 | x2 | y1 | y2 | m | b' | b |
|---|---|---|---|---|---|---|
| 32767 | -32768 | 200 | -50 | 0.0038147 | 125.0019 | 75.0019 |

## 12.4 Status messages

### 12.4.1 Most important status messages

Following message types are defined:

**I Information W Warning E Error D Debug**

**Type number: xxx-xxx-xxxx**

Number corresponds to the number of the type plate of M-LOG (entry in hw_descr.xml).

**Wait max. 3min for write permission (power good)**

M-LOG waits until CAPs are loaded (status message "Power good" from PIC). If status is “Power good“, data is written to flash. If this status is nor reached, (PIC) switches off M-LOG after 3 min.

**Power good**

Message (from PIC) that CAPs are loaded.

**Debounce remote signal 1000 ms**

Remote signal must be on at least for 1 s to reach “ON“ status. (Debouncing of remote signal)

Remote signal is detected as such if excitation > 6.5 V at PIN terminal:15 of PWR-IN/REM socket.

**Watchdog active**

PIC transfers control of M-LOG to TESTdrive.

Function: Testdrive cyclically describes a storage range in Powermanagement (PIC) (toggling bit). If this toggling fails for more than two minutes, M-LOG is switched off by Powermanagement (PIC).

**Free disk space: xxx/xxx**

Indicates the available total memory space. TestDrive 3.09 shows a "Free disk space: xxxx" at the left bottom display window. It shows the available space for data storage (40% of total memory space). Initially, this value fluctuates intensively but stabilizes with a longer acquisition because it is recalculated permanently.

**Time left: xx xx:xx:xx**

Display bottom left in monitor window. Meaning: d hh:mm:ss

**Power bad**

If excitation is too low, status is “Power bad“.

Info: This message is not concerned with CAPs contrary to "Power good" message.

**Can`t initialize communication mediums**

Dev_conf.xml is not included in Config, i.e. no data transfer activated.

**Shutdown in 55 min**

If data postprocessing (zipping, establishing, data transferring, data sending, etc.) is not completed within 55 min, Testdrive shuts down and data remains on logger.

**Emergency shutdown in 60 min**

Logger is unconditionally shut down after 60 min. Powermanagement of M-LOG controls the device.

**Time left: xx  xx:xx:xx**

Display bottom left in monitor window. Meaning:  d  hh:mm:ss

**Power bad**

If excitation is too low, status is “Power bad“.

Info: This message is not concerned with CAPs contrary to "Power good" message.

**Can`t initialize communication mediums**

Dev_conf.xml is not included in Config, i.e. no data transfer activated.

**Shutdown in 55 min**

If data postprocessing (zipping, establishing, data transferring, data sending, etc.) is not completed within 55 min, Testdrive shuts down and data remains on logger.

**Emergency shutdown in 60 min**

Logger is unconditionally shut down after 60 min. Powermanagement of M-LOG controls the device.

### 12.4.2 Warning and error messages after program update

#### Program options without license

TESTdrive Version 3.17 includes a license software to check the use of logger / TESTdrive options. This requires sending new license keys to the logger.

After a TESTdrive program update, potential warning messages can indicate locked options. TESTdrive checks if used functions are really unlocked on the logger. Following warning is written into log file if e.g. all inputs of a card with 4 CAN inputs is used but only 2 are unlocked.

**01.04.2009 14:11:45 W CAN1 : Upper limit of licenced CAN interfaces reached. Max= 2**

Please contact the sales team at +49 7221 / 9922 – 222 to assist you with warning and error messages, as well as, licensing.

*TESTdrive V03.22 deactivates the non-licensed functionality and warns with the red status LED (temporarily at acquisition start). Depending on the missing license, acquisitions are limited (e.g. possible calculations but no classification) or disabled (e.g. at missing license for interface).*

*If an extender is used at the logger, an additional  license for the extender is required!*

The system shows the following warning if an extender is used with a non-licensed CCP protocol:

**16.02.2011 15:36:01 E Extender.80200011: Error reading XML-Buffer at line 11**

## 12.5 Description of TESTdrive files

TESTdrive provides the measurement files as zip archives. TESTdrive creates separate zip files for every acquisition:

MEA_xxxx.zip (Acquisition data + header file + current configuration)

LOG_xxxx.zip (Protocol file for data acquisition)

A data set of an acquisition always includes a header file (AABBCCC.DAT), at least one acquisition file in DIAdem format, as well as, the corresponding acquisition configuration (e.g. IPEmotion.isf).

The names of the single acquisition files are generated according to the **AABBCCCC.DDD** structure:

| | | |
|---|---|---|
| **AA** | = | Data type |
| **BB** | = | Consecutive number allocated based on all data types used with the configuration |
| **CCCC** | = | Consecutive number of an acquisition |
| **DDD** | = | File extension |

After reaching the consecutive file number 9999 (CCCC), counting starts again with 0001!

If the file with this number still exists, it is overwritten with new data!

**AA** data type indicates the type of data:

| | | |
|---|---|---|
| **DO** | = | **D**ata **O**nline (storage group with time channel) |
| **PM** | = | **P**ost **M**ortem data of a ring buffer group (storage group with time channel) |
| **CO** | = | **(C)** Classification **O**nline (storage group with statistic data, without time reference) |
| **A0** | = | **A**udio **O**nline |
| **V0** | = | **V**ideo **O**nline |
| **J** | = | **J**ob data = Diagnostic data |
| **ST** | = | Min-Max list |
| **TBQS, T** | = | CAN/LIN traffic acquisition (during or after boot process) |
| **MV** | = | Maneuver recording |

The consecutive **BB** number clearly assigns data within one data type. Several storage groups for instance can be detected with real-time data, which are distinguished from each other by this number.

The consecutive **CCCC** number differentiates between single acquisitions. Each acquisition is therefore clearly defined.

The **DDD** file extension defines the data format as follows:

| | | |
|---|---|---|
| **DAT** | = | DIAdem header file |
| **T64** | = | DIAdem time channel with 64 bit resolution |
| **W8** | = | DIAdem data with 8 bit unsigned (BYTE) |
| **W16** | = | DIAdem data with 16 bit unsigned (WORD) |
| **W32** | = | DIAdem data with 32 bit unsigned (WORD) |
| **I16** | = | DIAdem data with 16 bit signed (INTEGER) |
| **I32** | = | DIAdem data with 32 bit signed (INTEGER) |
| **R32** | = | DIAdem data with 32 bit in floating point displaying (REAL) |
| **R64** | = | DIAdem data with 64 bit in floating point displaying (REAL) |
| **WAV** | = | Audio file in WAV format |
| **AVI** | = | Video file in AVI format |
| **CSV** | = | Comma Separated Values |
| **Jxx** | = | Binary file with job result |

Recorded signals are divided into different files accordingly to data typet, i.e. all 8 bit signals unsigned are in a *.W8 file, all 32 bit signals signed in a *.I32 Datei, etc.

**Important advices:**

TESTdrive defines the number of the storage group during the initialization process. An assignment of the storage group number to the order in the configuration interface is not given.

All information of a storage group is stored in the DAT file. The DAT file has a 8 bit ASCII format (ANSI code page 1252, ISO 8859-1).

The storage group name can be defined with IPEmotion and is also stored in the DAT file.

All project information are stored in the DAT file (vehicle no., project name, etc.). At working with classifications, the DAT file contains additional fields to define the classification in detail.

## 12.5.1 Data types

### Time-related data (Storage group)

The header file and the corresponding data files are created for every storage group (= signals with common storage rate).

**Example (Acquisition no. 699 > DOBBCCCC.DDD)**

| | | |
|---|---|---|
| **Storage group 1** | DO010699.DAT | Header |
| | DO010699.R32 | 32 Bit (Real) |
| | DO010699.W16 | 16 Bit (Word unsigned) |
| | DO010699.W32 | 32 Bit (Word unsigned) |
| **Storage group 2** | DO020699.DAT | Header |
| | DO020699.W8 | 8 Bit (Byte unsigned) |
| | DO020699.W32 | 32 Bit (Word unsigned) |
| **Storage group x** | DO0x0699.DATHeader | |
| | DO0x0699.W8 | 16 Bit (Word unsigned) |

### Statistics

TESTdrive saves classification data in DIAdem format, which includes all parameters in one header file. This header file includes additional information about classification description. Corresponding to the storage groups and the general DIAdem conventions, binary data is included in the same files. These files are separated accordingly to data type and are defined by an extension corresponding to data type, e.g. *.W32, *.R64.

The results of several classifications of the same data type are stored in one binary file. Most classification types create results of the W32 data type. The retention time classification can create different data types.

**Example (Acquisition no. 699 > COBBCCCC.DDD)**

| | | |
|---|---|---|
| **Header** | CO010699.DAT | Header |
| **Classification m … n** | CO010699.W32 | 32 Bit (Word unsigned), can include several classifications |
| | CO010699.R64 | 64 Bit (Real), can include several classifications |

## Audio recording

A DIAdem acquisition and one or several audio files (WAV format) are created with an audio recording.

The DIAdem acquisition contains the trigger event, the WAV file includes the audio data.

Every audio file is assigned to one triggering event. The names of the audio files include the counter value of the trigger channel to create a direct reference within the acquisition data. The respective audio file is recorded for the time of an activated trigger.

**Example DIAdem file (Acquisition no. 699 > A000CCCC.DDD)**

| | | |
|---|---|---|
| **Trigger channel** | A0000699.DAT | Header |
| | A0000699.W16 | 16 Bit (Word unsigned) |

**Example audio file (Acquisition no. 699 > ABBBCCCC.WAV)**

| | | |
|---|---|---|
| **Audio sequence 1** | A0010699.WAV | Audio data of 1. trigger event |
| **Audio sequence 2** | A0020699.WAV | Audio data of 2. trigger event |
| **Audio sequence x** | A00x0699.WAV | Audio data of 3. trigger event |

## Video recording

A DIAdem acquisition and one or several video files (JPG = single image or AVI = video sequence) are created with an video recording.

The DIAdem acquisition contains the trigger event, the JPG or AVI file includes the video data.

Every video file is assigned to one triggering event. The names of the video files include the counter value of the trigger channel to create a direct reference within the acquisition data. The respective video file is recorded for the time of an activated trigger. Exactly one single image is saved per trigger event, independent of the trigger duration.

**Example DIAdem file (Acquisition no. 699 > V000CCCC.DDD)**

| | | |
|---|---|---|
| **Trigger channel 1 (Video)** | V0000699.DAT | Header |
| | V0000699.W16 | 16 Bit (Word unsigned) |
| or | | |
| **Trigger channel 1 (Image)** | I0000699.DAT | Header |
| | I0000699.W16 | 16 Bit (Word unsigned) |

**Example video file (Acquisition no. 699 > VBBBCCCC.AVI)**

| | | |
|---|---|---|
| **Video sequence 1** | V0010699.AVI | Video data |
| **Video sequence 2** | V0020699.AVI | Video data |
| or | | |
| **Image 1** | I0010699.JPG | Image data |
| **Image 2** | I0020699.JPG | Image data |

## Diagnostic acquisition

**Error memory and non-recurring data**

TESTdrive creates additional files for error memory and non-recurring data. This CSV file (BDJDcccc.CSV) is an overview and includes all information about the completed jobs. Every successful job is saved in a binary file (BDS1cccc.Jxx), which contains the actual data.

A CSV file**, as well as, the corresponding jobs are created after the completed acquisition of UDS services.

**** A CSV file is only created in binary mode, not in trace modeT.**

The names of the single acquisition files are generated according to **BDPECCCC.DZZ** structure:

| | |
|---|---|
| BD**P**ECCCC.DZZ | with P = U (UDS protocol), P = K (KWP protocol) |
| BDP**E**CCCC.DZZ | with E = ECU number (1 ... 9) |
| BDPECCCC.**D**ZZ | with D = J (Job or binary file), D = T (Trace file) |
| BDPECCCC.D**ZZ** | with ZZ = Job number (01 ... 99) |

**Example non-recurring data (Acquisition no. 699 > BDPECCCC.DZZ)**

| | | |
|---|---|---|
| **Trace file** | BDU10699.T01 | UDS protocol, ECU no. 1, Acquisition no. 699, Job no. 1 |
| **Binary file** | BDU10699.J01 | UDS protocol, ECU no. 1, Acquisition no. 699, Job no. 1 |
| **CSV file** | BDJD0699.CSV | Job overview of binary files |

## Measurement status file

Measurement status file in XML format offers information about the process of a completed acquisition. This includes:

- Acquisition start and stop (… in standardized XML format “DataTime”)
- General system information (hardware, TESTdrive version,…)
- Storage group trigger
- Acquisition / Diagnosis information
- Limit violations
- Maneuver detection

Measurement status file is created after acquisition stop and is stored in the zip container of the acquisition (MEA_xxxx.zip) or attached to the status e-mail if this file creation has previously been activated in IPEmotion **Options > PlugIns > IPETRONIK LOG > PlugIn specific settings > Options > General > Create measurement status file**.

**Example Measurement status file (measurement number 699 > MSxxyyyy.xml)**

xx = Append number, yyyy = Measurement number

| | |
|---|---|
| **Measurement status file 1** | MS010699.xml |
| **Measurement status file 2** | MS020699.xml |

If an acquisition is later continued (Append mode), TESTdrive increases the append number in the file name.

## Min-Max list (STG file)

TESTdrive creates a separate statistic file (STG file) to record minimum and maximum values as well as first and last valid value of selected signals. The STG file is created at the end of each measurement, if the **Logger processing** contains a **Statistic group** containing at least one channel. The statistic file is stored in the zip container of the acquisition (MEA_xxxx.zip)

**Example STG file (Acquisition no. 699 > STBBCCCC.STG)**

| | | |
|---|---|---|
| **STG file 1** | ST010699.STG | Min / Max data of storage / process group 1 |
| **STG file 2** | ST020699.STG | Min / Max data of storage / process group 2 |

## Traffic recording (CAN, LIN)

TESTdrive saves traffic acquisitions in binary format. A traffic acquisition can include two binary files:

- Traffic data recorded by TESTdrive during boot time

- Traffic data recorded by TESTdrive during run time

Advice: Storage of CAN traffic data and LIN traffic data in the same file

The names of the single acquisition files are generated according to **TBBBCCCC.BIN** structure:

| | |
|---|---|
| **T**BBBCCCC.BIN | Traffic data recorded by TESTdrive |
| T**BBB**CCCC.BIN | Consecutive number within an acquisition |
| **TBQS**CCCC.BIN | Traffic data recorded by micro controller |

## Maneuver recording

TESTdrive creates a file in ASCII format for maneuver recordings.

The maneuver file functions like a storage group, i.e. the file receives a consecutive number within the acquisition.

The names of the single acquisition files are generated according to **MVBBCCCC.ASC** structure:

| | |
|---|---|
| MV**BB**CCCC.ASC | Name of respective storage group |

**Example (Acquisition no. 699 > MVBBCCCC.ASC)**

| | | |
|---|---|---|
| **Storage group 1** | DO010699.DAT | Header |
| | DO010699.I16 | 16 Bit (Integer signed) |
| **Storage group 2** | DO020699.DAT | Header |
| | DO020699.R32 | 32 Bit (Real) |
| **Maneuver file** | MV030699.ASC | |