TEMES 1.0

USER MANUAL

Installation

1 Installation

1.1 Requirements

In order to install TEMES 1.0 or run vTEMES 1.0 correctly, following items are required:

- Operating System: Windows 10/8.1/8/7/Vista/XP (x86 or x64 Edition)
- Free hard disk space: at least 1300 MiB (about 400 MiB are allocated by TEMES after installation)
- An installed printer driver
- Optional: Adobe Acrobat Reader
- Optional: Microsoft .NET Framework 4 (for Windows 7/Vista/XP)
- Optional: Microsoft .NET Framework 4 Full Language Pack (German) (for Windows 7/Vista/XP)
- Recommended: at least 8 GiB RAM (despite only up to 2-4 GiB are used by TEMES 1.0 since TEMES 1.0 is a 32-bit application)
- Recommended: monitor resolution at least 800 x 600
- Linux (experimental): Wine 4.0 (A serial communication does not work for SICOLOG/USBDL1/SICO3 at the moment (timeout issue). Also, *TEMES View.exe* must be marked as a Windows 7 application, and the SICOLOG USB stick initialization does not work due to an empty drive letter box.)

The parallel adapter PAD is only supported on Windows x86 Editions. (The USB adapter USBAD can be used instead on Windows x64 Editions.)

The setup application for TEMES (or vTEMES) can be downloaded from http://www.tellert.de/?product=temes

1.2 TEMES

TEMES is installed under the [x86] program folder in the path "*Tellert\TEMES 1.0*".

TEMES also requires a rewritable data folder which is by default the folder "*TEMES*" under the *documents* folder. It contains two sub-folders namely the "*CAN*" folder (where the CAN descriptions are stored), and the "*CALC*" folder (where user-definable calculation resources are stored).

1.3 vTEMES

The virtualized TEMES version does not require an installation. It is started by executing *vTEMES.EXE*. Note that vTEMES is a restricted TEMES version, and that not all TEMES features are available.

1.4 Version History

V1.0.94

- **New:** Integration of *TIL_TMS.DLL* into TEMES (Tools > Convert Files > TEMES to ... format).
- **New:** Support for a slower analog signal averaging to allow very slow sample rates (BaseFrequencyDivisor 152).
- **New:** Support for analog signals without averaging (RawMask 153).
- **New:** Support for an automatic export 42 of measurements.

V1.0.93

- **New:** Support for a checksum 152 of SICOLOG configurations.
- **New:** Repair of a corrupt TEMES measurement 31.
- **Modified:** The checkboxes *Eject mass storage...* are now always checked by default.

V1.0.92

- **New:** Optional additional extended logging 151 of start and stop events.

V1.0.91

- **Modified:** Support for additional stream bytes in the measurement data which were introduced in SICOLOG V1.0.44. These additional stream bytes are recognized as unexpected stream error bytes in previous TEMES versions. Hence, if the new extended logging features of SICOLOG V1.0.44 or better are used, it is strongly recommended to also install TEMES V1.0.91 or better.

V1.0.90

- **New:** Maintenance of an odometer 159.
- **New:** Storing of the device number 159 in a measurement variable for SICOLOG/USBDL1 V1.0.44 or newer.
- **Bugfix:** The corresponding GPS AutoCalc signals are no longer evaluated if their names are empty.
- **Bugfix:** The storage from mass media is more robust (and can now handle all unexpected bytes).

V1.0.89a

- **Bugfix:** *firlp.dll* produced a zero line for the remaining measurement of a long measurement. This issue is now resolved.

V1.0.89

- **New:** Experimental and rudimentary support for XCP 63 V1.1 and CCP V2.1 on CAN (Internal tests have been passed but external tests with physical XCP/CCP devices are still pending).
- **New:** Display TEMES directories 30.
- **New:** TEMES creates now the *CAN* and *Calc* subdirectories if they do not exist.
- **New:** vTEMES uses now the TEMES directory if it is found in the same path as the executable vTEMES file.
- **Modified:** The GPS AutoCalc function is also executed for *GpsSpeed* and *GpsSpeedInt* (in place of *POS_Speed*).
- **Modified:** The Analog Input Info 47 tab and Clutch Events 75 tab have been removed (due to resource limitations).
- **Bugfix:** The CAN-Editor import of DBC files works now correctly for multiplexed signals.
- **Bugfix:** The script functions *strtrim* and *strrtrim* work now correctly even for one letter strings.

V1.0.88

- **New:** Additional GPS parameter *GpsSpdMsgCompareTicks* 146.
- **New:** Additional parameter to mirror the User LED 151 onto a digital channel.
- **Modified:** The GPS signal *POS_Time* for SICOLOG/SICO3/SICO3M/USBDL1 was changed from scaling factor 1800 h/bit to 1/1800 h/bit.

V1.0.87

- **New:** Support for additional option for frequency input signals. See also Frequency Divider 149.
- **New:** Initialization of an USB stick (SICOLOG/USBDL1) via menu item *Tools → Additional Tools → Hardware → Initialize USB stick (SICOLOG/USBDL1).*

V1.0.86

- **New:** Support for SICO3M.

V1.0.85

- **New:** Optional real time clock synchronization with GPS. See also RTC Synchronization 149.

V1.0.84

- **Modified:** Update of the preview version of *TEMES View* 187 (TEMES: *Tools → TEMES View*).

V1.0.83

- **New:** Integration of a preview version of *TEMES View* 187 (TEMES: *Tools → TEMES View*).

V1.0.82

- **New:** Optional automatic file name generation for SICOLOG/USBDL1 measurements. See also File Name Generation 158.

V1.0.81

- **New:** Synchronization of a recording for multiple SICOLOGs/USBDL1s. See also Record Synchronization of Multiple Devices 156.

V1.0.80

- **Bugfix:** Backward compatibility with SICOLOG/SICO3 V1.0.26 or older restored for TEMES V1.0.71 or newer (The new CAN FD functions had also been used for the CAN node instead of the CAN FD node only for CAN signals which require bit masking. This was no problem for SICOLOG/SICO3 V1.0.27 or newer, or for aligned 8-bit/16-bit/32-bit CAN signals).

V1.0.79

- **New:** Changed digital output default values for SICOLOG/SICO3/USBDL1. See also Parameter Tree Node: Digital Output 72.
- **New:** Additional CAN signal data type (float) for a scanned CAN only. See also Parameter tree node: CAN/CAN FD/LIN signal 91.

V1.0.78

- **New:** Support for CAN scan mode for SICOLOG/USBDL1. See also CAN Scan Mode (Overview) 153.

V1.0.77

- **New:** Additional signed GPS signals *GpsLeanAngle/GpsLeanAngleInt* for SICOLOG/SICO3/USBDL1.
- **New:** New menu item *Tools* → *Reset Diagram* to start the diagram view again from the beginning.
- **New:** Automatic default signal delay for the GPS signals in the diagram view (when starting the diagram view from the beginning).
- **Bugfix:** The timestamps are now painted correctly at the right place when the first signal after the time signal was delayed. Also the timestamp at the current cursor position is now correctly calculated for delayed signals.
- **Bugfix:** A problem with Non-ASCII characters in signal/message names for the SICOLOG/SICO3/USBDL1 has been fixed. This problem affected mostly users with signal/message names in the device comment (e.g. for the external display ED12).

V1.0.76

- **Modified:** The *CAN sample point/listen only mode* for SICOLOG/SICO3/USBDL1 can now also be entered again via the device comment.

- **Modified:** Set stored directories to the TEMES data directory when they no longer exist.
- **Bugfix:** The first argument of external application calls is now placed in double quotes.

V1.0.75

- **New:** Additional GPS signal *GpsLeanAngleInt* for SICOLOG/SICO3/USBDL1.
- **New:** The menu item *SICOLOG/SICO3 configurations* has now an option to ignore the NEXT/INS (= B4) button.
- **Modified:** The menu item *Diagram without Programming* does not temporarily change the *use unused inputs* setting.
- **Modified:** Better version of the SICOLOG/SICO3/USBDL1 display 29.

V1.0.74

- **New:** Additional GPS signals are calculated (such as acceleration, distance and slope) for SICOLOG/SICO3/USBDL1.
- **New:** Timeout filter, and timeout for OBD signals.
- **Modified:** The OBD auto detection has been changed from combined OBD-II/WWH-OBD to separate OBD-II/WWH-OBD.
- **Modified:** New display item type (source value as bar display) for SICOLOG/SICO3/USBDL1.
- **Modified:** Better version of the SICOLOG/SICO3/USBDL1 display 29.
- **Bugfix:** The filters did not work as expected when also OBD signals were defined.

V1.0.73

- **Modified:** The SICOLOG/SICO3/USBDL1 display items now support *sample rate* and *memory usage*.
- **Modified:** Better version of SICOLOG/SICO3/USBDL1 display 29.

V1.0.72

- **New:** Support for USBDL1 / USBDL1 FD / USBDL1 A16.
- **New:** New LED output 74 for the SICOLOG/SICO3 L4 LED / USBDL1 USER LED.
- **New:** SICOLOG/SICO3/USBDL1 display 29.
- **Bugfix:** The CAN export of TEMES files was wrong in V1.0.71 and is now working correctly.
- **Bugfix:** The calibrate signals form 35 works now with activated unused input signals and with *DT_...* signals for SICOLOG/SICO3/USBDL1.

V1.0.71

- **New:** Support for CAN FD of a SICOLOG/SICO3 (with FD option).

V1.0.70

- **Modified:** The GPS items of a SICOLOG/SICO3 do now support the NMEA ZDA sentence (UTC date and time).

V1.0.69

- **New:** Support for detection of encrypted SICOLOG USB sticks. (The standard version of TEMES, and the standard SICOLOG do not support the SICOLOG encryption.)

V1.0.68

- **Bugfix:** SICOLOG measurements with multiple configurations are now recognized correctly.

V1.0.67

- **New:** Table editor 29.

V1.0.66

- **Modified:** OBD 56 editing and filter 65 editing for SICOLOG/SICO3. Additional *ExParameter* settings for SICOLOG/SICO3: *OBDRateRatio* 145 which describes the *n*-th OBD active cycle of the calculations, *OBDRepetitions* 146 which describes the number of OBD protocol tests before the next protocol is tested, and *FloatLin* 146 which (when enabled) switches to floating point operations for the Op04 (linear transform) 67.
- **Modified:** The DL16K clutch events 75 are now no longer supported within the German GUI. Please use the English TEMES version or an older TEMES version instead.

V1.0.65

- **Modified:** A void set of parameters is now transferred to a SICOLOG/SICO3 device prior a parameter set (for firmware ≥V1.0.24).

V1.0.64

- **Modified:** The time adjustment 30 of a SICOLOG/SICO3 device has been replaced by an application which supports both COM ports and USB sticks.
- **Modified:** The time zone is optionally refreshed in the USB stick dialog boxes (or more precisely in the mass storage dialog boxes).

V1.0.63

- **Modified:** On x64 systems, the sample cache is as large as the file size if possible.
- **Bugfix:** A rarely occurring read exception during displaying a chart curve has been fixed.

V1.0.62

- **Modified:** The SICOLOG/SICO3 floating point display is more flexible concerning the output text 77.
- **Bugfix:** The size of the online recording was wrongly limited too much on x64 systems. The check of available physical memory is now turned off on x64 systems for limiting the online recording size.
- **Bugfix:** Display items with a signed CAN/LIN signal source had a wrong offset for SICOLOG/SICO3.

V1.0.61

- **Bugfix:** The display of the recordings and the timestamps were eventually wrong when the time unit was different from seconds.

V1.0.60

- **New:** SICOLOG/SICO3 configurations editor.
- **New:** Keyboard shortcuts for fast switching of the time unit (seconds, minutes, hours, days).

V1.0.59

- **New:** Support for SICO3 V1.0.

V1.0.58

- **New:** Pulse width measurement for the first three digital channels of the SICOLOG.
- **Modified:** The TEMES online help has been rewritten.

V1.0.57

- **Bugfix:** The online chart without programming also works for non-SICOLOG devices.
- **Bugfix:** The name for further AutoCalc measurement blocks is now "AutoCalc".

V1.0.56

- **New:** SICOLOG has a new parameter tree node "LIN".

V1.0.55

- **New:** The automatic calculation of certain signals after reading out a measurement can be disabled via menu item "Tools" > "Disable AutoCalc".
- **Bugfix:** The automatic calculation after reading out a measurement was triggered under certain circumstances despite there was nothing to calculate.

V1.0.54

- **Modified:** During programming the corresponding ports were occasionally not freed. This bug is fixed.

V1.0.53

- **Modified:** Integration of ED12 V1.1.0 instead of ED12 V1.0.0. ED12 V1.1.0 supports signed SICOLOG signals, optionally no averaging, and automatic number of decimals.

V1.0.52

- **New:** Integration of the software for the external display ED12.
- **New:** Support for the listen-only-mode of the SICOLOG's CAN modules. This mode is activated with the following entry in the device comment and presumes a SICOLOG firmware of at least V1.0.7:

```
<DPF>
[ExParameters]
Can0ListenOnlyMode=1
Can1ListenOnlyMode=1
</DPF>
```

V1.0.51

- **New:** Overly large SICOLOG parameter sets can now also be transmitted serially. This feature works only for SICOLOG V1.0.6 or newer.
- **Modified:** Treatment of SICOLOG measurement files during a transfer of a parameter set onto a USB stick.
- **Bugfix:** Now, the block size of SICOLOG files during the transfer via the serial interface has been again reduced (for better handling of transmission errors).
- **New:** The beginning, end and cursor position(s) are now marked at the exported path during export to Google Earth.
- **New:** Display of the real time of the view's beginning and the view's end in the status bar.
- **New:** New keyboard assignment in diagram view if no cursor is visible, and if the time is selected as x axis (Then, a new recording always starts with a new timestamp):

  Page up: Display of the previous recording.
  Page Down: Display of the next recording.
  Ctrl+Home: Display of the first complete recording.
  Ctrl+End: Display of the last complete recording.
  Home: Movement to the beginning of the current recording.
  End: Movement to the end of the current recording.
  Shift+Home: Movement to the beginning of the first recording.
  Shift+End: Movement to the end of the last recording.

V1.0.50

- **Bugfix:** The module TE_TOOL1.DLL was differently included in TEMES.

V1.0.49

- **Bugfix:** Now, the block size of SICOLOG files during the transfer via the serial interface has been reduced (for better handling of transmission errors).

V1.0.48

- **New:** For SICOLOG parameter sets, during a transfer, if the generated CONFIG.TSF is greater than 64 KB, the corresponding TEMES parameter set is no longer embedded but copied separately as CONFIG.TMS. Unfortunately, those parameter sets cannot be transferred via a serial connection. If necessary, F9 has to be used for online parameter view.
- **New:** The device comment of parameter sets for SICOLOG V1.0.5 or newer can be used to activate the additional averaging of analog inputs:

```
<DPF>
[ExParameters]
AIAvg=1
</DPF>
```

V1.0.47

- **Bugfix:** Now, the mapping to request bytes of the f6 protocol works correctly for complex parameter sets.

V1.0.46

- **New:** Mapping of the signal addresses to request bytes of the buffered serial f6 protocol for SICOLOG V1.0.4 or newer. New TEMES device comment variable *F6Map*:

```
<DPF>
[ExParameters]
F6Map=SignalName#1,SignalName#2,SignalName#3
</DPF>
```

  Mapping of the signal names to request bytes: SignalName#1 → 0x80, SignalName#2 → 0x81, Signalname#3 → 0x82.

- **New:** Up to 16 serial input signals for SICOLOG V1.0.4 or newer. For this purpose the GPS signals must be turned off, the slider on the SICOLOG's bottom must be switched (or the jumper in the inside of the SICOLOG must be switched), and the following TEMES device comment variables must be defined:

```
<DPF>
[ExParameters]
SIBaudrate=57600
SITimeout=500
SIMode=4
SICount=3
SI0=0x80,SignalName#1,2
SI1=0x81,SignalName#2
```

```
SI2=0x82,SignalName#3
</DPF>
```

SIBaudrate: baud rate in bps; *SITimeout*: timeout in msec; *SIMode*: 1 - f6 protocol, 2 - f6 protocol with zero signal on timeout, 3 - f6 protocol with checksum, 4 - f6 protocol with checksum and zero signal on timeout or checksum error; *SICount*: Number of the following signal definitions; *SI<Number>*: request byte, signal name of an already defined TEMES calculation constant, optionally signal size in bytes.

V1.0.45

- **Bugfix:** Calculated signals, operation #19 (delay) did not work as expected for CFDL1/SICOLOG.

V1.0.44

- **New:** Optional asynchronous sample rate for the recording of online signals for SICOLOG V1.0.3 or newer. The asynchronous sample rate is taken as the next integer multiple of the fast sample rate.

V1.0.43

- **Bugfix:** A signal which was recorded during both the fast cycle and the slow cycle, was for the SICOLOG always refreshed with the slow sample rate instead of the fast sample rate. This error is fixed.

V1.0.42

- **New:** Menu item for recovery of marked as deleted SICOLOG measurements on the USB stick.
- **Bugfix:** After storing a SICOLOG measurement from the USB stick into a TEMES file, it was up to now impossible to "eject" the USB stick.
- **Bugfix:** The TEMES device manager removes not connected ports, and displays these ports after a program reset as not connected.

V1.0.41

- **Bugfix:** Sometimes the timestamp of the first sample was swallowed for CFDL1 measurements at read out time. This error is fixed.

V1.0.40

- **New:** Visibility of the write speed for SICOLOG parameter sets in the cycle definition view.

V1.0.39

- **Bugfix:** During calibration of the SICOLOG, all signals to be recorded were reset with the effect that an autarkic recording was not possible. (Remedy: a repeat programming of the parameter set). This bug is fixed.

V1.0.38

- **New:** Automatic switch of the serial port up to 230400 baud for SICOLOG devices.
- **New:** Online chart without programming (which allows a chart view of live signals during an autarkic measurement).
- **New:** Automatic cutting of SICOLOG measurements during reading out from a USB stick. The files are cut at a timestamp from 400 MB onwards. When no timestamp is found, the file is cut at 500 MB at the latest. Then, a new additional timestamp for the first sample is mentioned in the variables.

V1.0.37

- **New:** Complete support for the signal converter and data logger SICOLOG. Only the first 4,294,967,296 samples are transfered during reading out. The reading out of the SICOLOG only works via mass media (= USB stick).
- **Bugfix:** There was a memory size error at reading out CFDL1 measurements.

V1.0.36

- **New:** Support for the signal converter part of the SICOLOG.

V1.0.35

- **Changed:** Minimum required OS is Windows XP.
- **Changed:** List & Label 19 instead of List & Label 14.
- **New:** The Tellert Device Manager now detects COM1 to COM256 automatically.

TEMES

2 TEMES

2.1 General Concept

TEMES combines those tools which are required to setup devices (like *USBDL1*, *SICOLOG*, *SICO3*, *SICO2B*, *CFDL1*, *DL16CAN*) and to get signal data – either online or offline. A TEMES data file contains all parameters which are required to set up a device completely. The parameters are organized as a tree structure – showing only those tree nodes, or parameter paths, which are available for the corresponding device. The properties of a device parameter set are distinguished into four groups:

General Properties: General properties like device's sample rate.

Input Signals: Input signals represent the input data which can be used for further processing. To define an input signal, the corresponding input source (e. g. a voltage channel) must be given a unique name. Within the entire parameter definition this name is used to refer to the corresponding input source. By deleting the name of an input signal, the corresponding input source is deleted/deactivated as well.

Calculations: Sometimes different input signals needs to be combined in order to get the desired result. This can be realized with calculated signals. Basically, a calculated signal is derived from the class of input signals. This allows to refer to the result via the corresponding input source name. Further properties of a calculated signal are the type of operation, up to three names of input sources and up to three constants.

Output: Outputs are e. g. display items, voltage outputs or digital outputs. To enable an output, the input signal needs to be selected as output source. Further parameters may be used to scale the output signal correspondingly.

Note to read the section USBSER and TEMES 1.0 162 if the device is connected to the PC via a serial/RS232 cable.

2.2 Navigating Through the Parameter Tree

A TEMES document is divided into three areas: the parameter tree, the node/path name and an embedded form. By selecting a tree node, the corresponding tree path is displayed above the embedded form, and the corresponding embedded form is shown. If a tree node is represented by a plus symbol, the node can be expanded by clicking the plus symbol or by using the *right arrow* key. Correspondingly, a tree node can be collapsed if it is represented by a minus symbol instead. Some tree leaves can be moved within the same level. This is done with the *up* and *down arrow* keys pressed together with the *Alt* key (Helpful, e. g. to swap voltage channels, or if the processing order of calculated signals should be changed). Some tree leaves can be removed by pressing the *Del* key (e. g. filters, CAN signals or calculated signals), and some tree leaves can be inserted as a child leave by pressing the *Ins* key (e. g. filters, calculated signals or signal groups).

Though the embedded forms do not display the default buttons *OK* and *Cancel*, the functionality remains available via the keyboard: The *Enter* key represents the *OK* button, whereas the *Escape* key represents the *Cancel* button. By selecting a new tree node, the made changes are accepted before displaying the new embedded form. If changes are invalid, a new tree node cannot be selected. If a field's background is painted with red color it indicates that the made changes are invalid for this field. Then, the input needs to be corrected or to be discarded with the *Escape* key.

2.3 Main Menu

2.3.1 Main Menu: File

2.3.1.1 New

Creates a new document. A void and unnamed set of parameters will be created. The *Add Device* 34 dialog box will be displayed to ask the user which kind of hardware to add to the void set of parameters.

2.3.1.2 Open

Opens or imports an existing TEMES document. If a file with an extension other than *.tms* is chosen, the corresponding file will be converted into a temporary TEMES document. Such temporary documents are treated as unnamed TEMES documents.

The CFDL1 file "*<DriveLetter>\Tellert\CFDL\00000001.tmf*" can be directly opened with TEMES.

The SICOLOG/USBDL1 file "*<DriveLetter>\Tellert\SICOLOG\config.tsf*" can be directly opened with TEMES.

If a compressed TEMES file (= *.tma* file) does not open in TEMES, the file archiver (TEMES main menu: *Tools → File Archiver* 29) can be used to generate an uncompressed TEMES file. If an x-y recorder file (= *.xy* file) does not open in TEMES, a *.tms* file can directly be generated with "*<ProgramFiles>\Tellert\TEMES 1.0\xy2tms.exe*".

2.3.1.3 Open Measurement

Opens an existing TEMES document by using the current measurement folder as default folder. If the document contains measurement data, the view changes to the chart view.

2.3.1.4 Save

Saves active TEMES document.

2.3.1.5 Save as

Saves active TEMES document as a new file.

2.3.1.6 Close

Closes active TEMES document.

2.3.1.7 Import

The import menu item is depending on the selected node of the parameter set. If the selected node is part of a device branch, the import is done for a device. If the selected node is part of a measurement (block), the import is done for a measurement (block).

2.3.1.7.1 Import (Device)

Imports one set of parameters from another TEMES document. This menu item is helpful to use a TEMES set of parameters for another hardware type, e. g. if a TEMES file was made for the signal converter SICO2, and the parameter settings should be used for the data logger DL16CAN, the already defined parameter set for the SICO2 can be selected, and the DL16CAN can be chosen as target hardware type 34.

2.3.1.7.2 Import (Measurement)

Adds the measurements blocks of another TEMES file to the list of measurement blocks. Or, if a TEMES license for script controlled online recording is available, imports a text file (*ASC*, *TXT*, *CSV*) as an additional measurement block.

2.3.1.8 Export

The export menu item is depending on the selected node of the parameter set. If the selected node is part of a device branch, the export is done for a device. If the selected node is part of a measurement (block), the export is done for a measurement (block).

2.3.1.8.1 Export (Device)

.TXT: Exports as a text report. This is useful if detailed signal information is required (especially about the bit assignments or signal addresses).

.CAN: Exports as a CAN description (the best destination location is the *CAN* sub-folder of the TEMES data folder (which is usually "*documents\TEMES\CAN*"). Each input signal is mapped to a 16-bit CAN signal. The generated CAN description can then either be used directly by the CAN node 58 of the parameter tree, or it can be opened in the CAN editor for changing or to export it as a DBC file.

.XY: Exports as an x-y recorder parameter set.

2.3.1.8.2 Export (Measurement)

This menu item is less important since there are better methods to convert a TEMES measurement. Namely, the Tellert import library TIL TMS.DLL (e. g. Drag-and-drop the *.TMS* file in the Windows explorer onto the file "*TMS2TXT.EXE*" to obtain an Excel importable *.TXT* file in the same directory as the *.TMS* file). Or the TEMES menu item *Tools → Convert Files → TEMES to DIAdem...* 32

.TXT: Converts into a text file without timestamps.

.XY: Converts into x-y recorder format.

.BIN: Saves the raw measurement block.

2.3.1.9 Exit

Quits TEMES.

2.3.2 Main Menu: Hardware

2.3.2.1 Program Device

Sends selected parameter set to the corresponding device.

2.3.2.2 Calibrate Device

Calibrates connected voltage signals inside the *Calibrate Signals* 35 form.

2.3.2.3 Write to Mass Storage

Writes the parameter set onto a mass storage (USB stick or CF card) with the Select Mass Storage 36 form.

2.3.2.4 Save Measurement

Copies measurement data from the data logger DL16CAN into a file with the Save Measurement 36 form.

2.3.2.5 Save Measurement from Mass Storage

Saves measurement from a mass storage (USB stick or CF card) into a file with the Save Measurement from Mass Media 37 form.

2.3.2.6 Restore Measurement on Mass Storage

Restores a previously marked-as-deleted SICOLOG/USBDL1 measurement onto the USB stick with the Recovery 38 form.

A more powerful tool to unerase/repair a SICOLOG/USBDL1 measurement is TSD.EXE.

2.3.2.7 Device Manager

Shows Tellert Device Manager 82 window.

2.3.3 Main Menu: View

2.3.3.1 Diagram

Opens the chart view of the active document. If the active document contains no measurement data, the chart form is opened in either offline or online mode from which the user can display or record the signals which are currently available from the corresponding device. If the active document contains at least one measurement block, the measured data will be displayed instead. If a measurement block node, or one of its branches is selected, only the corresponding measurement block is displayed, otherwise the signals of all measurement blocks are shown – assuming that the first signal of a measurement block represents the time signal which is used to synchronize the signals of different measurement blocks.

2.3.3.2 Diagram without Programming

Opens the chart view of the active document. This menu item does not reprogram the device, instead the device is assumed to be programmed with the active document. This menu item is used to snoop into an already running autark measurement.

2.3.3.3 Variables

Opens the variable editor. The variable editor is a mean to edit user defined variables which belong to the active document (e. g. *driver name* or *weather conditions*). Those variables can be used as placeholders in the layout module.

2.3.4 Main Menu: Tools

2.3.4.1 CAN Editor

Starts a new instance of the CAN editor. An instance of a CAN editor can be used to create, to show, to export or to modify a CAN description file.

2.3.4.2 XCP/CCP Editor

Starts a new instance of the XCP/CCP editor. An instance of a XCP/CCP editor can be used to create, to show or to modify a XCP/CCP description file.

2.3.4.3 Table Editor

Starts a new instance of the table editor. An instance of a table editor can be used to create, to show or to modify a table file (or characteristic curve).

In order to change a characteristic curve within a TEMES file completely, following steps are necessary:

1. Select a different characteristic curve and display another node.
2. Go back to the node with the characteristic curve and change the characteristic curve back to the original name.

2.3.4.4 SICOLOG/SICO3/USBDL1 Display

Starts a new instance of the SICOLOG/SICO3/USBDL1 display application.

2.3.4.5 SICOLOG/SICO3 Configurations

Starts a new instance of the SICOLOG/SICO3 configurations editor.

2.3.4.6 TEMES View

Starts a new instance of TEMES View 187.

2.3.4.7 External Display ED12

Starts a new instance of the External Display ED12 editor.

2.3.4.8 File Archiver

Starts the file archiver to compress/expand TEMES documents (Compressed files have the extension *.tma*). The used lossless compression algorithm has two passes. The first one re-orders the measurement data to sort the data by signal and then tries to reduce the signal entropy by simple signal transformations. The second pass is ZIP compatible. Note that compressed files can be directly opened with *File → Open* but do not save into an archive format. Thus, compressed files are somewhat write protected.

2.3.4.9 Additional Tools

2.3.4.9.1 Hardware

2.3.4.9.1.1 Set SICOLOG/SICO3/USBDL1 real time clock

Sets SICOLOG's, SICO3's or USBDL1's real time clock.

2.3.4.9.1.2 Initialize USB stick (SICOLOG/USBDL1)

Initializes a USB stick for SICOLOG/USBDL1.

2.3.4.9.1.3 Set CFDL1 real time clock

Sets CFDL1's real time clock. The CFDL1 must be connected via a USB cable with a type B plug (and **not** with the virtual COM port USBSER).

2.3.4.9.1.4 Initialize CF Card (CFDL1)

Initializes CF card for CFDL1 or unerases a marked-as-deleted CFDL1 measurement.

2.3.4.9.1.5 Set DL16 real time clock

Sets DL16CAN's real time clock.

2.3.4.9.1.6 Change SICO2/DL16 Settings

Changes the SICO2/DL16CAN settings like baud rate, device number or display type. Please connect the device, turn all other connected devices off, and wait 5 seconds for the window.

2.3.4.9.1.7 Monitor

Opens a monitor window for SICO2/DL16CAN (for debugging purposes).

2.3.4.9.2 Directories

Display all directories which are used by TEMES. The entries can only be altered when TEMES is not running and the program *TemesDirectories\TemesDirectories.exe* is called directly from e. g. the Windows Explorer.

2.3.4.9.3 Select Corrupt TEMES File

Selection of the corrupt TEMES file whose measurement data will be used for each opened TEMES document. Please mind to delete this entry after a recovery attempt. A corrupt TEMES file can be obtained, if the USB stick is unplugged too fast from the PC and if it is not ejected beforehand.

A recovery attempt of a corrupt TEMES measurement can be done as follows:
1. Call the TEMES menu item *Select corrupt TEMES file* and select the corrupt TEMES file.
2. Open a similar intact TEMES measurement which has the same number of measurement blocks and the same configuration as the corrupt TEMES file.
3. Save the (merged) TEMES measurement under a new name.
4. Rename/Delete the corrupt TEMES file and remove its name from being selected (*Select corrupt TEMES file*).

2.3.4.9.4 Edit menu item "Additional Tools"

This menu item allows to edit the *Additional Tools* menu item. Note that write access is needed for the files "*C:\Program Files (x64)\Tellert\TEMES 1.0\ToolsMenu_eng.ini*" and "*C:\Program Files (x64)\Tellert\TEMES 1.0\ToolsMenu_de.ini*".

2.3.4.10 Convert Files

2.3.4.10.1 TEMES to Textfile format

Converts the signals of the selected TEMES files into corresponding text files.

2.3.4.10.2 TEMES to MDF3 format

Converts signals of selected TEMES files into corresponding MDF3 files.

2.3.4.10.3 TEMES to M format

Converts signals of selected TEMES files into corresponding M files (Matlab/Octave).

2.3.4.10.4 TEMES to R format

Converts signals of selected TEMES files into corresponding R files.

2.3.4.10.5 TEMES to JL format

Converts signals of selected TEMES files into corresponding JL files (Julia).

2.3.4.10.6 TEMES to SCE format

Converts signals of selected TEMES files into corresponding SCE files (Scilab).

2.3.4.10.7 TEMES to DIAdem

Converts signals of selected TEMES files into one single DIAdem import document. Those converted files can be opened within DIAdem with *Import via Header File*.

2.3.4.10.8 Make report (Acceleration/Top Speed)

Makes a text file with a report of acceleration and top speed of the selected TEMES file. The settings can be altered in the text file "*C:\Program Files (x64)\Tellert\TEMES 1.0\SpeedReport.cvs*", if you have the necessary write access rights.

2.3.4.10.9 Split TEMES files

Splits selected TEMES file. The file is split at the corresponding timestamps or according to the settings in the text file "*C:\Program Files (x64)\Tellert\TEMES 1.0\SplitTms.cvs*". Note that the necessary write access rights might be needed to alter this file.

When a file has timestamps the file is split at the corresponding timestamps and the resulting TEMES documents have no timestamps. When a CFDL1 measurement has timestamp exceptions these exceptions are ignored.

When a file has no timestamps, the file is split according to a trigger signal (which is specified in the *.CVS* file).

2.3.4.11 Configure TEMES

2.3.4.11.1 Language

Changes the TEMES language to German/English.

2.3.4.11.2 License

Displays or registers a license for script controlled online recordings. This license also includes the import of ASCII measurement files.

2.3.4.11.3 Discard answers

Discards all remembered TEMES answers. A message box will then be shown again to query the corresponding answer.

2.3.4.12 Execute Script

Executes a script file. A script file is provided by *Rudy Tellert Elektronik* on request.

2.3.4.13 Disable AutoCalc

Disables the automatically execution of the *AutoCalc* function after a measurement is read out, if this menu item is checked.

2.3.4.14 Update AutoCalc Signals

Deletes the *AutoCalc* measurement blocks and execute the calculation of the *AutoCalc* signals (which are multiplexed temperatures and additional GPS signals).

2.3.4.15 Reset Diagram

Reset the diagram to its default view. This menu item also sets the signal delay of GPS signals to their default value.

2.3.5 Main Menu: Window

2.3.5.1 Cascade

Organizes the child windows in a cascaded way.

2.3.5.2 Tile

Organizes the child windows side by side.

2.3.5.3 Arrange Icons

This menu item has no function at the moment.

2.3.5.4 Minimize all

Minimizes all child windows.

2.3.6 Main Menu: Help

2.3.6.1 TEMES help

Opens the TEMES help.

2.3.6.2 Data Sheets

Shows the data sheets (which are PDF files), if the Adobe Acrobat Reader is installed.

2.3.6.3 Info

Shows information about TEMES.

2.4 Dialog Boxes

2.4.1 Add Device Form

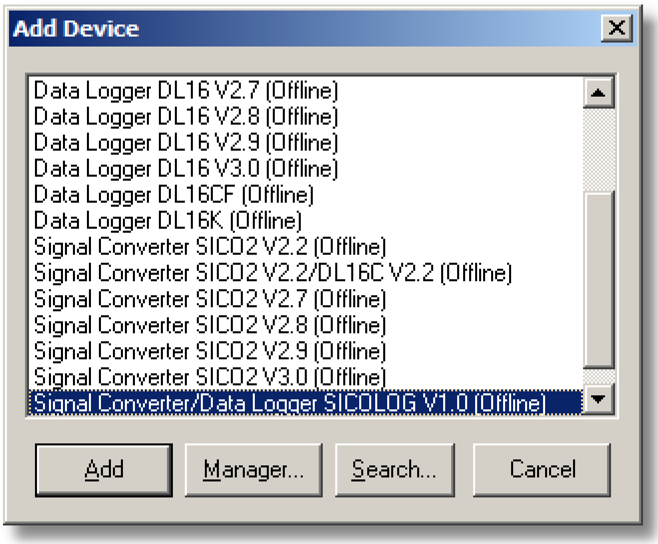


Figure 2-1: Add Device Form.

The *Add Device* form displays all device types which are supported by TEMES. The first lines show all devices which are currently registered with the *Tellert Device Manager*. The following lines show all devices which are supported by TEMES. Depending on whether the device is already registered with the *Tellert Device Manager*, the corresponding port name is displayed just after the type name in parenthesis. In case of a not registered but supported device, the text *Offline* is displayed instead.

Add: Add parameter set for the selected device.

Manager: Show the *Tellert Device Manager* 82 window.

Search: Ask the *Tellert Device Manager* to update all of its ports.

Cancel: Close the *Add Device* form.

General Device Type (newest generation)	Add Device List Box Entry	Device tree node
Data Logger *CFDL1*	Data Logger CFDL1	CFDL1
Data Logger *DL16* or *DL16CAN*	Data Logger DL16 V3.0	DL16C V3.0
Data Logger *USBDL1*	Data Logger USBDL1 V1.0	USBDL1 V1.0
Signal Converter *SICO2*, *SICO2B*, or *SICO2I*	Signal Converter SICO2 V3.0	SICO2 V3.0

General Device Type (newest generation)	Add Device List Box Entry	Device tree node
Signal Converter *SICO3, SICO3I or SICO319I*	Signal Converter SICO3 V1.0	SICO3 V1.0
Signal Converter/Data Logger SICOLOG	Data Logger SICOLOG V1.0	SICOLOG V1.0
Thermoelement multiplexer *TH16MI/CAN*	Signal Converter SICO2 V3.0	SICO2 V3.0

2.4.2 Calibrate Signals Form

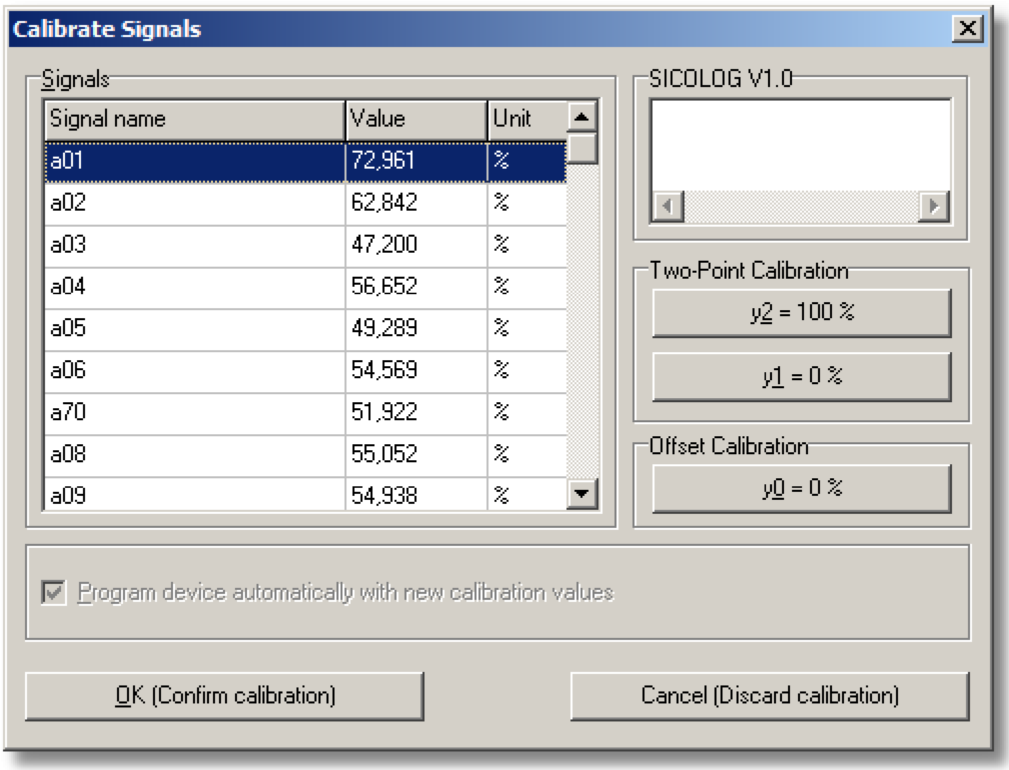


Figure 2-2: Calibrate Signals Form.

There are two methods to calibrate a voltage signal. One method is the two-points calibration which measures the signal factor and the signal offset. The latter method is a movement of the zero point which is used to measure the signal offset only. The procedure of applying the methods is the same in both cases. First you make sure that the signal to be measured equals the reading of a calibration button. Then, by pressing this calibration button, the currently measured signal value is defined as to be equal with the reading of this button. In case of a two-points calibration, a pressing of both buttons is required in order to calibrate a signal completely.

The readings of the calibration buttons are taken from the assignment of the corresponding voltage input 43.

The top-right group box (with the device name) is reserved for the SICO2/DL16CAN start-up text 76, otherwise this box is empty.

2.4.3 Select Mass Storage Form

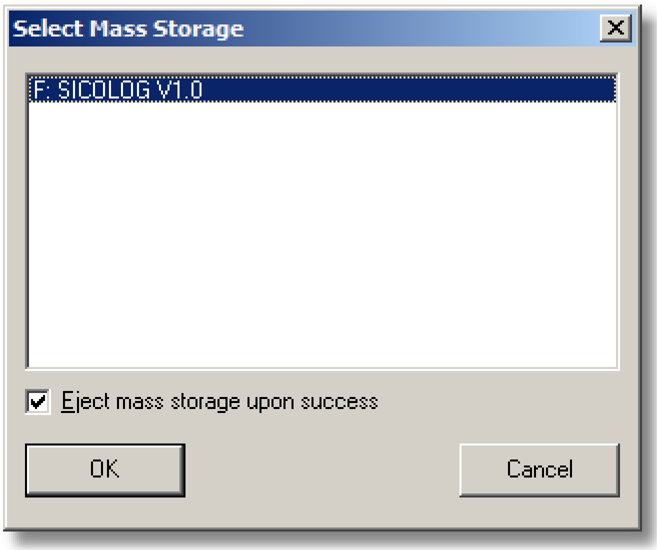


Figure 2-3: Select Mass Storage Form.

Note to check the checkbox *Eject mass storage upon success*, otherwise the media must be manually ejected in the Windows Explorer before it is physically ejected.

2.4.4 Save Measurement Form

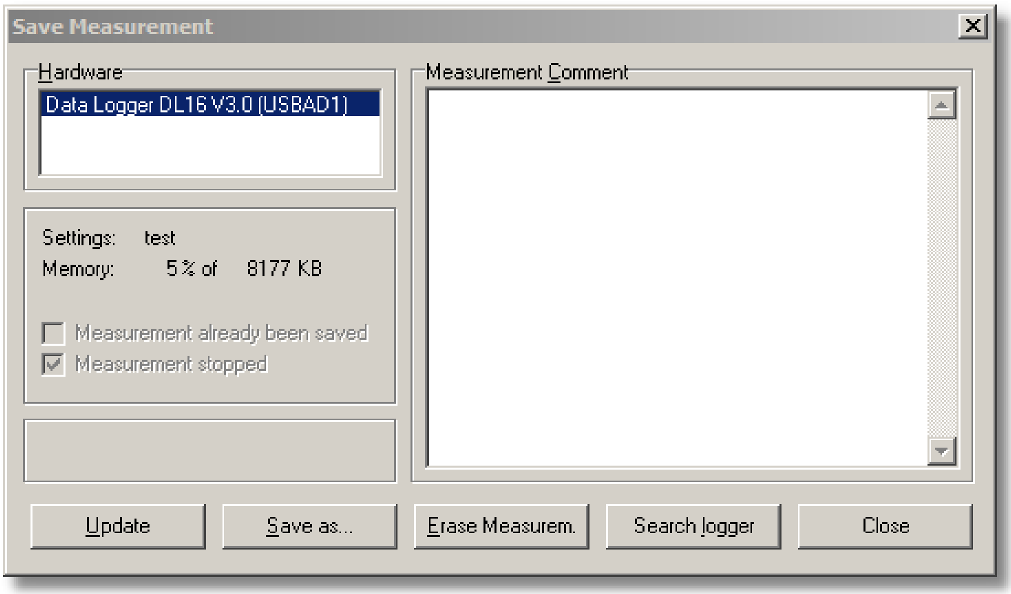


Figure 2-4: Save Measurement Form.

The *Hardware* list box displays all connected devices which are registered with the *Tellert Device Manager*. The text in the *Measurement Comment* memo field is saved together with the measurement.

Update: The *update* button retrieves the current information about the selected hardware.

Save as: Transfers the measurement to the PC and saves it as a TEMES file.

Erase Measurem.: Mark measurement as already been transferred to the PC.

Search logger: Ask the *Tellert Device Manager* to update all of its ports.

Close: Close dialog box.

2.4.5 Save Measurement from Mass Storage

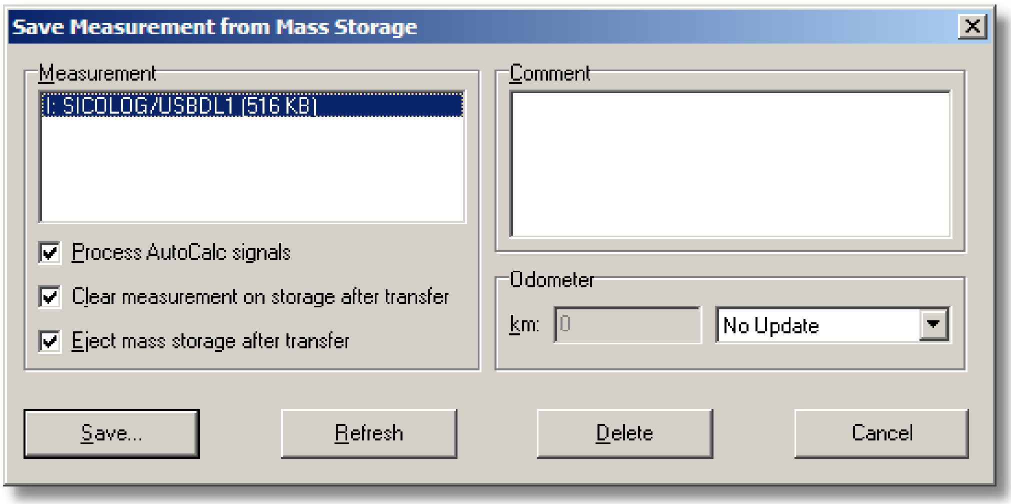


Figure 2-5: Save Measurement from Mass Storage Form.

The comment is saved together with the measurement. Note to check the check-box *Eject mass storage after transfer*, otherwise the media must be manually ejected in the Windows Explorer before it is physically ejected.

km: Reading of the odometer (in units of the *GPS_Distance* signal which is usually km). This edit field is enabled if both the checkbox *Process AutoCalc signals* is checked and the odometer combo box shows either *Measurement Begin* or *Measurement End*.

Save: Save the measurement as a TEMES file.

Refresh: Look for new mass storages.

Delete: Mark the measurement as deleted.

Cancel: Close *Save Measurement from Mass Storage* form.

2.4.6 Recovery Form

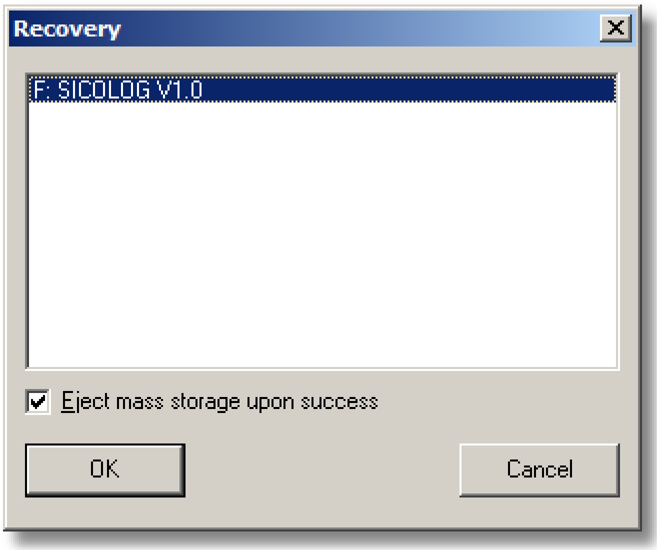


Figure 2-6: Recovery Form.

Only those mass storages are listed which have a marked-as-deleted SICOLOG/USBDL1 USB stick inserted.

Note to check the checkbox *Eject mass storage upon success*, otherwise the media must be manually ejected in the Windows Explorer before it is physically ejected.

A more powerful tool to unerase/repair a SICOLOG/USBDL1 measurement is TSD.EXE.

2.4.7 Variables Editor

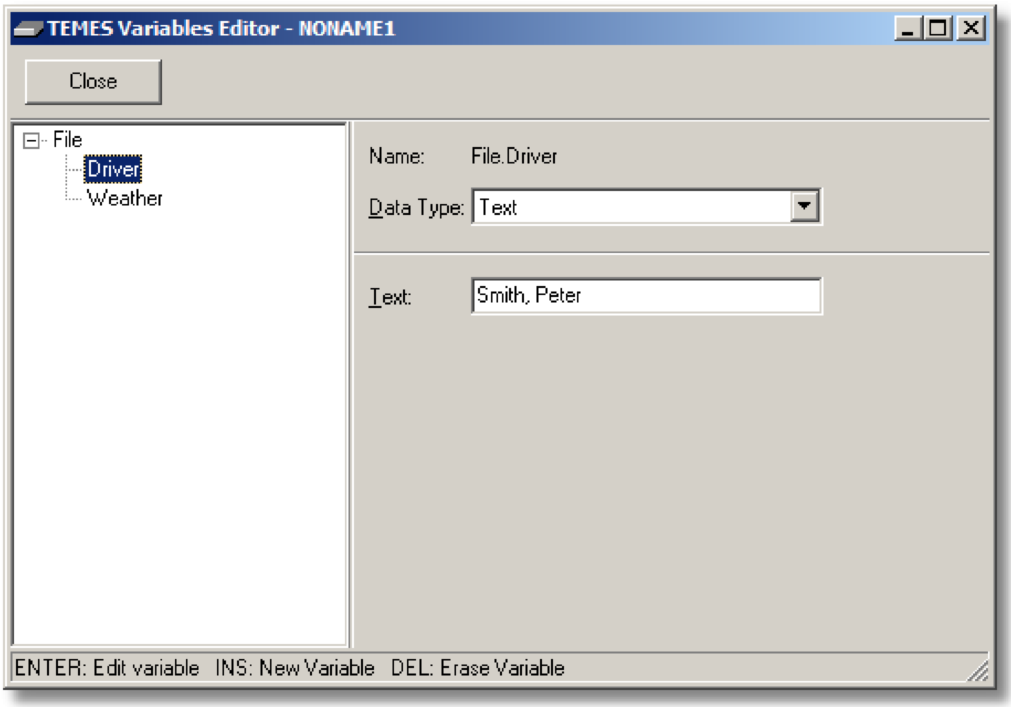
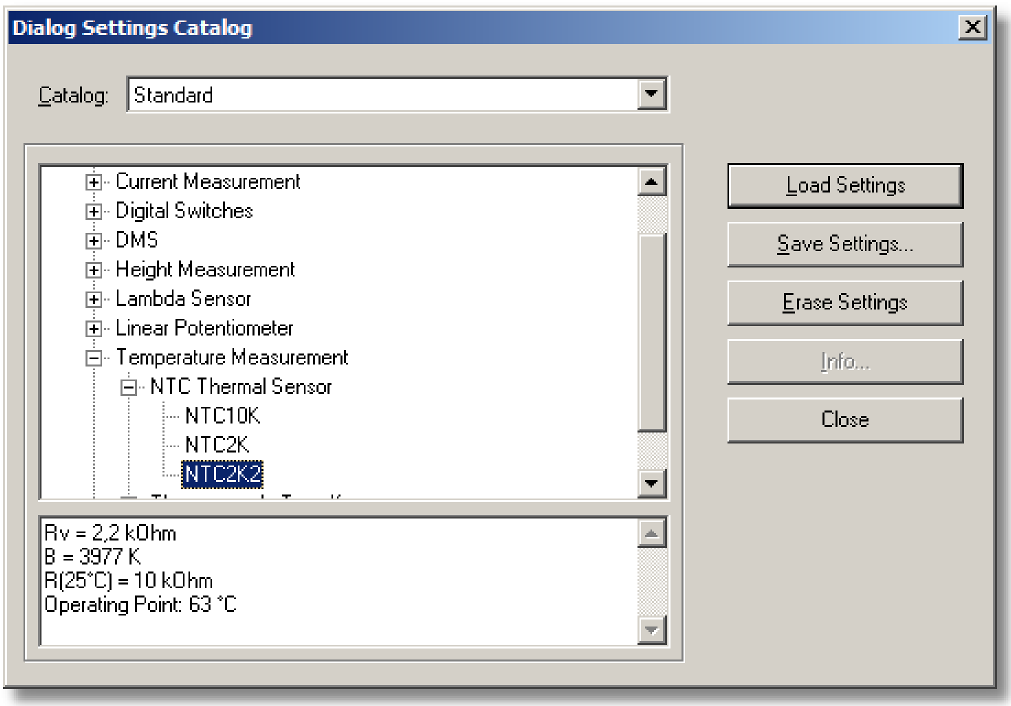


Figure 2-7: Variables Editor.

The variables editor can be used to define variables. A dot can be used to categorize the variables more deeply. These variables are inserted into the print layout module via placeholders.

2.4.8 Catalog Form



Form 2-8: Catalog Settings Form.

The **Catalog** drop-down list box offers *Standard*, *Tellert*, and *Tellert RS232*. Where *Standard* is the default catalog, *Tellert* is a read-only catalog, and where *Tellert RS232* is a read-only catalog for serial inputs.

The settings are accepted with the **Load Settings** button.

The current settings are saved with the **Save Settings** button.

The marked entry is erased with the **Erase Settings** button.

The **Info** button displays a corresponding PDF file (if available).

The dialog box is closed with the **Close** button.

2.5 Child Window

2.5.1 Parameter Tree Node: <device>

2.5.1.1 General Tab

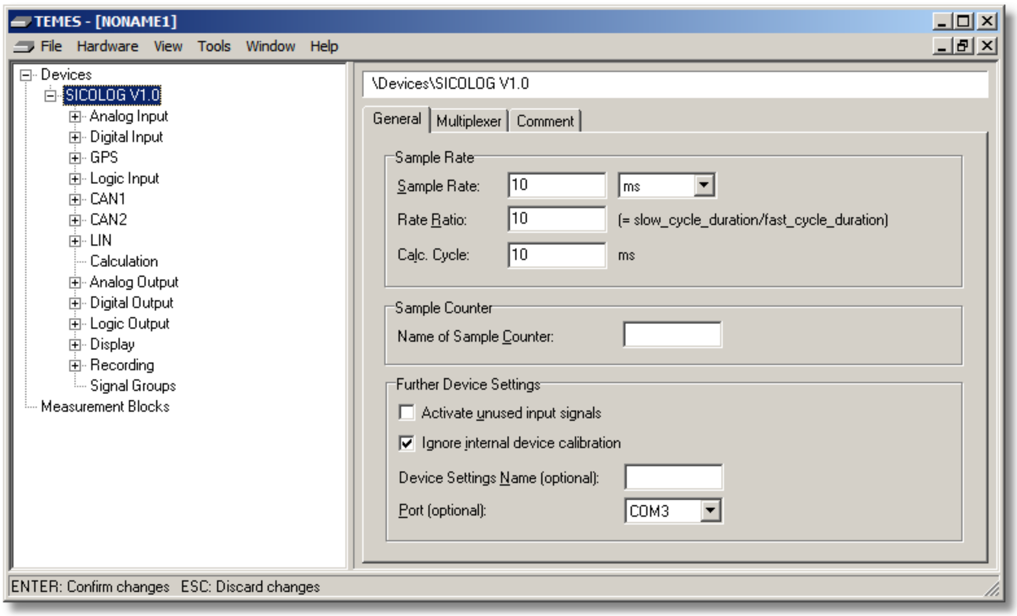


Figure 2-9: Parameter tree node <*device*> (*General* tab).

The **sample rate** is used for the fast recording cycle, and it is the rate which is used for the average values of the voltage and digital input signals (if the corresponding signal is not only recorded in the slow cycle).

The **rate ratio** is an integer multiple of the *sample rate* which is used for the slow recording cycle. Slowly recorded voltage and digital input signals are averaged with the rate of the slow recording cycle in order to get smoother signals.

Note that only those fast samples are recorded by a data logger which also have a corresponding slow sample.

The **calc. cycle** represents the rate used for calculated signals and CAN/LIN signals.

The **sample counter** in this context is a 16-bit free-running wrap-around counter based on the actual fast sample rate and not based on a recording.

Activate unused input signals: This checkbox is used for additional TEMES parameters 139 to enable the calculated signals even if these signals are not used as outputs. Otherwise, TEMES discards/deactivates unused input signals and simplifies the parameter set automatically. Alternatively, you can also add the unused signal to a signal group 81 and transform it to a used input.

Ignore internal device calibration: This checkbox is used to ignore the internal device calibration. The SICOLOG/SICO3/USBDL1 are the only three devices with an additional calculation complexity when this checkbox is turned off. On the other hand, when this checkbox is turned on, the same look-up tables for the SICO2 and DL16CAN can be reused (yielding a smaller parameter set on the device).

The **device settings name** is an optional entry with no further effect in the parameter set.

The **port** has also almost no effect except that the programming is slightly faster (because the port has not to be searched).

2.5.1.2 Multiplexer Tab

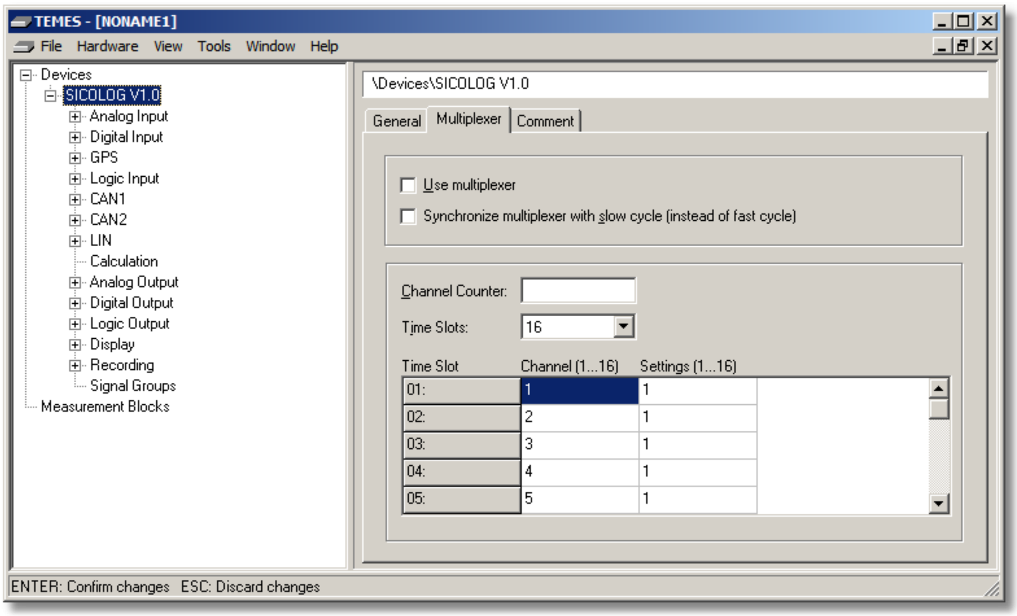


Figure 2-10: Parameter tree node <*device*> (*Multiplexer* tab).

The device broadcasts a control byte to connected multiplexers over its *Logic I/O* port if the **Use multiplexer** checkbox is checked.

The broadcast (of the control byte) is synchronized with the slow recording cycle instead of the fast recording cycle if the **Synchronize multiplexers with slow cycle (instead of fast cycle)** checkbox is checked.

The **Channel Counter** allows to define the current multiplexer index as an input signal.

The number of **Time Slots** (= number of used multiplexer channels) can be chosen from 1 to 16.

The **Settings** (= operating range) can be specified for each time slot or **Channel** (see also *multiplexer TH16MI* 128).

2.5.1.3 Comment Tab

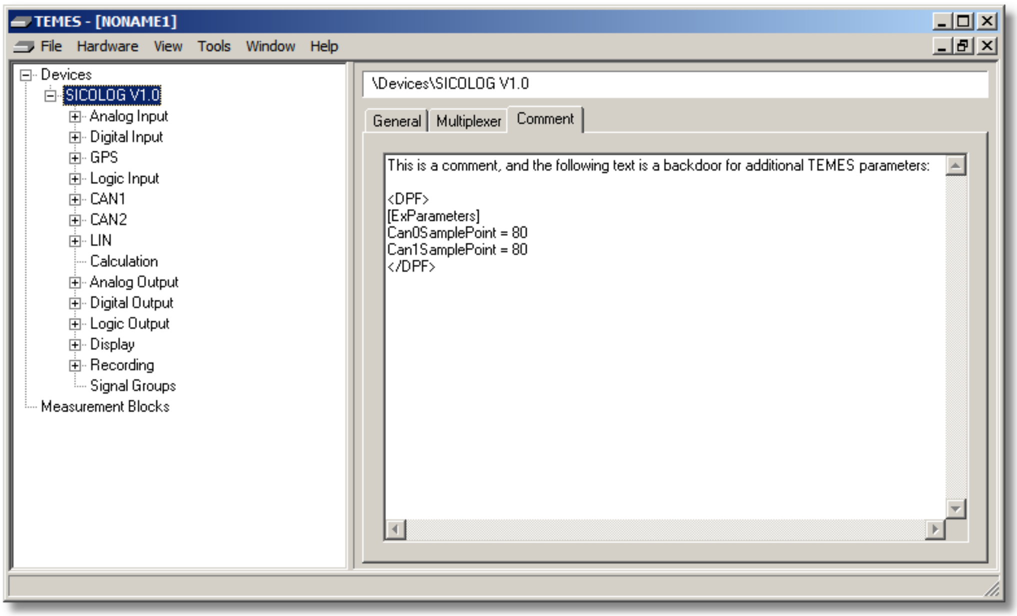


Figure 2-11: Parameter tree node <*device*> (*Comment* tab).

The multi-line comment is not only a comment for the parameter set. This comment is also used as a backdoor for additional TEMES parameters 139.

AutoCalc statements can be inserted between the one-off <*DPF*> and <*/DPF*> tags in the section *[Calc]*. E. g.:

<DPF>
[Calc]
CdfCount=1
Cdf0=append,*ResultName*,*BlockName*,*CdfName*

```
AutoSave=1
AutoExport=4
</DPF>
```

where *ResultName* is the name of the resulting measurement block, *BlockName* is the source measurement block, and *CdfName* is the name of the offline calculation 171 without file extension, which is located in the "*Calc*" subfolder of the TEMES data folder (which is usually "*documents\TEMES\Calc*"). When *AutoSave* is specified as 1, the AutoCalc signals are automatically saved after calculation. When *AutoExport* is specified, an automatic export is invoked. The value is hereby a bitmask consisting of:

AutoExport	Export Format
1	Text
2	MDF3
4	M (Matlab/Octave)
8	R
16	JL (Julia)
32	SCE (Scilab)

2.5.2 Parameter Tree Node: Analog Input

2.5.2.1 Settings Tab

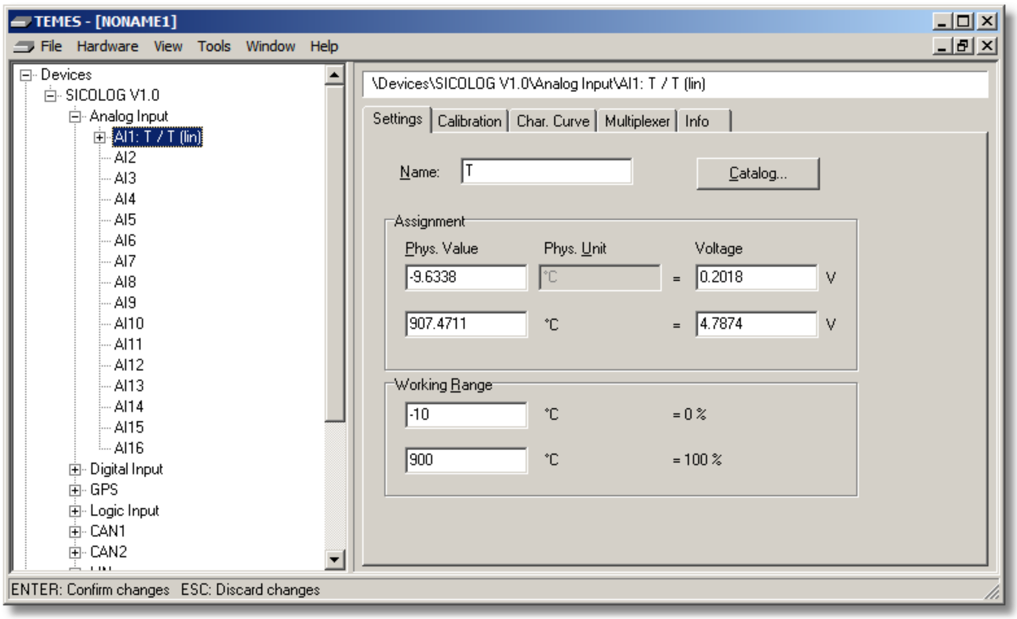


Figure 2-12: Parameter tree node Analog Input (*Settings* tab).

Name: Unique name of the input signal. If this field is empty the corresponding input signal is deactivated.

Catalog: Opens the catalog settings form 39.

Assignment: Definition of the linear relationship. The left side represents the physical signal p which depends on the measured signal m. The relationship is defined by two different physical signal values $p_0 = p(m_0)$ and $p_1 = p(m_1)$, and their corresponding measured signal values m_0 and m_1. Take care that p_0 does not equal p_1, and that m_0 should not equal m_1. Note that the values p_0 and p_1 are also used for the online calibration 35.

$Factor = (p_1 - p_0) / (m_1 - m_0)$
$Offset = p_0 - Factor \cdot m_0$

Final Relationship:
$p(m) = Factor \cdot m + Offset$

If *Factor* and *Offset* of the linear relationship are already known, the following values for m_0, m_1, p_0 and p_1 can be used:

$m_0 = 0$
$m_1 = 1$
$p_0 = Offset$
$p_1 = Factor + Offset$

The following image shows where the variables have to be put into the form. Note that the input fields for m_0, m_1, p_0 and p_1 allow floating point values only:

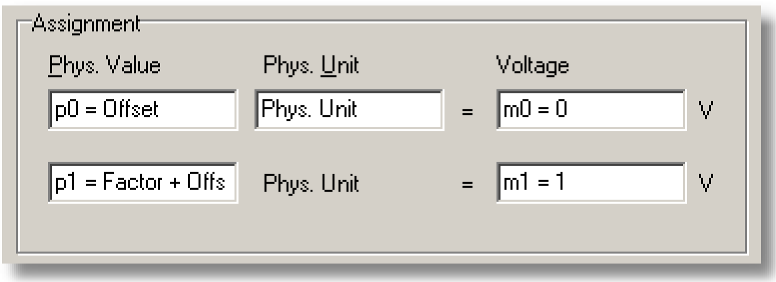


Figure 2-13: Assignment.

Physical Value: The first line defines p_0, the second line defines p_1.

Physical Unit: Unit of the physical signal $p(m)$.

Voltage: The first line defines m_0 in volts, the second line defines m_1 in volts.

Working Range: Definition of the nominal physical signal range. These values will only be used as default values for chart axes, and as default values for output assignments.

2.5.2.2 Calibration Tab

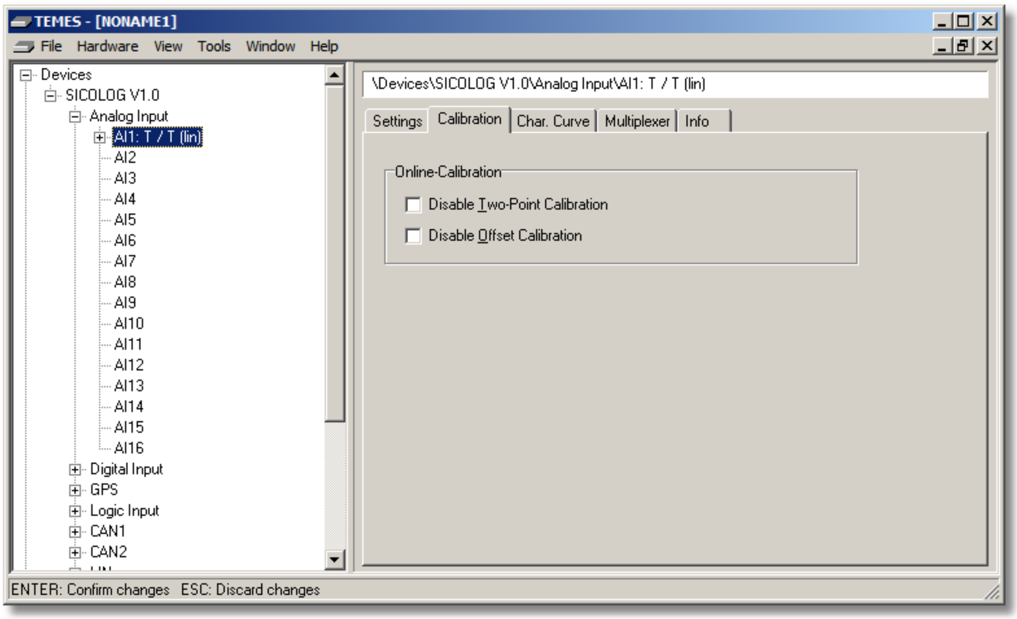


Figure 2-14: Parameter tree node Analog Input (*Calibration* tab).

This tab gives control over what can be changed with the online calibration. If both checkboxes are checked then the signal definition cannot be changed with an online calibration.

2.5.2.3 Char. Curve Tab

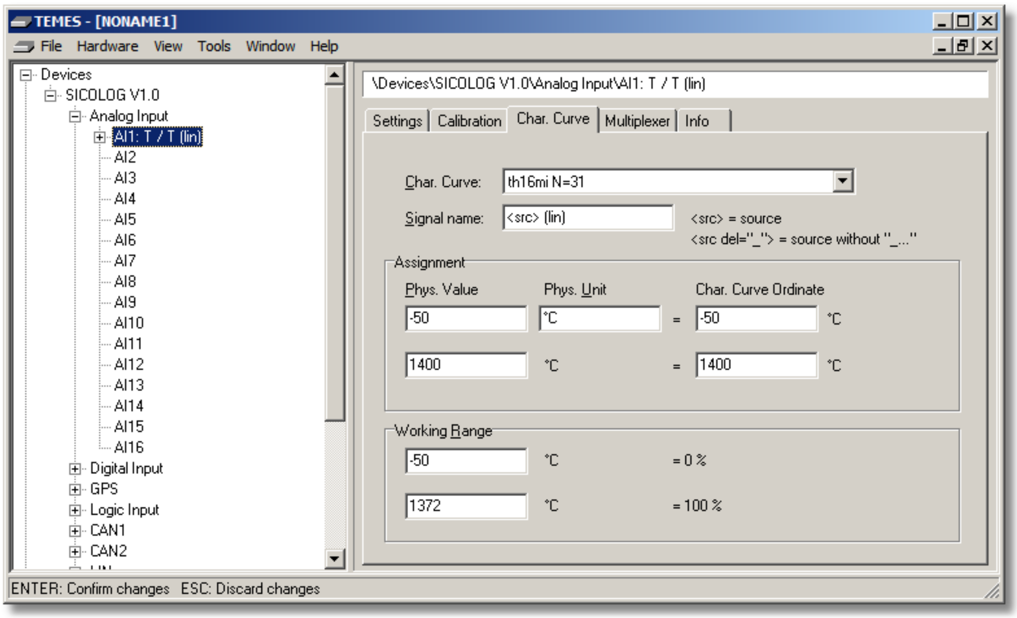


Figure 2-15: Parameter tree node Analog Input (*Char. Curve* tab).

The characteristic curve tab generates a new linearized signal from the voltage signal. The look-up table 183 is selectable in the field **Char. Curve**. The name of the linearized signal is entered in the field **Signal name** where the text <*src*> is used as a placeholder for the name of the voltage signal. The assignment is usually a

1:1 relationship, but it can also be used to transform the unit of the physical signal (e. g. *K*, or °*F* instead of °*C*). The working range is only used as the default value for the axes, and for the default values for output assignments.

2.5.2.4 Multiplexer Tab

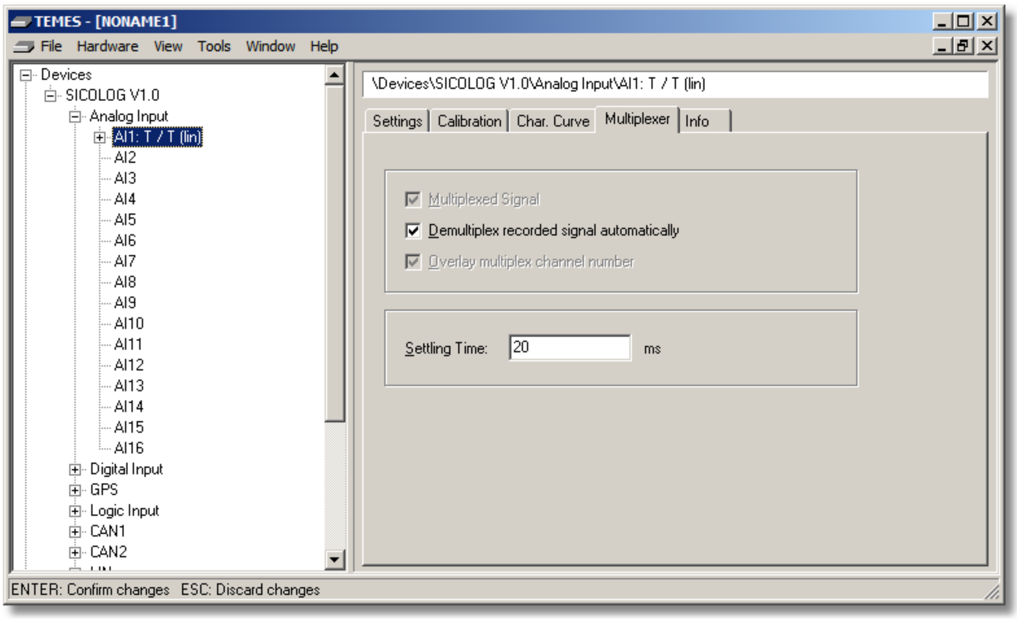


Figure 2-16: Parameter tree node Analog Input (*Multiplexer* tab).

The checkbox **Multiplexed Signal** is checked if a multiplexer is connected to the corresponding voltage input.

The checkbox **Demultiplex recorded signal automatically** is checked if the multiplexed signal shall automatically be demultiplexed after the recording by the *AutoCalc* function.

The checkbox **Overlay multiplex channel number** is checked to overlay the 4 least significant bits of the multiplexed signal with the channel number (to allow a mapping of the demultiplexed signal to the correct channel).

The **Settling Time** is the time between two time slices during which the averaging is turned off (to reduce/avoid signal cross talk). The settling time should be 20 ms for the thermocouple multiplexer *TH16MI*.

See also TH16MI 131 setup.

2.5.2.5 Info Tab

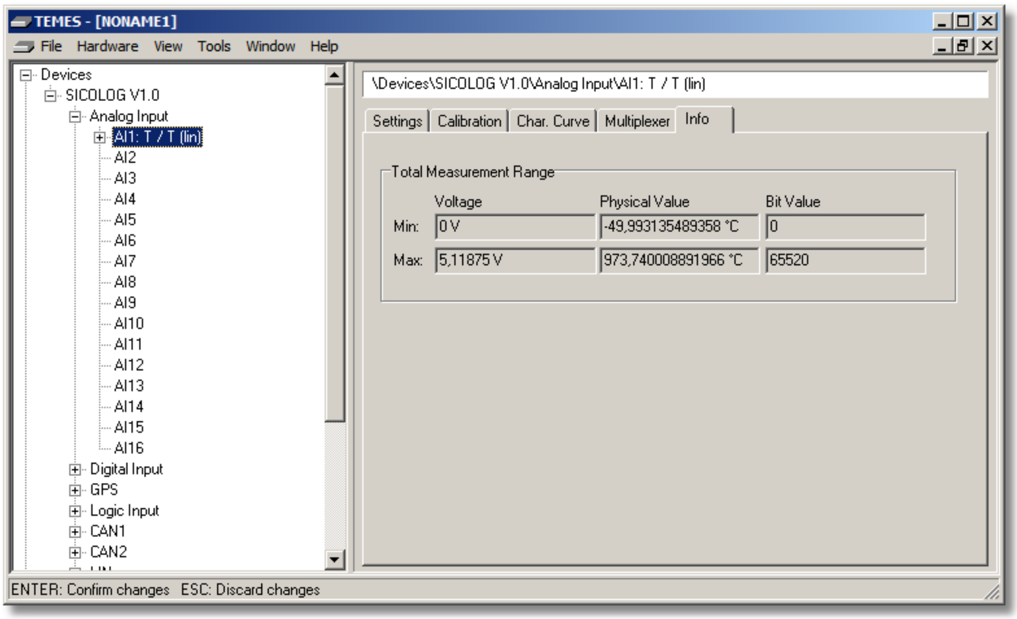


Figure 2-17: Parameter tree node Analog Input (*Info* tab).

The Info tab shows the relationship to the input signal's bit values. This info can be useful for calculated signals which are based on the bit values.

Note, that this tab has been removed from TEMES V1.0.89 onwards due to resource limitations. A protocol file 27 can be generated instead.

2.5.3 Parameter Tree Node: Digital Input

2.5.3.1 Settings Tab

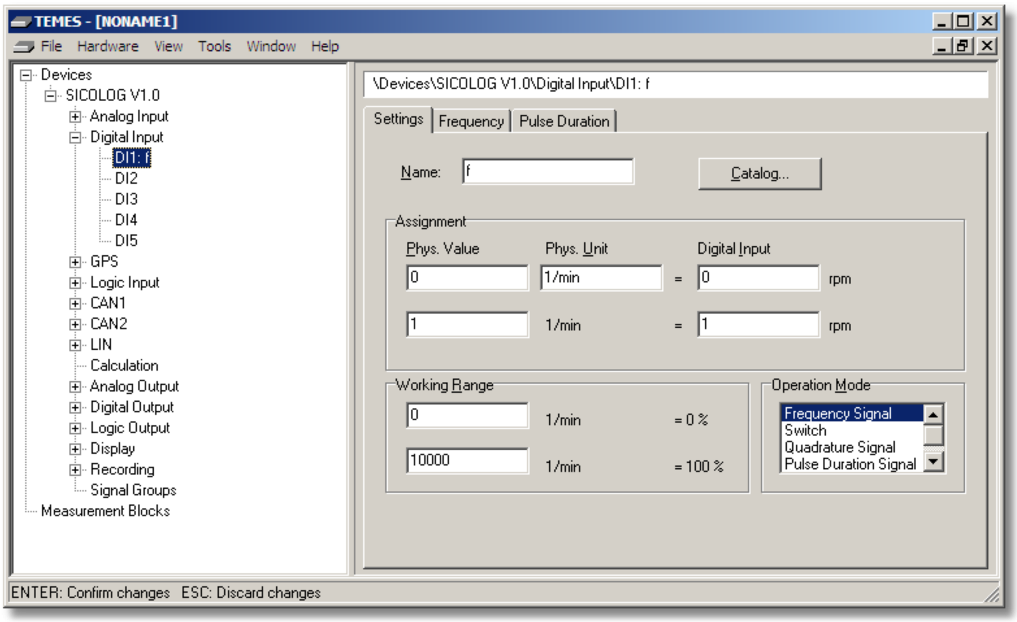


Figure 2-18: Parameter tree node Digital Input (*Settings* tab).

Name: Unique name of the input signal. If this field is empty, the corresponding input signal is deactivated.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

Working Range: Definition of the nominal physical signal range. These values will only be used as default values for chart axes, and as default values for output assignments.

Operation Mode: Definition of the operating mode.

Frequency Signal – The signal is a frequency signal. The frequency tab 48 is used for further parameters.

Switch – The signal is a logical value yielding only 0 or 1.

Quadrature Signal – The signal is a quadrature signal (= phase counting mode signal). This kind of signals is only supported by the *SICO2* family.

Pulse Duration Signal – The signal is the pulse width. The pulse duration tab 49 is used for further parameters.

Counter – The signal is a 16-bit wrap-around counter which is incremented with each pulse.

2.5.3.2 Frequency Tab

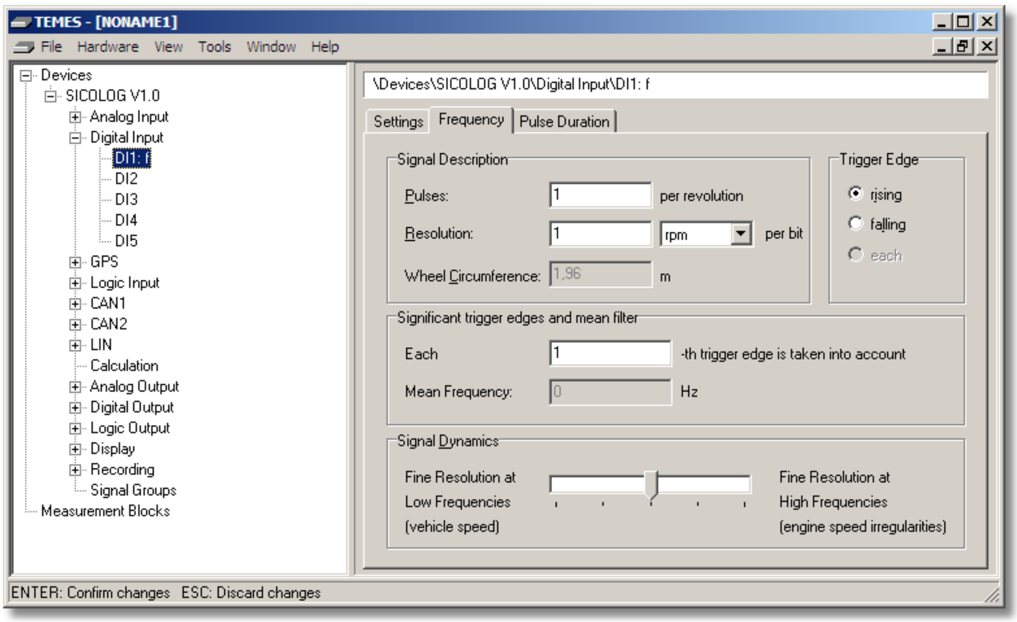


Figure 2-19: Parameter tree node Digital Input (*Frequency* tab).

Pulses: Number of pulses per revolution.

Resolution: Resolution per bit of the resulting frequency signal. Note that frequency signals are represented by 16-bit-values. Therefore, the maximum frequency is received by *Resolution Per Bit* times 65535.

Wheel Circumference: Wheel circumference in meters.

Trigger Edge: Selection whether the signal should be triggered at rising edges or falling edges.

Significant trigger edges: The significant trigger edges are a means to trigger only at a certain number of trigger edges. This is useful e. g. if you have a 7 teeth wheel with uneven teeth, you might want to trigger only at the 7th trigger edge to obtain a smooth signal.

Signal Dynamics: Defines the most important frequency range of interest. This frequency range will be quantized in smaller steps than the less important ones. Moving the wheel to the left side sets the focus on low frequencies (like vehicle speed). Whereas moving the wheel to the right side sets the focus on high frequencies (like engine speed irregularities).

2.5.3.3 Pulse Duration Tab

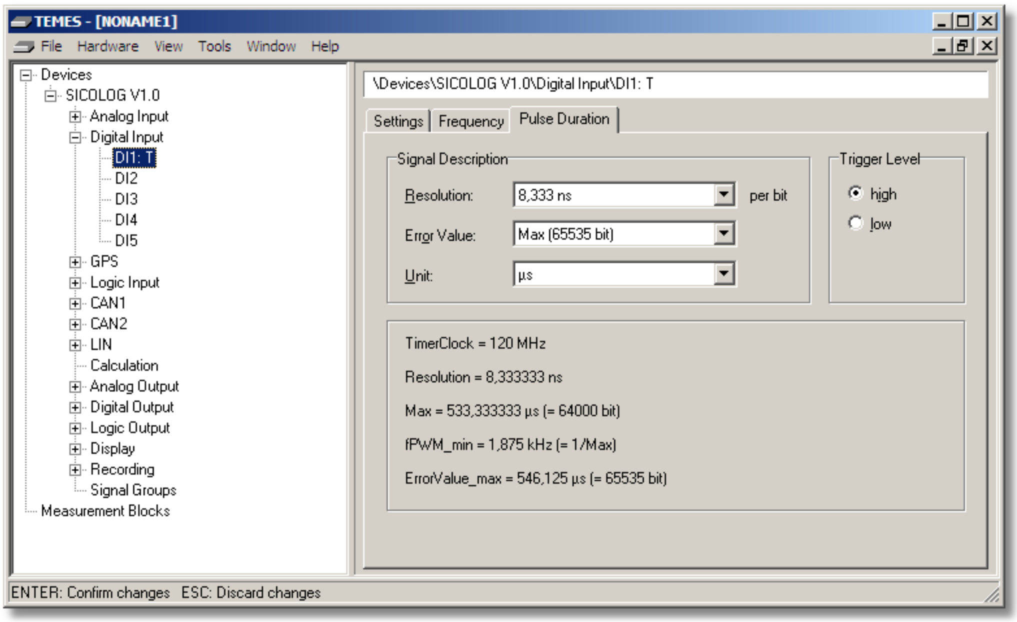


Figure 2-20: Parameter tree node Digital Input (*Pulse Duration* tab).

Resolution: Defines the resolution per bit.

Error Value: This value will be used when the signal is too long to be captured.

Unit: Defines the unit of the pulse width measurement.

2.5.4 Parameter Tree Node: Serial Input

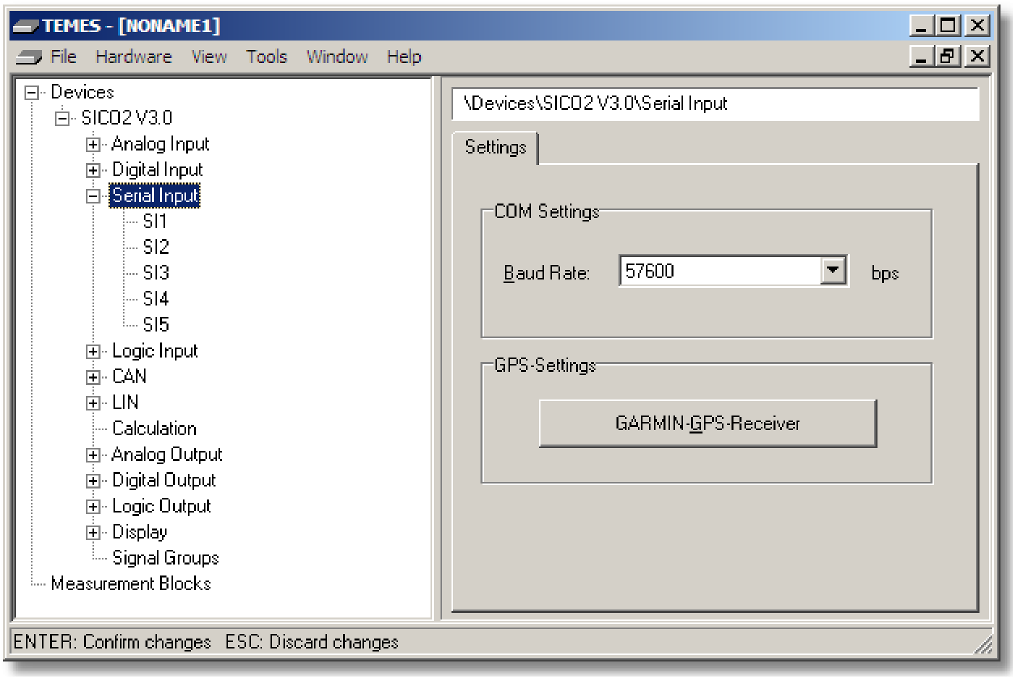


Figure 2-21: Parameter tree node Serial Input.

The **Baud Rate** can be 4800, 9600, 19200, 28800, 38400, 57600 or 115200. If the baud rate is set either to 4800 or 38400, the serial connection is considered as NMEA sentences source from a GPS receiver (= VTG, RMC, GGA sentences), otherwise it is considered as f6 protocol 185 source.

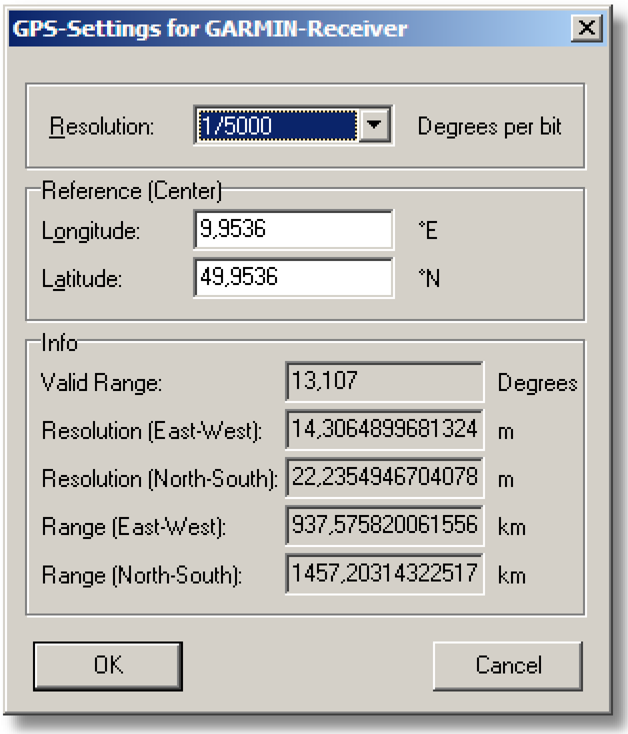


Figure 2-22: GPS Settings form.

Resolution: Internal resolution for the 16-bit-values of longitude and latitude in degrees per bit.

Reference: Longitude and latitude of the representable area's center.

Longitude: Eastern longitude of the center in degrees.

Latitude: Northern latitude of the center in degrees.

2.5.4.1 Settings Tab

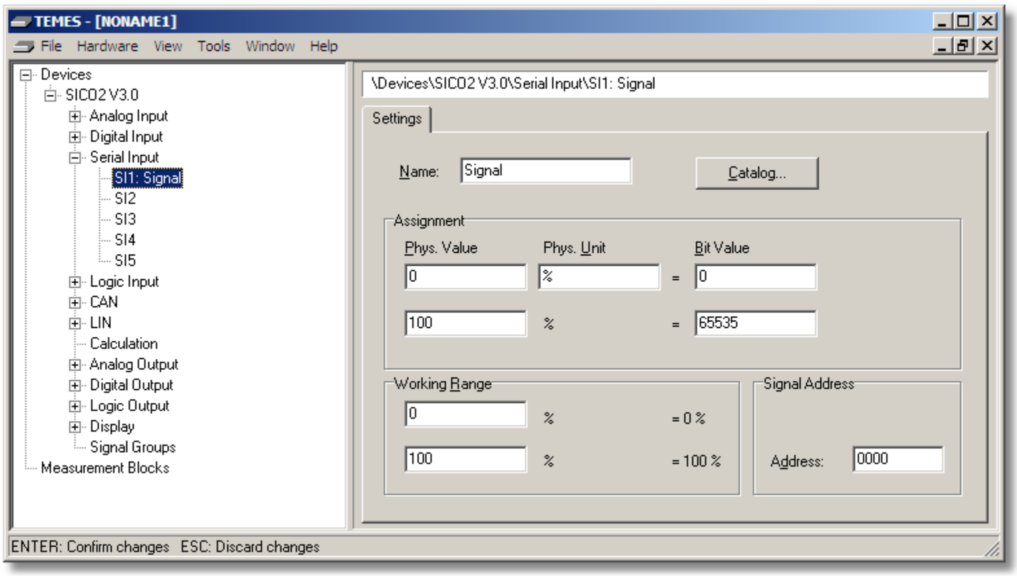


Figure 2-23: Parameter tree node Serial Input (*Settings* tab).

Name: Unique name of the input signal. If this field is empty the corresponding input signal is deactivated.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

Working Range: Definition of the nominal physical signal range. These values will only be used as default values for chart axes, and as default values for output assignments.

Signal Address: Either a hexadecimal 16-bit parameter for the GPS Settings form, or the hexadecimal addresses of the f6 protocol with the high byte being the request byte for the high byte (which is usually $f6_{16}$), and the low byte being the request byte for the low byte.

2.5.5 Parameter Tree Node: GPS

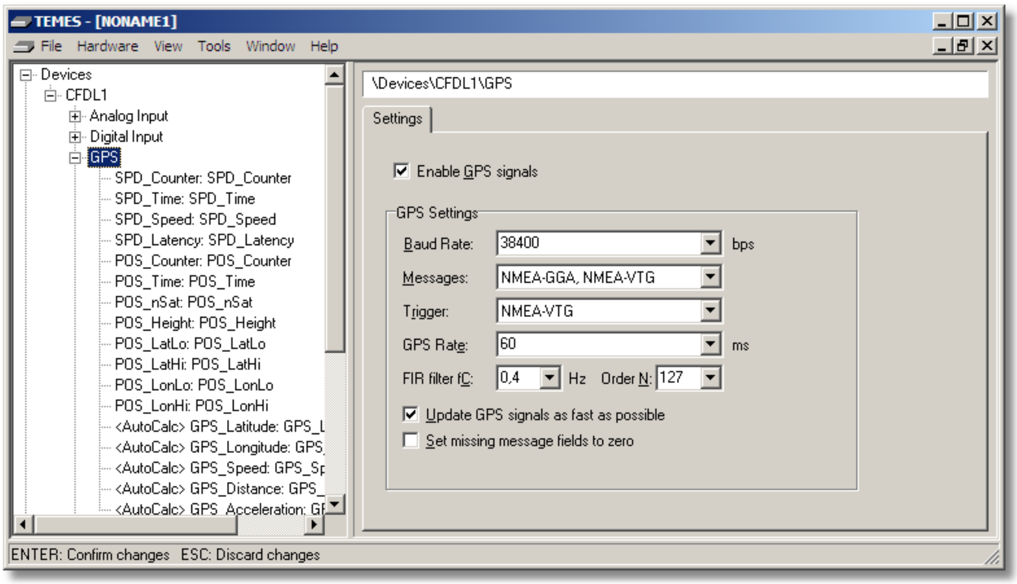


Figure 2-24: Parameter tree node GPS.

The **Enable GPS signals** checkbox is used to turn on the NMEA sentence processing (= GPS signals).

The **Baud Rate** is usually (115200,) 38400 or 4800 bd.

Messages (*CFDL1* only): Defines the NMEA sentences which are recognized by the device.

Trigger (*CFDL1* only): Defines the NMEA sentence which will be used to recognize a new set of GPS signals.

GPS Rate (*AutoCalc* only): Defines the rate of the GPS signals (especially that of the GPS speed signal). This value is also used for the initial GPS signal delay of the curves.

FIR filter fC (*AutoCalc* only): Defines the cut-off frequency of the low-pass filter for the GPS speed signal.

Order N (*AutoCalc* only): Defines the order of the FIR filter for the GPS speed signal.

Update GPS signals as fast as possible (*CFDL1* only): The processed GPS signals are transmitted to the signal MCU as fast as possible. Otherwise the CF card writing data has priority.

Set missing message fields to zero: Set the corresponding field to zero if the message field (of the corresponding NMEA sentence) is missing. Otherwise, keep the previous value.

2.5.5.1 Parameter Tree Node: GPS Signal

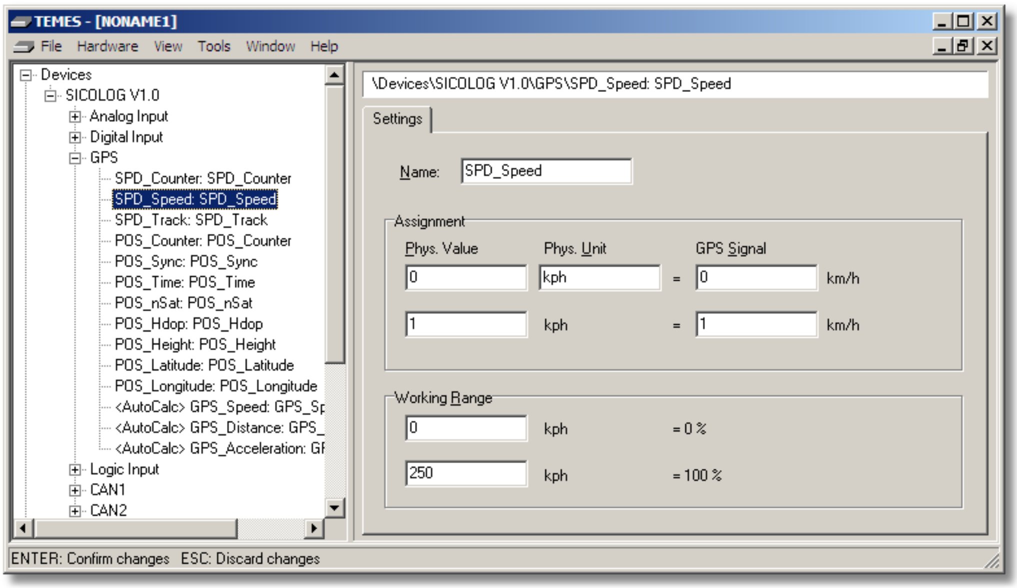


Figure 2-25: Parameter tree node GPS signal.

Name: Unique name of the input signal. If this field is empty the corresponding input signal is deactivated.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

Working Range: Definition of the nominal physical signal range. These values will only be used as default values for chart axes, and as default values for output assignments.

The following SICOLOG/SICO3/USBDL1 signals increase the computational complexity (if used):

Signal	Description
GpsSpeed	GPS speed with a minimal speed value and with limited acceleration (which is 2 *g* by default). This value requires also a minimal amount of visible satellites and a maximal value of HDOP.
GpsSpeedInt	Linear interpolated values of the *GpsSpeed* signal. The signal is delayed by the fix GPS refresh rate.
GpsTrack	GPS track signal with a minimal amount of visible satellites and a maximal value of HDOP and a required minimum speed.
GpsTrackEx	Extended angle of *GpsTrack*. This value corrects the modulo 360° operation of *GpsTrack*.
GpsAcc	Longitudinal acceleration
GpsAccInt	Linear interpolated values of the *GpsAcc* signal. The signal is delayed by the fix GPS refresh rate.

GpsTAcc	Transversal acceleration
GpsTAccInt	Linear interpolated values of the *GpsTAcc* signal. The signal is delayed by the fix GPS refresh rate.
GpsTotalAcc	Square root of ($GpsAcc^2 + GpsTAcc^2$)
GpsLeanAngle	Theoretical lean angle of a motorbike. This value is extracted from the signal *GpsTAcc*: *GpsLeanAngle* = arctan(*GpsTAcc* / *g*) = arctan(*GpsTAcc* / 9.81 m/s²), and is valid for |*GpsTAcc*| ≤ 5 *g*.
GpsLeanAngleInt	Linear interpolated values of the *GpsLeanAngle* signal. The signal is delayed by the fix GPS refresh rate.
GpsAbsLeanAngle	Theoretical magnitude of the lean angle of a motorbike. This value is extracted from the signal *GpsTAcc*: *GpsAbsLeanAngle* = |arctan(*GpsTAcc* / *g*)| = |arctan(*GpsTAcc* / 9.81 m/s²)|, and is valid for |*GpsTAcc*| ≤ 2 *g*.
GpsAbsLeanAngleInt	Linear interpolated values of the *GpsAbsLeanAngle* signal. The signal is delayed by the fix GPS refresh rate.
GpsRadius	Curve radius
GpsTrackDiff	Track difference
GpsAngularSpeed	Angular speed (of the track)
GpsDistance	Integration of the *GpsSpeed* value
GpsDistanceInt	Linear interpolated values of the *GpsDistance* signal. The signal is delayed by the fix GPS refresh rate.
GpsSlope	Distance slope
GpsLatitude	GPS latitude with a minimal number of visible satellites and with a maximal value of HDOP.
GpsLongitude	GPS longitude with a minimal number of visible satellites and with a maximal value of HDOP.

The following signals were introduced in the corresponding firmware:

GpsLeanAngle	SICOLOG/SICO3/USBDL1 firmware V1.0.33
GpsLeanAngleInt	SICOLOG/SICO3/USBDL1 firmware V1.0.33
GpsAbsLeanAngleInt	SICOLOG/SICO3/USBDL1 firmware V1.0.32
All other *Gps*... signals	SICOLOG/SICO3/USBDL1 firmware V1.0.31

2.5.6 Parameter Tree Node: Logic Input

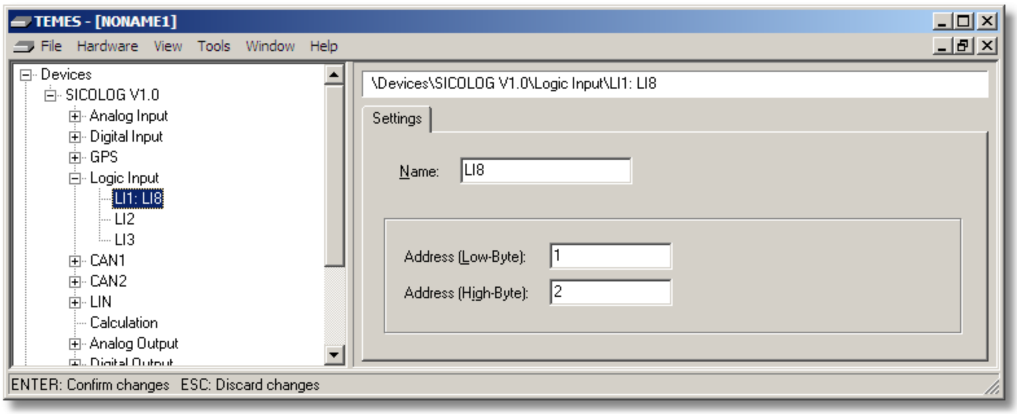


Figure 2-26: Parameter tree node Logic Input.

Name: Unique name of the input signal. If this field is empty, the corresponding input signal is deactivated.

Address (Low-Byte): Definition of a_l.

Address (High-Byte): Definition of a_h.

The input signal is received by ByteValue(a_h) · 256 + ByteValue(a_l), or by ByteValue(a_l) when a_h is empty, with ByteValue(x) equals the 8-bit value of the normalized I^2C address x.

2.5.7 Parameter Tree Node: OBD

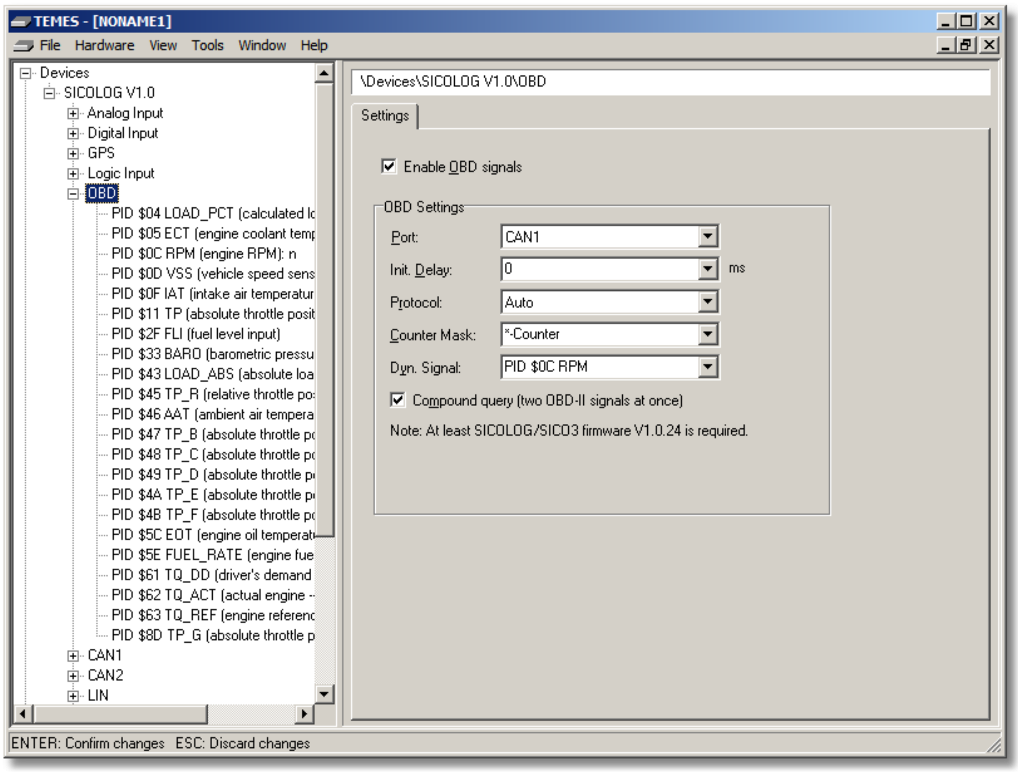


Figure 2-26a: Parameter tree node OBD.

The **Enable OBD signals** checkbox is used to turn on the OBD processing (= OBD signals).

The **Port** is used to select either CAN1 or CAN2 for OBD access.

Init. Delay: Defines the initial waiting period before protocol detection.

Protocol: Defines the used OBD protocol. *Auto* is used to automatically detect the OBD protocol.

Counter Mask: Each OBD signal has an associated counter signal whose name is derived from the name of the OBD signal. A star symbol (*) is a placeholder for the original OBD signal name.

Dyn. Signal: Optionally defines the OBD signal which is requested each 2nd ti-me.

Compound query: In case of OBD-II, the request is done for two OBD signals at once.

Note that there shouldn't be a *Code* definition referring to OBD access within the device comment. Also note that the request cycle is aligned to the cycle of the calculated signals. Also, for some cars, the refresh rate of the OBD signals may be too fast. In this case, the calculation cycle should be enlarged (to e. g. 30 ms).

2.5.7.1 OBD Signal

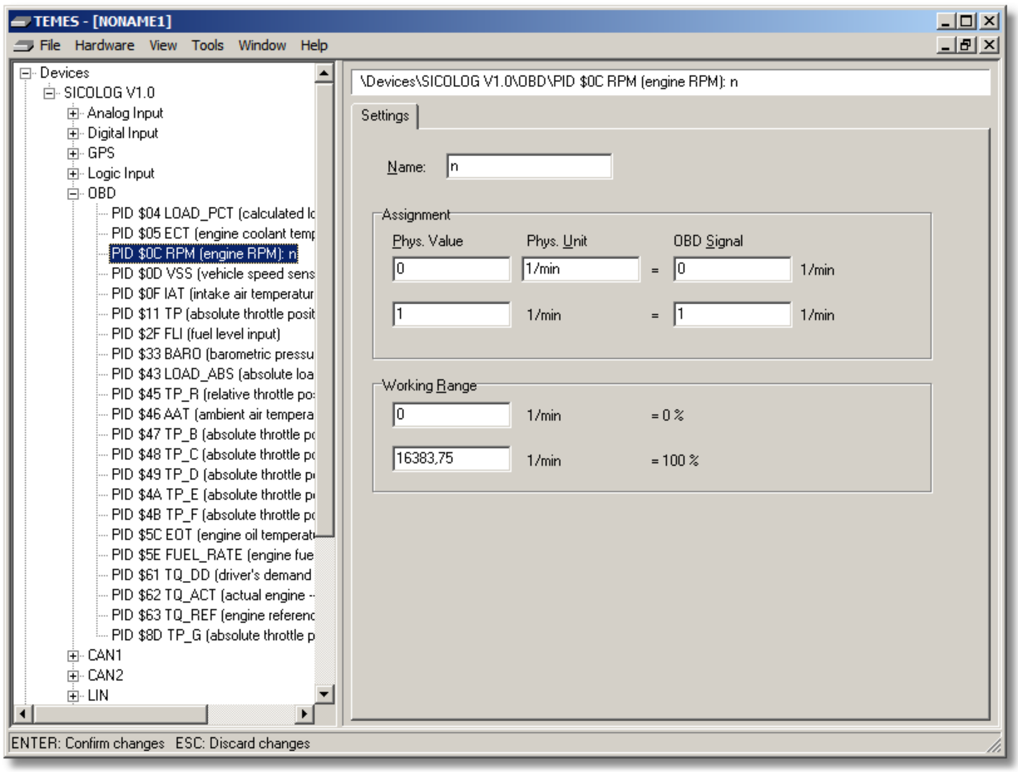


Figure 2-26b: Parameter tree node OBD signal.

Name: Unique name of the input signal. If this field is empty the corresponding input signal is deactivated.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

Working Range: Definition of the nominal physical signal range. These values will only be used as default values for chart axes, and as default values for output assignment.

2.5.8 Parameter Tree Node: CAN

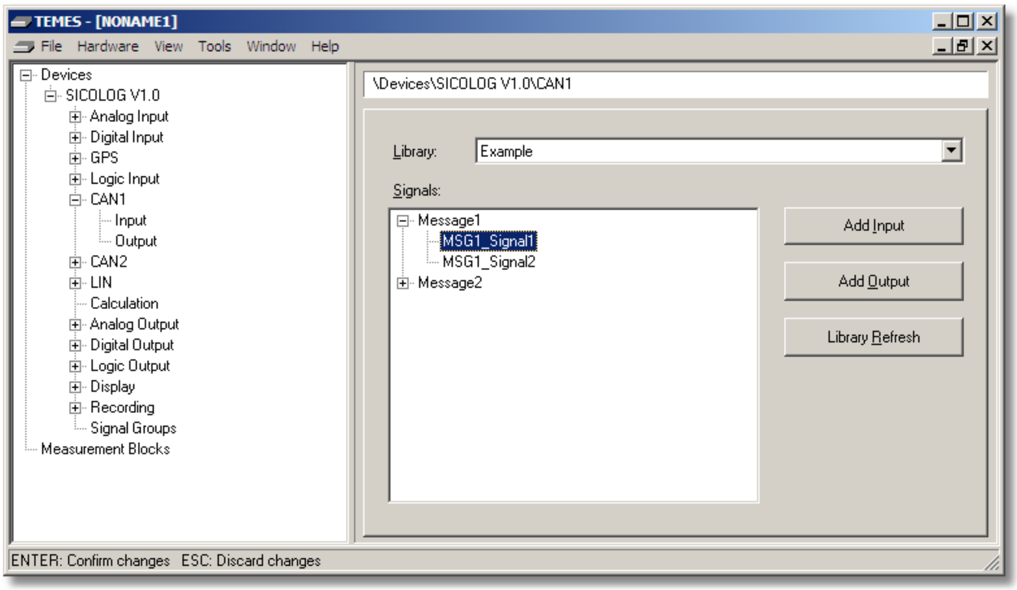


Figure 2-27: Parameter tree node CAN.

Library: Drop-down list box with all known CAN descriptions (= *.CAN* files which are in the *CAN*-subfolder of the TEMES data folder which is usually "*documents\TEMES\CAN*"). The Signals list box is filled with the corresponding CAN messages and signals when a new library is selected.

Signals: Supported CAN messages and CAN signals (below CAN messages). When a CAN signal is selected, also the Add input/output buttons are enabled. Note that CAN signals are suppressed if they are not supported by the corresponding device (meaning if the signal has a data length of 32, then it will only be shown in case of a SICOLOG/SICO3/USBDL1 device).

Add Input: The add input button generates a new input in the parameter tree (as long as the CAN signal has not already an output tree node).

Add Output: The add output button generates a new output in the parameter tree (as long as the CAN signal has not already an input tree node).

Library Refresh: The CAN definition is overtaken as soon as possible from the CAN library (at the moment when a parameter tree node is generated). However, to change the CAN settings of the TEMES document according to a new/changed CAN description, this button must be pressed: The CAN identifier of the same CAN message names are changed, and the CAN message names of the same CAN message identifiers are changed. CAN signals are treated the same way (with start bit and bit length instead of the id respectively). CAN signals which are not mappable are removed in the TEMES document.

2.5.8.1 CAN Input Message

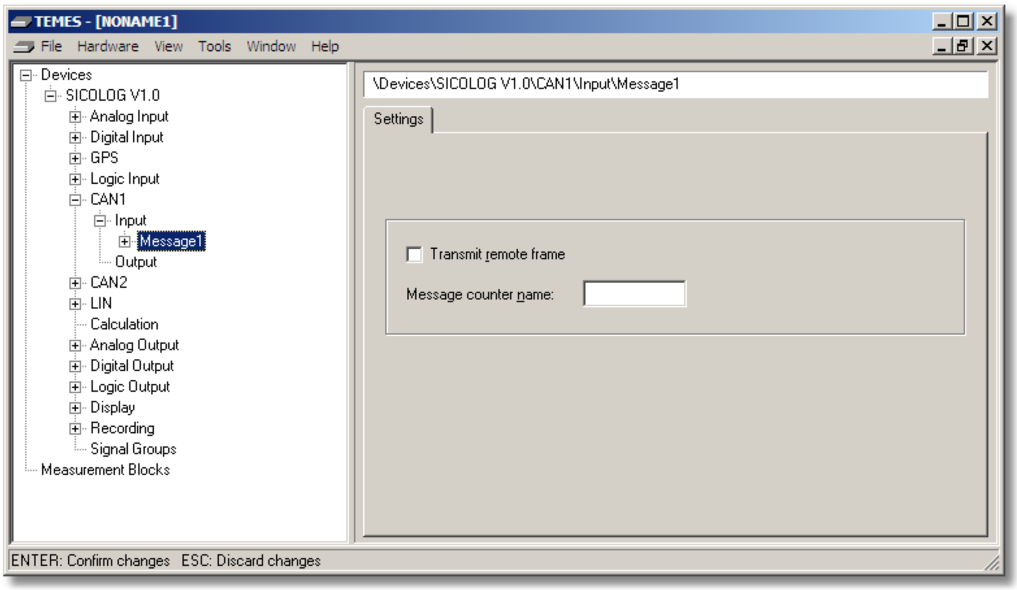


Figure 2-28: Parameter tree node CAN input message.

Transmit remote frame: The *transmit remote frame* checkbox generates the header of a remote frame whenever the CAN message is requested.

Message counter name: TEMES creates a 16-bit wrap-around counter with the message count when the message counter name field is not empty.

2.5.8.2 CAN Input Signal

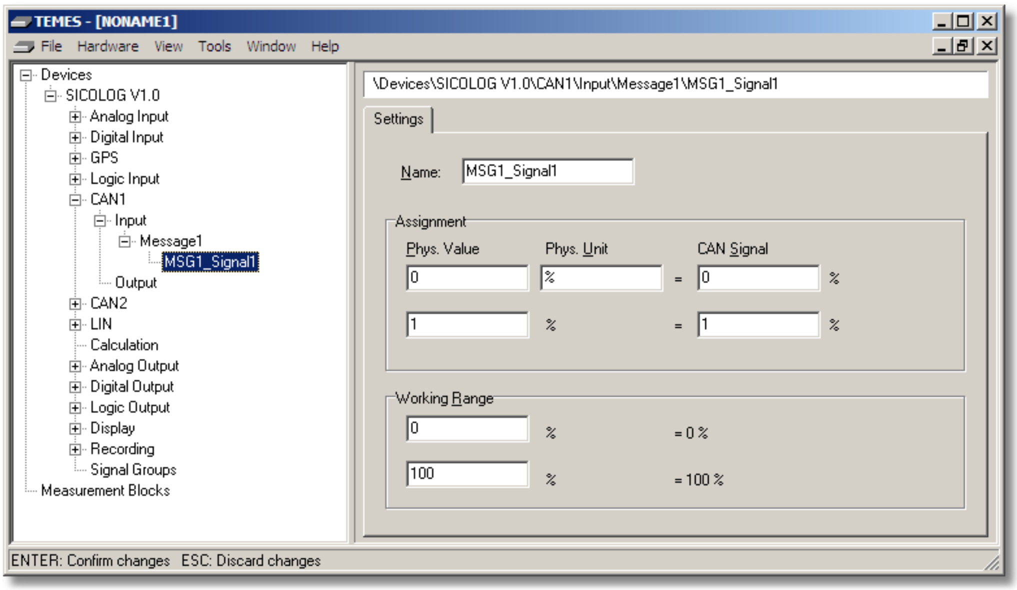


Figure 2-29: Parameter tree node CAN input signal.

Name: Unique name of the input signal. If this field is empty, the corresponding input signal is deactivated.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

Working Range: Definition of the nominal physical signal range. These values will only be used as default values for chart axes, and as default values for output assignments.

2.5.8.3 CAN Output Message

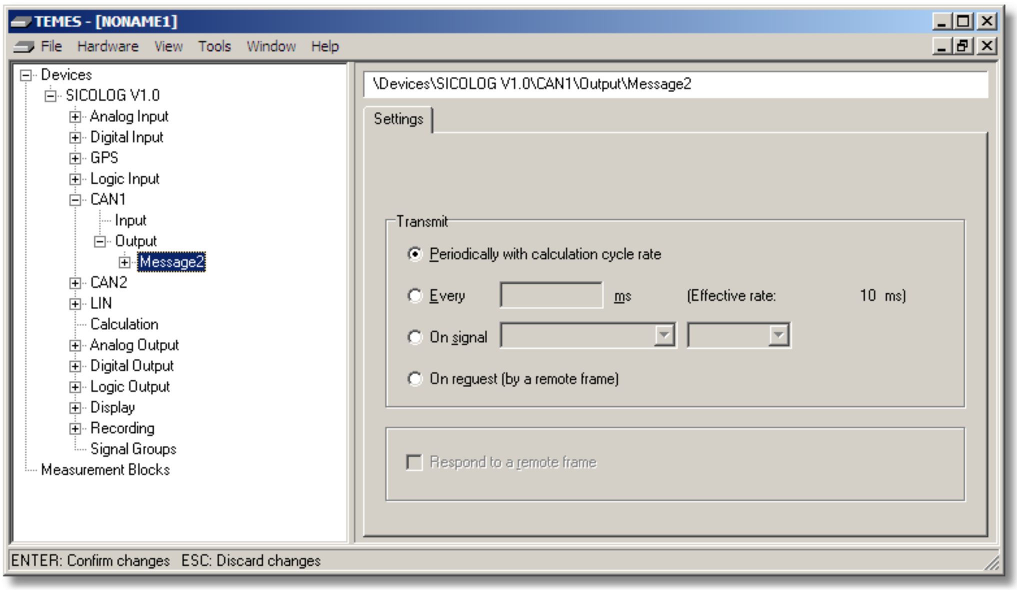


Figure 2-30: Parameter tree node CAN output message.

The CAN message is sent **Periodically with calculation cycle rate** if this radio button is selected.

The **every *x* ms** radio button, where *x* is an integer multiple of the calculation cycle rate, causes the device to send a CAN message with the *effective rate*.

The **On signal** radio button queries a value during the calculation cycle, and if this value is true, the device sends the corresponding CAN message.

The **On request (by a remote frame)** radio button sends a CAN message during the calculation cycle on the request by a remote frame.

The **Respond to a remote frame** checkbox allows the device to answer a header of a remote frame on the CAN bus with the message.

2.5.8.4 CAN Output Signal

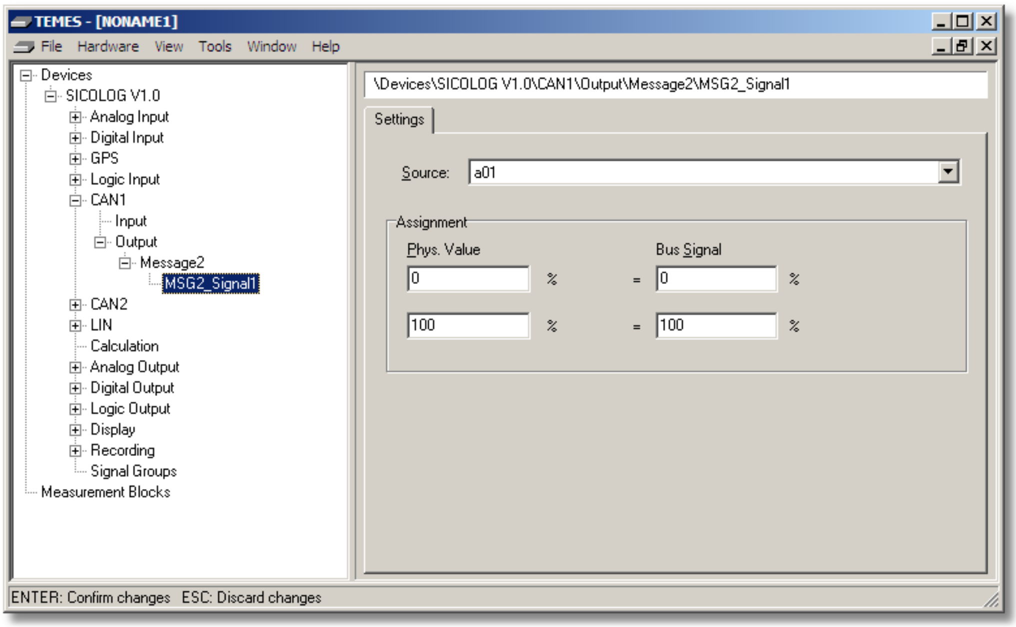


Figure 2-31: Parameter tree node CAN output signal.

Source: List of all available input signals.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

2.5.9 Parameter Tree Node: CAN_FD

The CAN FD settings are almost the same as those of the CAN settings 58.

The CAN editor can be used to create a CAN FD description.

2.5.10 Parameter Tree Node: LIN

The LIN settings are almost the same as those of the CAN settings 58.

The CAN editor can be used to create a LIN (= CAN) description.

The number of LIN messages is device dependent and is limited to 8 messages for USBDL1, SICOLOG, SICO3, SICO2 and DL16CAN.

A LIN master unit has usually an associated schedule. Which is described in the next section.

2.5.10.1 LIN Schedule

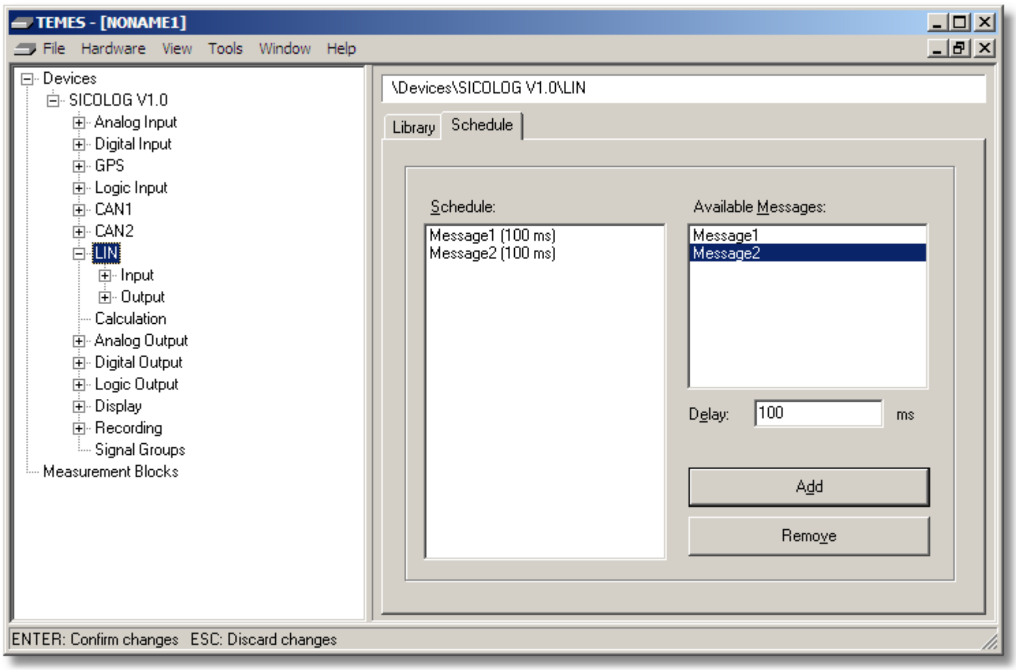


Figure 2-32: Parameter tree node LIN (*schedule* tab).

The **Delay** is the time gap between two adjacent LIN messages (for SICO2 and DL16CAN), or between two adjacent beginnings of the LIN messages (for SICOLOG/USBDL1 and SICO3).

The **Available Messages** is a list of all LIN messages of the corresponding LIN (= CAN) description (tab *Library → Library* 58).

The number of **Schedule** list items is limited depending on the device. This limit is 16 entries by the supported devices.

2.5.11 Parameter Tree Node: XCP/CCP

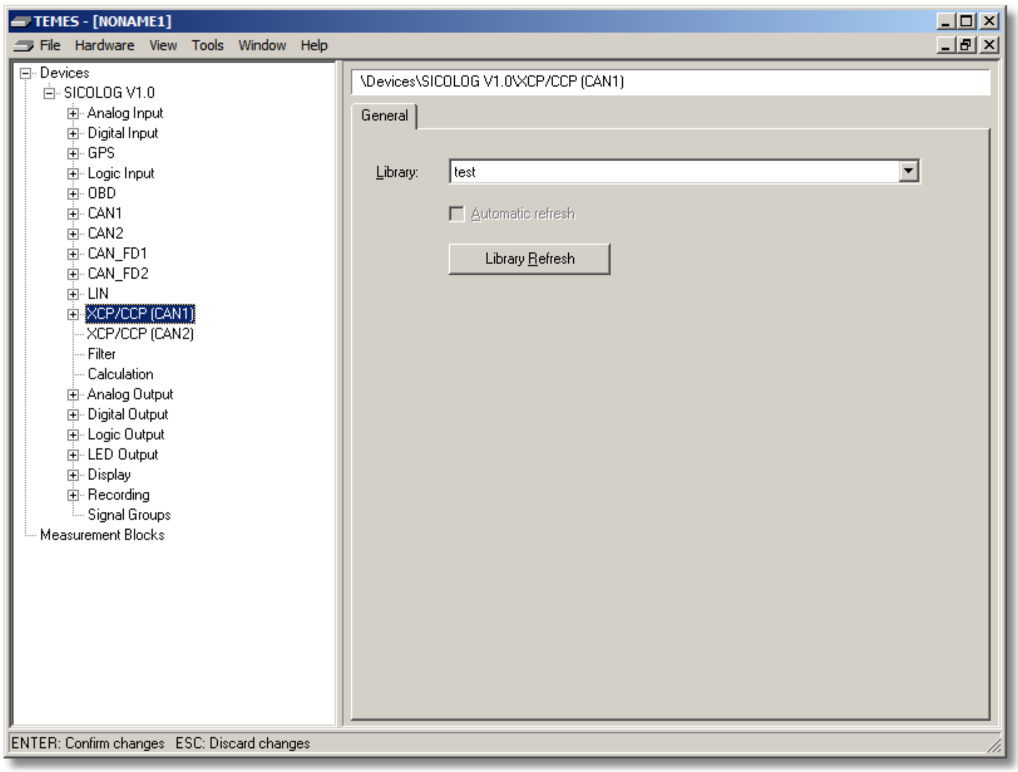


Figure 2-31a: Parameter tree node XCP/CCP.

Library: Drop-down list box with all known XCP/CCP descriptions (= *.XCP* files which are in the *CAN*-subfolder of the TEMES data folder which is usually "*documents\TEMES\CAN*").

Automatic Refresh: This checkbox is reserved for future use. (When checked, the XCP/CCP definition is updated before the configuration is programmed and the description file has been changed.)

Library Refresh: The XCP/CCP definition is overwritten from the XCP/CCP description file. In order to change the XCP/CPP settings of the TEMES document according to a new/changed XCP/CCP description file, this button must be pressed.

2.5.11.1 XCP/CCP Signal

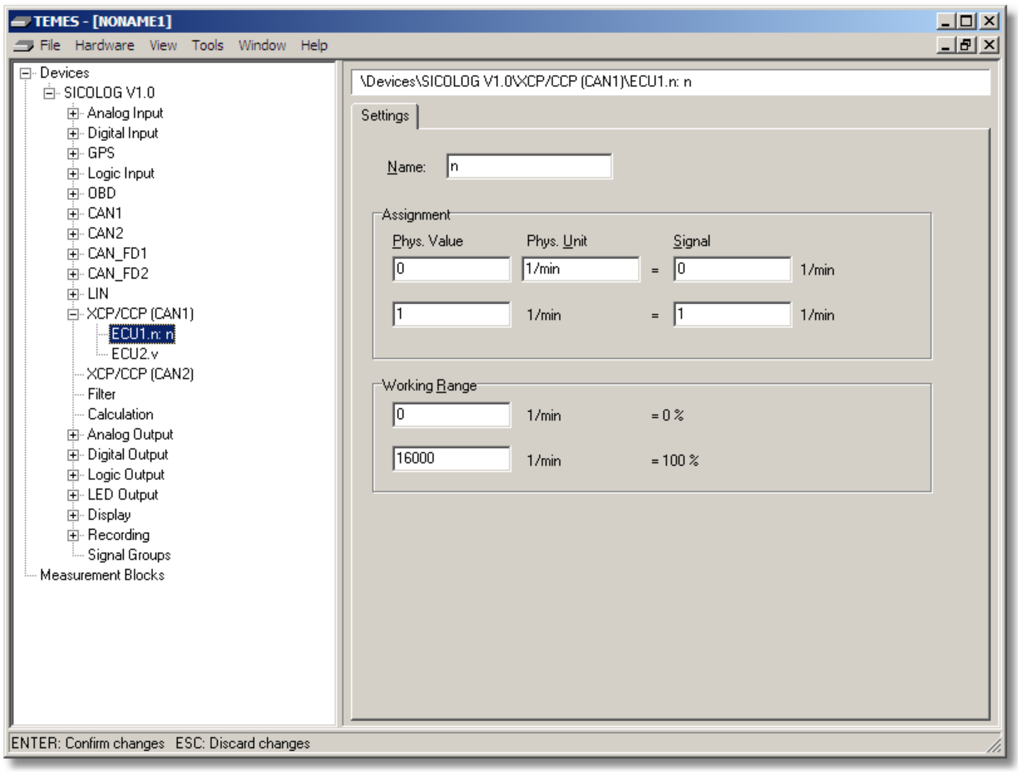


Figure 2-31b: Parameter tree node XCP/CCP signal.

Name: Unique name of the input signal. If this field is empty, the corresponding input signal is deactivated. Note that a second signal with appending *Counter* to its signal name will be created as well. The second signal is incremented by one, each time, the first signal becomes refreshed.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

Working Range: Definition of the nominal physical signal range. These values will only be used as default values for chart axes, and as default values for output assignment.

2.5.12 Parameter Tree Node: Filter

A new filter signal can here be inserted with the *Ins* key.

2.5.12.1 Filter

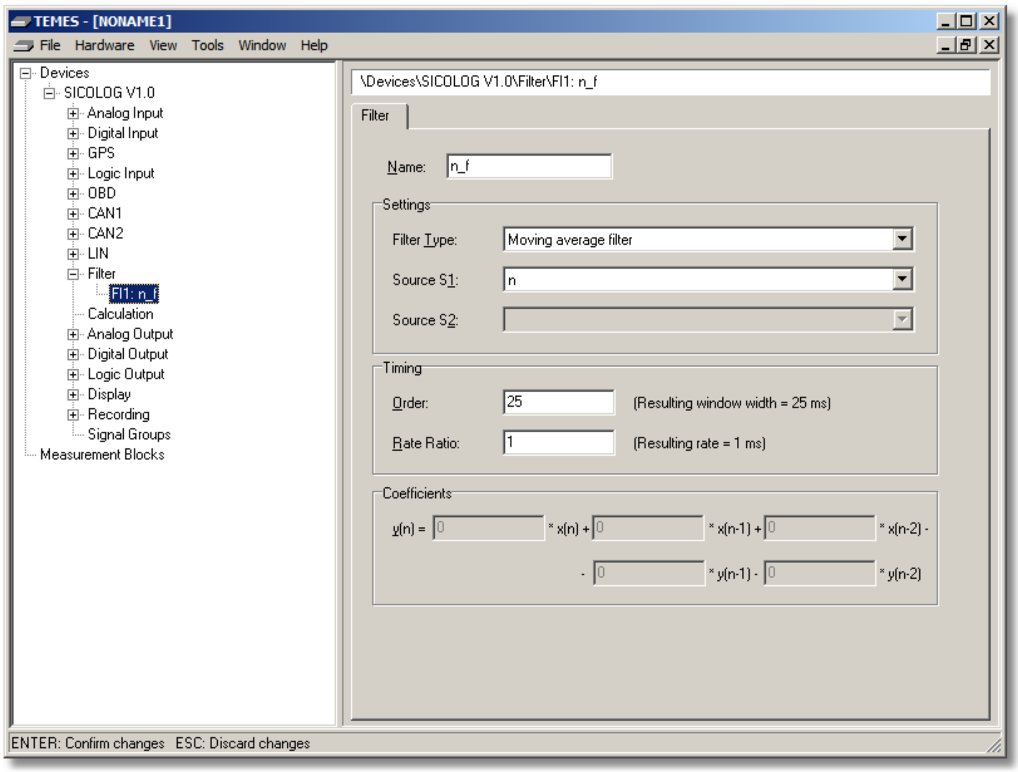


Figure 2-32a: Parameter tree node Filter (*Filter* tab).

Name: Unique name of the input signal. If this field is empty the corresponding input signal is deactivated.

Filter Type: The filter type is either a *moving average filter*, *delay filter*, *IIR filter 2nd order (biquad)*, *maximum filter*, *minimum filter*, *edge filter*, *rising edge filter*, or *falling edge filter*.

Source S1: The name of the input signal.

Source S2: Optionally the resetting signal when S_2 changes.

Order: Defines the window width (and also the number of storage locations). Note that the storage is scarce and the order number should be ≤ 100 for the *moving average filter* and the *delay filter*. For the *moving average filter*, the signal delay is given by half of the window width.

Rate Ratio: Defines the filter rate as a multiple of the cycle of the calculation signals. This value usually equals 1, and is used only to extend the window width without increasing the order.

Coefficients: The floating point coefficients of the *IIR filter 2nd order (biquad)* which can be looked up online via a biquad calculator.

Note that at least SICOLOG/SICO3/USBDL1 firmware version 1.0.25 is required for the filter implementations.

2.5.13 Parameter Tree Node: Calculation

A new calculated input signal can here be inserted with the *Ins* key.

2.5.13.1 Settings Tab

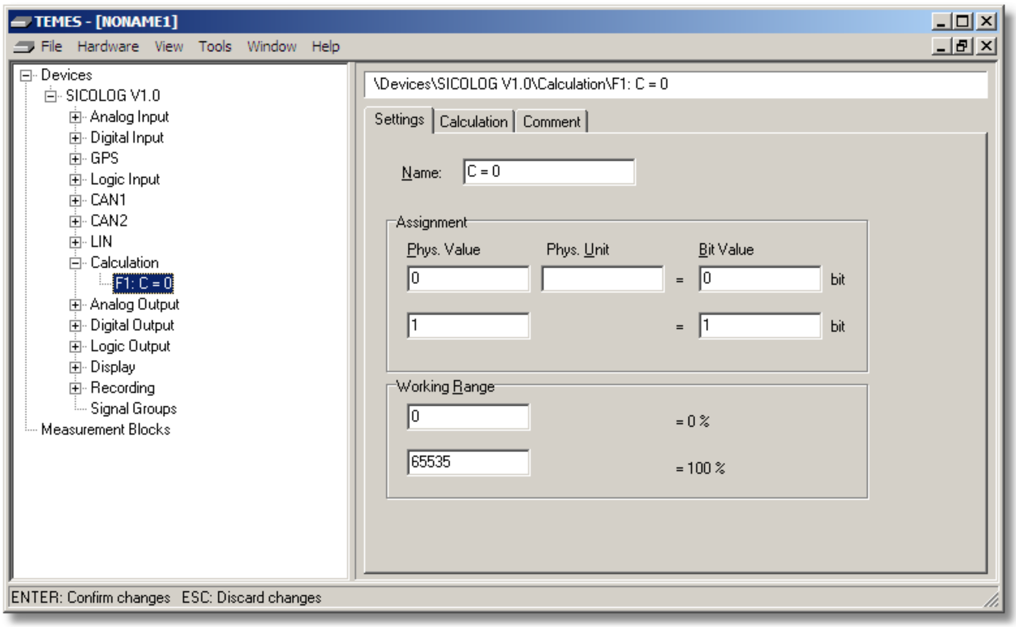


Figure 2-33: Parameter tree node Calculation (*Settings* tab).

Name: Unique name of the input signal. If this field is empty, the corresponding input signal is deactivated.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

Working Range: Definition of the nominal physical signal range. These values will only be used as default values for chart axes, and as default values for output assignments.

2.5.13.2 Calculation Tab

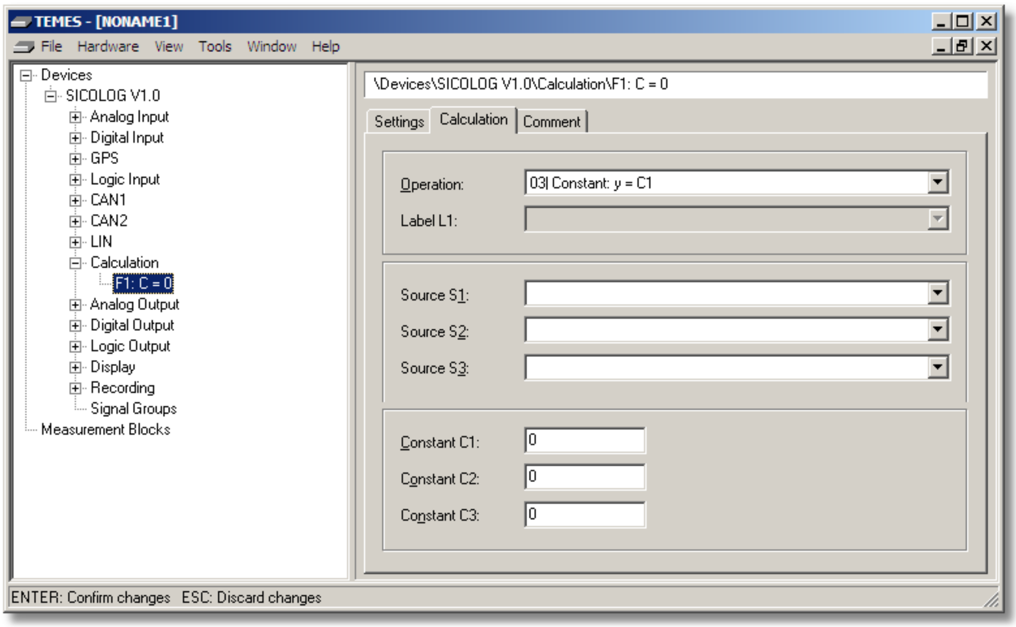


Figure 2-34: Parameter tree node Calculation (*Calculation* tab).

The **Operation** drop-down list selects the operation of the input signal (see below).

The **Source S1** drop-down list represents source S_1.

The **Source S2** drop-down list represents source S_2.

The **Source S3** drop-down list represents source S_3.

The **Constant C1** edit box represents the constant C_1.

The **Constant C2** edit box represents the constant C_2.

The **Constant C3** edit box represents the constant C_3.

The operations are defined as follows:

Op01 – External Function: (not yet supported)

Op02 – Char. Curve Lookup: (not yet supported)

Op03 – Constant: $y = C_1$
Constant definition.

Op04 – Linear Transform: $y = C_1 \cdot S_1 + C_2$

Both constants may be floating point values. For SICO2 and DL16CAN: The constants are automatically adjusted for a calibration transformation if S_1 is a voltage input signal.

Op05 – Sum: $y = (S_1 + S_2) / \lfloor C_1 \rfloor$ with $\lfloor x \rfloor$ = (int) x

Op06 – Difference: $y = (\lfloor C_1 \rfloor + S_1 - S_2) / \lfloor C_2 \rfloor$

Op07 – Product: $y = (S_1 \cdot S_2) / \lfloor C_1 \rfloor$

Op08 – Ratio: $y = (S_1 \cdot \lfloor C_1 \rfloor) / S_2$

Op09 – Equality: $y = 1$ if $S_1 = S_2$ else 0

Op10 – Less Than: $y = 1$ if $S_1 < S_2$ else 0

Op11 – Less Than or Equal To: $y = 1$ if $S_1 \leq S_2$ else 0

Op12 – Condition: $y = S_2$ if $S_1 \neq 0$ else S_3

Op13 – Set constant conditionally: $S_2 = S_3$ if $S_1 \neq C_1$

Op14 – And: $y = S_1 \wedge S_2$ (bitwise)

Op15 – Or: $y = S_1 \vee S_2$ (bitwise)

Op16 – Exclusive Or: $y = S_1 \oplus S_2$ (bitwise)

Op17 – Intel ↔ Motorola

Op18 – Mirror bits (lsb → msb, msb → lsb)

Op19 – Delay: $y(k) = S_1(k-1)$

Op20 – Define counter

A counter is internally represented as 32-bit value. If the counter is directly accessed by its name, the lower 16-bit word of the counter is returned for the devices SICO2, DL16CAN and CFDL1.

Op21 – Increment counter: $S_1 = S_1 + S_2$

Op22 – Decrement counter: $S_1 = S_1 - S_2$

Op23 – Set counter conditionally: $S_2 = C_2$ if $S_1 \neq C_1$

Op24 – Read counter: $y = S_1 / S_2$

Op25 – Set quadrature position conditionally: $S_2 = S_3$ if $S_1 \neq C_1$

About calculated signals: All operations are performed with the quantized values of the corresponding input signal (= internal unsigned 16-bit representation of the physical signal for SICO2, DL16CAN and CFDL1. And signed 32-bit representation for the SICOLOG/SICO3/USBDL1 respectively). If operations on physical signals are required, then first a transform is needed from physical signal into bit domain. Then, the actual operation can be performed. And finally, the result can be transformed back from bit domain to physical domain. The transform between the two domains is done via a linear transform with the two constants *factor* and *offset*:

$physical\_value = factor \cdot b + offset$, where b is the corresponding bit value

Example 1: Two distances shall be added. The distance d_1 and d_2 are defined by

$d_1(b_1) = factor_1 \cdot b_1 + offset_1$, where b_1 is the bit value of d_1
$d_2(b_2) = factor_2 \cdot b_2 + offset_2$, where b_2 is the bit value of d_2

The addition of d_1 and d_2 yields

$$d_1(b_1) + d_2(b_2) = (factor_1 \cdot b_1 + offset_1) + (factor_2 \cdot b_2 + offset_2)$$

To simplify this equation we need to make the assumption that factor1 equals factor2. This assumption yields

$$d_1(b_1) + d_2(b_2) = factor_1 \cdot (b_1 + b_2) + (offset_1 + offset_2)$$

The final equation can be calculated with *Op05* since b_1 and b_2 are in bit domain. The required steps can be summarized by:

1) Take care that both summands have the same factor. If not, use a linear transformation to adjust the factor of one of the summands.
2) Apply operation *Op05* to add the bit representation of both summands.
3) The resulting bit value can be transformed back to physical domain by using the same factor as the summands and the sum of the summand's offset as result offset.

4) Get the physical assignment (p_0, m_0, p_1 and m_1) of the calculated signal from the result's *factor* and *offset*. (see also Assignment 43).

Example 2: Calculation of the power of the accelerated mass *m* (Constant moments of inertia of rotated masses can be taken into account by increasing the mass *m* correspondingly).

Work *W* is defined by

$W(k) = ½ \cdot m \cdot v(k) \cdot v(k)$, where $v(k)$ is the speed signal in m/s²

Power *P* is defined by

$P(k) = (W(k) - W(k\text{-}1)) / (t(k) - t(k\text{-}1))$

The time period $t(k) - t(k\text{-}1)$ is constant for every k, since all calculations are done with a constant loop rate of *T*. Combining all constants together yields:

$P(k) = C \cdot (v(k) \cdot v(k) - v(k\text{-}1) \cdot v(k\text{-}1))$, with $C = m / (2 \cdot T)$

Using bit values for v(k) instead of physical values yields:

$P(k) = C \cdot ((factor \cdot b(k) + offset)^2 - (factor \cdot b(k\text{-}1) + offset)^2)$

To simplify this equation we need to make the assumption that offset equals zero. This assumption yields:

$P(k) = C \cdot factor \cdot factor \cdot (b(k) \cdot b(k) - b(k\text{-}1) \cdot b(k\text{-}1))$

We have to take into account that the result of the product $b(k) \cdot b(k)$ does not produce a 16-bit overflow. Therefore, we divide the product by the constant *D* (e. g. *D* = 65535). Finally, the equation reads:

$P(k) = G \cdot (b(k) \cdot b(k) / D - b(k\text{-}1) \cdot b(k\text{-}1) / D)$
where $G = C \cdot factor \cdot factor \cdot D = (m \cdot factor \cdot factor \cdot D) / (2 \cdot T)$

The final equation can be calculated with the following operations where the bit value of the speed is given by input signal *v*:

1) v^2 = Op07($S_1 = v$, $S_2 = v$, $C_1 = D$)
2) $v(k\text{-}1)^2$ = Op19($S_1 = v^2$)
3) diff = Op06($S_1 = v^2$, $S_2 = v(k\text{-}1)^2$, $C_1 = 0$, $C_2 = 1$)
4) P = Op04(S_1 = diff, $C_1 = G$, $C_2 = 0$)

2.5.13.3 Comment Tab

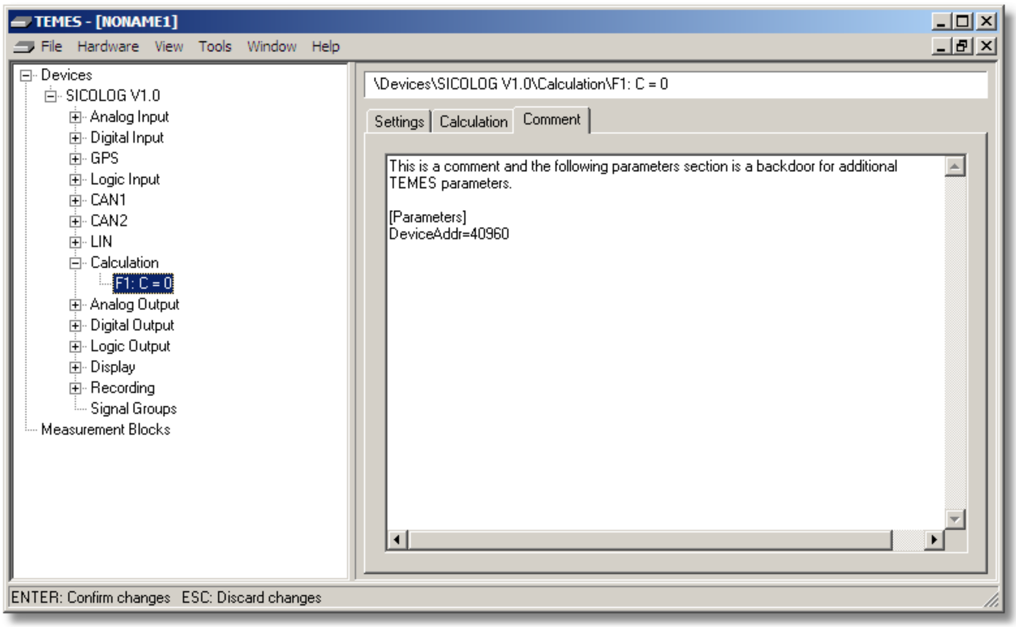


Figure 2-35: Parameter tree node Calculation (*Comment* tab).

The multi-line comment is not only a comment for the calculated signal. This comment is also used as a backdoor for additional TEMES parametes. See also the byte code 141 usage as an example.

2.5.14 Parameter Tree Node: Analog Output

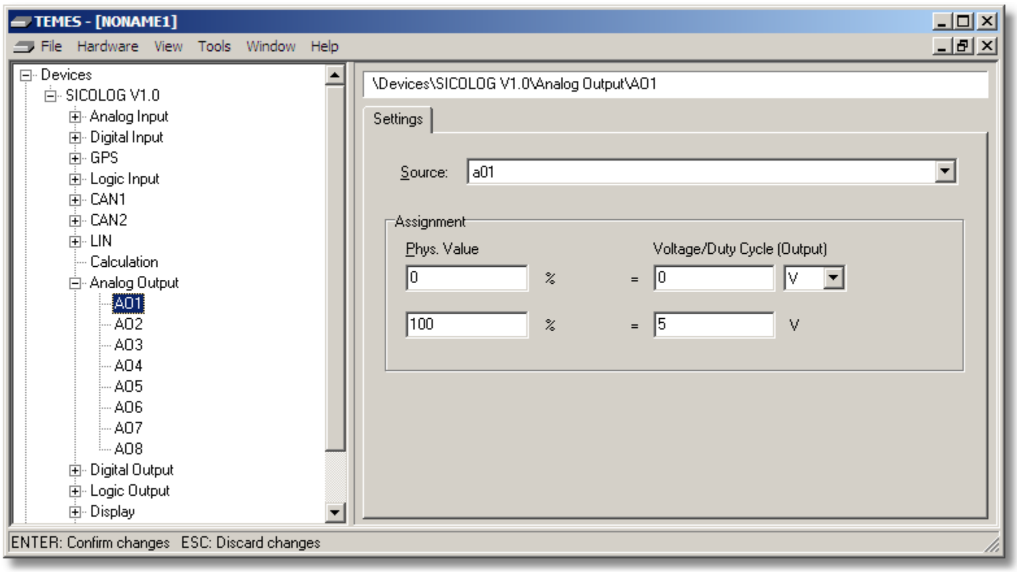


Figure 2-36: Parameter tree node Analog Out.

Source: List of all available input signals.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

2.5.15 Parameter Tree Node: Digital Output

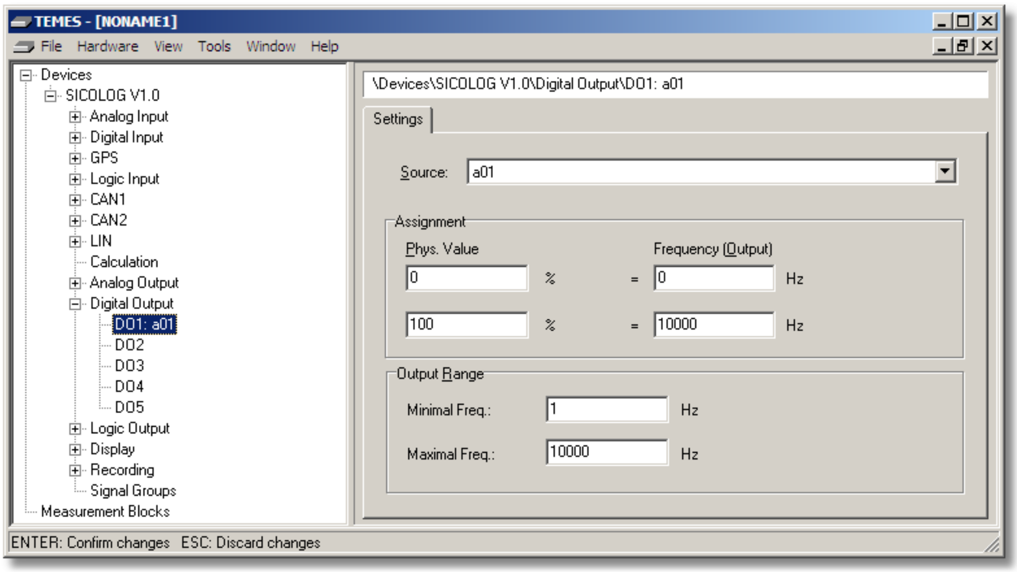


Figure 2-37: Parameter tree node Digital Output.

Source: List of all available input signals.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

Output Range: Definition of the minimal and maximal frequency. Note that 0 Hz for the minimal frequency means that the minimal frequency is not important, and that the timer clock setting is optimized for the maximal frequency only. Otherwise it is optimized for the minimal frequency. For SICOLOG/SICO3/USBDL1 it is usually required to enter a minimal frequency greater than 0 Hz. From TEMES V1.0.79 onwards, the SICOLOG/SICO3/USBDL1 default output frequency range is 1 Hz to 1000 Hz.

A digital port can be used in both directions.
For SICO2/DL16CAN: If a digital channel is used to output a signal, only switch states 47 can be selected as an input signal for the same channel.
For SICOLOG/SICO3/USBDL1: if a digital channel is used to output a signal, then it cannot be used as an input channel at the same time.

2.5.15.1 PWM Output (SICOLOG/SICO3/USBDL1)

The SICOLOG/SICO3/USBDL1 outputs a PWM signal if the minimal frequency (≠ 0 Hz) equals the maximal frequency (= PWM switching frequency) of the output range. Note that the unit of the frequency column in the assignment is in this case no longer in Hz, but in % of the duty cycle, with the minimal duty cycle in the first line, and the maximal duty cycle in the second line.

2.5.15.2 Logical Signal Output (SICOLOG/SICO3/USBDL1)

The SICOLOG/SICO3/USBDL1 outputs a logical signal if the minimal frequency is greater than the maximal frequency of the output range.

2.5.16 Parameter Tree Node: Logic Output

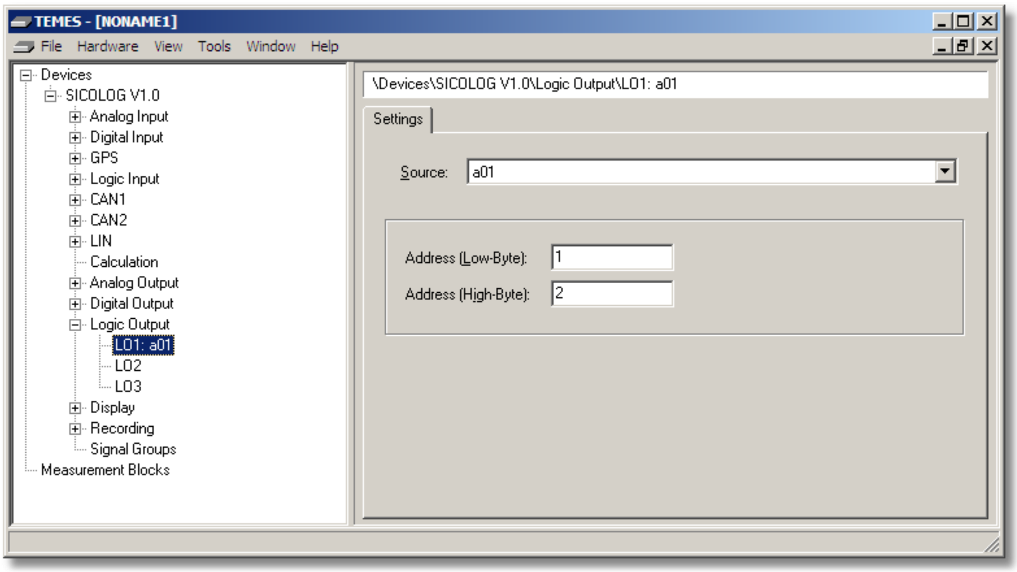


Figure 2-38: Parameter tree node Logic Output.

Source: List of all available input signals.

Address (Low-Byte): Definition of a_l.

Address (High-Byte): Definition of a_h.

The values to be output are a_h = HighByte(*source*) and a_l = LowByte(*source*).

2.5.17 Parameter Tree Node: Port Output

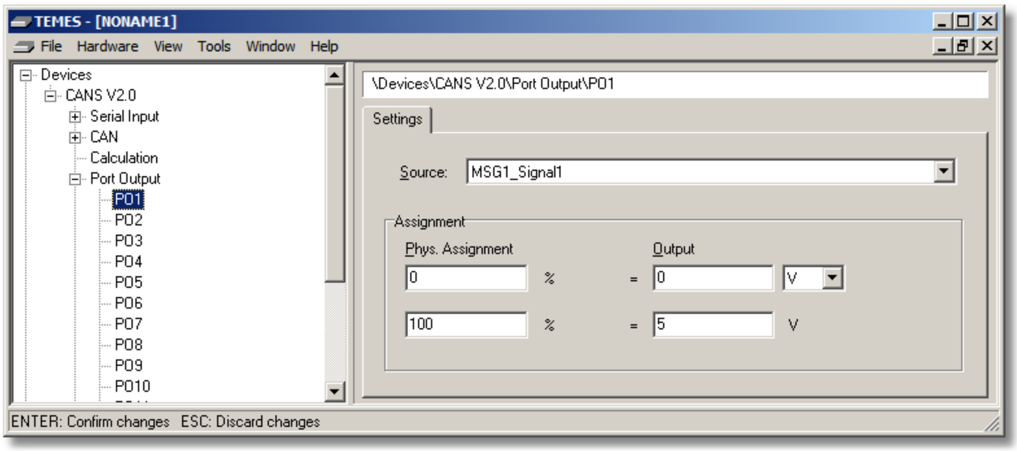


Figure 2-39: Parameter tree node Port Output.

Source: List of all available input signals.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

2.5.18 Parameter Tree Node: LED Output

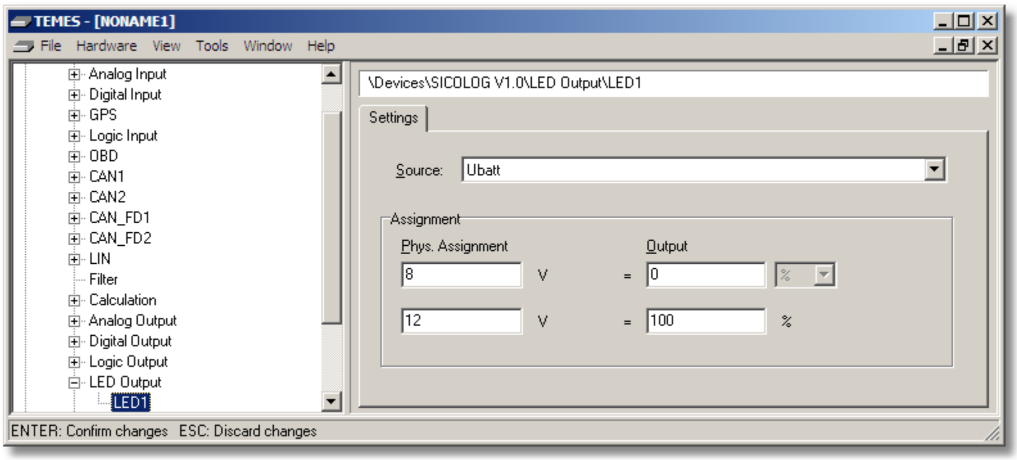


Figure 2-39: Parameter tree node LED Output.

Source: List of all available input signals.

Assignment: Definition of the linear relationship. See also Analog Input, Settings tab 43.

2.5.19 Parameter Tree Node: Clutch Events

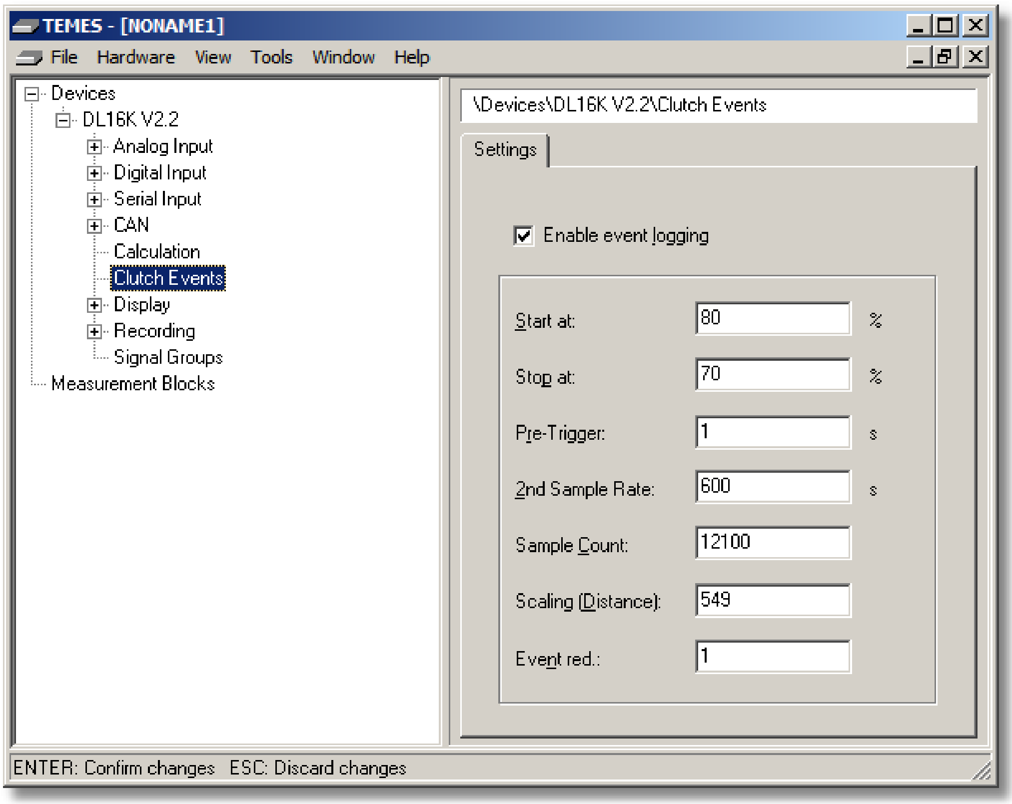


Figure 2-40: Parameter tree node Clutch Events.

The parameters within this form are the same as used within the MS-DOS program for the DL16K. Details about them can be looked up in the online help of the MS-DOS program.

Note, that clutch events are no longer supported within the German GUI from TEMES V1.0.66 onwards. Note also, that clutch events are no longer supported in TEMES V1.0.89 onwards due to resource limitations.

2.5.20 Parameter Tree Node: Display

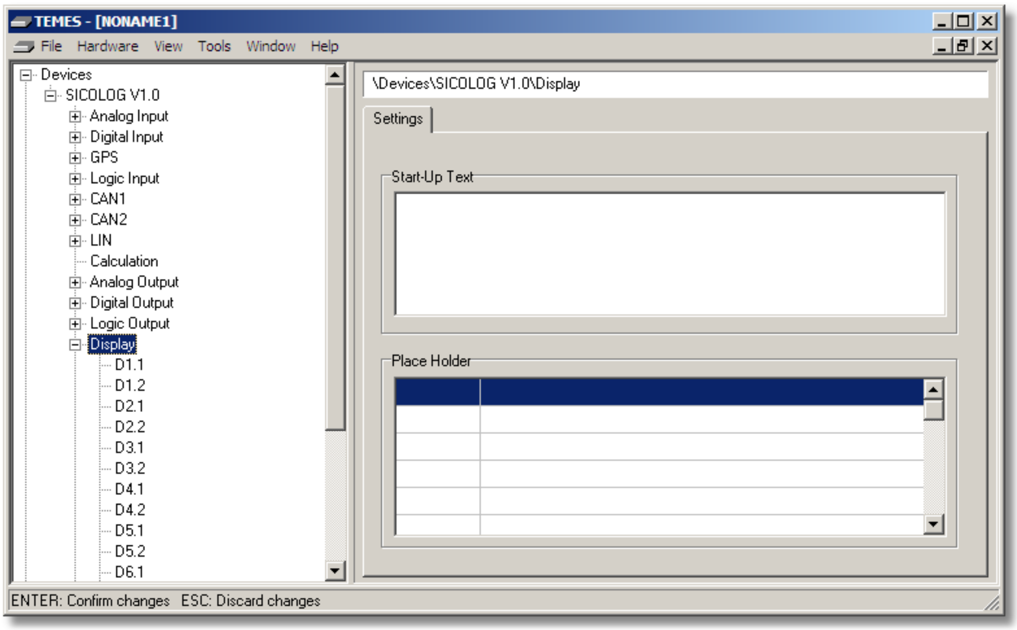


Figure 2-41: Parameter tree node Display.

2.5.20.1 SICOLOG/USBDL1, SICO3, CFDL1

The **Start-Up Text** is displayed just after a parameter set is loaded, or when the page slider is used, and stays visible for about two seconds.

(USBDL1: The start-up text will not be displayed but it can be accessed via a PC.)

2.5.20.2 SICO2, DL16CAN

The **Start-Up Text** is displayed just after a reset of the device and stays visible for about two seconds. The start-up text can be used to identify the device and the current parameter set. If the start-up text is undefined, a default start-up text will be used instead which shows the name of the parameter file, the device number and the date of setup. Following placeholders can be used:

Placeholder	**Description**
@@	Single @
@n	Device's internal serial number
@N	User defined device number
@D	Setup day
@M	Setup month
@Y	Setup year

2.5.20.3 Parameter Tree Node: Display Item

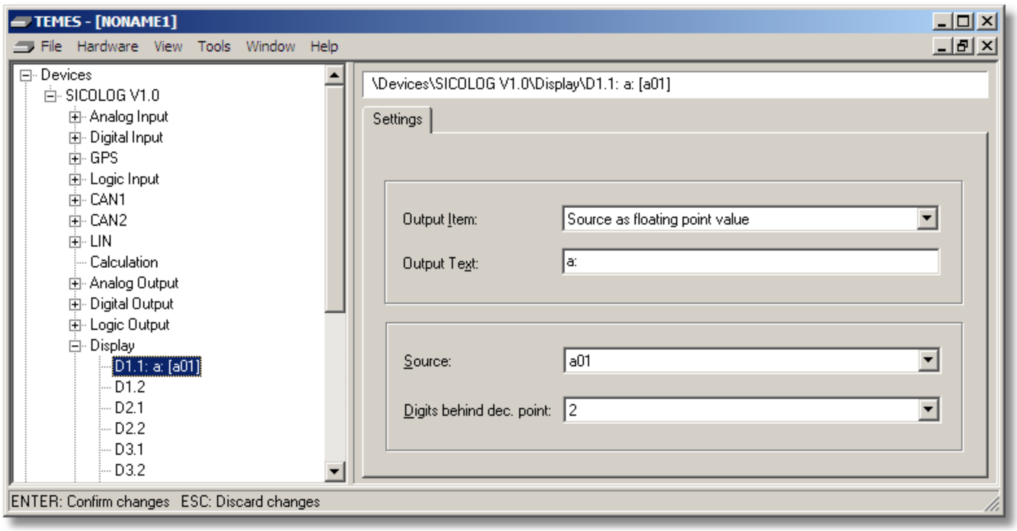


Figure 2-42: Parameter tree node Display Item.

The display items will be shown just after the start-up text disappears (or directly after a parameter set is loaded). Display items are arranged in rows and columns. The position of a display item is given by its name (D*r*.*c*, with row *r* and column *c*). The rows beyond 4 (row ≥ 5) are shown on the second page starting with the first row. The second page can be made visible by switching the page slider from 1 to 2.

Output Item: The following types are supported.

Text – The specified **Output Text** will be displayed.

Source as floating point value – The first two characters of the specified output text will be displayed. Followed by the source's physical value in floating point format.
For SICOLOG/SICO3/USBDL1 V1.0.18 or better, the output text can be shorter/longer, and optionally be formatted with a ##[#...] placeholder or with a **[*...] placeholder for the floating point value, where each placeholder character represents one character of the floating point value.

Source as hex value – The first two characters of the specified output text will be displayed. Followed by the hexadecimal bit value of the source.

Source as binary value (Low-Byte) – The first two characters of the specified output text will be displayed. Followed by the binary low byte of the source's bit value.

Source as binary value (High-Byte) – The first two characters of the specified output text will be displayed. Followed by the binary high byte of the low word of the source's bit value.

CPU-Load – Displays the load of the CPU in percent. This value is temporarily insignificant for the SICOLOG/USBDL1 during USB stick I/O accesses (it will temporarily show a value of up to 100 %).

Sample rate – Displays the sample rate 40.

Memory usage – Displays the memory usage.

Memory block – Displays the memory block.
Date – Displays the date.
Time – Displays the time.
Bar (left) – Displays the source as bar (starting from left)
Bar (middle) – Displays the source as bar (starting from the middle)
Bar (right) – Displays the source as bar (starting from right)

Note that not all output items are necessarily implemented by the correspon-ding device.

Source: List of all available input signals.

Digits behind dec. point: Number of decimals.

2.5.21 Parameter Tree Node: Recording Memory

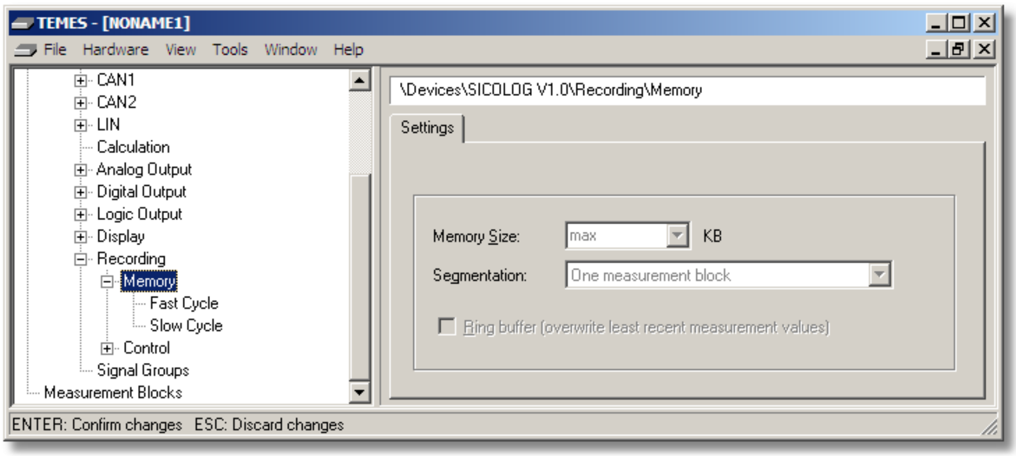


Figure 2-43: Parameter tree node Recording Measurement.

Memory Size: Desired maximal size of memory for one measurement block.

Segmentation: If enabled, this setting splits the memory in equally sized blocks.

Ring buffer: If this checkbox is checked, the oldest samples will be overwritten by the recent ones. If this checkbox is not checked, the measurement stops after the record memory is completely used.

2.5.21.1 Parameter Tree Node: Recording Memory Fast/Slow Cycle

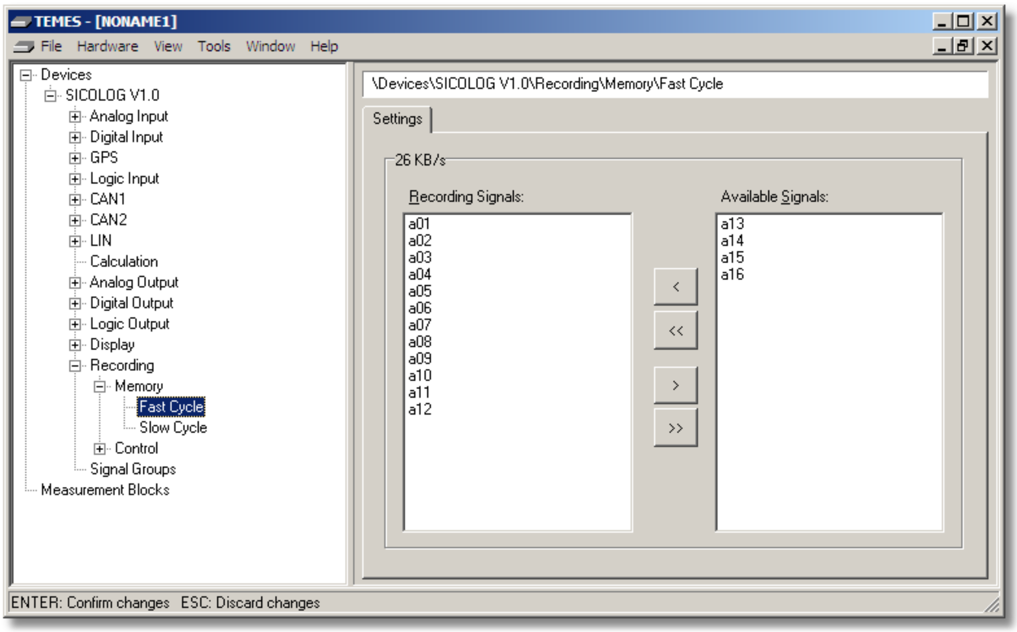


Figure 2-44: Parameter tree node Recording Memory Fast/Slow Cylce.

Select the signals to be recorded into the left list box. The writing speed (of both the fast and the slow recording cycle) is displayed at the top of the list boxes for the SICOLOG/USBDL1.

2.5.22 Parameter Tree Node: Recording Control

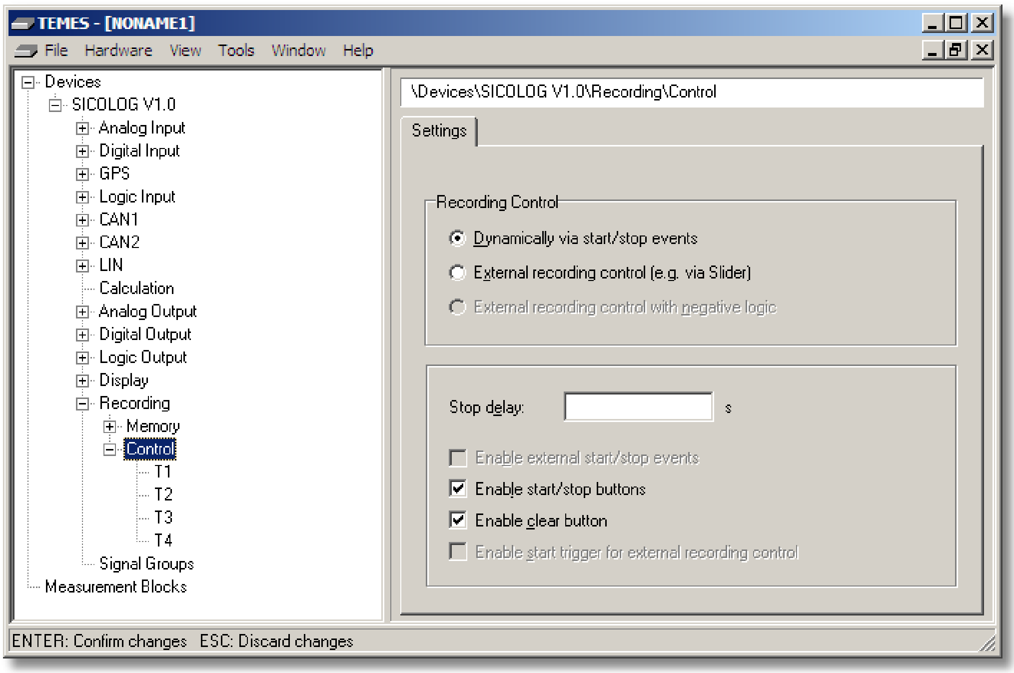


Figure 2-45: Parameter tree node Recording Control.

The **recording control** defines the recording mode: Either **dynamically via start/stop events**, **external recording control**, or **external recording control**

with negative logic. Start/stop events are either the start/stop buttons or a trigger event[80]. The external recording control is usually triggered by hardware via clamp 15 (DIN 72552).

The **stop delay** is the time duration when the logger keeps recording after a stop event.

The **start/stop buttons** can be disabled by unchecking the corresponding checkbox.

The **clear button** can be disabled by unchecking the corresponding checkbox.

TEMES nomenclature	SICOLOG, USBDL1, CFDL1	DL16CAN
Start button	Rec button (= B1 button)	Start button
Stop button	Stop button (= B2 button)	Stop button
Clear button	Button B1 and button B2 pressed together for at least 2 seconds until the LED L4 (or USER LED) is unlit again (with the REC slider being in off position).	Start and stop buttons pressed together for at least 2 seconds.
External recording control	POWER slider in off position (when clamp 15 is connected to the ON pins).	RUN-EXT-slider in EXT position.

2.5.22.1 Parameter Tree Node: Recording Control Trigger

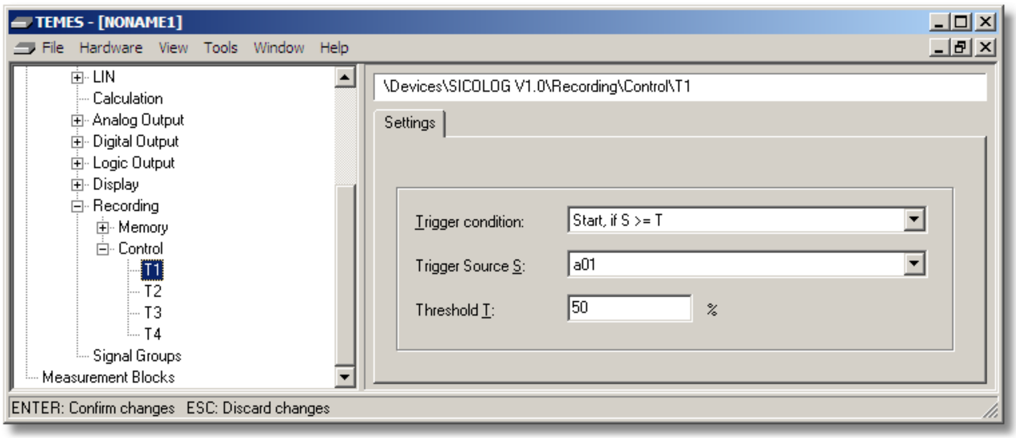


Figure: 2-46: Parameter tree node Recording Control Trigger.

Defines a start or stop condition. Note that a condition is only true for the moment when the condition becomes fulfilled from unfulfilled.

2.5.23 Parameter Tree Node: Signal Groups Item

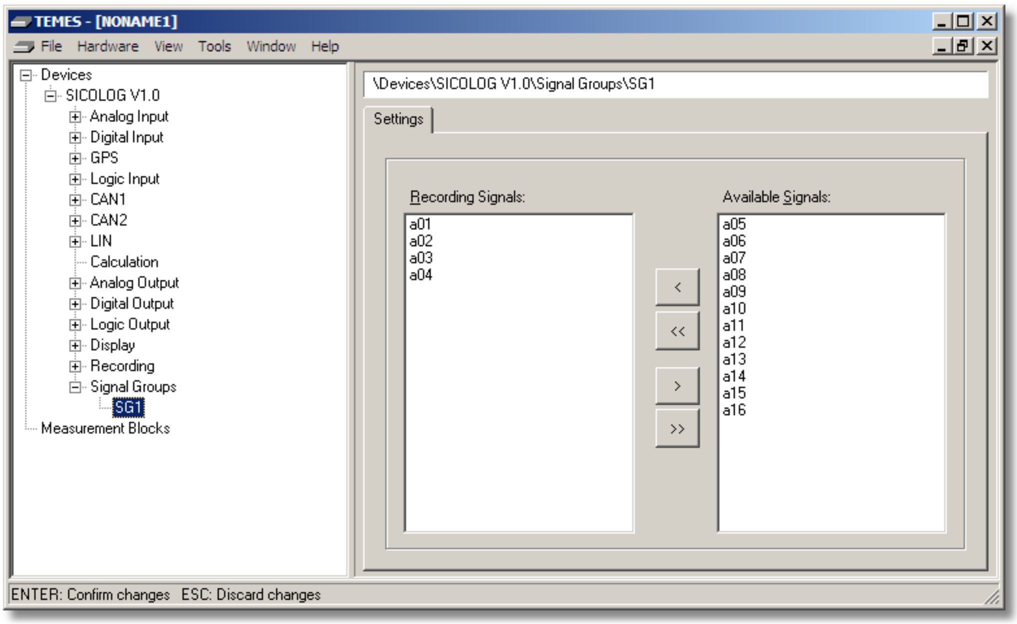


Figure 2-47: Parameter tree node: Signal Groups Item.

A signal group can be inserted by pressing the *Ins* key on the tree node *Signal Groups*. A signal group is useful to select those signals which should be visible in an online recording, or to select also those signals which should be marked as used (e. g. for only displaying a signal via external display but not using the signal otherwise).

2.5.24 Parameter Tree Node: Measurement Block

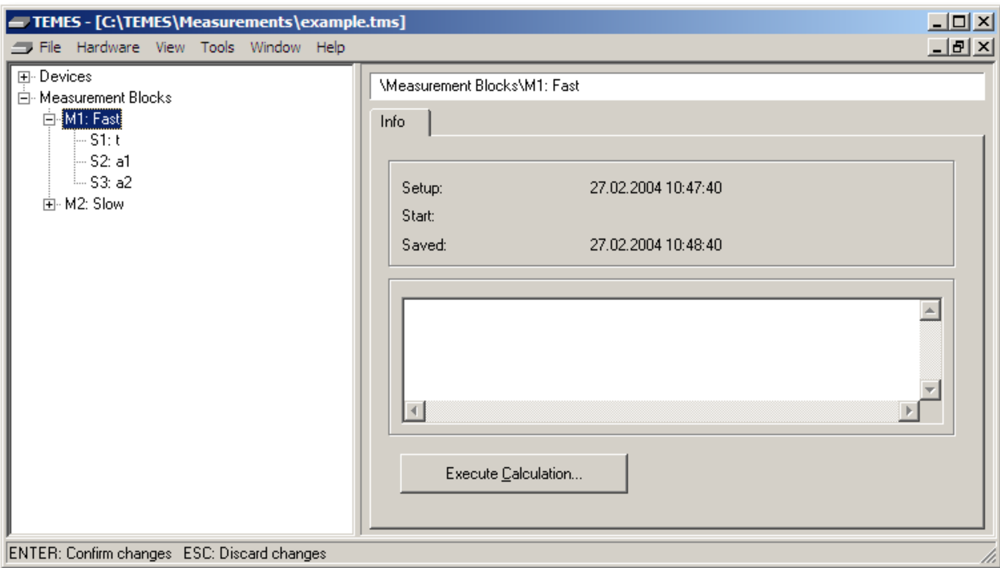


Figure 2-49: Parameter tree node Measurement Block.

Execute Calculation: Select a calculation definition file (*.CDF*) and execute offline calculations 171.

2.6 Tellert Device Manager

The *Tellert Device Manager* is an OLE2 automation server. Applications like *TEMES* and *Tellert Script Files* use this automation server for communicating with devices from Rudy Tellert Elektronik.

2.6.1 Device Manager Tab

Applications which use the *Tellert Device Manager* have only access to those devices which are registered with the *Tellert Device Manager* by the user. For doing so, the user selects the corresponding port and assigns the correct device type. Usually, the user needs only to press the *Update port* button. In this case, the Device Manager tries to find the correct device type and transfer settings on its own.

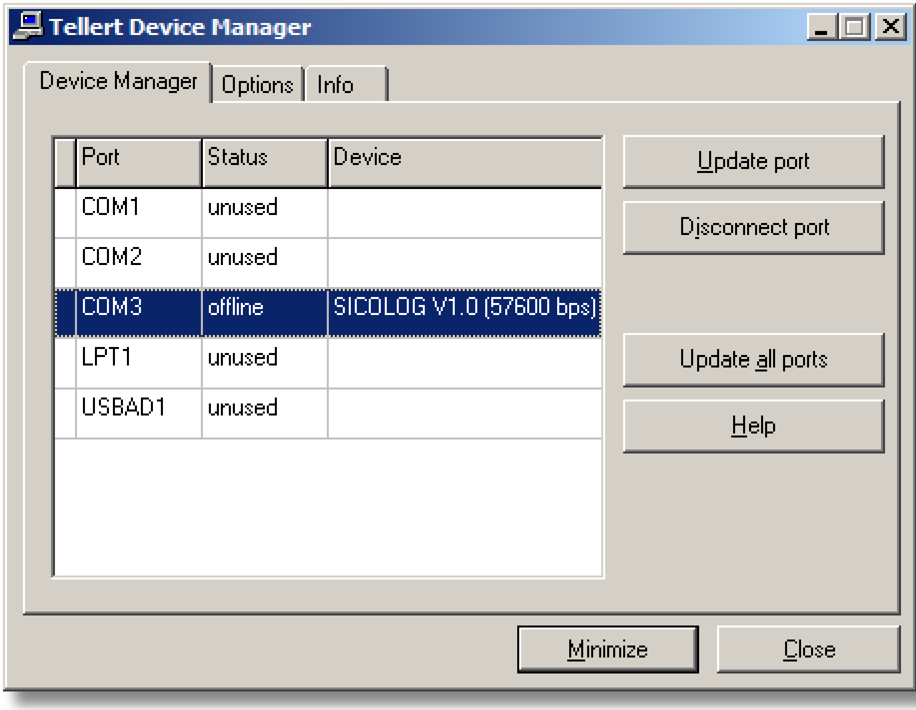


Figure 2-50: Tellert Device Manager (*Device Manager* tab).

First column: A star symbol within the first (unnamed) column of the list box indicates an active communication.

Port: Name of the PC port (COM*x*, LPT*x* or USBAD1).
in use – The port cannot be used by the Device Manager, since another application has already taken control over the port. The port must first be released by

the controlling application before it is again accessible by the Device Manager. Note: If the port is used by a DOS program, it is not enough to exit the DOS program. It is also necessary to close the command line window in which the DOS program did run.

unused – No device is registered with this port.

offline – A device is registered with this port, but there is currently no communication between device and any Device Manager client applications.

online – An application is communicating with the corresponding device. Note: If a communicating application is closed with the Windows Task Manager, the Tellert Device Manager will not be notified that it lost a client. In this case, the status of the port remains *online,* even the application has already been terminated. To reuse the port again, it is necessary to select the corresponding port and to press the *Disconnect port* button.

Update port: First, the status of the port will be updated. Then, the Device Manager tries to find the correct device type and transfer settings.

Disconnect port: The device type will be removed for the selected port.

Update all ports: All ports will be updated. Note: It is not possible to use ports which use the same IRQ number (e. g. COM1 and COM3, COM2 and COM4) at the same time. In this case, select the corresponding port, and press the *Update port* button instead.

2.6.2 Options Tab

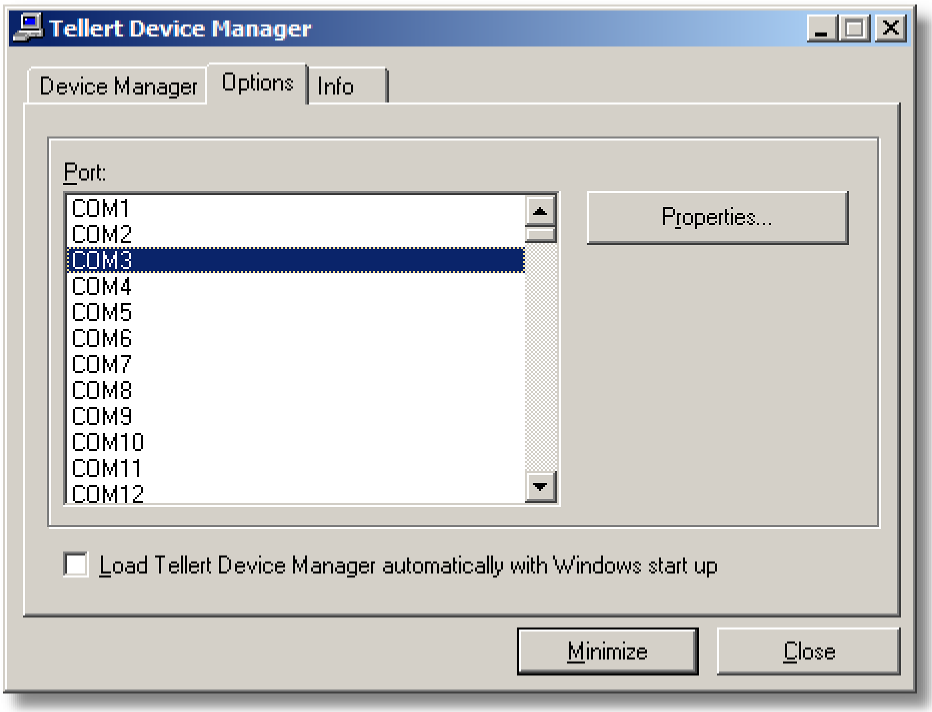


Figure 2-51: Tellert Device Manager (*Options* tab).

Properties: Edit properties of the selected **Port**.

Load Tellert Device Manager automatically with Windows start up: This setting determines whether the Tellert Device Manager is automatically started with Windows. The auto-start has the advantage that Tellert Device Manager client applications start up faster. On the other hand, the Tellert Device Manager allocates system resources like memory even if no application uses the Device Manager (this may reduce the performance for other applications).

2.6.2.1 Port Properties (General Tab)

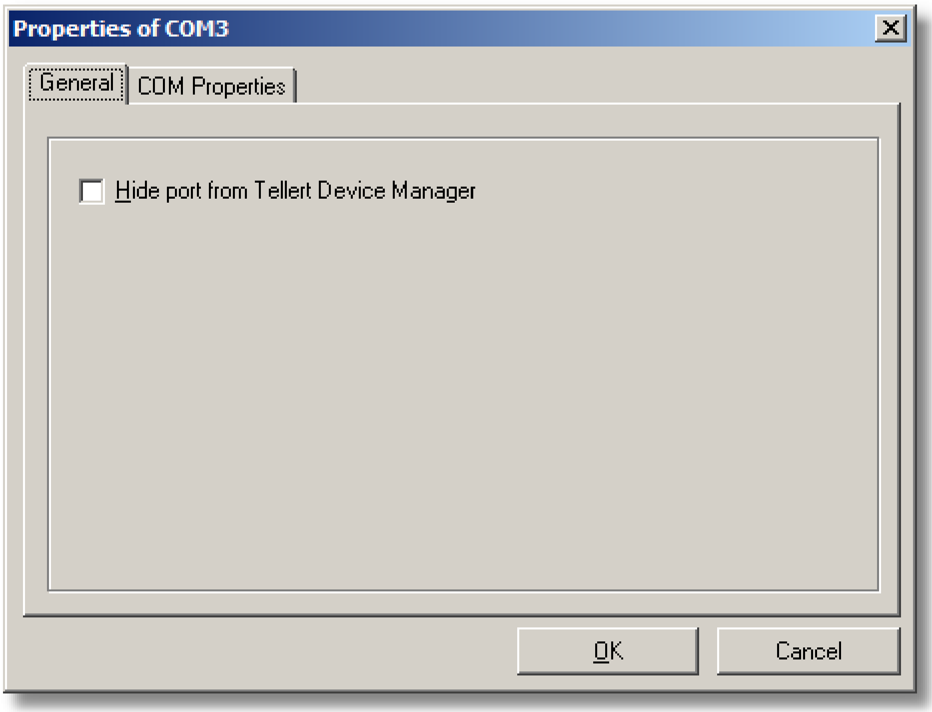


Figure 2-52: Port properties (*General* tab).

Hide port from Tellert Device Manager: If a port is not hidden from the Tellert Device Manager, the Tellert Device Manager determines the availability of a port by opening the port via the Windows File System.

2.6.2.2 Port Properties (COM Properties Tab)

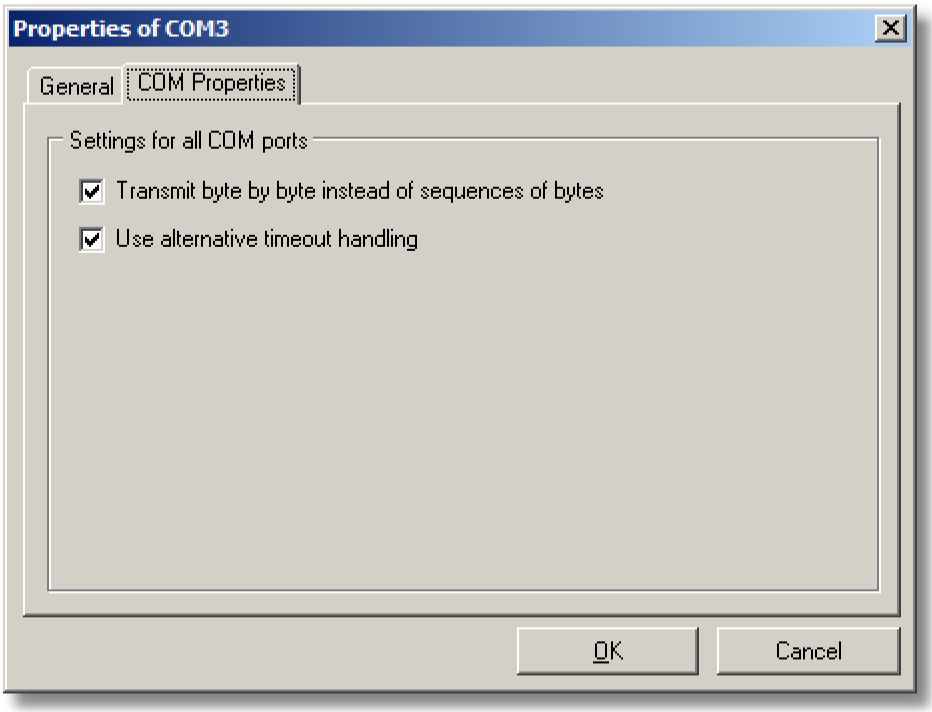


Figure 2-53: Port properties (*COM Properties* tab).

The two checkboxes should be checked – especially if an USBSER is used.

Transmit byte by byte instead of sequences of bytes: If this checkbox is unchecked, a sequence of bytes is send directly to the device. Regardless whether the device can cope with this sequence or not.

Use alternative timeout handling: If this checkbox is unchecked, then the timeout is done by the Windows API. Otherwise, the timeout is disabled for the Windows API, and the timing is done by the Device Manager instead.

2.6.2.3 Port Properties (LPT Properties Tab)

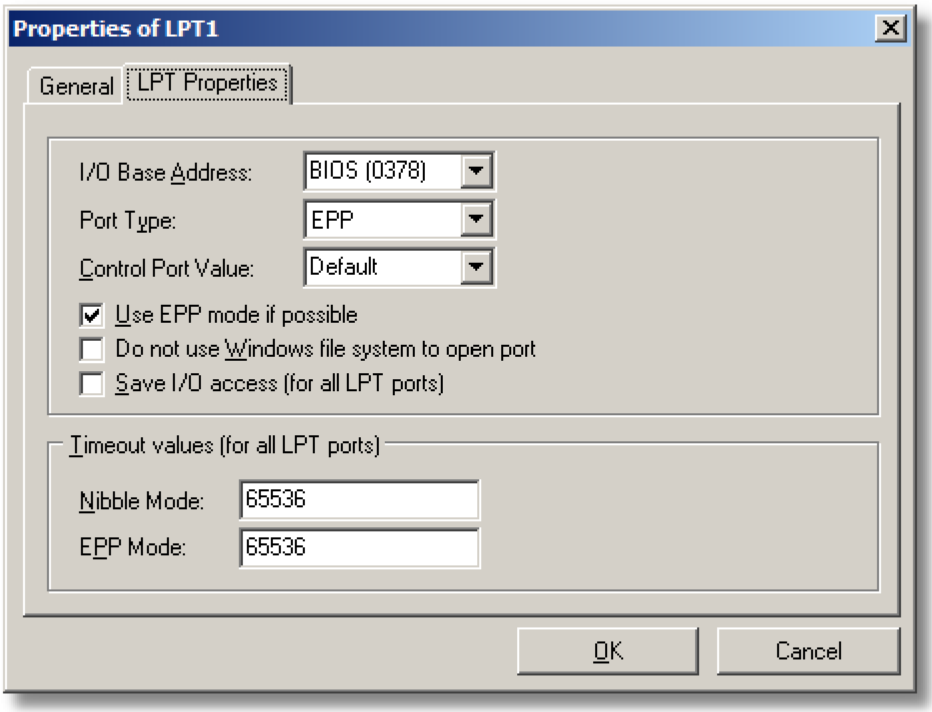


Figure 2-54: Port properties (*LPT Properties* tab).

The LPT properties are only supported by a 32-bit Windows Edition. A USBAD can be used instead an LPT port if the Windows Edition is 64-bit.

I/O Base Address: Hexadecimal base address of the interface unit. The entry *BIOS* is a placeholder for that base address which is given by the BIOS for the corresponding LPT port.

Port Type: Type of the LPT port (*Standard*, *EPP* or *ECP*).

Control Port Value: Hexadecimal initial value for the control register of the LPT port. If a bidirectional port is wanted to be run in uni-directional mode (e. g. for the parallel adapter PAD), the data pins must be used as outputs instead of inputs. This can be done by using the corresponding (manufacturer dependent) value for the control register (which is hexadecimal 04_{16} or 44_{16} in most cases).

Use EPP mode if possible: Uncheck this setting if an EPP transmission mode is not wanted. Note: Some ECP do not support the EPP mode.

Do not use Windows file system to open port: This item has always to be checked if the status of the corresponding port indicates that the port is *used* and if no application uses this port at the moment.

Save I/O access: A save I/O access is done indirectly via the HAL (= hardware abstraction layer) of Windows. A save I/O access takes more time than a direct one.

Timeout value: These values define the initial value of the timeout counters for data transmission via an LPT port. If the computer does not react instantly after updating a free LPT port, the corresponding timeout value may be reduced. On the other hand, if the data transmission fails often, the timeout value may be too low. In this case, a higher timeout value should be used.

2.7 CAN/CAN_FD/LIN Editor

2.7.1 CAN/CAN_FD/LIN Editor Main Menu

2.7.1.1 CAN/CAN_FD/LIN Editor Main Menu: File

2.7.1.1.1 New

Creates a new CAN/CAN_FD/LIN definition.

2.7.1.1.2 Open

Opens a CAN/CAN_FD/LIN definition.

2.7.1.1.3 Save

Saves a CAN/CAN_FD/LIN definition. The preferred path is the *CAN*-subfolder of the TEMES data folder which is usually "*documents\TEMES\CAN*".

2.7.1.1.4 Save as

Saves CAN/CAN_FD/LIN definition as a new file. The preferred path is the *CAN*-subfolder of the TEMES data folder which is usually "*documents\TEMES\CAN*".

2.7.1.1.5 Import

Imports a CAN/CAN_FD/LIN definition.

.DBC: Import the CAN/CAN_FD/LIN definition from a vector *DBC* file. Note that the CAN/CAN_FD/LIN baud rate is set to the fixed value of 500 kbps, and that the data rate is set to the fixed value of 2000 kbps for a CAN FD definition.

.TMS: Import the CAN/CAN_FD/LIN definition from a TEMES document.

.TXT: Import the CAN/LIN definition from an *ASCII* file. This is only a legacy file format. Note that the CAN/LIN baud rate is set to the fixed value of 500 kbps.

2.7.1.1.6 Append

Appends those CAN/CAN_FD/LIN messages from a CAN/CAN_FD/LIN definition file to the current CAN/CAN_FD/LIN definition which have a new CAN/CAN_FD/LIN id.

2.7.1.1.7 Export

Exports the current CAN/CAN_FD/LIN definition as a vector *DBC* file.

2.7.1.1.8 Exit

Quits the CAN/CAN_FD/LIN editor.

2.7.1.2 CAN/CAN_FD/LIN Editor Main Menu: Edit

2.7.1.2.1 Sort Messages

2.7.1.2.1.1 By Name

Sorts all CAN/CAN_FD/LIN messages by name.

2.7.1.2.1.2 By Identifier

Sorts all CAN/CAN_FD/LIN messages by identifier.

2.7.1.2.1.3 Reverse Order

Reverses the order of all CAN/CAN_FD/LIN messages.

2.7.1.2.2 Sort Signals

2.7.1.2.2.1 By Name

Sorts all CAN/CAN_FD/LIN signals by name.

2.7.1.2.2.2 By Position

Sorts all CAN/CAN_FD/LIN signals by the start bit.

2.7.1.2.2.3 Reverse Order

Reverses the order of all CAN/CAN_FD/LIN signals.

2.7.1.3 CAN/CAN_FD/LIN Editor Main Menu: Help

2.7.1.3.1 Info

Shows information about the CAN/CAN_FD/LIN editor.

2.7.2 Parameter tree node: CAN/CAN_FD/LIN

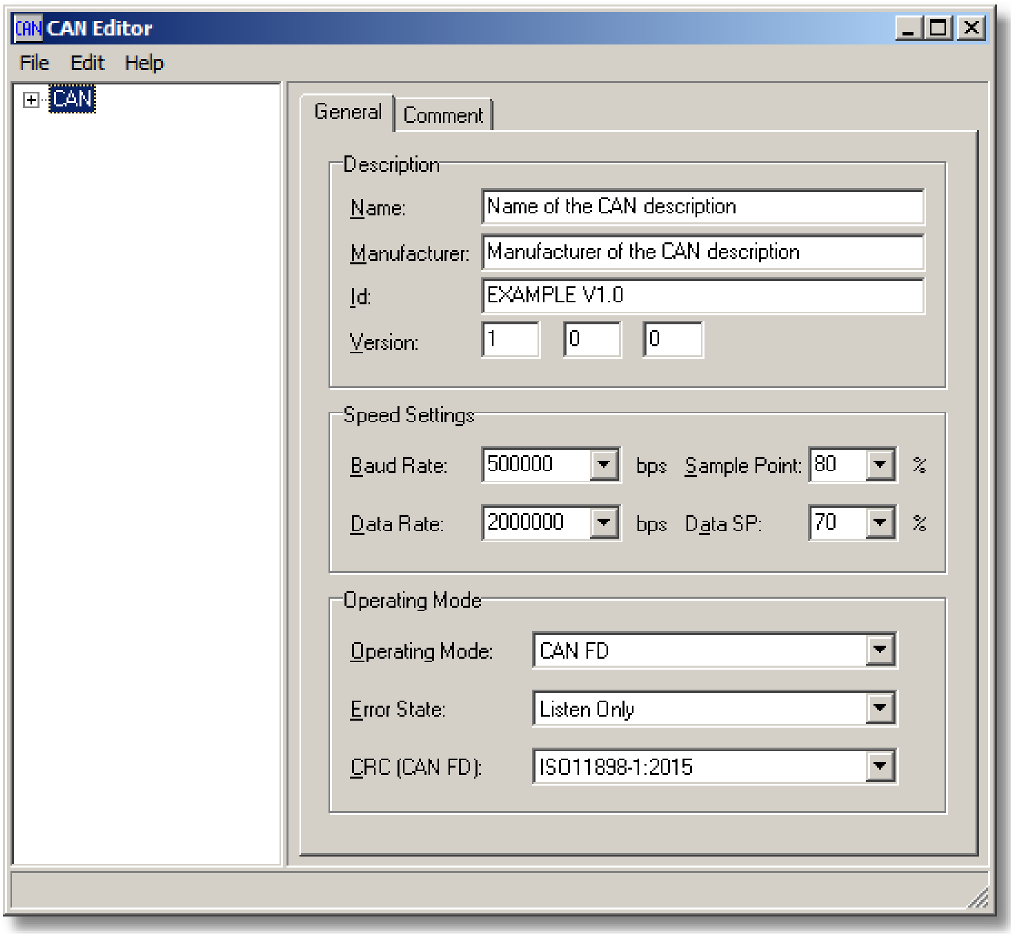


Figure 2-55: Parameter tree node CAN/CAN_FD/LIN.

The **description** is only significant for the user who fills in the **name** of the CAN/CAN_FD/LIN description, the **manufacturer** of the CAN/CAN_FD/LIN description, the **id** (= identification) of the CAN/CAN_FD/LIN description and the **version** of the CAN/CAN_FD/LIN description (consisting of major version, minor version and release).

The **baud rate** of the CAN/CAN_FD/LIN bus is specified in bits per seconds. Note that the baud rate is set to 500 kbps after a *DBC/TXT* file import.

The **sample point** gives a hint for the CAN/CAN FD bus. If left to 0, a standard sample point of about 80 % is assumed.

The **data rate** of the CAN FD bus is specified in bits per second. The data rate is set to 2000 kbps after a *DBC* file import of a CAN FD description. Note that this value is set to 0 for a CAN/LIN description, and set to a non-zero value for a CAN FD description.

The **data sample point** gives a hint for the CAN FD bus. If left to 0, a standard data sample point of about 70 % is assumed.

The **operating mode** can be set to either CAN FD, CAN 2.0 (= Classic CAN), Restricted (= transmission access only for correctly received frames), Listen only (= no transmission access).

The **error state** shows the operating mode when the CAN FD bus is in error state.

The **CRC (CAN FD)** defines the method how the CRC value of the CAN FD is calculated.

The *Comment* tab provides a multi-line comment for the user.

2.7.3 Parameter tree node: CAN/CAN_FD/LIN message

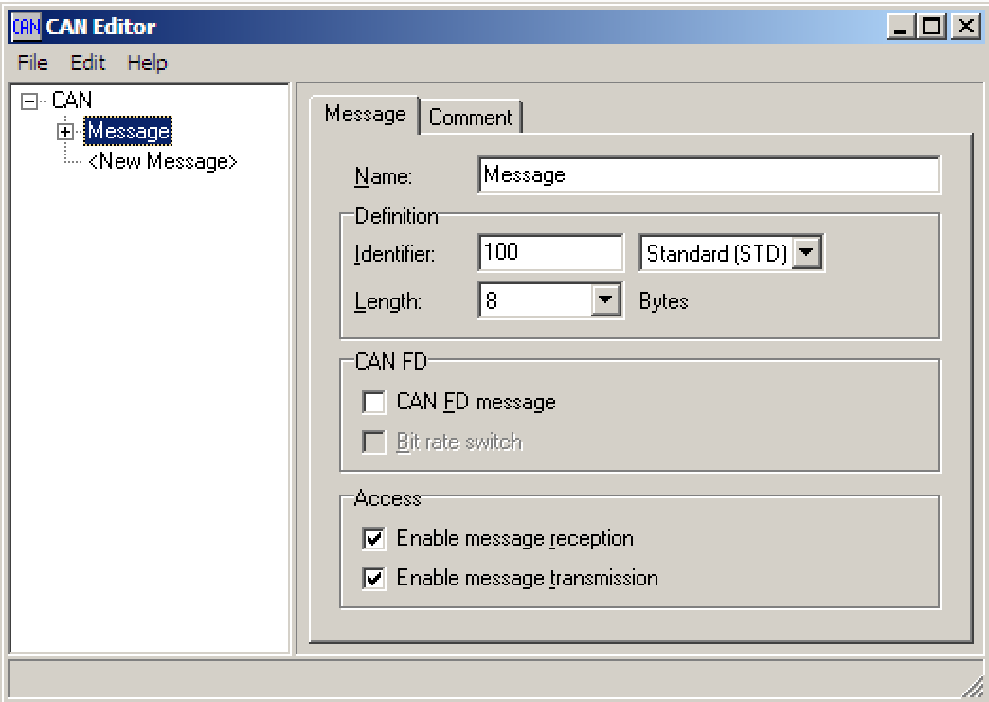


Figure 2-56: Parameter tree node CAN/CAN_FD/LIN message.

The parameter tree node *<New Message>* is a placeholder for a new CAN/CAN_FD/LIN message. As soon as a unique message name is given, the message is listed as a new CAN/CAN_FD/LIN message node in the tree structure.

Name: The name is a unique and non-empty name of the CAN/CAN_FD/LIN message.

Identifier: The CAN/CAN_FD/LIN identifier is a unique hexadecimal 11-bit or 29-bit value.

Identifier Type: *Standard* represents an 11-bit identifier, and *Extended* represents a 29-bit identifier.

Length: The CAN/CAN_FD/LIN length in bytes.

CAN FD message: Defines whether the message is a CAN FD message or not. Note that this check box is only enabled, if the data rate is non-zero.

Bit rate switch: Defines whether the data bits are transmitted at the data rate or at the arbitration bit rate.

The access controls whether this CAN/CAN_FD/LIN message can be added as a receive message to the TEMES parameter tree (or as a transmit message respectively).

The *Comment* tab provides a multi-line comment for the user.

The *Del* key can be used to remove an already defined CAN/CAN_FD/LIN message. The *Alt* + *Down/Up* key can be used to move the message in the tree structure.

2.7.4 Parameter tree node: CAN/CAN_FD/LIN signal

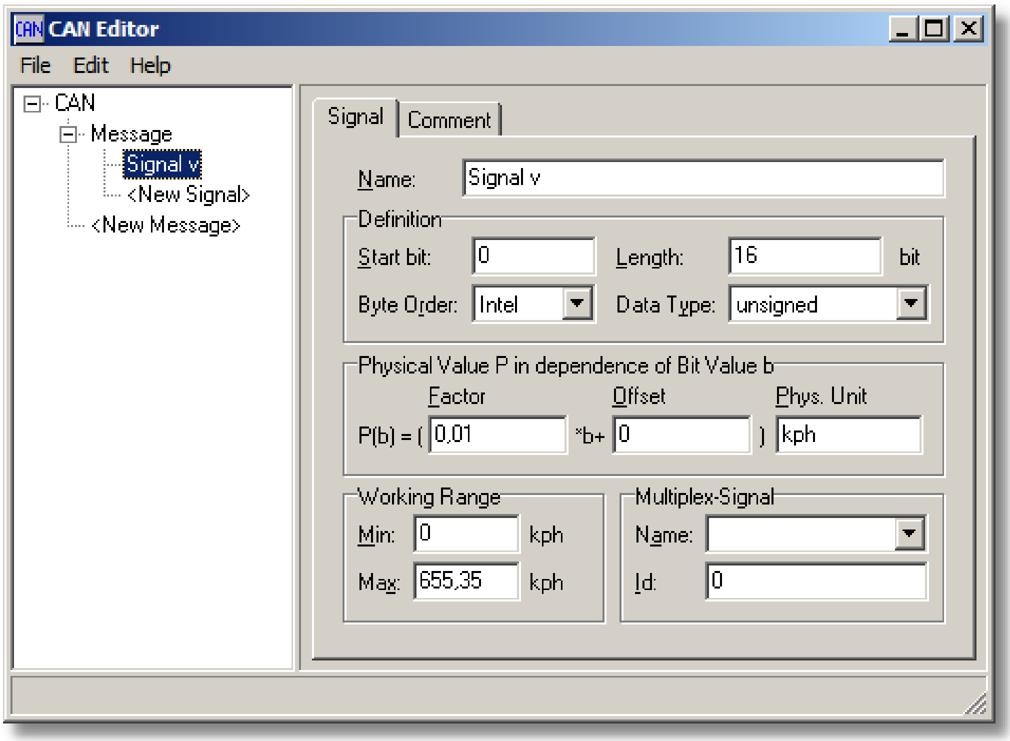


Figure 2-57: Parameter tree node CAN/CAN_FD/LIN signal.

The parameter tree node <*New Signal*> is a placeholder for a new CAN/CAN_FD/LIN signal. As soon as a unique signal name is given, the signal is listed as a new CAN/CAN_FD/LIN signal node in the tree structure.

Name: The name is a unique and non-empty name of the CAN/CAN_FD/LIN signal.

The **start bit**, bit **length**, **byte order** and **data type** define the bit layout/value of the CAN/CAN_FD/LIN signal (see also Bit Position of CAN Signals 165). The data type *float* is only supported by *LOG2TMS.EXE* at the moment.

The **factor**, **offset** and **phys. unit** describe the physical relationship of the the CAN/CAN_FD/LIN signal's bit value.

The **Min** and **Max** values are only for the TEMES working range (used by the chart axes and default value for the outputs).

A signal can be defined as a multiplexed signal. In this case the same CAN/CAN_FD/LIN bit positions are used by different multiplexed signals. The signals are then distinguished by a unique multiplex **Id** of a multiplex signal (= **Name**).

The *Comment* tab provides a multi-line comment for the user.

The *Del* key can be used to remove an already defined CAN/CAN_FD/LIN signal. The *Alt* + *Down*/*Up* key can be used to move the signal in the tree structure.

2.8 XCP/CCP Editor

Some tree nodes can be deleted with the *DEL* button.

The SICOLOG implementation of the XCP/CCP protocol uses polling of the CAN frames. Following points must be satisfied to achieve a similar DAQ list performance as with CAN interrupts:
- Only one ODT per DAQ list
- Own CAN identifiers for DAQ lists in case of more than one DAQ list per ECU

2.8.1 XCP/CCP Editor Main Menu

2.8.1.1 XCP Editor Main Menu: File

2.8.1.1.1 New

Creates a new XCP/CCP definition.

2.8.1.1.2 Open

Opens a XCP/CCP definition.

2.8.1.1.3 Save

Saves a XCP/CCP definition. The preferred path is the *CAN*-subfolder of the TEMES data folder which is usually "*documents\TEMES\CAN*".

2.8.1.1.4 Save as

Saves a XCP/CCP definition. The preferred path is the *CAN*-subfolder of the TEMES data folder which is usually "*documents\TEMES\CAN*".

2.8.1.1.5 Exit

Quits the XCP/CCP editor.

2.8.1.2 XCP Editor Main Menu: Edit

This menu item is reserved for future use.

2.8.1.3 XCP Editor Main Menu: Help

2.8.1.3.1 Info

Shows information about the XCP/CCP editor.

2.8.2 Parameter Tree Node: CAN

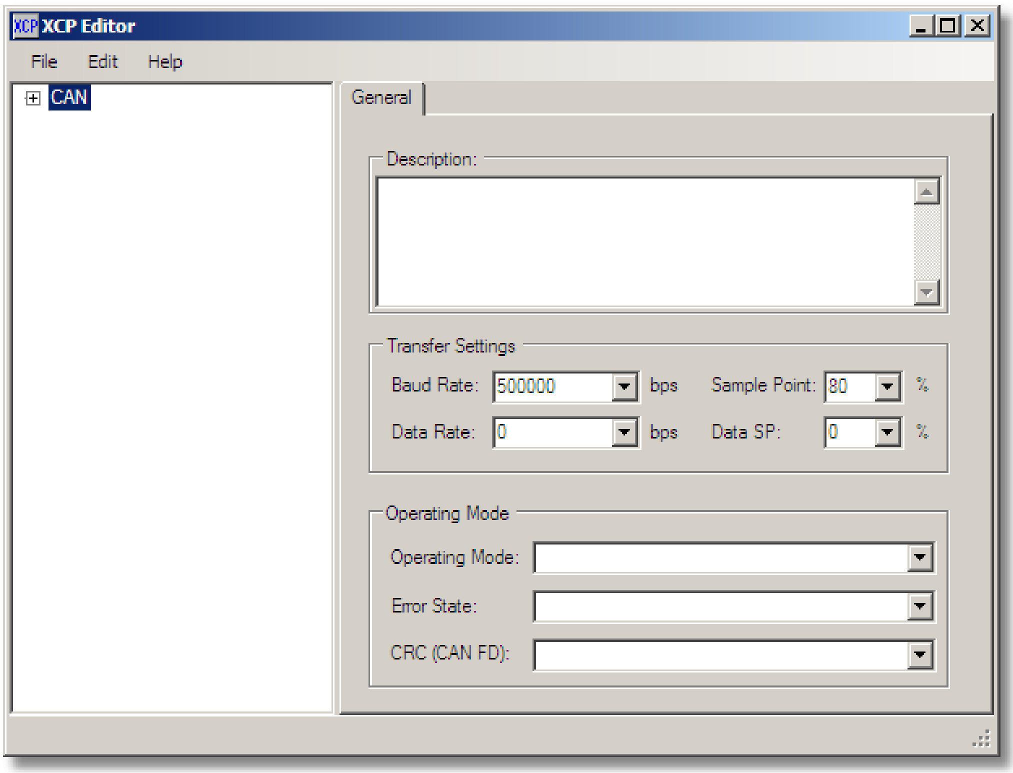


Figure 2-57a: Parameter tree node XCP/CCP CAN.

The **description** is only significant for the user and will be ignored by TEMES.

The **baud rate** of the CAN/CAN_FD bus is specified in bits per seconds.

The **sample point** gives a hint for the CAN/CAN FD bus. If left to 0, a standard sample point of about 80 % is assumed.

The **data rate** of the CAN FD bus is specified in bits per second. Note that this value is set to 0 for a CAN description, and set to a non-zero value for a CAN FD description.

The **data sample point** gives a hint for the CAN FD bus. If left to 0, a standard data sample point of about 70 % is assumed.

The **operating mode** can be set to either CAN FD, CAN 2.0 (= Classic CAN), Restricted (= transmission access only for correctly received frames), Listen only (= no transmission access).

The **error state** shows the operating mode when the CAN/CAN FD bus is in error state.

The **CRC (CAN FD)** defines the method how the CRC value of the CAN FD is calculated.

Note that the settings within this tab are ignored, if the corresponding CAN node has already a non-zero baud rate.

2.8.3 Parameter Tree Node: ECU

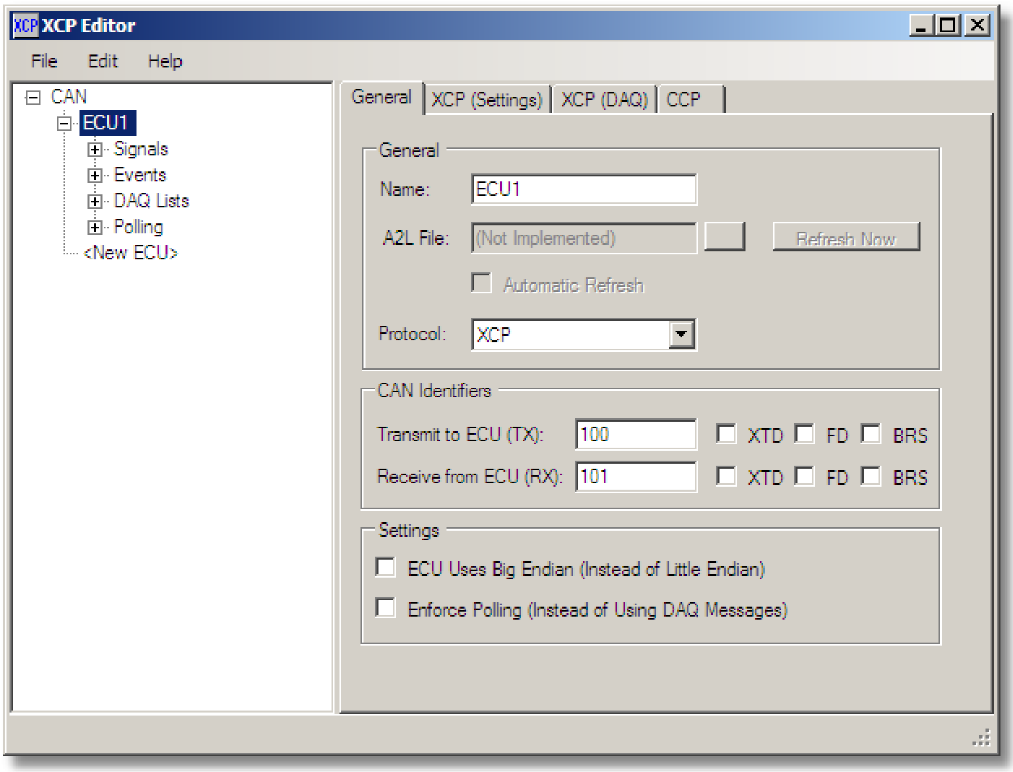


Figure 2-57b: Parameter tree node XCP/CCP editor: ECU.

Name: The name is a unique and non-empty name of the ECU.

Protocol: The protocol is either XCP or CCP.

CAN Identifiers: The identifiers are specified in hexadecimal. When the XTD checkbox is checked, the identifier is treated as an extended 29-bit-identifier, otherwise as a standard 11-bit-identifier. When the FD checkbox is checked, the identifier specifies a CAN-FD-message. When the BRS checkbox is checked, the identifier specifies a message with bit rate switching (which is only supported by CAN FD).

ECU uses big Endian: This checkbox is used to define the byte order which is used internally by the ECU. Note that this value is also provided by the ECU during the beginning of the XCP communication.

Enforce polling: This checkbox is used to use polling instead of DAQ messages. The polling method is simpler but also slower. If a definition with DAQ lists does not show the desired signal, please try polling instead, to see whether a valid signal is available.

2.8.3.1 XCP Settings Tab

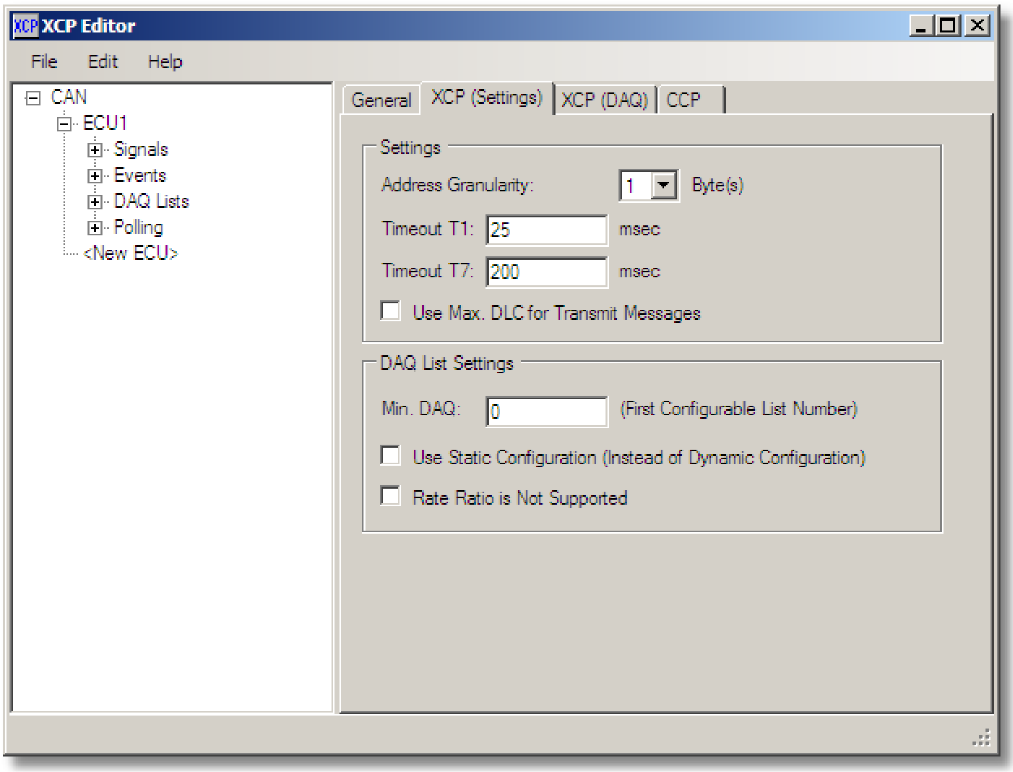


Figure 2-57c: Parameter tree node XCP/CCP editor: ECU (XCP Settings tab).

Address Granularity: This field specifies the address granularity which defines the address access type of the ECU: An address granularity of 1 means that there is no restriction in accessing the address. An address granularity of 2 means that addresses are only accessible in multiples of 2 bytes. And an address granularity of 4 means that addresses are only accessible in multiples of 4 bytes.

Timeout T1: This timeout value defines the general timeout.

Timeout T7: This timeout value is used in a further command attempt when the ECU is busy.

Use max. DLC for transmit messages: When checked, the transmit messages are extended to the max. possible field width with zero bytes. The max. possible field width is 8 bytes for a CAN message and 64 bytes for a CAN FD message.

Min. DAQ: This entry specifies the minimal list number for a configurable list. If the list numbers are greater or equal to min. DAQ than the ODT setup is always exchanged between XCP master (= data logger) and XCP slave (= ECU). Note that this value is usually provided by the ECU during the XCP communication.

Rate Ratio is not supported: This checkbox specifies whether the rate ratio (= sample rate prescaler) is supported or not. Note that this value is usually provided by the ECU during the XCP communication.

2.8.3.2 XCP DAQ Tab

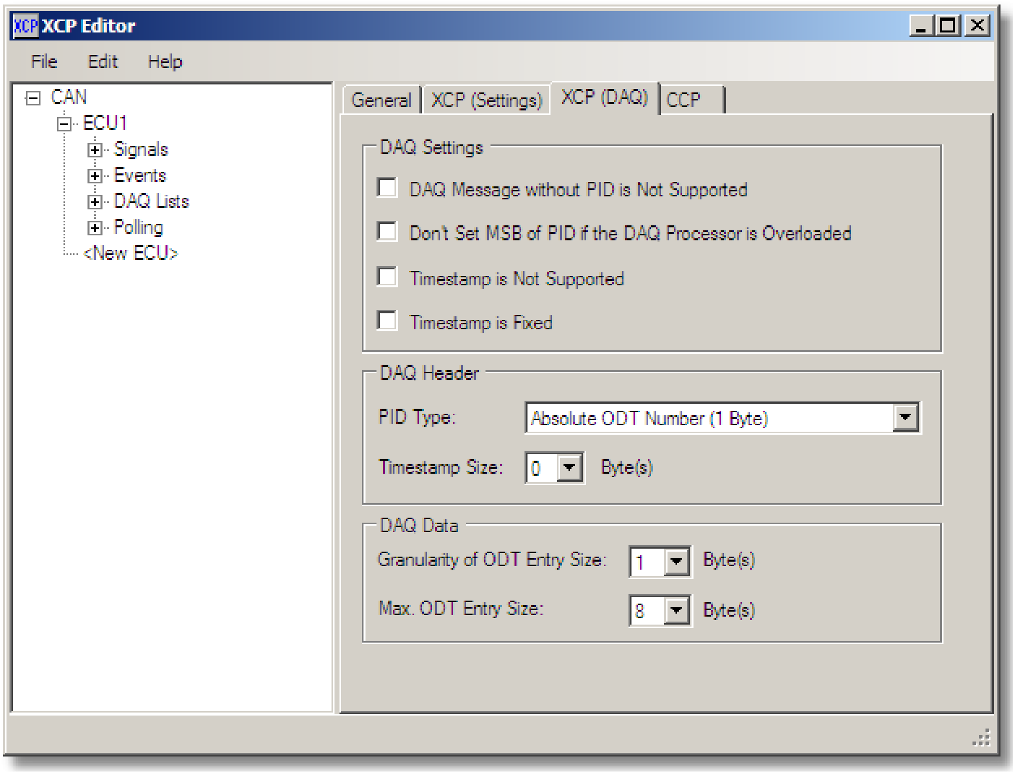


Figure 2-57d: Parameter tree node XCP/CCP editor: ECU (XCP DAQ tab).

Don't set MSB of PID: This checkbox specifies whether the MSB of the first byte of the PID is used to indicate a DAQ processor overload.

Granularity of ODT entry size: This value is almost like the address granularity. It refers to all ODT entries. Note that the granularity of the ODT entry size must be greater or equal to the address granularity.

Max. ODT entry size: This value is used to split one ODT entry into multiple smaller entries.

Note that the values of this tab are usually provided by the ECU during the XCP communication.

2.8.3.3 CCP Tab

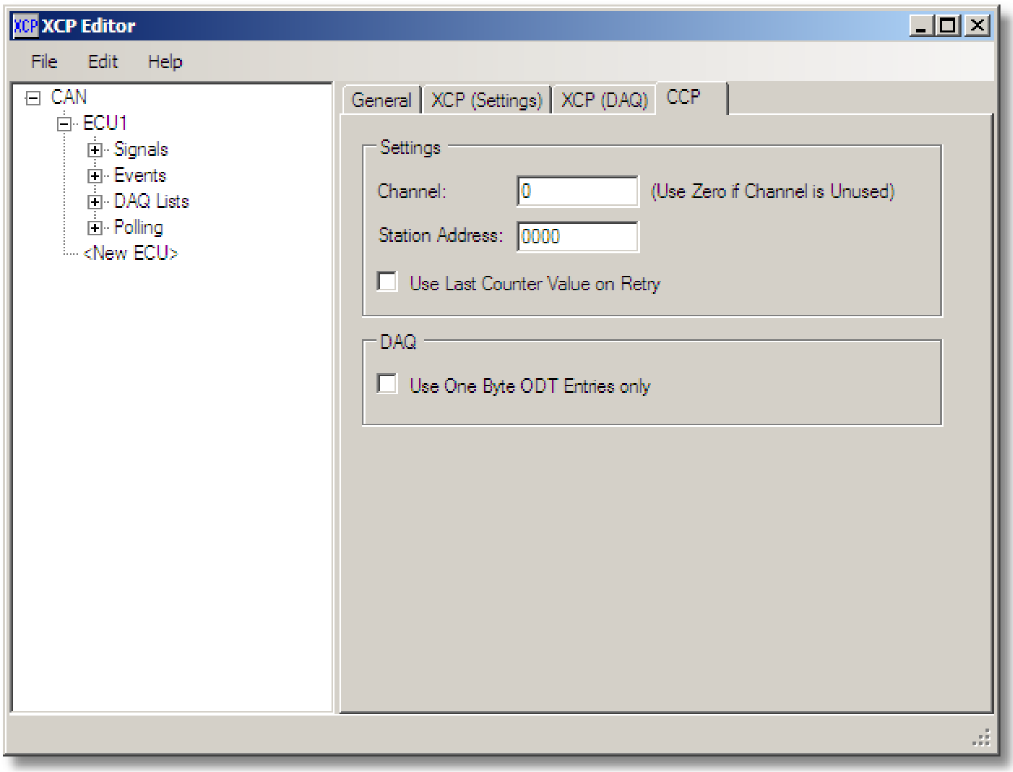


Figure 2-57e: Parameter tree node XCP/CCP editor: ECU (CCP tab).

Channel: This value is not used at the moment. It is reserved for CCP ECUs with common CAN identifiers. The CCP master must always use a (slow) CONNECT attempt when the next CCP ECU with common CAN identifiers is addressed.

Station Address: This field is a 16-bit hexadecimal value. It is used to distinguish between different ECUs in case of common CAN identifiers.

Use one byte ODT entries only: This entry specifies whether an ODT entry is split into multiple one-byte-entries.

2.8.4 Parameter Tree Node: Signal

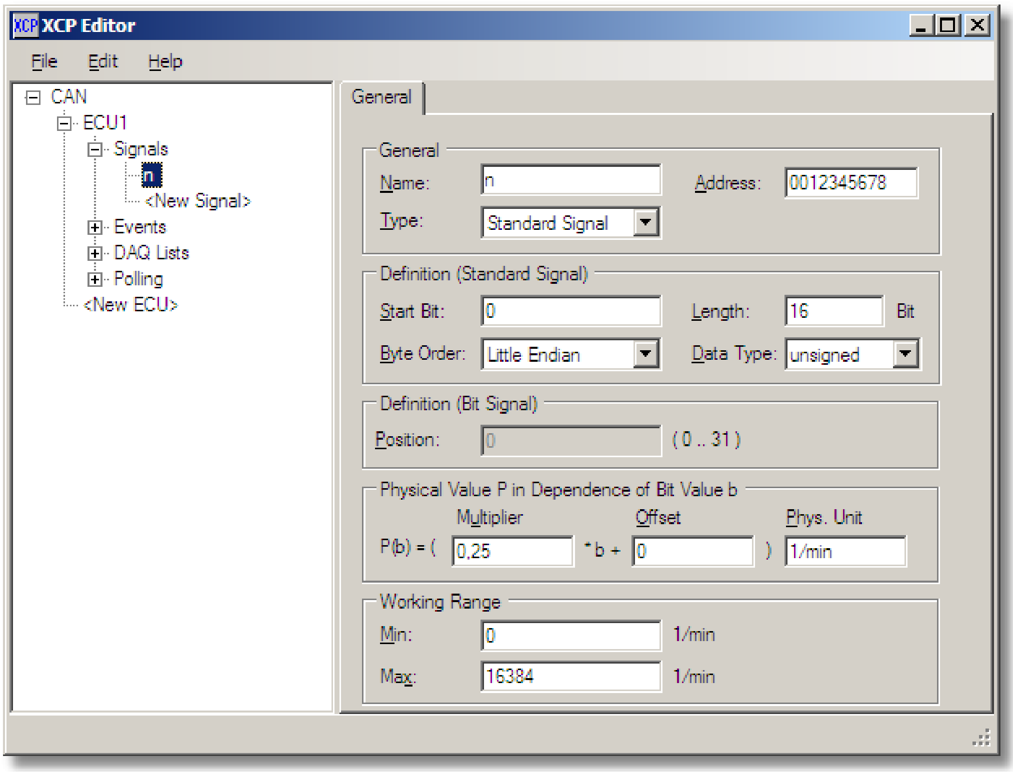


Figure 2-57f: Parameter tree node XCP/CCP editor: Signal.

Name: Unique and non-empty signal name.

Address: Hexadecimal 40-bit value with the first two digits as the segment, and the last eight digits as the 32-bit address.

Type: Either a standard signal or a bit signal. Bit signals are only supported by the XCP protocol.

Start Bit: Lowest bit position of the signal. This value is usually 0. The total number of signal bytes is the integer of (start bit + bit length + 7) / 8. Signals with Big Endian byte order use the last byte as start byte for the numbering of the bits (similar to CAN signals in Motorola 165 format).

Length: Bit length. Note that a value greater than 32 is not supported at the moment by the SICOLOG firmware.

Data Type: Note that *float* is unsupported at the moment by the SICOLOG firmware.

Position: Bit position (with using the ECU's byte order) between 0 and 31.

2.8.5 Parameter Tree Node: Event

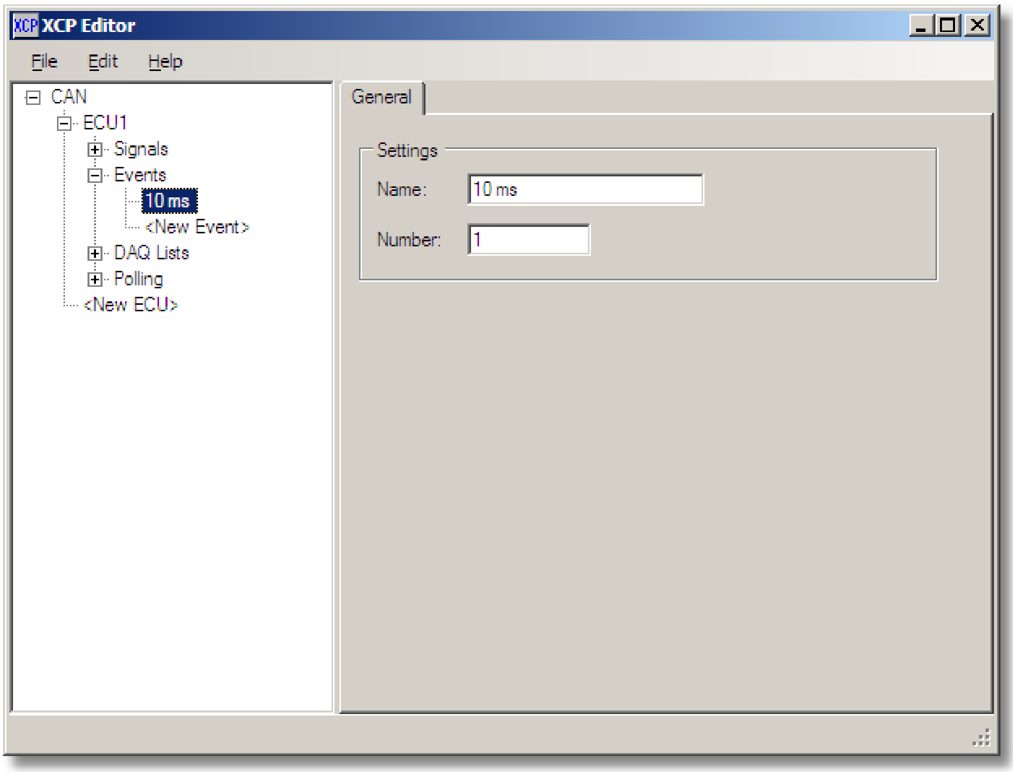


Figure 2-57g: Parameter tree node XCP/CCP editor: Event.

Name: Unique and non-empty name.

Number: Event number between 0 and 255.

2.8.6 Parameter Tree Node: DAQ List

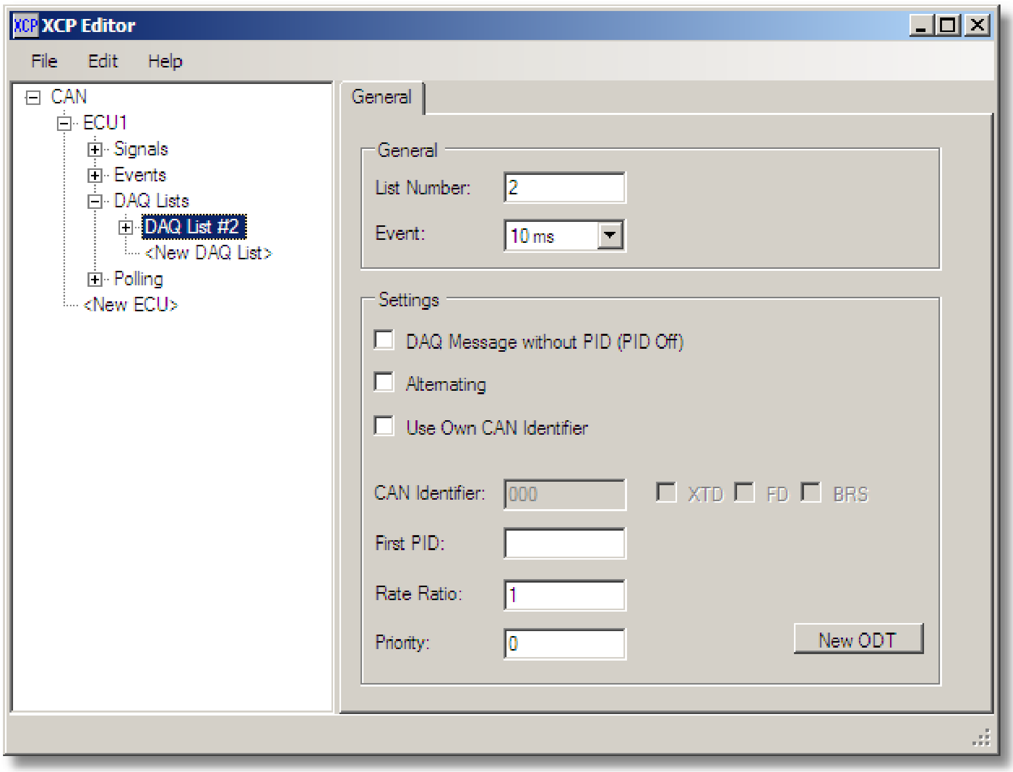


Figure 2-57h: Parameter tree node XCP/CCP editor: DAQ list.

List number: Unique list number. This value must be different for a specific ECU.

Event: Selection of the event name.

CAN Identifier: The identifier is specified in hexadecimal. When the XTD checkbox is checked, the identifier is treated as an extended 29-bit-identifier, otherwise as a standard 11-bit-identifier. When the FD checkbox is checked, the identifier specifies a CAN-FD-message. When the BRS checkbox is checked, the identifier specifies a message with bit rate switching (which is only supported by CAN FD).

First PID: This field may only be important for the XCP protocol with a one-byte PID type (= absolute ODT number). It is entered in decimal. Usually, the XCP protocol automatically detects the first PID.

Rate Ratio: This value is the transmission rate prescaler.

Priority: This value is the priority between 0 and 255. It is possible that an ECU refuses a priority greater than 0.

New ODT: A new ODT node is created for this DAQ list (by pressing the corresponding button).

2.8.7 Parameter Tree Node: Signal Reference (DAQ List)

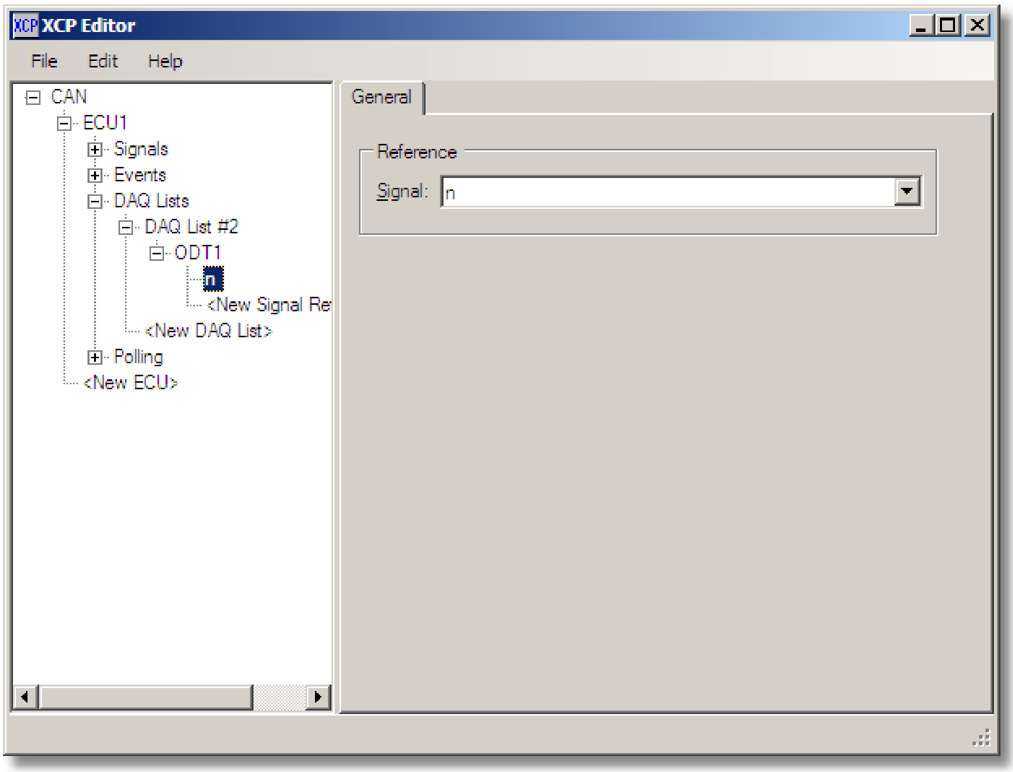


Figure 2-57i: Parameter tree node XCP/CCP editor: Signal reference (DAQ list).

A signal is only queried if there is a corresponding *signal reference* defined.

2.8.8 Parameter Tree Node: Signal Reference (Polling)

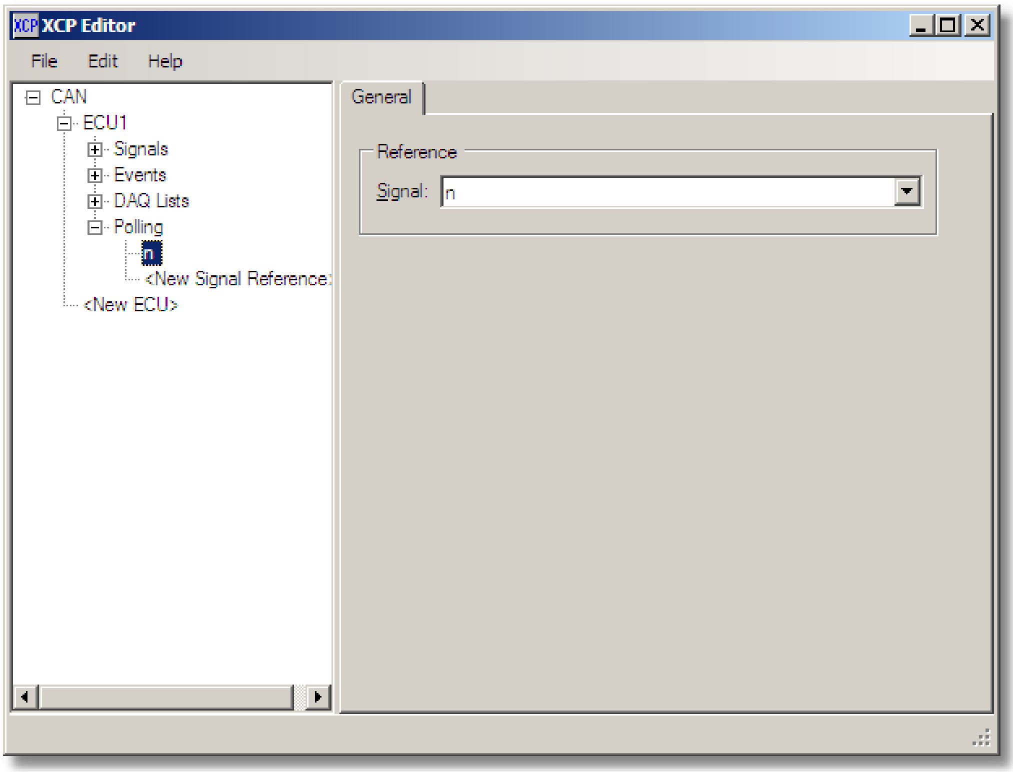


Figure 2-57j: Parameter tree node XCP/CCP editor: Signal reference (Polling).

Polling signal references are only meant as auxiliary method: Usually the signal references are specified inside DAQ lists. Then, they can be switched between DAQ access and polling access, either automatically or manually. If no DAQ list is specified, the signal references can also be specified here (but only for polling purpose).

2.9 Diagram View

2.9.1 Diagram's Main Menu

2.9.1.1 Diagram's Main Menu: Diagram

2.9.1.1.1 Preview

Opens a printer preview window. This window can also be used to export the chart as *.PDF* file or graphics file (like a *.PNG* file).

2.9.1.1.2 Print

Prints out the charts. This menu item can also be used to export the print to a *.PDF* file or a graphics file (like *.PNG* file).

2.9.1.1.3 Variables

Opens the variables editor39.

2.9.1.1.4 Overview Window

Opens a second window with the selected curve. The overview window can be handy when it displays the curve in map view (latitude over longitude). The print layout can also have both the overview window and the chart.

2.9.1.1.5 Export visible range

.TMS: The *export visible range* clips the curves to the visible range (especially when displayed over time) and saves only the clipped curves to a file.

.KML: The longitude and latitude of the visible range is exported as *.KML* file (e. g. to be displayed in Google Earth).

.GPX: The longitude and latitude of the visible range is exported as *.GPX* file.

2.9.1.1.6 Close File

Closes the chart window and also the TEMES document.

2.9.1.1.7 Close

Closes the chart window.

2.9.1.2 Diagram's Main Menu: Edit

2.9.1.2.1 Signal

2.9.1.2.1.1 Properties

Opens the signal properties dialog box.

2.9.1.2.2 Curve

2.9.1.2.2.1 Play Sound

Plays a curve segment as sound (defined by the difference cursor).

If the signal is either a frequency signal, or it has the unit *kph*, *km/h*, *rpm*, *1/min*, *m/s*, *mph*, *mps*, *Hz*, or *kHz*, then the *wave settings for frequency signal*107 dialog box is opened, otherwise it is interpreted as *PCM* data.

2.9.1.2.2.2 Play Sound again

Plays the sound again.

2.9.1.2.2.3 Stop paying Sound

Stops playing the sound.

2.9.1.2.2.4 Save Sound

Saves the sound as *.WAV* file.

2.9.1.2.2.5 Repeat Sound

When this menu item is checked, the sound is repeated after it reaches the end of the curve segment.

2.9.1.2.2.6 Properties

Opens the curve properties dialog box.

2.9.1.2.3 Axis

2.9.1.2.3.1 Properties

Opens the axis properties dialog box.

2.9.1.2.4 Diagram

Opens the diagram dialog box.

2.9.1.2.5 Layout

Opens the layout editor. TEMES uses only the layout file "*default.lbl*" which is stored in the TEMES data folder (which is usually "*documents\TEMES\default.lbl*").

2.9.1.3 Diagram's Main Menu: View

2.9.1.3.1 Default View

Shows the initial view of the window.

2.9.1.3.2 Save

Shows the Save View 112 form.

2.9.1.4 Diagram's Main Menu: Cursor

2.9.1.4.1 Cursor mode

Selects/Deselects the cursor mode.

2.9.1.4.2 Statistics

Shows the statistics 113 for the current point of the selected signal in cursor mode.

2.9.1.5 Diagram's Main Menu: Analysis

2.9.1.5.1 Acceleration Measurement

Sets the markers for an acceleration measurement.

2.9.1.5.2 Draft Measurement

Sets the markers for a draft measurement.

2.9.1.5.3 Brake Measurement

Sets the markers for a brake measurement.

2.9.1.5.4 incl. Details

2.9.1.5.4.1 Acceleration Measurement (incl. interim values)

Sets the markers for an acceleration measurement incl. interim values.

2.9.1.5.4.2 Draft Measurement (incl. interim values)

Sets the markers for a draft measurement incl. interim values.

2.9.1.5.5 Clear Markers

Clears the markers.

2.9.1.5.6 Settings

This menu item allows to edit the *Analysis* menu item. Note that write access is needed for the files "*C:\Program Files (x64)\Tellert\TEMES 1.0\ChartMenu_eng.ini*" and "*C:\Program Files (x64)\Tellert\TEMES 1.0\ChartMenu_de.ini*".

2.9.1.6 Diagram's Main Menu: Recording

2.9.1.6.1 Online

Sets the device's status either online or offline.

2.9.1.6.2 Start

Starts a measurement.

2.9.1.6.3 Stop

Stops a measurement.

2.9.1.6.4 Save/Discard

Shows the save or discard measurement 115 form.

2.9.1.6.5 Settings

Shows the recording settings 114 form.

2.9.1.7 Diagram's Main Menu: Help

2.9.1.7.1 Temes help

Opens the TEMES help.

2.9.2 Dialog Boxes (Diagram)

2.9.2.1 Wave Settings Form

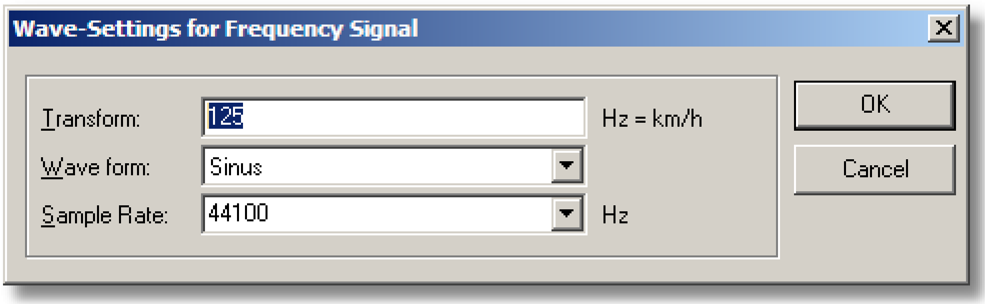


Figure 2-58: Wave Settings Form.

Requires the difference cursor and that the selected signal is either a frequency signal, or it has the unit *kph*, *km/h*, *rpm*, *1/min*, *m/s*, *mph*, *mps*, *Hz*, or *kHz*.

Transform: The transform defines a *factor* for translating the signal into the wave form.

Wave form: A wave form can be chosen from the drop-down list. Possible built-in wave forms are *Sinus*, *Square*, *Sawtooth*, or *Triangle*. Other wave forms can be selected by defining a *.WF* file in the "*Calc*" subfolder of the TEMES data folder (which is usually "*documents\TEMES\Calc*"). A *.WF* file is a text file which consists of the header *[Waveform]* followed by the monotonically increasing support points (*t;y*) with normalized time $t \in [0; 2]$ and normalized values $y \in [-1; 1]$.

v2_90.wf:
[Waveform]

0;0
0.0625;0.8
0.125;1
0.1875;0.8
0.3125;-0.8
0.375;-1
0.4375;-0.8
0.5;0
0.75;0
0.8125;0.8
0.875;1
0.9375;0.8
1.0625;-0.8
1.125;-1
1.1875;-0.8
1.25;0
2;0

Sample Rate: The sample rate is the number of generated samples per second.

2.9.2.2 Properties Form (Signal tab)

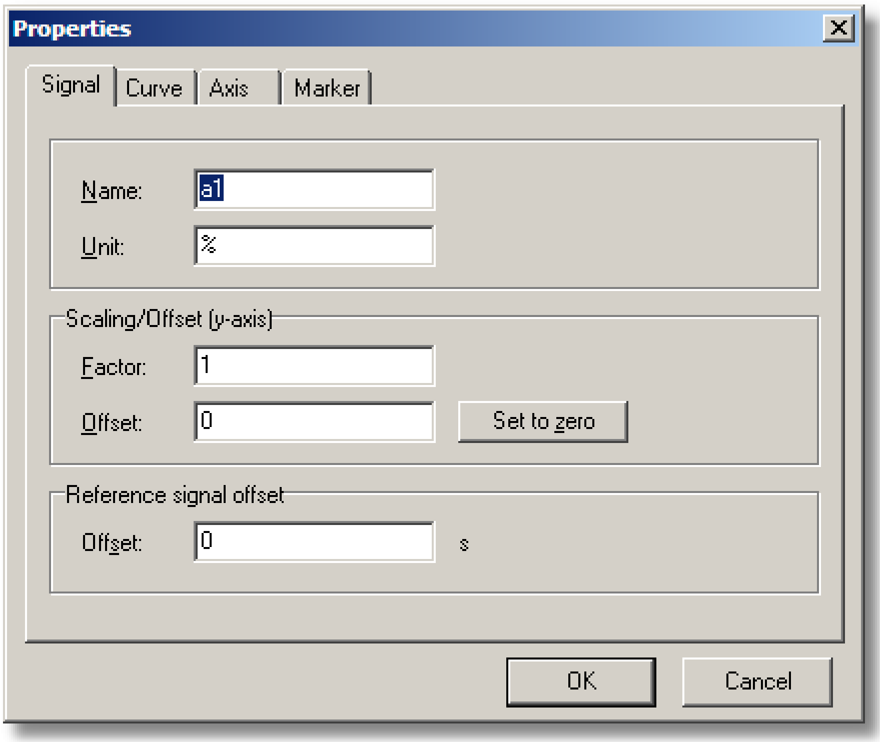


Figure 2-59: Properties Form (*Signals* tab).

All settings in this tab are only used for displaying the chart. That means that a signal can be renamed/rescaled, but the changes will only be displayed in the chart window and not necessarily in the signal values which are exported.

The **Set to zero button** requires a cursor mode and inserts an offset with which the selected signal is zero at the selected point.

The **Reference signal offset** is negative for a delayed signal.

2.9.2.3 Properties Form (Curve tab)

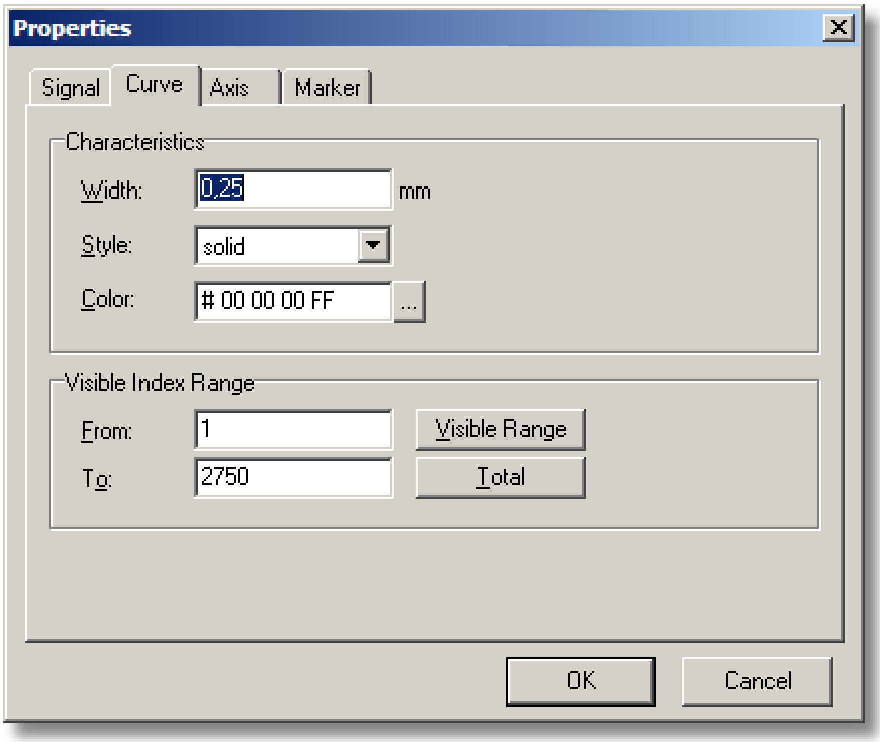


Figure 2-60: Properties Form (*Curve* tab).

The curve must be visible in order to also see the *Curve* tab.

The curve's **Characteristics** can be set in this tab.

The **Visible Index Range** is the first visible sample and the last visible sample. Samples before the first visible sample, and samples behind the last visible sample are hidden.

The **Visible Range** button inserts the values for the curve's visible samples.

The **Total** button inserts the max. possible range for the curve.

2.9.2.4 Properties Form (Axis tab)

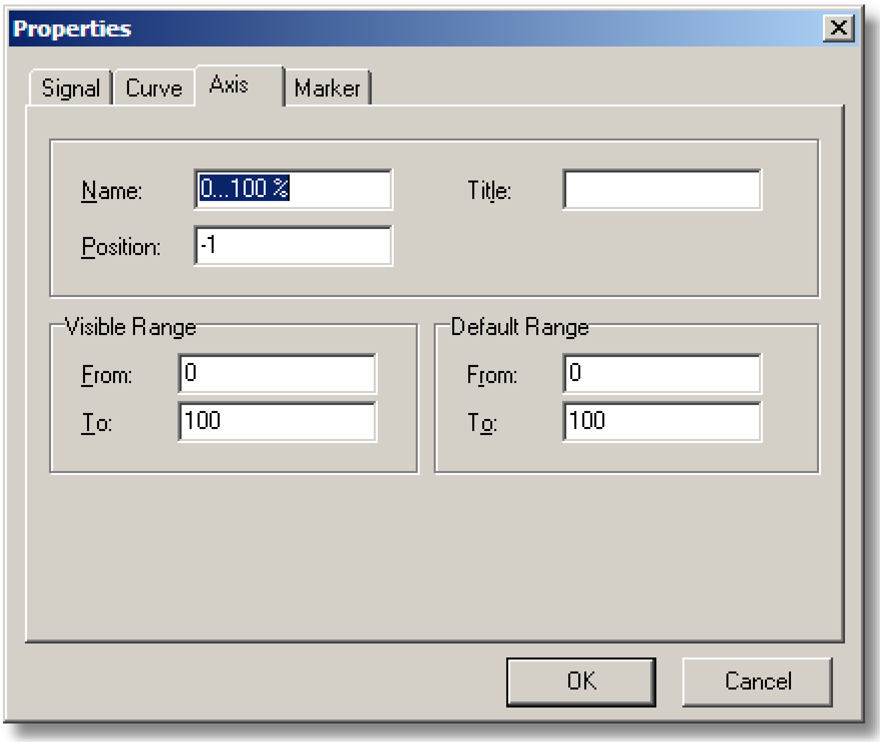


Figure 2-61: Properties Form (*Axis* tab).

If the **Name** of an axis is non-empty, all signals with this name share the same axis. To split the axes again, the name must be temporarily deleted (in order to have separate axes).

The **Title** is the axis' caption.

The **Position** is negative for the left side and closer to the chart with a smaller non-zero absolute value.

2.9.2.5 Properties Form (Marker tab)

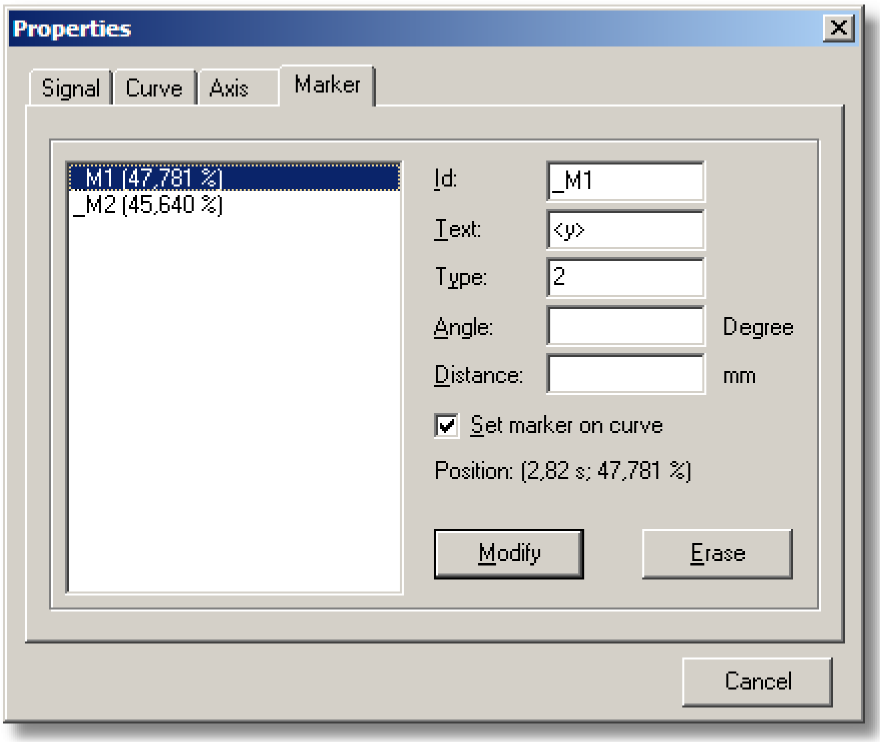


Figure 2-62: Properties Form (*Marker* tab).

In order to see the *Marker* tab, at least one marker needs to be existing (which is done by pressing the *m* key when the difference cursor is active).

The **Id** is the identification for the marker and used as a program internal reference for this marker.

The **Text** is the displayed text for this marker. The placeholders *<x>* and *<y>* are used for the corresponding curve's coordinate at the marked point.

The **Type** specifies the type which is 2 for a standard marker, and 1 for a bigger marker. A zero (or empty field) represents an invisible marker with text only.

The **Angle** can be specified in degrees (0° ... 360°).

The **Distance** between text and marker point can be specified in millimeters.

If the **Set marker on curve checkbox** is unchecked, the marker is invisible.

The **Modify** button (= OK tab button) accepts the changes.

The **Erase** button removes the selected marker.

2.9.2.6 Diagram Properties Form

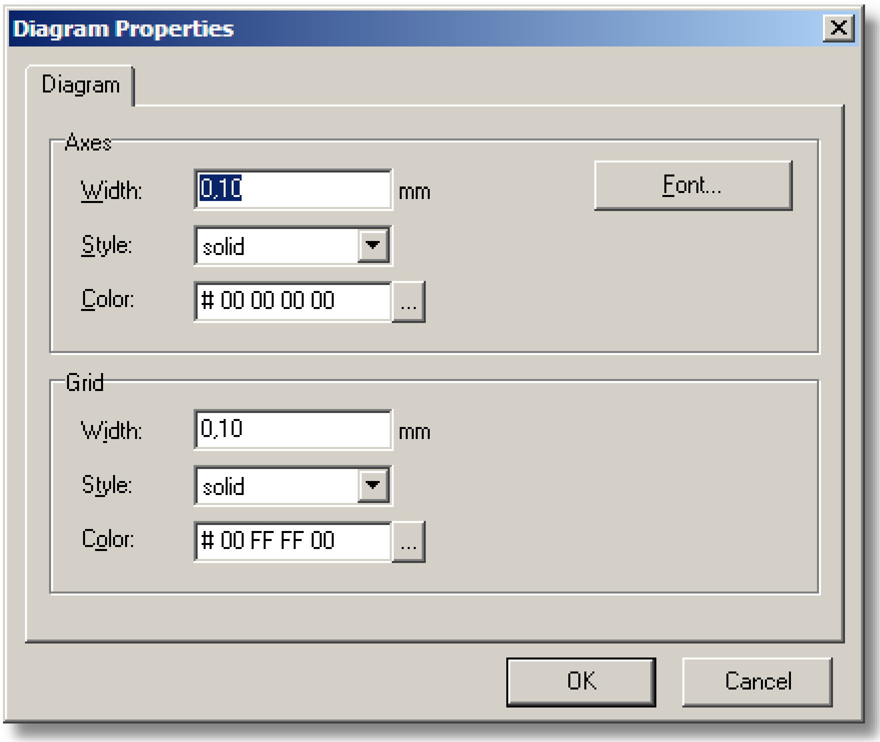


Figure 2-63: Diagram Properties Form.

The settings in the *Diagram* tab are made for the axes and grid.

2.9.2.7 Save View Form

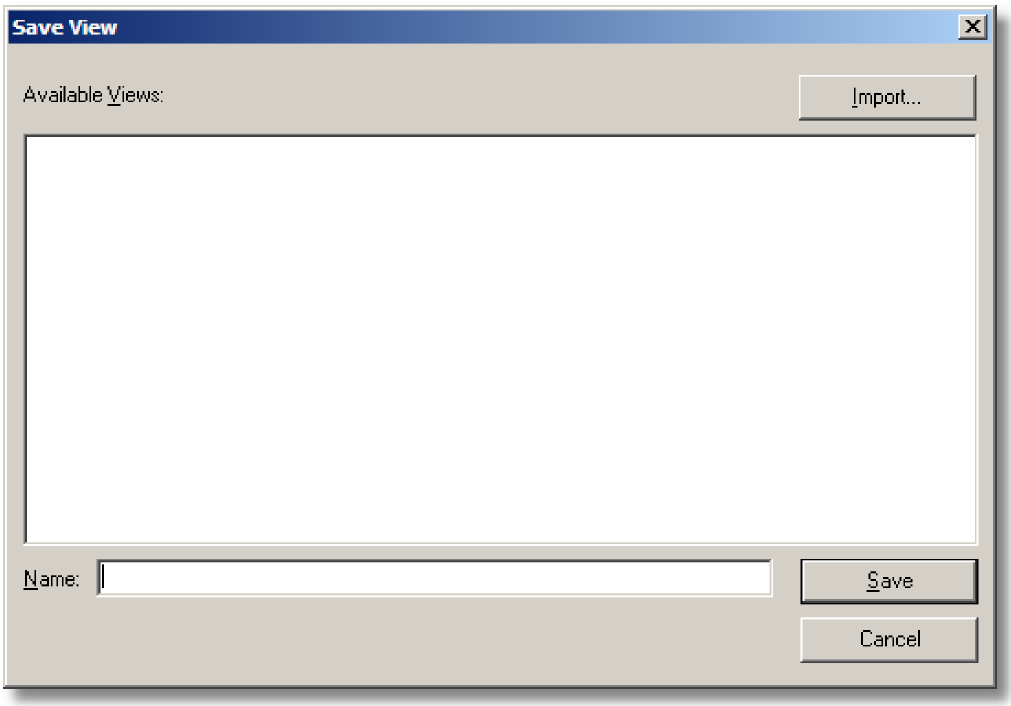


Figure 2-64: Save View Form.

The tree structure shown in **Available Views** is reflected by the menu item *View*. A sub menu is created by using the backslash (\) as a separator. The *Del* key deletes an item. The *Shift* + *Del* key deletes an item without questions. A tree item can be renamed with the *F2* key. A **name** can be entered which is used as a new tree/menu item.

With the **Import** button, a TEMES file can be selected whose views are used to append or to replace the tree/menu items.

2.9.2.8 Statistics Form

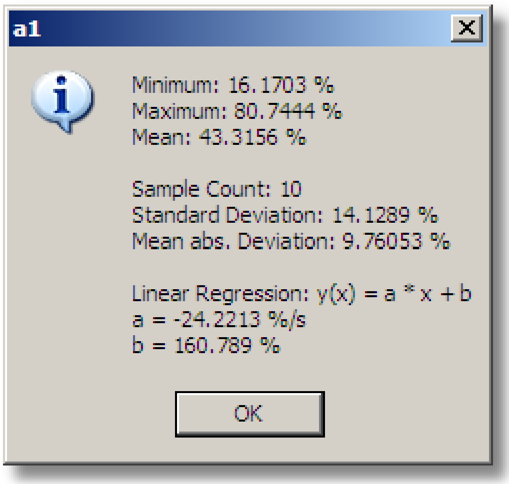


Figure 2-65: Statistics Form.

The statistics form shows statistic values for the samples between the difference cursor.

2.9.2.9 Recording Settings Form

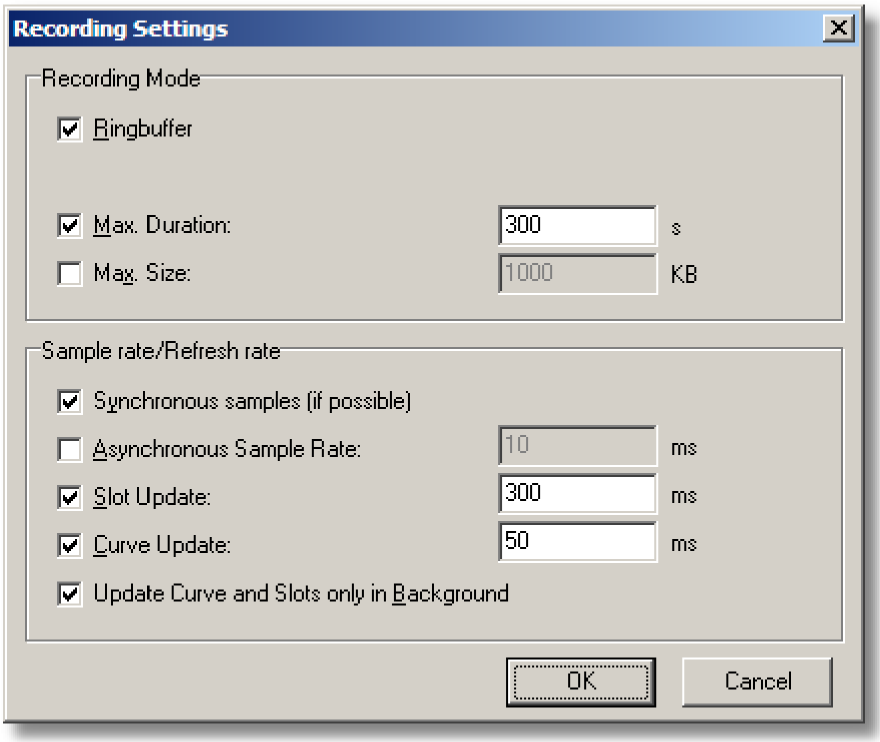


Figure 2-66: Recording Settings Form.

The **Ringbuffer** specifies whether the oldest samples shall be overwritten with the newest samples when the max. recording duration is reached.

The **Max. Duration** specifies the max. duration of the recording.

Synchronous samples (if possible): This is a mode which chooses the signals to be sampled in advance and asks the device to do the sampling at the given rate if checked.

Asynchronous Sample Rate: If checked the SICOLOG/USBDL1 or SICO3 is asked to do the sampling at another rate. In case of a DL16CAN or SICO2, no sampling is done in the device, but the signals are queried by polling the current signal values instead.

Update Curve and Slots only in Background: If checked, TEMES will only update the curve and slots when it has time to do so. The signal recording has priority in this case.

2.9.2.10 Save or Discard Measurement Form

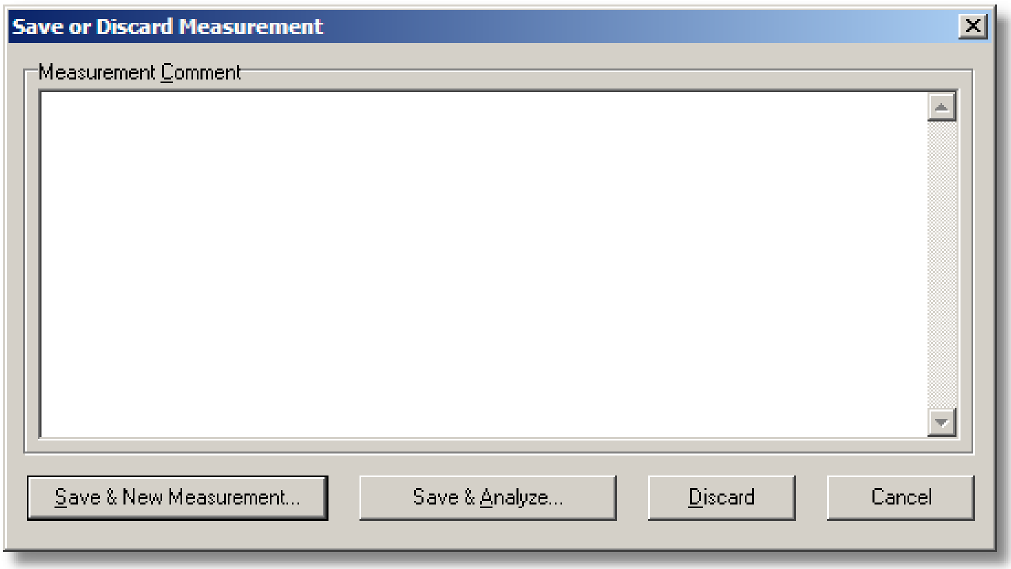


Figure 2-67: Save or Discard Measurement Form.

Save & New Measurement: Saves a measurement together with the **Measurement Comment** and starts a new measurement.

Save & Analyze: Saves a measurement together with the **Measurement Comment** and returns back to analyzing mode.

Discard: Discards the measurement.

Cancel: Closes the dialog box.

2.9.3 Diagram Pane

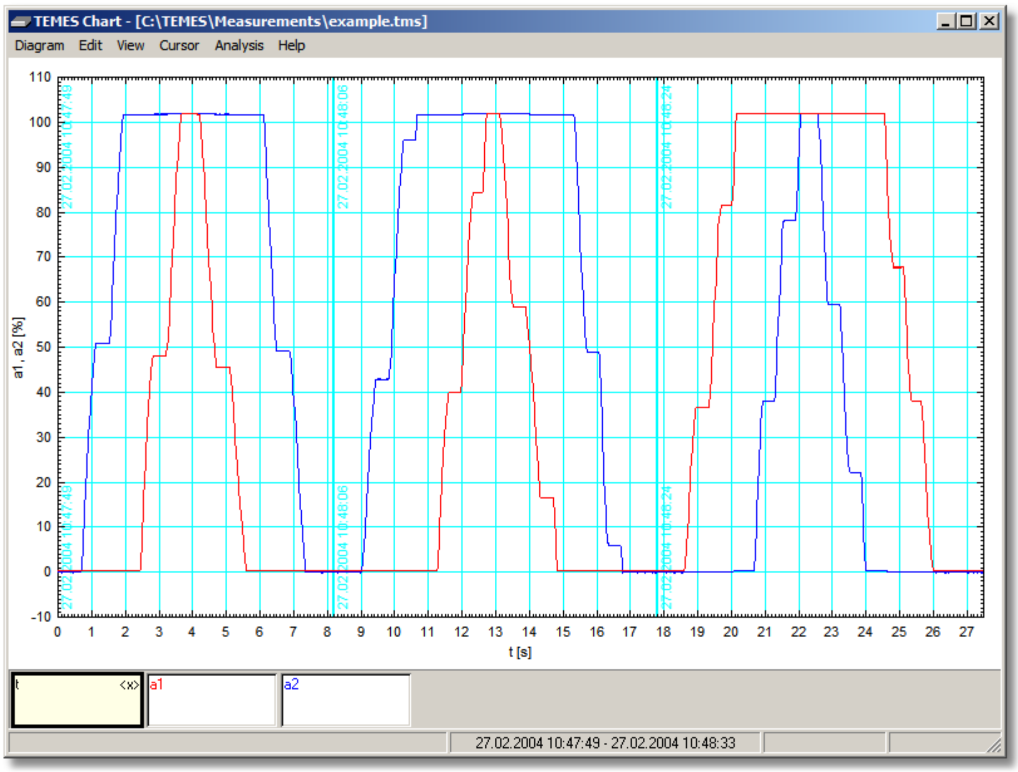


Figure 2-68: Diagram pane.

The diagram pane has usually one or more timestamps embedded in the grid. A timestamp marks a new measurement, and the samples between two time-stamps are usually considered to be continuously sampled.

Below the diagram pane are the slots where one slot can be selected (marked with a black border and a yellowish background) with the *Tab* and *Shift* + *Tab* keys.

2.9.3.1 Single Cursor

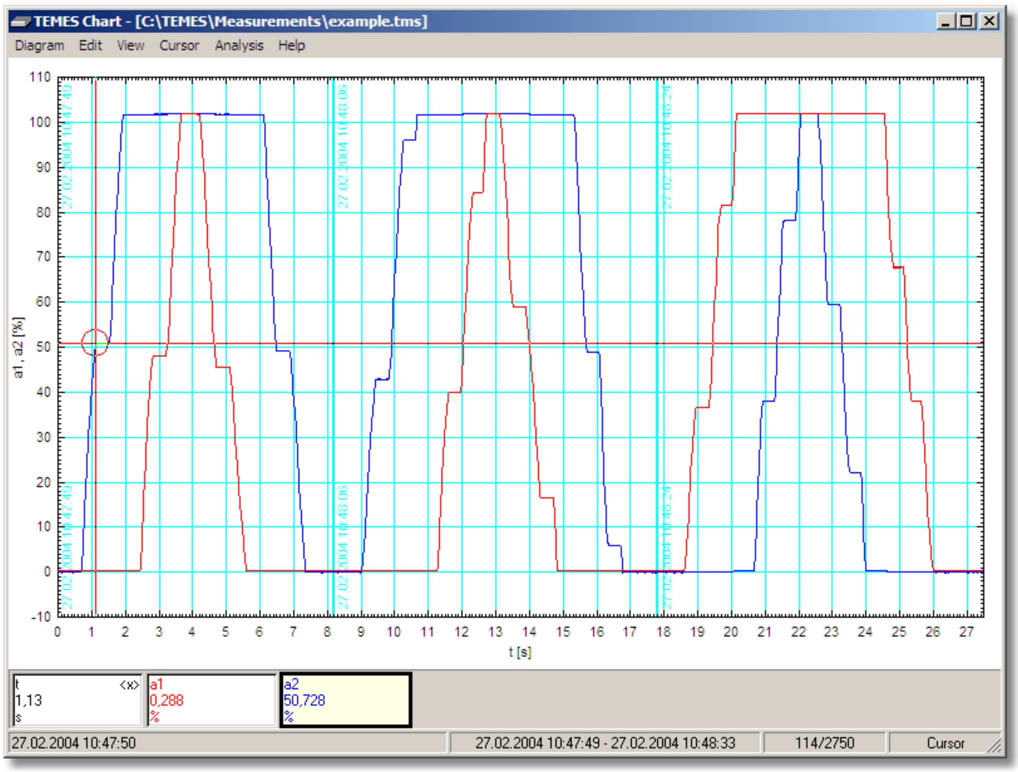


Figure 2-69: Diagram Pane (Single Cursor).

The single cursor mode is entered by pressing the *F9* key and then by selecting a curve point with the left mouse button.

2.9.3.2 Difference Cursor

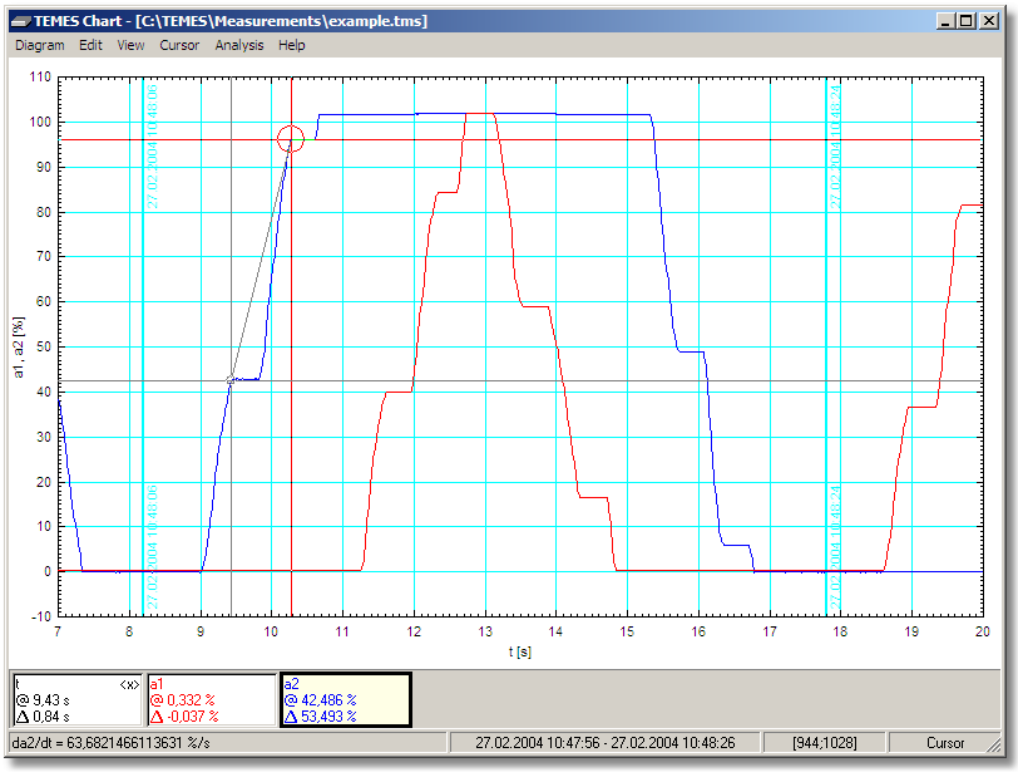


Figure 2-70: Diagram Pane (Difference Cursor).

The difference cursor mode is entered from the single cursor by pressing the *n* key, or by clicking the middle mouse button (= *Ctrl* key + left mouse button).

2.9.4 Keyboard Shortcuts

2.9.4.1 Standard Mode

d	Default range for x direction
D	Default range for y direction
t	Maximal x range for current signal
T	Maximal y range for current signal
z	Zoom in x direction
Z	Zoom in y direction
u	Unzoom in x direction
U	Unzoom in y direction
i	Isotropic x axis adjustment while displaying Earth coordinates (x range [km] := y range [km])
I	Isotropic y axis adjustment while displaying Earth coordinates (y range [km] := x range [km])
Arrow keys	Move image section by half of the visible range

Ctrl + Arrow	Move image section by full visible range
v	Display/Hide curve for current signal. This operation is done for all signals, if the current slot represents the x axis
a	Display/Hide axis for current signal. This operation is done for all signals, if the current slot represents the x axis
Shift + Arrow	Move the axis of the current signal to the left or right
Tab	Select the next slot (to the right)
Shift + Tab	Select the next slot (to the left)
Del	Remove slot
Ins	Show all slots including removed slots
Alt + Arrow	Move slot
x	Make current slot to the x axis
0	1st point of the two-point-calibration (in online view)
1	2nd point of the two-point-calibration (in online view)
Esc	Cancel curve drawing
F8	Repaint all curves
Shift + G	If °N/°E signals exist: Make a KML file of the visible image section and call *Google Earth*. If *Google Earth* is installed but does not start, then the *KML-Fixer* (https://www.tellert.de/?product=temes) can be helpful.
Ctrl + G	If °N/°E signals exist: Make a GPX file of the visible image section and call the associated application
Page Up[1]	Display of the previous recording (delimited by timestamps)
Page Down[1]	Display of the next recording (delimited by timestamps)
Ctrl + Home[1]	Display of the first complete recording
Ctrl + End[1]	Display of the last complete recording
Shift + Home[1]	Movement to the beginning of the first recording
Shift + End[1]	Movement to the end of the last recording
Ctrl + 1	Unit change of the time (= x-axis) in "s" (= seconds)
Ctrl + 2	Unit change of the time (= x-axis) in "min" (= minutes)
Ctrl + 3	Unit change of the time (= x-axis) in "h" (= hours)

Ctrl + 4	Unit change of the time (= x-axis) in "d" (= days)

[1] Only valid when the time signal is used as *x*-axis. A recording in this context is defined as the duration between two adjacent timestamps or the end of the measurement.

2.9.4.2 Cursor Mode

0	Move curve by setting the current point to zero
PgUp	Set cursor to the previous sample
PgDn	Set cursor to the next sample
Ctrl + PgUp	Set cursor to the 10th previous sample
Ctrl + PgDn	Set cursor to the 10th next sample
Home	Set cursor to the first sample
End	Set cursor to the last sample
c	Center x range to cursor
C	Center y range to cursor
m	Open local menu to set/move/hide a marker. A marker is being set when the cursor is on the current curve. A marker is being removed when the cursor is outside the current curve.
n	Change to difference cursor, or reset of the zero point of the difference cursor, or change to normal cursor
g	If °N/°E signals exist: Copy coordinates of the selected point into the clipboard
>	Replays a marked section (= difference cursor)

2.9.5 Mouse Usage

Left mouse button: Execution of the standard action.

Right mouse button: Open a local context menu.

Middle mouse button: Execution of the alternative action. The middle mouse button can be simulated by pressing the left mouse button together with Ctrl.

Standard action (Axis): Move of visible range. The *Shift* key removes the gra-ting.

Alternative action (Axis): Zoom/Unzoom of the visible range. The clicked point remains at its position. A movement to greater numbers triggers an unzoom. A movement to smaller numbers triggers a zoom.

Standard action (diagram area): A rectangle from top left to down right represents the new displayable range (zoom in). If the mouse is dragged from bottom left to top right, then the x axis is zoomed. If the mouse is dragged from top right to bottom left, then the y axis is zoomed.
There is no grating when the Shift key is pressed. The Alt key triggers an unzoom. (The selected range represents the current visible range).

Alternative action (diagram area): Move visible range. There is no grating when the *Shift* key is pressed.

Standard action (diagram area/Cursor mode): If no curve is currently selected, the closest curve will be selected. With the *Shift* key, also the in-between sample values are searched. The *Alt* key activates the alternative cursor. There is no selected curve in the alternative cursor mode and no curve binding of the cursor.

Alternative action (diagram area/Cursor Mode): Activate or position the difference cursor.

Mouse wheel: Paging in x direction

Shift + Mouse wheel: Paging in y direction

Ctrl + Mouse wheel: Zoom/Unzoom in x direction

Shift + Ctrl + Mouse wheel: Zoom/Unzoom in y direction

2.9.6 Status Bar in Cursor Mode

A conversion of the current signal is shown in the status bar.

Normal Cursor:

y axis		Conversion
km/h, kph		Conversion in m/s

Difference Cursor:

y axis	x axis	Conversion
km/h, kph	s, sec	Conversion in acceleration (m/s^2)
°N, °S	°E, °O, °W	Length of the full circle segment in km; start course angle;
m	km	Slope in percent; inclination angle;
arbitrary	arbitrary	d*y*/d*x*; same units: angle;
arbitrary	s, sec	Current signal is time: 1/*x* (to estimate frequencies)

2.10 Trouble Shooting

If a specific TEMES file cannot be overwritten, please choose a new non-existing file instead. If a specific TEMES parameter set cannot be loaded, please try to import the parameter set instead. If TEMES behaves unexpectedly, please save the current parameter set and restart TEMES.

First Steps

3 First Steps

3.1 Displaying a Voltage Signal

1. Call the TEMES menu item *File → New*.
2. Select the corresponding device in the *Add Device* dialog box and press the *Add* button.
3. Select the path \Devices\<**device**>\Analog Input\AI1 and enter the signal name (e. g. *a1*) in the *name* edit box.
4. Set the following assignment in order to get a voltage signal instead of a percentage value:

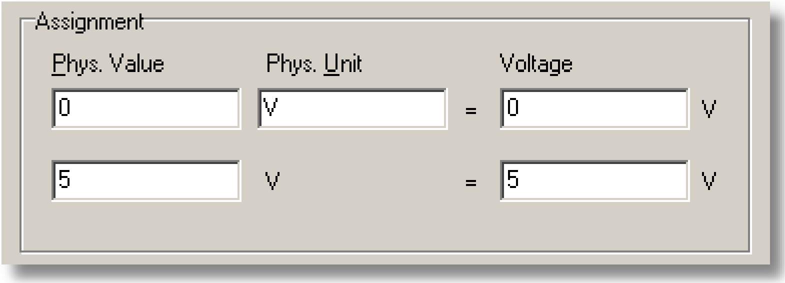


Figure 3-1: Voltage assignment.

5. Select the path \Devices\<**device**>\Display\D1.1.
6. Select the output item: *Source as floating point value*.
7. Enter the text *1:* in the field output text.
8. Select *a1* as source.
9. Call the TEMES menu item *Hardware → Program device.*

3.2 Recording a Voltage Signal

1. Call the TEMES menu item *File → New*.
2. Select the corresponding device in the *Add Device* dialog box and press the *Add* button.
3. Select the path \Devices\<**device**>\Analog Input\AI1 and enter the signal name (e. g. *a1*) in the name edit box.
4. Set the following assignment in order to get a voltage signal instead of a percentage value:

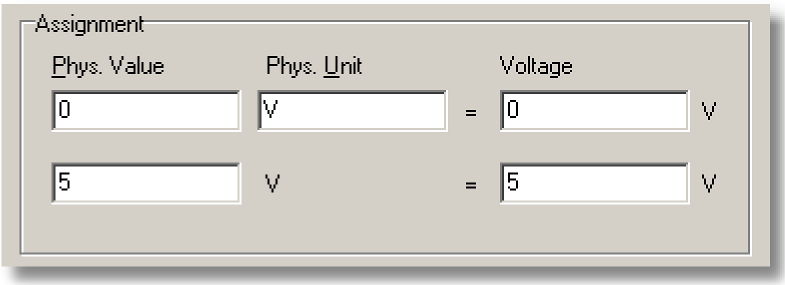


Figure 3-2: Voltage assignment.

5. Select the path *Devices****<device>***Recording**Memory**Fast Cycle*.
6. Select the *a1* signal
7. Press the < button to bring the a1 signal into the *recording signals* list box.
8. Select the path *Devices****<device>**
9. Select the tab *General*
10. Enter the desired sample rate.
11. Call the TEMES menu item *Hardware → Program device*.

3.3 Making a Measurement

3.3.1 Making a Measurement with SICOLOG/USBDL1

1. Record a voltage signal 124.
2. Move the *REC* slider on.
3. Press the *REC* button.
4. Wait 5 seconds.
5. Press the *STOP* button.
6. Optionally, discard the measurement: Move the *REC* slider off. Press both the *STOP* button and the *REC* button, and hold these buttons down for at least 3 seconds until the *L4 LED* is unlit again.
   Make a new measurement: Move the *REC* slider on. Press the *REC* button. Wait 5 seconds. Press the *STOP* button.
7. Remove the USB stick and insert it into the PC
8. Call TEMES menu *Hardware → Save Measurement from Mass Storage...*
9. Keep both checkboxes checked.
10. Select the USB stick and press the *Save...* button.

3.3.2 Making a Measurement with CFDL1

1. Record a voltage signal 124.
2. Move the *REC* slider on.
3. Press the *REC* button.
4. Wait 5 seconds.
5. Press the *STOP* button.
6. Optionally, discard the measurement: Move the *REC* slider off. Press both the *STOP* button and the *REC* button, and hold these buttons down for at least 3 seconds until the *L4 LED* is unlit again.
   Make a new measurement: Move the *REC* slider on. Press the *REC* button. Wait 5 seconds. Press the *STOP* button.
7. Remove the CF card and insert it into the PC
8. Call TEMES menu *Hardware → Save Measurement from Mass Storage...*
9. Keep both checkboxes checked.
10. Select the CF card and press the *Save...* button.

3.3.3 Making a Measurement with DL16/DL16CAN

1. Record a voltage signal 124.
2. Press the *START* button.
3. Wait 5 seconds.
4. Press the *STOP* button.
5. Optionally, discard the measurement: Press both the *STOP* and the *START* buttons and hold these buttons down for at least 3 seconds.
   Make a new measurement: Press the *START* button. Wait 5 seconds. Press the *STOP* button.
6. Connect the data logger with the PC.
7. Call TEMES menu *Hardware → Save Measurement...*
8. Select the data logger and press the *Save as...* button.

3.4 Choosing a CAN signal from a DBC file

1. Run *TEMES*.
2. Call menu item *Tools → CAN Editor...*
3. Call CAN editor's menu item *File → Import...*
4. Select the desired DBC file.
5. Verify the CAN baud rate in the root/CAN node.
6. Call CAN editor's menu item *File → Save*, and choose the *CAN* sub-folder of the TEMES data folder (which is usually "*documents\TEMES\CAN*") as storage location for the CAN file.
7. Close the CAN editor.
8. Call TEMES' menu item *File → New*.
9. Select the corresponding device in the *Add Device* dialog box and press the *Add* button.
10. Select path *\Devices**<device>**\CAN*, *\Devices**<device>**\CAN1* or *\Devices**<device>**\CAN2*.
11. Click inside the drop-down list box *library* and select the saved CAN file.
12. Press on a plus symbol inside the signals list box.
13. Select a CAN signal (which is below a CAN message).
14. Press the *Add Input* button.

More Advanced Steps

4 More Advanced Steps

4.1 Multiplexed Temperatures (TH16MI)

Following steps are required to setup one (or more) thermocouple multiplexer(s).

4.1.1 Setup of the Multiplexer(s)

The control code is broadcast commonly to all connected multiplexers at the same time. Hence, TEMES only knows one multiplexer (and treats the other multiplexers the same way). The settings are made in the path *Devices*\ ***<device>*** in the tab *Multiplexer* 41. The first checkbox must be checked to inform TEMES that a control byte has to be broadcast to connected multiplexers. The second checkbox must be checked if the multiplexer(s) are synchronized with the slow recording cycle instead of the fast recording cycle. A name can be given to the channel counter edit box which allows to refer to it inside TEMES. Then, the number of time slots must be entered (from 1 to 16, and 16 being the default value). The table consists of three columns: The first column is the time slot, the second column is the channel (from 1 to 16), and the third column is the operating range (from 1 to 16).

Operating Range	Min. Temperature [°C]	Max. Temperature [°C]
1	66.7	1372.0
2	50.3	1200.0
3	40.4	943.7
4	20.4	474.0
5	10.2	240.5
6	6.8	159.3
7	5.1	119.1
8	4.1	95.4
9	17.0	1372.0
10	0.0	1144.9
11	-10.4	892.6
12	-31.7	426.0
13	-42.8	189.0
14	-46.5	109.0
15	-48.4	70.0
16	-49.5	46.3

With linearized temperatures.

4.1.2 Choosing a Voltage Input

A voltage input must be chosen at which the thermocouple amplifier is connected. This is done in the path *\Devices***<device>***\Analog Input\A***x**, and in the tab *Multiplexer*: It is recommended to check all three checkboxes and set the settling time to 20 ms (Note that the channel sample rate must be slower than 20 ms to obtain at least one value). A plus symbol appears in the parameter tree after pressing the *ENTER* key. By clicking this plus symbol further inputs are available which will be explained in the next section. However, the multiplexed signal needs a parameter definition for the main temperatures: This is done by selecting the tab *Char. Curve* and by choosing the table *th2 N=29* (or *th16mi N=31*). The nonlinearized parameters are specified in the tab *Settings*, where first a name must be entered and an assignment according to the main temperature range must be made (see next section).

4.1.3 Subchannel of the Multiplexed Voltage Input

In the tab *Char. Curve*, choose again a table (like *th2 N=29* or *th16mi N=31*), and enter a name for the multiplexed signal in the tab *Settings*. Then, also in the tab *Settings*, the assignment must be made according to the corresponding operating range.

Operating Range	$T_{c1,n}$ [°C]	V_{c1} [V]	$T_{c2,n}$ [°C]	V_{c2} [V]
1	70.0418	0.2101	1346.9403	4.0408
2	59.8673	0.2395	1200	4.8
3	49.6928	0.2485	956.3775	4.7819
4	29.5650	0.2957	475.8663	4.7587
5	19.6117	0.3922	239.4937	4.7899
6	9.7567	0.2927	150.8233	4.5247
7	9.7567	0.3903	110.7889	4.4316
8	9.7567	0.4878	90.4645	4.5232
9	19.6117	0.2088	1346.9403	4.1908
10	0	0.2000	1145.5394	4.7822
11	-9.6338	0.2018	907.4711	4.7874
12	-28.4099	0.2159	423.7159	4.7372
13	-37.5276	0.2494	180.3391	4.6068
14	-37.5276	0.3742	100.6390	4.5192
15	-37.5276	0.4989	59.8673	4.3947
16	-37.5276	0.6236	39.5920	4.4796

With first nonlinearized temperature $T_{c1,n}$, second nonlinearized temperature $T_{c2,n}$, voltage V_{c1} at $T_{c1,n}$, and voltage V_{c2} at $T_{c2,n}$.

4.1.4 Choosing an Output Signal

An output has to be used for the multiplexed voltage input in order to activate the multiplexed voltage input, e. g. a display item (*Devices*\ ***<device>***\ *Display*\ *D1.1*) or a recording signal (like \ *Devices*\ *Recording*\ *Memory*\ *Fast Cycle*).

4.1.5 Calibrating a Temperature Multiplexer

For convenience reasons, a separate parameter set is created with one multiplexer time slot which is set to the operating range to be calibrated. The sample rate is set to 200 ms (to obtain a very smooth signal averaging). Then, the multiplexed voltage input (and its first sub-channel) must be defined as described as before. And a display output for the multiplexed voltage input should be defined as well. Then, call menu item *Hardware → Calibrate device...* and select the nonlinearized temperature signal, attach the voltage for the linearized temperature (e. g. with a thermocouple calibrator) and press the corresponding button of the two-point calibration. Attach the other linearized temperature and press the corresponding button of the two-point calibration. Then press *OK*. The device is now calibrated.

Operating Range	$T_{c1,n}$ [°C]	T_{c1} [°C]	$T_{c2,n}$ [°C]	T_{c2} [°C]
1	70.0418	70	1346.9403	1370
2	59.8673	60	1200	1200
3	49.6928	50	956.3775	940
4	29.5650	30	475.8663	470
5	19.6117	20	239.4937	240
6	9.7567	10	150.8233	150
7	9.7567	10	110.7889	110
8	9.7567	10	90.4645	90
9	19.6117	20	1346.9403	1370
10	0	0	1145.5394	1140
11	-9.6338	-10	907.4711	890
12	-28.4099	-30	423.7159	420
13	-37.5276	-40	180.3391	180
14	-37.5276	-40	100.6390	100
15	-37.5276	-40	59.8673	60
16	-37.5276	-40	39.5920	40

With first nonlinearized temperature $T_{c1,n}$, first linearized temperature T_{c1}, second nonlinearized temperature $T_{c2,n}$ and second linearized temperature T_{c2}.

4.1.6 Recording Multiplexed Temperatures

The recommended way is to check all three checkboxes in the tab *Multiplexer* 46 like:

Figure 4-1: Recommended Multiplexer Settings.

Then, only the multiplexed main temperature signal needs to be recorded.

After the measurement is read out, the parameter set is searched for defined (but not necessarily used) sub-channels. If these sub-channels are found then they are automatically demultiplexed and linearized by the AutoCalc function (first by looking for a characteristic curve without the "*N*=" part, and only if this curve is not found then the characteristic curve is taken as specified).

4.2 Calculated Signals: Bang-Bang Controller

A bang-bang controller is realized in this section to demonstrate the calculated signals. This example requires only one voltage input signal called *T (lin)* for the temperature. The output signal is *Cnt* which is either 0 or 1 to control a relay.

- F1: C = 0 (*op*3, C_1 = 0)
- F2: C = 1 (*op*3, C_1 = 1)
- F3: C = 7°C (*op*3, C_1 = 12454)
- F4: C = 8°C (*op*3, C_1 = 12670)
- F5: T <= 7°C (*op*11, S_1 = "T (lin)", S_2 = "C = 7°C")
- F6: T >= 8°C (*op*11, S_1 = "C = 8°C", S_2 = "T (lin)")
- F7: Cnt (*op*20)
- F8: Cnt == 0 (*op*9, S_1 = "Cnt", S_2 = "C = 0")
- F9: Cnt == 1 (*op*9, S_1 = "Cnt", S_2 = "C = 1")
- F10: Cnt == 0 && T >= 8°C (*op*14, S_1 = "Cnt == 0", S_2 = "T >= 8°C")
- F11: *Procedure* (*op*21, S_1 = "Cnt", S_2 = "Cnt == 0 && T >= 8°C")
- F12: Cnt == 1 && T <= 7°C (*op*14, S_1 = "Cnt == 1", S_2 = "T <= 7°C")
- F13: *Procedure* (*op*22, S_1 = "Cnt", S_2 = "Cnt == 1 && T <= 7°C")

Devices

5 Devices

5.1 CFDL1

The home page of the CFDL1 is located at http://www.tellert.de/?product=cfdl1.

Adjustment of the real time clock: The CFDL1 must be connected via a USB cable with a type B plug (and **not** with the virtual COM port USBSER). Then, there must be a parameter set running on the CFDL1 (meaning a CF card with a valid parameter set is inserted in the CFDL1). The synchronize process is then started with *TEMES → Tools → Additional Tools → Hardware → Set CFDL1 real time clock*. When the CFDL1 cannot hold the real time after a power cycle, then keep the CFDL1 on power for at least 2 hours.

CF card initialization: The initialization of a CF card is done with *TEMES → Tools → Additional Tools → Hardware → Initialize CF card (CFDL1)*.

Display remains empty (while turning the power switch): Please verify whether the device is connected to a power supply. The RS232/USB connector does not supply the device with (enough) current.

5.2 DL16/DL16CAN

The home page of the DL16/DL16CAN is located at http://www.tellert.de/?product=dl16can.

The buffer battery should have a voltage of ≥ 3 V, otherwise it should be replaced which is described as follows.

5.2.1 Changing the Buffer Battery

5.2.1.1 Procedure

Figure 5-1: Unscrew the four M2.5 screws from the lid and the two M2.5 screws from the parallel interface.

Figure 5-2: Turn the lid to the top. (Pay attention that the LED is adjusted correctly, and that the wires don't occupy the space for the MCU when closing the lid).

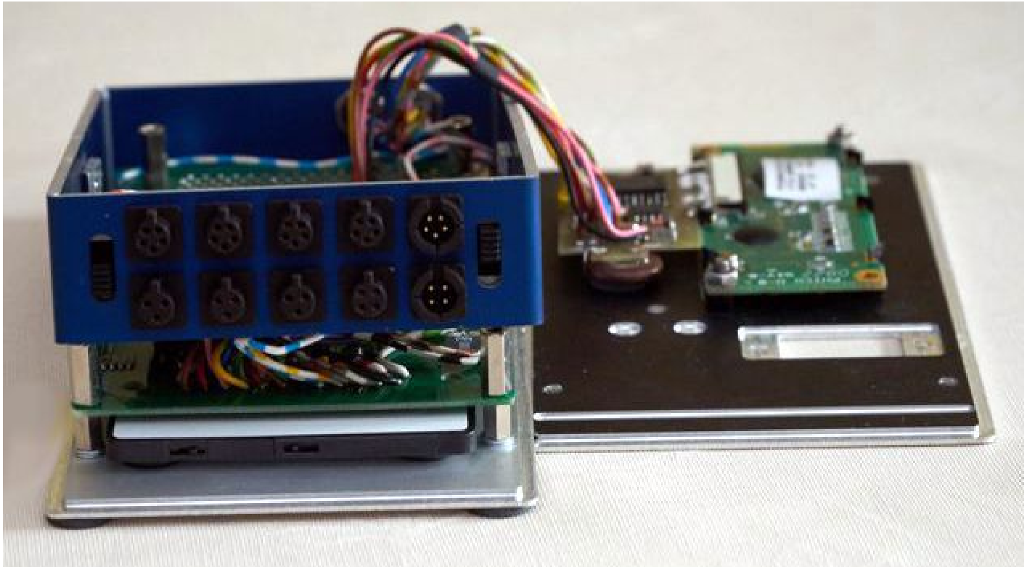


Figure 5-3: Lift the right side of the DL16CAN until the memory card can be seen.

Figure 5-4: Remove the card. The card can be levered with a screwdriver (by using the notches).

Battery replacement:

- Recommended Batteries:
  PAYOVAC BR2325
  PANASONIC BR2325
  PANASONIC CR2025
- Setting the write-protect switch: This switch is similar to the write-protect switch on the floppy disk. Write is inhibited when the switch is toggled to the write-protect position, and vice versa otherwise.
- Unlock the battery cover: Push the switch left and pull out using a pen.
- Take off the cover with the old battery.

- Insert a new (recommended) battery into the card.
- Put the battery cover on and lock-up.
- Note: Once the battery is removed from the card, it must be replaced within 20 minutes, or the data will be lost.

5.2.1.2 Trouble Shooting (Changing the Buffer Battery)

If the DL16CAN does not communicate with TEMES after inserting a new SRAM card, then the following procedure might be helpful:

- Turn DL16CAN off
- Bridge pin 2 and pin 3 (= two lower pins) of the connector *LOGIC I/O*, e. g. with a straightened electrically conducting paper clip
- Turn DL16CAN on (the display is invalid due to a blocked communication with the LCD)
- Wait 5 seconds
- Remove the bridge (e. g. remove the paper clip)
- Program the DL16CAN with TEMES

5.2.2 Trouble Shooting

If the device does not show an initial text onto its display while turning the switch from off to on position, then please verify whether the device is connected to a power source. The RS232 connection does not supply the device with current.

If the USBAD is not displayed in TEMES after a TEMES 1.0.35 or newer update, then the service TE_SPTI.EXE might be deactivated. This service can be activated again by deinstalling TEMES, followed by a computer restart, and followed by an installation of TEMES.

If a measurement is read via the USBAD, and if in rare cases there are echos (= replays of the same values concerning all signals) in the measurement, then please verify whether your USBAD has the newest firmware. If your USBAD firmware is not at least V1.3.3, then please upgrade the firmware and repeat the read out of the measurement. The homepage of the USBAD is located at http://www.tellert.de/?product=usbad where a software to lookup/upgrade the firmware is available.

If the DL16/DL16CAN cannot hold the real time after a power cycle, please power the DL16/DL16CAN for at least 2 hours and then reprogram the DL16/DL16CAN with TEMES.

If the DL16/DL16CAN does not communicate with TEMES then the following procedure might be helpful:

- Turn DL16/DL16CAN off

- Bridge pin 2 and pin 3 (= two lower pins) of the connector *LOGIC I/O*, e. g. with a straightened electrically conducting paper clip
- Turn DL16/DL16CAN on (the display is invalid due to a blocked communication with the LCD)
- Wait 5 seconds
- Remove the bridge (e. g. remove the paper clip)
- Program the DL16/DL16CAN with TEMES

5.3 SICO2

The home page of the SICO2 is located at http://www.tellert.de/?product=sico2.

5.3.1 Trouble Shooting

If the device does not show an initial text onto its display while turning the switch from off to on position, then please verify whether the device is connected to a power source. The RS232 connection does not supply the device with current.

If the SICO2 does not communicate with TEMES then the following procedure might be helpful:

- Turn SICO2 off
- Bridge pin 2 and pin 3 (= two lower pins) of the connector *LOGIC I/O*, e. g. with a straightened electrically conducting paper clip
- Turn SICO2 on (the display is invalid due to a blocked communication with the LCD)
- Wait 5 seconds
- Remove the bridge (e. g. remove the paper clip)
- Program the SICO2 with TEMES

5.4 SICO3

The properties of the signal converter are almost the same as for the SICOLOG138.

The home page of the SICO3 is located at http://www.tellert.de/?product=sico3.

5.5 SICO3M

The properties of the signal converter are almost the same as for the SICOLOG138.

The home page of the SICO3M is located at http://www.tellert.de/?product=sico3m.

5.6 SICOLOG

Some of the SICOLOG tools require the Microsoft .NET Framework 4 which is part of Windows 8 or newer, or which can be downloaded from https://www.microsoft.com/en-us/download/details.aspx?id=17718. And the Microsoft .NET Framework 4 Full Language Pack can be downloaded from https://www.microsoft.com/de-de/download/details.aspx?id=3324.

The home page of the SICOLOG is located at http://www.tellert.de/?product=sicolog.

5.6.1 Trouble Shooting

If the device does not show an initial text onto its display while turning the switch from off to on position, then please verify whether the device is connected to a power source. The RS232 connection does not supply the device with current.

If the SICOLOG cannot hold the real time after a power cycle, please power the SICOLOG for at least 2 hours and reprogram the SICOLOG via a programming cable with TEMES.

If the SICOLOG does not communicate with TEMES then the following procedure might be helpful (note, that the SICOLOG does not communicate during USB stick preallocation):

- Turn SICOLOG off
- Remove USB stick
- **SICOLOG/USBDL1 Firmware ≥ V1.0.29 only:** Move the *Record* slider in off position.
- **Firmware ≥ V1.0.16:** Hold both buttons B3 and B4 and startup the SICOLOG. (Respectively, hold both buttons *NEXT* and *INS* and startup the SICO3, or hold both buttons B1 and B2 and startup the USBDL1/SICO3M.)
  Firmware ≥ V1.0.0: Bridge pin 2 and pin 3 (= two lower pins) of the connector *LOGIC I/O*, e. g. with a straightened electrically conducting paper clip
- Turn SICOLOG on (note that a programmed parameter set won't be loaded)
- Wait 5 seconds
- Remove the bridge (e. g. remove the paper clip)
- Program the SICOLOG with TEMES

At the moment, the slow sample rate must not be longer than 5 seconds, otherwise a 32-bit-overflow can occur for averaging of analog input signals. (The sampling rate of raw analog input signals is fixed in TEMES to 10000 Hz at the moment: the raw input signals are left-shifted to obtain a 16-bit-value and then accumulated inside a 32-bit-value.)

5.6.2 Additional TEMES Parameters

Additional TEMES parameters are defined in the device comment 42 between the one-off *<DPF>* and *</DPF>* tags, and into the *[ExParameters]* section:

<DPF>
[ExParameters]
Additional TEMES parameters
</DPF>

5.6.2.1 Device Signals

The device signals are already defined by the device and can be made accessible by TEMES constants 71.

Please mind that the comment of a device signal's TEMES constant must include:

[Parameters]
DeviceAddr=*address*

Where *address* is entered as decimal value and belongs to the device address.

5.6.2.1.1 Unsigned 8-bit Signals

Name	Address	Description
Marker	8	Default address for a marker.
RecStatus8	12	Recording status (synchronized with the fast sample rate — only available if there is at least one digital output used).
TimeSync	13	If 0, the time is not synchronized with GPS, otherwise it is synchronized each time when the *TimeSync* signal changes.

5.6.2.1.2 Unsigned 16-bit Signals

Name	Address	Description
CounterFast	8194	Counter of the fast cycle.
CounterSlow	8195	Counter of the slow cycle.
SupplyVoltage	8214	Supply voltage in 0.51575091 mV/bit. Assignment example: 0 V ≡ 0 bit, 0.00051575091 V ≡ 1 bit.
CounterCalc	8229	Counter of the calculation cycle.

Name	Ad-dress	Description
CpuLoad	8231	CPU load in percent.
Buttons	8235	& 1 – B4 & 2 – B2 (stop button) & 4 – B3 & 8 – B1 (rec button) & 16 – Slider 1 & 32 – L4 LED & 64 – REC LED & 128 – USB LED & 256 – Slider 2
RecStatus	8236	& 1 – Recording & 2 – Recording condition true
CounterTick	8237	0.01 s wrap-around counter
FileNumber	8238	Current *DATA9999.TSD* file number.
CounterDiskBuf-ferError	8239	Number of measurement exceptions.
SampleNumber	8240	Recorded sample number.
DiskBufferUsage	8243	Disk buffer usage in sectors (with 512 bytes per sector).
Counter1ms	8248	Counter in milliseconds (if enabled)

5.6.2.1.3 Unsigned 32-bit Signals

Name	Address	Description
UnixTime	16384	UTC time in seconds since 1970-01-01.
SampleNum-berLow	16385	Low DWORD of the recorded sample number.
SampleNum-berHigh	16386	High DWORD of the recorded sample number.
FilePos	16387	Current *DATA9999.TSD* file position.
CounterDiskBuf-ferError32	16389	Number of measurement exceptions.
ZeroTime	16390	Time stamp of power-on event (= seconds since 1970-01-01), or of the moment when the real time clock has been set.
RunningTime	16391	Time in seconds since power-on event, or since the moment when the real time clock has been set.
CounterTick32	16392	0.01 s wrap-around counter.
CounterFast32	16393	Counter of the fast cycle.
CounterSlow32	16394	Counter of the slow cycle.

Name	Address	Description
CounterCalc32	16395	Counter of the calculation cycle.
StopDelay	16398	Current value of the stop delay counter in 0.01 s starting from 0 s.
Counter1ms32	16400	Counter in milliseconds (if enabled)

5.6.2.1.4 Signed 32-bit Signals

Name	Ad-dress	Description
TimeZone	20482	Timezone in seconds.

5.6.2.2 Byte Code

The SICOLOG implements calculated signals with interpreted byte code. If a function cannot be implemented with TEMES calculations, then it might possibly be implemented with byte code instead. A resettable maximal value is such a function which will be realized with byte code in the following:

The interface for the byte code to TEMES is done via calculated signals which are assigned to fixed addresses:

- The output signal: Constant definition with signal address $a000_{16}$. Following calculated signal comment is required:

  [Parameters]
  DeviceAddr=40960

- The input signal: Linear transform of the input signal with signal address $a001_{16}$. Following calculated signal comment is required:

  [Parameters]
  DeviceAddr=40961

- The test signal (on which the output signal should be reseted): Equality operation of the test signal with signal address $a002_{16}$. Following calculated signal comment is required:

  [Parameters]
  DeviceAddr=40962

The function is realized in C as follows:


```
extern int32_t a000, a001, a002;
if (a002 != 0) {
    a000 = 0;
}
else {
    if (a000 < a001) a000 = a001;
}
```

The function is realized in byte code as follows:

```
da 02 a0 00 00   PushS4 $a002
22               IsNotZero
0e 0c            JumpShortIfZero 12
82 00 00 00 00   Push4 0
ea 00 a0 00 00   PopS4 $a000
0c 17            JumpShort 23
da 00 a0 00 00   PushS4 $a000
da 01 a0 00 00   PushS4 $a001
24               IsLessThan
0e 0a            JumpShortIfZero 10
da 01 a0 00 00   PushS4 $a001
ea 00 a0 00 00   PopS4 $a000
```

Following additional TEMES parameters are required:

CodeCount = 3
Code0 = da 02 a0 00 00   22   0e 0c   82 00 00 00 00   ea 00 a0 00 00   0c 17
Code1 = da 00 a0 00 00   da 01 a0 00 00   24   0e 0a   da 01 a0 00 00
Code2 = ea 00 a0 00 00

5.6.2.3 Byte Code (2nd Example)

Another example is data type conversion from a 32-bit unsigned variable with an offset to a 32-bit signed variable:

The interface for the byte code to TEMES is done via calculated signals which are assigned to fixed addresses:

- The input signal: Linear transform of the input signal with signal address $c000_{16}$. Following calculated signal comment is required:

  [Parameters]
  DeviceAddr=40960

- The output signal: Constant definition with signal address $d000_{16}$. Following calculated signal comment is required:

[Parameters]
DeviceAddr=53248

The function is realized in C as follows:

```
extern uint32_t c000;
extern int32_t d000;
uint32_t offset = 0x80000000;
d000 = c000 - offset; /* = c000 + (~offset + 1) */
```

The function is realized in byte code as follows:

```
da 00 c0 00 00   PushS4 $c000
82 00 00 00 80   Push4 $80000000
54               Add
ea 00 d0 00 00   PopS4 $d000
```

Following additional TEMES parameters are required:

CodeCount=4
Code0 = da 00 c0 00 00
Code1 = 82 00 00 00 80
Code2 = 54
Code3 = ea 00 d0 00 00

5.6.2.4 CAN Sample Point

The CAN sample point can be entered with

Can0SamplePoint = 80
Can1SamplePoint = 80

where *Can0SamplePoint* refers to the 1st CAN, and where *Can1SamplePoint* refers to the 2nd CAN. The value is given in percent, and its default value is 80.

For SICOLOG/SICO3/USBDL1:
Required versions: up to TEMES V1.0.70 and from TEMES V1.0.76 onwards
(Note, that the CAN/CAN_FD sample point is preferably entered via the CAN editor from TEMES V1.0.71 onwards. The definition via the device comment is only for backward compatibility.)

5.6.2.5 CAN Listen only Mode

The CAN node is set into listen only mode with

Can0ListenOnlyMode=1
Can1ListenOnlyMode=1

where *Can0ListenOnlyMode* refers to the 1st CAN, and where *Can1ListenOnlyMode* refers to the 2nd CAN.

Minimum required versions: SICOLOG V1.0.7 / TEMES V1.0.52

For SICOLOG/SICO3/USBDL1:
Required versions: TEMES V1.0.52 to TEMES V1.0.70 and from TEMES V1.0.76 onwards
(Note, that the CAN/CAN_FD listen only mode is preferably selected via the CAN editor from TEMES V1.0.71 onwards. The definition via the device comment is only for backward compatibility.)

5.6.2.6 CAN Scan Mode (Definition)

The CAN node is set into scan mode with

Can0ScanMode=1
Can1ScanMode=1
Can2ScanMode=1
Can3ScanMode=1

where *Can0ScanMode* refers to the 1st CAN, and where *Can1ScanMode* refers to the 2nd CAN, and where *Can2ScanMode* refers to the 1st CAN FD, and where *CAN3ScanMode* refers to the 2nd CAN FD.

Note, that the corresponding CAN description file is selected as *library* for the communication settings (see Parameter Tree Node: CAN 58).

Please also see also CAN Scan Mode (Overview) 153.

Minimum required versions: SICOLOG V1.0.34 / TEMES V1.0.78 / TSD.EXE V1.09

5.6.2.7 Additional Analog Signal Averaging

The analog signals can be further smoothed by an additional 4-fold averaging (for up to 5 signals plus up to one AIN1-AIN8 signal), or by an additional 2-fold averaging (for up to 14 signals) with the definition of

AIAvg = 1

Minimum required versions: SICOLOG V1.0.5 / TEMES V1.0.48

5.6.2.8 Support for F6 Protocol (Output)

The input signals can be mapped to addresses of the F6 protocol with the following definition:

F6Map=SignalName#1,SignalName#2,SignalName#3,...

where *SignalName#1* is mapped to 80_{16}, *SignalName#2* is mapped to 81_{16} and *SignalName#3* is mapped to 82_{16}, and so on.

Minimum required versions: SICOLOG V1.0.4 / TEMES V1.0.46

5.6.2.9 Support for F6 Protocol (Input)

Up to 16 serial input signals via the f6 protocol can be defined. For this purpose the slider on the SICOLOG's bottom must be switched to RS232, and the following additional TEMES parameters are required:

SIBaudrate=57600
SITimeout=500
SIMode=4
SICount=3
SI0=0x80,SignalName#1,2
SI1=0x81,SignalName#2
SI2=0x82,SignalName#3

where

SIBaudrate: baud rate in bps;
SITimeout: timeout in msec;
SIMode:
  1 - f6 protocol,
  2 - f6 protocol with zero signal on timeout,
  3 - f6 protocol with checksum,
  4 - f6 protocol with checksum and zero signal on timeout or checksum error;
SICount: Number of the following signal definitions;
SI<Number>: request byte (with optional *0x* hex prefix), signal name of an already defined TEMES calculation constant, optionally signal size in bytes.

Minimum required versions: SICOLOG V1.0.4 / TEMES V1.0.46

5.6.2.10 OBD Rate Ratio

The OBD signals can be delayed by awaiting the *n*-th cycle of the calculation cycle:

OBDRateRatio = 1

Minimum required versions: SICOLOG V1.0.25 / TEMES V1.0.66

5.6.2.11 OBD Repetitions

The OBD detection phase can be shortened by reducing the number of protocol retries (1-6):

OBDRepetitions = 5

Minimum required versions: SICOLOG V1.0.25 / TEMES V1.0.66

5.6.2.12 OBD Timeout

The OBD signals can be set to bit value 0 on a timeout:

OBDZeroOnTimeout = 1

Minimum required versions: SICOLOG V1.0.31 / TEMES V1.0.74

5.6.2.13 Floating Point Linear Operation

The operation Op04 (linear transform) 67 is implemented with 32-bit floating point operations with:

FloatLin = 1

Minimum required versions: SICOLOG V1.0.25 / TEMES V1.0.66

5.6.2.14 User LED definition

The user LED L4 can be configured with

LedTicks = 200
LedMax = 5
LedGap = 6

where *LedTicks* is the time period in milliseconds for the on duration (and also for the off duration), *LedMax* is the highest number used for blinking, and *LedGap* is the gap in LedTicks between two adjacent output values.

Minimum required versions: SICOLOG V1.0.29 / TEMES V1.0.72

5.6.2.15 GPS Parameters

Following GPS parameters can be modified:

Parameter	**Default value** unit	**Description**
GpsSpdMsg	0	0: When the GGA timestamp is used, process NMEA sentence VTG after a received

		GGA message. Otherwise process VTG message immediately. 1: When the GGA timestamp is used, process NMEA sentence VTG immediately. Otherwise process VTG message after a received GGA message.
GpsSpdMsgRate	0 s	Message rate hint in s (with a resolution of 1 ms).
GpsSpdMsgAlignment	0 s	Message rate alignment in s with a resolution of 1 ms. Note that the time stamp of the GGA sentence is taken instead if both *GpsSpdMsgRate* and *GpsSpdMsgAlignment* equal 0 s. In this case, the following must be given: • There is exactly one VTG message between two GGA messages, and exactly one GGA message between two VTG messages. • The VTG and GGA messages must be refreshed with the same constant rate. • The timestamp of the GGA message must be different from one GGA message to the next GGA message, and must reflect the refresh rate. • The refresh rate must be an integer multiple of 0.01 s
GpsSpdMsgCompareTicks	3	This value is only valid for non-zero values, and if the GGA timestamp is taken for estimating the duration to the previous speed message. It represents the max. allowed sample number offset ratio between the GGA timestamp and real measured time. Minimum required versions: SICOLOG V1.0.42 / TEMES V1.0.88
GpsSpdWindowWidth	0.42 s	Window size for acceleration measurements in s (The internal moving average buffer size is limited to 51 samples).
GpsSpdMinSpeed	0.5 kph	Minimal speed of the *GpsSpeed* signal in kph. A speed of 0 kph is assumed for lower speeds.
GpsSpdMaxAcc	19.62 m/s^2	Limit for the magnitude of the *GpsSpeed* acceleration in m/s^2, and deactivated if 0 m/s^2.
GpsSpdMinSat	3	Minimal visible satellites for a valid *GpsSpeed* signal.
GpsSpdMaxHdop	9.8	Maximal HDOP value for a valid *GpsSpeed* signal.

GpsSpdMinTrackSpeed	2 kph	Minimal speed for the *GpsTrack* signal in kph. The previous value is taken if *GpsSpeed* is below this threshold.
GpsSpdMinRadius	0.1 m	Minimal radius in m
GpsSpdMaxRadius	999.9 m	Maximal radius in m
~~GpsSpdRadiusMinDifferenceAngle~~	~~0.5 °~~	~~Limit for the magnitude of the difference angle to calculate the radius.~~
GpsPosMsg	0	0: Process NMEA sentence GGA
GpsPosMinSat	3	Minimal visible satellites for a valid *GpsLatitude* and *GpsLongitude* signal.
GpsPosMaxHdop	9.8	Maximal HDOP value for a valid *GpsLatitude* and *GpsLongitude* signal.
GpsHgtMsg	0	0: Process NMEA sentence GGA
GpsHgtDistanceDelta	50 m	Distance delta for the measurement of the distance slope in m.
GpsDelay	0.12 s	General delay of the GPS receiver
GpsNoSignalDelay	0	0: Set the default signal delay to the corresponding GPS delay. 1: Set the default signal delay to zero.

Following signal delays have been measured:

Signal Description	**Signal Delay**
GPS16G via SICOLOG: interpolated GPS speed	180 ms
GPS16G via SICOLOG: interpolated GPS distance	150 ms
GPS16G via SICOLOG: autocalc GPS acceleration	150 ms
HS16G via CAN: interpolated GPS speed	150 ms

Minimum required versions: SICOLOG V1.0.31 / TEMES V1.0.74

5.6.2.16 Record Start Delay

The start of the recording with external recording control can be delayed by

RecStartDelay = 2

where *RecStartDelay* is the time period in seconds for the delay between reading the parameter set and measurement start.

Minimum required versions: SICOLOG V1.0.36 / TEMES V1.0.81

5.6.2.17 Record Synchronization Channel

The recording can be made synchronously depending on a master device with

RecSyncChan = 1

where *RecSyncChan* is the digital input channel number (1-5) of the record synchronization.

Minimum required versions: SICOLOG V1.0.36 / TEMES V1.0.81

5.6.2.18 RTC Synchronization

The RTC (= real time clock) can be synchronized with the ZDA sentence of a GPS receiver:

RtcSync = 1
RtcSyncTimeWindow = 0
RtcSyncPeriod = 86400

where *RtcSync* enables the synchronization if not null, and *RtcSyncTimeWindow* is the allowed absolute time difference between RTC and GPS time in seconds to skip the synchronization. The *TimeSync* 139 signal can be used to observe the synchronization. Note that a synchronization will only occur at most once after a power cycle (if *RtcSyncPeriod* is 0).

Note, that a value of *RtcSync* = 2 also waits until a GPS fix value exists before enabling synchronization. This property works for a minimum version SICOLOG V1.42.

RtcSyncPeriod is only available for at least SICOLOG V1.43. It is the duration in seconds after a time sync, or after loading a new configuration, when the synchronization is repeated. *RtcSyncPeriod*=0 turns the repetition off. The *TimeSync* signal will be incremented each time a synchronization was successful and skip an increment to 0 (and use 1 instead).

Minimum required versions: SICOLOG V1.0.37 / TEMES V1.0.85

5.6.2.19 Frequency Divisor

Up to two used frequency input signals can be set into frequency divider mode or into TDC (= top dead centre) detection mode with

DigitalInEx0=*timer*,*mode*,*port*,*logic*,*teeth*,*duration*,*scaleMin*,*scaleMax*,*delay*,*scaleFreq*

DigitalInEx1=*timer*,*mode*,*port*,*logic*,*teeth*,*duration*,*scaleMin*,*scaleMax*,*delay*,*scaleFreq*
DigitalInEx2=*timer*,*mode*,*port*,*logic*,*teeth*,*duration*,*scaleMin*,*scaleMax*,*delay*,*scaleFreq*
DigitalInEx3=*timer*,*mode*,*port*,*logic*,*teeth*,*duration*,*scaleMin*,*scaleMax*,*delay*,*scaleFreq*
DigitalInEx4=*timer*,*mode*,*port*,*logic*,*teeth*,*duration*,*scaleMin*,*scaleMax*,*delay*,*scaleFreq*

where

DigitalInEx0 refers to digital input DI1 (*DigitalInEx1* refers to DI2, *DigitalInEx2* refers to DI3, …)

timer refers to
0 – default timer resolution
1 – timer resolution of 120 MHz
2 – timer resolution of 120/2 MHz (= 60 MHz)
3 – timer resolution of 120/4 MHz (= 30 MHz)
4 – timer resolution of 120/8 MHz (= 15 MHz)
5 – timer resolution of 120/16 MHz (= 7.5 MHz)
6 – timer resolution of 120/32 MHz (=3.75 MHz)
7 – timer resolution of 120/64 MHz (= 1.875 MHz)
8 – timer resolution of 120/256 MHz (= 468.75 kHz)
9 – timer resolution of 120/1024 MHz (= 117.1875 kHz)

mode refers to
0 – frequency signal
1 – frequency divider
2 – TDC detection

port refers to
1 – Digital output DO1
2 – Digital output DO2
3 – Digital output DO3
4 – Digital output DO4
5 – Digital output DO5
8 – SCL of LIO (SICO3/SICO3M only)
9 – SDA of LIO (SICO3/SICO3M only)
Note, that a LIO port has the following hardware restriction: *R*=4.7 kΩ against 5 V.

logic refers to
0 – positive voltage level
1 – negative voltage level

teeth refers to the pulses per revolution

duration refers to the pulse duration in pulses

scaleMin refers to the comparison factor for TDC detection (= current time period ≥ *scaleMin* • previous time period)

scaleMax refers to the comparison factor for end of TDC detection

delay refers to the start delay of TDC detection in pulses (Note, that a value different from 0 is experimental only).

scaleFreq refers to the pulse period correction divider for the TDC pulse of a TDC detection measurement. This correction makes it possible to measure the frequency from tooth-to-tooth without disrupting the signal on a TDC. E.g. *scaleFreq*=3 for two missing teeth of a wheel. *scaleFreq* is activated if its value is different from 0. Note, that the pulse measurement skips the *f_predicted*=1/*T_current* correction part for missing pulses if *scaleFreq* is activated. (An activated *scaleFreq* requires at least SICOLOG V1.0.42 and TEMES V1.0.88).

Note, that unused frequency input signals must be at least added to a signal group 81.

Minimum required versions: SICOLOG V1.0.41 / TEMES V1.0.87

5.6.2.20 Mirror the User LED

The status of the user LED can be mirrored onto a digital port:

LedMirrorPort = 1
LedMirrorInverted = 0

where *LedMirrorPort* represents the otherwise unused digital channel (1...5). The logic level can be inverted with *LedMirrorInverted* equals 1.

Minimum required versions: SICOLOG V1.0.42 / TEMES V1.0.88

5.6.2.21 Extended Logging of Start and Stop Events

The extended logging of start and stop events can be activated with

LogMode = 1

where *LogMode* represents an integer number with the following meaning:

0: Disabled (default)

1: Extended logging of the first start events after a new configuration has been loaded
2: Extended logging of all start events
3: Extended logging of all start and stop events (Stop events are not reported by TEMES V1.0.91)

The log is enclosed in the measurement variable *Measurement.M1.info*. The extended logging of start events includes the fast sample index, the device number, the device version number, the timestamp, the timezone, and the supplying voltage of the SICOLOG. The extended logging of stop events includes the fast sample index, the timestamp, the timezone, and the supplying voltage of the SICOLOG.

ATTENTION: When using a *LogMode* greater than 0, then at least TEMES V1.0.91 is required to read the measurement.

Minimum required versions: SICOLOG V1.0.44 / TEMES V1.0.92 / TSD.EXE V1.13

5.6.2.22 No Checksum Verification

The verification of the checksum of the SICOLOG configuration can be disabled with

NoChecksum = 1

The checksum verification is enabled by default. If enabled, the checksum will be verified each time a new configuration is loaded by SICOLOG V1.0.45 or better.

Minimum required versions: SICOLOG V1.0.45 / TEMES V1.0.93

5.6.2.23 Base Frequency Divisor

The base frequency (which is 10 kHz) is divided by

BaseFrequencyDivisor = 1

This is a necessary adjustment for slow measurements to still allow averaging of the analog inputs. The averaging, which is limited by 65535 samples, would otherwise overflow. This entry has an impact of how fast the samples are done for averaging.

Note that this entry will automatically be set to a decent value by TEMES V1.0.94 or better if required.

Minimum required versions: TEMES V1.0.94

5.6.2.24 RawMask

The entry

RawMask = 1

is a bit mask to disable the averaging of the specified analog inputs. The maximal value is 65535 which disables the averaging of all 16 analog input signals.

Channel	Value
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128
9	256
10	512
11	1024
12	2048
13	4096
14	8192
15	16384
16	32768

Minimum required versions: TEMES V1.0.94

5.6.3 CAN Scan Mode (Overview)

In CAN scan mode, all CAN messages are logged (except remote frames and error frames, and assuming the SICOLOG is fast enough). However, CAN signals of a scanned CAN are not updated (and thus input CAN signals always yield 0, and output CAN signals are never output). It is possible to also make a standard recording at the same time. Then, saving the measurement with *TEMES* will only store the samples and ignore the CAN samples, and *TSD.EXE* must first be called at the command line in order to extract the recorded CAN samples. E. g. with

```
C:\>TSD s m:
```

(or with

```
C:\>TSD s m: filename.log
```

to also redirect the output to file *filename.log*) which yields

```
2018-09-02 10:31:54
2018-09-02 10:31:56.953910 1          7df  3 1b 1c 1d
2018-09-02 10:31:57.463868 1          7df  3 1b 1c 1d
2018-09-02 10:31:57.983868 1          7df  3 1b 1c 1d
2018-09-02 10:31:58.503928 1     18db33f1  2 2a 2b
2018-09-02 10:31:59.023906 1     18db33f1  2 2a 2b
2018-09-02 10:31:59.543906 1     18db33f1  2 2a 2b
00000000011111111112222222222333333333344444444445555
12345678901234567890123456789012345678901234567890123

Column  1-10: Date
Column 12-26: Time (start marker or sync time without fracti-
on)
Column    28: CAN number (1-4)
Column    30: 'F' if message is a CAN FD message
Column    31: 'B' if message uses bit rate switching
Column    32: 'R' is reserved for remote frames
Column 34-41: Message identifier (hex)
Column 43-44: Message size (decimal)
Column 46-  : Message data (hex)
```

The timestamps have an accuracy of about 1 s (because they are derived from the real time clock), and only the relative timestamps (= the differences of absolute timestamps) are exact (for the corresponding CAN number) within a contiguous section which is delimited by start markers.

Following warning or error lines can appear:

```
W: Missing measurement data (USB stick access too slow)!
W: Missing messages for CAN #1!
W: Missing messages for CAN #2!
W: Missing messages for CAN #3!
W: Missing messages for CAN #4!
E: Unknown data = stream data (hex)
E: Wrong data = stream data (hex)
```

And one warning line which can mostly be ignored for the CAN samples:

```
W: Missing measurement data (CPU too slow)!
```

However, the timestamp might be out of synchronization when the cycle processing was delayed by at least 30 ms (for CAN1 or CAN2) or 35 minutes (for CAN3 or CAN4), or when a start marker happens to be in the missing measurement data. Also, there is a blind spot after *missing messages* for a CAN that further missing messages will be ignored for the next following 4 CAN messages (for CAN1 or CAN2) or for the next following 26 CAN messages (for CAN3 or CAN4).

The ASCII file *filename.log* can be converted to a TEMES measurement *filename.tms* with the command line program *LOG2TMS.EXE*, e. g. with

```
C:\>LOG2TMS -1 library filename
```

which creates signals for all defined CAN signals of the library *library.can* with the adjustable default sample rate of 10 ms and only the samples of CAN number 1.

Please see also CAN Scan Mode (Definition) 144.

Minimum required version: TSD.EXE V1.09

5.6.3.1 CAN (CAN number 1 or 2)

The fast sample rate should not be slower than 10 ms (due to the timestamp synchronization).

The CAN messages are sampled during the fast recording cycle. Four CAN messages are buffered. A fast sample frequency of 10 kHz (= sample rate of 0.1 ms) should allow the recording of all CAN messages of a fully loaded CAN at 1 Mbit/s (assuming that both the CPU and the USB stick access is fast enough). If there are no 0-data-byte-messages, then the sample frequency can also be reduced to 5 kHz (= sample rate of 0.2 ms). If additionally the CAN bit rate is 500 kbit/s, then the sample frequency can be reduced to 2.5 kHz (= sample rate of 0.4 ms). If additionally the CAN messages are with 8 data bytes only, then the sample frequency can be reduced to 1.2 kHz (= sample rate of 0.8 ms).

CAN message size ≥ 47 bits (for any CAN message)
CAN message size ≥ 55 bits (for CAN messages with at least 1 data byte)
CAN message size ≥ 111 bits (for CAN messages with 8 data bytes only)

For maximal CAN bit rate of 1 Mbit/s:
$f_{\text{CAN message}}$ < 22 kHz (for any CAN message)
$f_{\text{CAN message}}$ < 19 kHz (for CAN messages with at least 1 data byte)
$f_{\text{CAN message}}$ < 9.1 kHz (for CAN messages with 8 data bytes only)

For CAN bit rate of 500 kbit/s:
$f_{\text{CAN message}}$ < 11 kHz (for any CAN message)
$f_{\text{CAN message}}$ < 9.1 kHz (for CAN messages with at least 1 data byte)
$f_{\text{CAN message}}$ < 4.6 kHz (for CAN messages with 8 data bytes only)

Recommendation:
Sample rate $T \leq 4 / f_{\text{CAN message}}$

Sample rate *T*	CAN bit rate: 500 kbit/s	CAN bit rate: 1 Mbit/s	CAN message type

T ≤	0.3 ms	0.1 ms	Arbitrary CAN message (incl. 0-byte-messages)
T ≤	0.4 ms	0.2 ms	CAN messages with at least 1 byte
T ≤	0.8 ms	0.4 ms	CAN messages with only 8 bytes

5.6.3.2 CAN FD (CAN number 3 or 4)

The fast sample rate should not be slower than 15 minutes (due to the time-stamp synchronization).

The CAN FD messages are sampled during the fast recording cycle. 26 CAN FD messages are buffered.

Recommendation: Sample rate $T \leq 26 / f_{\text{CAN message}}$

Sample rate *T*	CAN bit rate: 500 kbit/s	CAN bit rate: 1 Mbit/s	CAN message type
T ≤	2.4 ms	1.2 ms	Arbitrary CAN message (incl. 0-byte-messages)

5.6.4 Record Synchronization of Multiple Devices

The signals of a recording can be divided onto several devices.

To merge the signals into one single TEMES file, first the TEMES file of the master measurement must be opened. Then the node *Measurement Blocks* has to be selected. And then, the menu item *File → Import* has to be chosen, and the TEMES file of the slave measurement has to be selected.

A synchronized measurement with several devices has following restrictions:

- Only one measurement block should be recorded (namely the measurement block *Fast*). However, the slow signals can be assigned to one or more slave units.
- One digital channel per device has to be reserved for the synchronization: An output channel for the master unit, and an input channel for the slave unit(s).
- The number of slave devices is limited to 5 (with resistor R_M = 10 kΩ) or to 10 (with resistor R_M = 5 kΩ).
- The recording should only start after all slave unit(s) are ready for recording.

5.6.4.1 Setup of the Master Unit

- The rate ratio [40] of the sample rate should be set to 1
- The recording start delay [148] can optionally be set to a non-zero value with the following device comment [42]:

```
<DPF>
[ExParameters]
RecStartDelay=2
</DPF>
```

(When the recording start delay is non-zero, then also an external recording control [79] has to be chosen.)

- The signal *rec* has to be defined as a constant with device address 12. This is done by pressing the *Ins*-Button on the node *Calculation* [66]. And then, by defining the signal name to *rec* and the comment [71] to:

```
[Parameters]
DeviceAddr=12
```

- The signal *rec* has to be assigned to a digital output [72]:

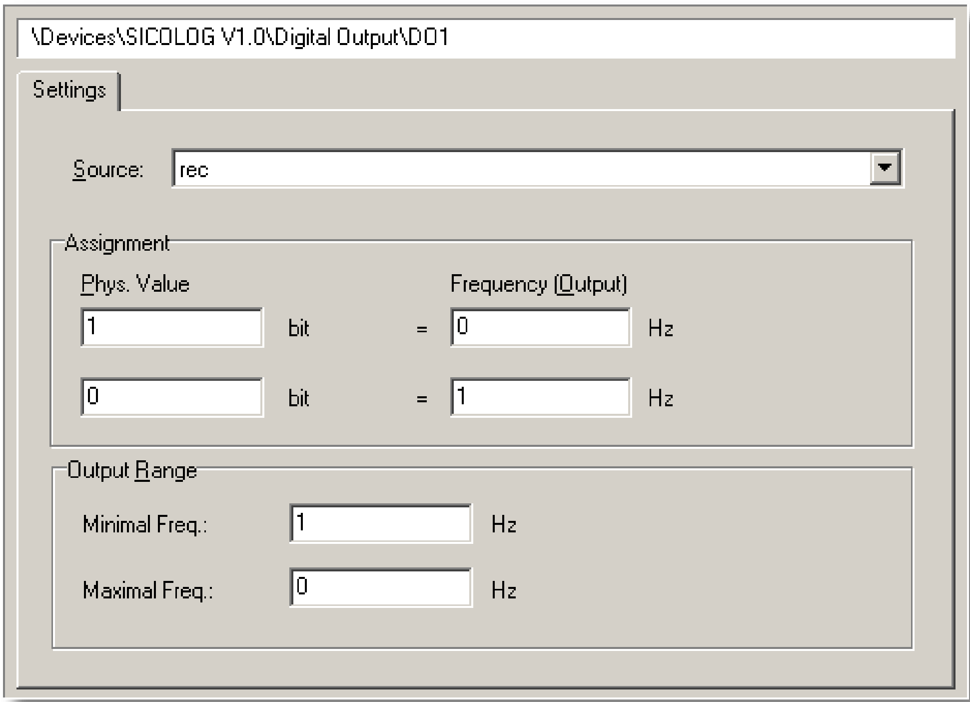


Figure 5-4a: Settings of the digital output for synchronization.

- A measurement should be terminated with the REC slider into off position.
- The digital out cable should be assigned as follows:

Pin	Assignment

1	not connected
2	bridged with master pin 3 — connected with slave pin 2
3	bridged with master pin 2, and bridged with master pin 4 via R_M (= 5 kΩ or 10 kΩ)
4	bridged with master pin 3 via R_M (= 5 kΩ or 10 kΩ) — connected with slave pin 3

5.6.4.2 Setup of the Slave Unit(s)

- The rate ratio[40] of the sample rate should be set to 1.
- The sample rate[40] should optionally be set to an integer multiple of the master unit.
- Following entry has to be added to the device comment[42] (see also Record Synchronization Channel[149]):

```
<DPF>
[ExParameters]
RecSyncChan=1
</DPF>
```

- The recording control[79] should be set to external recording, and the start/stop buttons should be disabled.
- The REC slider should be turned on to activate the recording readiness of the slave unit.
- The digital in cable should be assigned as follows:

Pin	Assignment
1	not connected
2	connected with master pin 2
3	connected with master pin 4
4	not connected

5.6.5 File Name Generation

If the file \TELLERT\SICOLOG\CONFIG.INI exists onto the SICOLOG USB stick, then it is treated as profile file[186] with section *Settings* and the key *Dest*. The key *Dest* defines the default name for the TEMES output filename. If it is a relative filename, then it will be prefixed with the TEMES data path. The folders of the filename are created if necessary. Following placeholders will be replaced accordingly:

<year>	2-digit century (00...99)
<month>	2-digit month (01...12)
<day>	2-digit day (01...31)
<hour>	2-digit hours (00...23)
<min>	2-digit minutes (00...59)
<sec>	2-digit seconds (00...59)

<num>	Empty string, or if filename already exists an incremented number with a prefixed underscore.

Example:

\TELLERT\SICOLOG\CONFIG.INI:

```
[Settings]
Dest=Measurements\A_20<year>-<month>-<day><num>.tms
```

5.6.6 Odometer

The odometer is calculated by updating/changing the offset of the GPS AutoCalc signal *GPS_Distance*.

The odometer settings can be maintained automatically by defining variable *Odometer* in section *Settings* of the profile file CONFIG.INI 158. The respective variables are as follows:

Odometer = Odometer reading in kilometers of measurement begin
OdometerEnd = Odometer reading in kilometers of measurement end. (This value becomes the new *Odometer* when the measurement is deleted by TEMES.)
OdometerBackup = previous content of variable *Odometer*. (This value becomes the new *Odometer* when the measurement is restored within TEMES.)

5.6.7 Device Number

The SICOLOG firmware V1.0.44 or better automatically stores the device number on a freshly configured USB stick together with the measurement data. Once the device number is written onto the USB stick, it is frozen until a new configuration is written onto the USB stick. TEMES updates the variable *Measurement.M1.deviceNo* to identify the corresponding recording device.

5.6.8 Repair of a Measurement

This section describes an example of how to repair a measurement. The necessary program to repair a measurement onto a corrupted USB stick is *TSD.EXE* which can be downloaded from http://www.tellert.de/?product=sicolog. The steps assume that the configuration TEMES file is stored somewhere on the computer.

5.6.8.1 Create a Backup Copy of the Corrupted USB Stick

First, an already (possibly) erased measurement needs to be unerased from the one inserted corrupted USB stick with

TSD u

Then, the backup copy is created with

TSD c * d:\data.bin

where *d:\data.bin* is the destination target file for the backup copy.

5.6.8.2 Create a New USB stick for SICOLOG

Unplug the corrupted USB stick, and insert an unused SICOLOG USB stick into the PC. Alternatively, create a new SICOLOG USB stick within TEMES: *Tools → Additional Tools → Hardware → Initialize USB stick (SICOLOG/USBDL1)*. Open the same TEMES configuration as the corrupted measurement and save the configuration onto the USB stick (TEMES: *Hardware → Write to Mass Storage*...).

5.6.8.3 Copy the file \TELLERT\SICOLOG\CONFIG.TSF

Copy the file *\TELLERT\SICOLOG\CONFIG.TSF* from the newly created SICOLOG USB stick (which is already configured the same as the corrupted measurement) onto the corrupted USB stick.

5.6.8.4 Repair the USB Stick Header

Make sure that only the corrupted USB stick is inserted into the PC, then run TSD.EXE with the following command:

TSD h * -f

This command writes a new header into the file *DATA0000.TSD* with the new identification of the configuration. Note that if you have run the SICOLOG in CAN scan mode, then you need also the CAN baud rates as additional arguments (e.g. CAN#1 at 500 kBd and CAN#2 at 1 MBd):

TSD h * -f 500000 1000000

5.6.8.5 Try to Read the Measurement Again

The final step is to run TEMES and try to read the measurement again (TEMES: *Hardware → Save Measurement from Mass Storage*...). But this time, hopefully, without the problems as during the 1st time.

5.6.8.6 Restore the Corrupted USB Stick

Optionally, create a new SICOLOG USB stick within TEMES: *Tools → Additional Tools → Hardware → Initialize USB stick (SICOLOG/USBDL1)*. Then, make sure that only the new SICOLOG USB stick is inserted into the PC. The corrupted USB stick can finally be replicated with:

TSD i

and then with

TSD D * d:\data.bin

5.6.8.7 Command window example

This subsection summarizes the repair steps. (Windows 7 is used here.):

1. Create a folder onto the desktop named *SICOLOG Recovery* and copy the file *TSD.EXE* into this folder.

2. Press *Windows*+*R* and enter *cmd* *Enter*. A command line window opens with the text *C:\Users\Username*>. Enter *cd* "*Desktop\SICOLOG Recovery*" followed by the *Enter* key.

3. Alternatively, as a replacement for step 2, right-click on the folder *SICOLOG Recovery* while holding the *Shift* key pressed, and select the command *Open command window from here* from the context menu.

4. Plug in the corrupt USB stick and wait until its drive letter appears inside the *Windows Explorer*. Goto the command line window and enter:

   C:\Users\Username\Desktop\SICOLOG Recovery>tsd u *Enter*
   C:\Users\Username\Desktop\SICOLOG Recovery>tsd c * data.bin *Enter*

5. Switch to TEMES.EXE, unplug the corrupt USB stick, plug in and create a new USB stick, and configure it with the measurement configuration.

6. Copy the file *\TELLERT\SICOLOG\CONFIG.TSF* from the new USB stick to the desktop folder *SICOLOG Recovery*.

7. Unplug the new USB stick and plug in the corrupted USB stick, and copy (= overwrite) the file *CONFIG.TSF* from the desktop folder *SICOLOG Recovery* to the folder *\TELLERT\SICOLOG* of the corrupted USB stick.

8. Switch to the command line window and enter (Assuming that the CAN scan mode is not used):

   C:\Users\Username\Desktop\SICOLOG Recovery>tsd h * -f *Enter*
   C:\Users\Username\Desktop\SICOLOG Recovery>exit *Enter*

9. Switch to TEMES.EXE, store the measurement from mass media (= the corrupted USB stick) into a TMS file.

5.7 USBDL1

The properties of the data logger are almost the same as for the SICOLOG 138.

The home page of the USBDL1 is located at http://www.tellert.de/?product=usbdl1.

5.8 USBSER and TEMES 1.0

The home page of the USBSER is located at http://www.tellert.de/?product=usbser.

5.8.1 USBSER Driver Installation

The device driver for USBSER is available either on the CD-ROM in the folder *\Setup\Drivers\USBSER* or *\Drivers\USBSER*, or from the Internet via http://www.tellert.de/?product=usbser. Best practice is to run the automatic FT232R device driver installer before the USBSER is plugged in for the first time.

5.8.2 Windows Device Manager

The Windows Device Manager is started as follows: Choose *System* from the *Control Panel*. Choose tab *Hardware* if it is available, then click on *Device Manager*. (A faster method is to right-click on the *Computer* icon and to choose menu item *Properties*. Further methods are to press the hotkey *Windows*+*Pause*,, or *Windows*+*x*, or to run *devmgmt.msc*).

5.8.3 USBSER Port Settings

The advanced port settings can be modified as follows: Choose the corresponding port in the Windows Device Manager. Right-click on the port and choose menu item *Properties*. Select tab *Port Settings* and then press button *Advanced...*

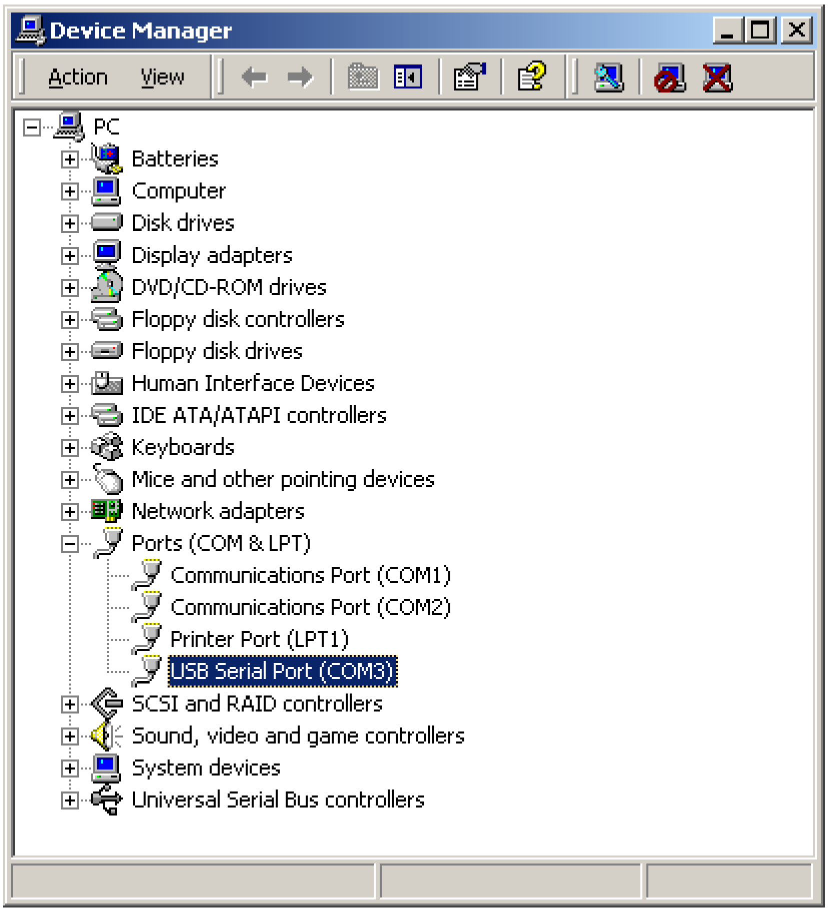


Figure 5-5: Windows Device Manager.

5.8.4 Recommended Settings

All values of the advanced port settings should be set to their corresponding minimum values, and all check boxes should be unchecked.

Appendix

6 Appendix

6.1 Bit Position of CAN Signals (Start Bit)

The start bit represents the least significant bit (lsb) of a signal. Please try to lay your signal into the following templates and look up the corresponding start bit (= the smallest bit number):

Example Signal 1: Intel (= Little Endian), message size = 8 bytes, start bit = 28, bit length = 16

```
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 07 06 05 04 03 02 01 00
Data Byte 2    | 15 14 13 12 11 10 09 08
Data Byte 3    | 23 22 21 20 19 18 17 16
Data Byte 4    | 31 30 29 28 27 26 25 24
Data Byte 5    | 39 38 37 36 35 34 33 32
Data Byte 6    | 47 46 45 44 43 42 41 40
Data Byte 7    | 55 54 53 52 51 50 49 48
Data Byte 8    | 63 62 61 60 59 58 57 56
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================
```

Example Signal 2: Motorola (= Big Endian), message size = 8 bytes, start bit = 28, bit length = 16

```
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 63 62 61 60 59 58 57 56
Data Byte 2    | 55 54 53 52 51 50 49 48
Data Byte 3    | 47 46 45 44 43 42 41 40
Data Byte 4    | 39 38 37 36 35 34 33 32
Data Byte 5    | 31 30 29 28 27 26 25 24
Data Byte 6    | 23 22 21 20 19 18 17 16
Data Byte 7    | 15 14 13 12 11 10 09 08
Data Byte 8    | 07 06 05 04 03 02 01 00
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================
```

```
Message Size: 8 Bytes
Byte Order: Intel (Little Endian)
=======================================
CAN Bus       | Bit Number
              | msb                  lsb
--------------+------------------------
Message ID    |
Control Field |
Data Byte 1   | 07 06 05 04 03 02 01 00
Data Byte 2   | 15 14 13 12 11 10 09 08
Data Byte 3   | 23 22 21 20 19 18 17 16
Data Byte 4   | 31 30 29 28 27 26 25 24
Data Byte 5   | 39 38 37 36 35 34 33 32
Data Byte 6   | 47 46 45 44 43 42 41 40
Data Byte 7   | 55 54 53 52 51 50 49 48
Data Byte 8   | 63 62 61 60 59 58 57 56
CRC Field     |
Ack. Field    |
End of Frame  |
=======================================
```

```
Message size: 8 Bytes
Byte Order: Motorola (Big Endian)
=======================================
CAN Bus       | Bit Number
              | msb                  lsb
--------------+------------------------
Message ID    |
Control Field |
Data Byte 1   | 63 62 61 60 59 58 57 56
Data Byte 2   | 55 54 53 52 51 50 49 48
Data Byte 3   | 47 46 45 44 43 42 41 40
Data Byte 4   | 39 38 37 36 35 34 33 32
Data Byte 5   | 31 30 29 28 27 26 25 24
Data Byte 6   | 23 22 21 20 19 18 17 16
Data Byte 7   | 15 14 13 12 11 10 09 08
Data Byte 8   | 07 06 05 04 03 02 01 00
CRC Field     |
Ack. Field    |
End of Frame  |
=======================================
```

```
Message Size: 7 Bytes
Byte Order: Intel (Little Endian)
=======================================
CAN Bus       | Bit Number
              | msb                 lsb
--------------+------------------------
Message ID    |
Control Field |
Data Byte 1   | 07 06 05 04 03 02 01 00
Data Byte 2   | 15 14 13 12 11 10 09 08
Data Byte 3   | 23 22 21 20 19 18 17 16
Data Byte 4   | 31 30 29 28 27 26 25 24
Data Byte 5   | 39 38 37 36 35 34 33 32
Data Byte 6   | 47 46 45 44 43 42 41 40
Data Byte 7   | 55 54 53 52 51 50 49 48
CRC Field     |
Ack. Field    |
End of Frame  |
=======================================
```

```
Message size: 7 Bytes
Byte Order: Motorola (Big Endian)
=======================================
CAN Bus       | Bit Number
              | msb                 lsb
--------------+------------------------
Message ID    |
Control Field |
Data Byte 1   | 55 54 53 52 51 50 49 48
Data Byte 2   | 47 46 45 44 43 42 41 40
Data Byte 3   | 39 38 37 36 35 34 33 32
Data Byte 4   | 31 30 29 28 27 26 25 24
Data Byte 5   | 23 22 21 20 19 18 17 16
Data Byte 6   | 15 14 13 12 11 10 09 08
Data Byte 7   | 07 06 05 04 03 02 01 00
CRC Field     |
Ack. Field    |
End of Frame  |
=======================================
```

```
Message Size: 6 Bytes
Byte Order: Intel (Little Endian)
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 07 06 05 04 03 02 01 00
Data Byte 2    | 15 14 13 12 11 10 09 08
Data Byte 3    | 23 22 21 20 19 18 17 16
Data Byte 4    | 31 30 29 28 27 26 25 24
Data Byte 5    | 39 38 37 36 35 34 33 32
Data Byte 6    | 47 46 45 44 43 42 41 40
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================

Message size: 6 Bytes
Byte Order: Motorola (Big Endian)
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 47 46 45 44 43 42 41 40
Data Byte 2    | 39 38 37 36 35 34 33 32
Data Byte 3    | 31 30 29 28 27 26 25 24
Data Byte 4    | 23 22 21 20 19 18 17 16
Data Byte 5    | 15 14 13 12 11 10 09 08
Data Byte 6    | 07 06 05 04 03 02 01 00
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================

Message Size: 5 Bytes
Byte Order: Intel (Little Endian)
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 07 06 05 04 03 02 01 00
Data Byte 2    | 15 14 13 12 11 10 09 08
Data Byte 3    | 23 22 21 20 19 18 17 16
Data Byte 4    | 31 30 29 28 27 26 25 24
Data Byte 5    | 39 38 37 36 35 34 33 32
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================
```

```
Message size: 5 Bytes
Byte Order: Motorola (Big Endian)
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 39 38 37 36 35 34 33 32
Data Byte 2    | 31 30 29 28 27 26 25 24
Data Byte 3    | 23 22 21 20 19 18 17 16
Data Byte 4    | 15 14 13 12 11 10 09 08
Data Byte 5    | 07 06 05 04 03 02 01 00
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================

Message Size: 4 Bytes
Byte Order: Intel (Little Endian)
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 07 06 05 04 03 02 01 00
Data Byte 2    | 15 14 13 12 11 10 09 08
Data Byte 3    | 23 22 21 20 19 18 17 16
Data Byte 4    | 31 30 29 28 27 26 25 24
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================

Message size: 4 Bytes
Byte Order: Motorola (Big Endian)
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 31 30 29 28 27 26 25 24
Data Byte 2    | 23 22 21 20 19 18 17 16
Data Byte 3    | 15 14 13 12 11 10 09 08
Data Byte 4    | 07 06 05 04 03 02 01 00
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================
```

```
Message Size: 3 Bytes
Byte Order: Intel (Little Endian)
=======================================
CAN Bus       | Bit Number
              | msb                 lsb
--------------+------------------------
Message ID    |
Control Field |
Data Byte 1   | 07 06 05 04 03 02 01 00
Data Byte 2   | 15 14 13 12 11 10 09 08
Data Byte 3   | 23 22 21 20 19 18 17 16
CRC Field     |
Ack. Field    |
End of Frame  |
=======================================

Message size: 3 Bytes
Byte Order: Motorola (Big Endian)
=======================================
CAN Bus       | Bit Number
              | msb                 lsb
--------------+------------------------
Message ID    |
Control Field |
Data Byte 1   | 23 22 21 20 19 18 17 16
Data Byte 2   | 15 14 13 12 11 10 09 08
Data Byte 3   | 07 06 05 04 03 02 01 00
CRC Field     |
Ack. Field    |
End of Frame  |
=======================================

Message Size: 2 Bytes
Byte Order: Intel (Little Endian)
=======================================
CAN Bus       | Bit Number
              | msb                 lsb
--------------+------------------------
Message ID    |
Control Field |
Data Byte 1   | 07 06 05 04 03 02 01 00
Data Byte 2   | 15 14 13 12 11 10 09 08
CRC Field     |
Ack. Field    |
End of Frame  |
=======================================
```

```
Message size: 2 Bytes
Byte Order: Motorola (Big Endian)
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 15 14 13 12 11 10 09 08
Data Byte 2    | 07 06 05 04 03 02 01 00
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================

Message Size: 1 Byte
Byte Order: Intel (Little Endian)
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 07 06 05 04 03 02 01 00
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================

Message size: 1 Byte
Byte Order: Motorola (Big Endian)
=======================================
CAN Bus        | Bit Number
               | msb                 lsb
---------------+------------------------
Message ID     |
Control Field  |
Data Byte 1    | 07 06 05 04 03 02 01 00
CRC Field      |
Ack. Field     |
End of Frame   |
=======================================
```

6.2 Offline Signal Processing

To automate offline signal processing, TEMES uses an ASCII file (which is called a *Calculation Definition File* in the following to describe the conversion action of a measurement block. Beneath the definition file you may also require the corresponding *Dynamic Link Libraries (.DLL)* and other resources like characteristic curves.

6.2.1 Calculation Definition File

A calculation definition file (.CDF) is a profile file 186 which describes the conversion rules. It looks as follows:

```
[Rules]
Include = <Signals to be included>
Exclude = <Signals to be excluded>
DefCount = <Number of new signals>
Def0 = <Definition data of new signal #0>
Def1 = <Definition data of new signal #1>
Defx = <Definition data of new signal #x>
```

Include: The *Include* entry contains the name of all signals of the input measurement block which are copied directly into the resulting measurement block. Signal names are separated by a comma. Note that the signal position of the new measurement block is taken from the position within this entry. You can use the star symbol '*' as a placeholder to include all signals of the input measurement block.

Exclude: The *Exclude* entry is only necessary, if you use the star symbol within the *include* entry. Signal names which are listed within the *exclude* entry are not copied into the resulting measurement even they are mentioned within the *include* entry.

DefCount: The entry *DefCount* defines the number of signals to be generated. Note that the number of digits should equal those contained in the entry name of the signal definitions (Use Proceeding zeros to adjust the number of digits).

Def*x*: This entry defines the *x*-th signal to be generated. It contains at least two comma separated parameters:

Position	Description
0	Name of the resulting signal
1	Name of the calculation module
2 ... *x*	Parameter 0 ... (*x*-2) of the corresponding calculation module

6.2.2 Calculation Modules

The working horses of offline signal processing are the calculation modules. They generate the floating point values for the new signals. TEMES does a lot work in the background to simplify their structure. Generally, the process is divided into three parts. First, all signals are dequantized by TEMES to convert all signal values into 64 bit floating point values. Second, TEMES calls the corresponding calculation module for each signal to be generated. Finally, TEMES quantizes the resulting signal values back from 64 bit floating point format into that bit representation which is desired by the calculation module. Usually, the desired format is the same as the input signal, but it may differ if the module explicitly requires a better bit representation to preserve signal accuracy. If not specified by the module, the quantization parameters are chosen automatically by TEMES to map the lowest resulting value to the lowest bit value, and to map the highest resulting value to the highest bit value.

Expression	**Description**
Arg	Argument or parameter of a calculation module call. If an argument number is written within brackets the corresponding argument may be omitted and a default value is used instead.
N	Number of samples
T	Sample rate in seconds
k	Loop variable (sample number)
x	Input signal
y	Resulting signal

6.2.3 Calculation Module AVG

The module *AVG* calculates the moving average of the input signal.

Arg	**Description**
0	Name of the input signal *x*
1	Order *N,* or time window width T_{window} when postfixed with "s" ($N = T_{window}/T$)
2	Factor, or 0 (to use the properties of the input signal for the output signal)
3	Physical unit of the output signal
[4]	Bit factor *m* of the output signal quantization
[5]	Offset *t* of the output signal quantization

6.2.4 Calculation Module AVGDIFFT

The module *AVGDIFFT* calculates the 1st derivative of the input signal by using a moving average approach.

Arg	**Description**
0	Name of the input signal *x*

Arg	Description
1	Order *N*, or time window width T_{window} when postfixed with "s" ($N = T_{window}/T$)
2	Factor
3	Physical unit of the output signal
[4]	Bit factor *m* of the output signal quantization
[5]	Offset *t* of the output signal quantization

6.2.5 Calculation Module BCD2DEC

The module *BCD2DEC* decodes a BCD encoded signal into its decimal value: *y*(*x*) = BCD2DEC((*x* XOR *XorMask*) AND *AndMask*, *ErrorValue)*.

Arg	Description
0	Name of the input signal *x*
1	*XorMask*
2	*AndMask*
3	*ErrorValue*
[4]	Min. value of viewing range
[5]	Max. value of viewing range

6.2.6 Calculation Module BIQUAD

The module *BIQUAD* calculates the value $y(n) = a_0 \cdot x(n) + a_1 \cdot x(n - 1) + a_2 \cdot x(n - 2) - b_1 \cdot y(n - 1) - b_2 \cdot y(n - 2)$

Arg	Description
0	Name of the input signal *x*
1	1 (reserved for sample rate ratio)
2	coefficient a_0
3	coefficient a_1
4	coefficient a_2
5	coefficient b_1
6	coefficient b_2

6.2.7 Calculation Module CANSIG

The module *CANSIG* extracts a physical signal out of a bit value.

Arg	Description
0	Name of the input signal *x*
1	reserved
2	reserved
3	reserved
4	Type (0 – Intel unsigned, 1 – Intel signed, 2 – Motorola unsigned, 3 – Motorola signed)
5	Start bit
6	Bit length
7	Factor
8	Offset
9	Unit
10	Min
11	Max
12	ModeType
13	ModeStartBit
14	ModeBitLength
15	ModeId

6.2.8 Calculation Module DEMPX

The module *DEMPX* demultiplexes a multiplexed signal where the last four bits are the multiplex index.

Arg	Description
0	Name of the input signal *x*
1	MpxNo
2	Start value for previous value

6.2.9 Calculation Module DERIVE

The module *DERIVE* calculates the 1st time derivative of the input signal.

Arg	Description
0	Name of the input signal *x*
1	Factor

Arg	Description
2	Physical unit of the output signal
[3]	Bit factor *m* of the output signal quantization
[4]	Offset *t* of the output signal quantization

6.2.10 Calculation Module DIFFT

The module *DIFFT* calculates the first time-derivative of the input signal:

$$y(k) = s \cdot \frac{x(k) - x(k-1)}{T}$$

Arg	Description
0	Name of input signal *x*
1	Factor *s*
2	Physical unit of the output signal
[3]	Bit factor *m* of the output signal quantization
[4]	Offset *t* of the output signal quantization

6.2.11 Calculation Module FIRLP

The module *FIRLP* calculates the output of a FIR low pass filter.

Arg	Description
0	Name of input signal *x*
1	Order *N*
2	Cut-off frequency (either normalized within the range [0; 0.5] or in Hz, when given as negative value.
3	Window: *RECT*, *BARTLETT*, *HANN* or *HANNING*, *HAMMING*, *BLACKMAN*, *BLACKMAN-HARRIS*
4	Time delay, or *AUTO*
5	Minimum axis value
6	Maximal axis value

6.2.12 Calculation Module FREQ

The module *FREQ* replaces missing frequency information with linear interpolated values. This is done by searching the following values of a considered frequency signal point within a time window. The time window is given by the reciprocal value of the current pulse frequency. If a different frequency value is found within the used time window, all values between the considered frequency value and the found one are assumed to be missing and therefore they are replaced by linear interpolation. To take signal dynamics into account, the time window to be used is multiplied with the relative time window *W*.

Arg	Description
0	Name of the input signal x
1	Relative time window W (zero represents infinitiy)
[2]	Time offset in seconds

6.2.13 Calculation Module GPSV

The module *GPSV* filters the GPS speed signal.

Arg	Description
0	Name of input signal *x* (with the unit km/h)
1	Optionally, name of trigger signal (which changes on a new speed value)
2	Refresh rate of the GPS signal in seconds
3	Max. acceleration in (km/h)/s
4	Max. distortion width in seconds

6.2.14 Calculation Module LATLON2M

The module *LATLON2M* calculates the distance of the latitude and longitude signal.

Arg	Description
0	Output type: *LATITUDE*, *LONGITUDE*, or *DISTANCE*
1	Name of the latitude signal
2	Name of the longitude signal
3	Factor (= 1)
4	Unit (= m)

6.2.15 Calculation Module LIN

The module *LIN* calculates the signal $y(x) = m \cdot x + t$

Arg	Description
0	Name of the input signal *x*
1	Unit of the output signal
2	Factor *m*
3	Offset *t*
4	Minimum of the visible range
5	Maximum of the visible range

6.2.16 Calculation Module MEAN2

The module *MEAN2* calculates the 1st derivative $y(x)$ = dx/d*r* of signal *x* to another signal *r*.

Arg	Description
0	Name of the input signal *x*
1	Name of the reference signal *r*
2	Window width of the reference signal
3	Factor
4	Offset

6.2.17 Calculation Module MERGE

The module *MERGE* combines two signals into one output signal.

Arg	Description
0	Name of the LOWORD of the input signal
1	Name of the HIWORD of the input signal
2	DataType (which is either 6 for unsigned 32-bit, or 7 for signed 32-bit)
3	Factor *m*
4	Offset *t*
5	Unit of the output signal

6.2.18 Calculation Module MOVE

The module *MOVE* moves the input signal: $y(t) = x(t - Displacement) + Offset$.

Arg	Description
0	Name of the input signal *x*
1	*Displacement* in samples, or, when postfixed with "s", in time.
2	*Offset*

6.2.19 Calculation Module MPX

The module *MPX* demultiplexes a multiplexed signal.

Arg	Description
0	Name of the input signal *x*
1	Name of the multiplex signal
2	Multiplex index
3	Default value

6.2.20 Calculation Module POLY

The module *POLY* calculates the polynomial $y(x) = a_n \cdot x^n + \ldots + a_1 \cdot x + a_0$

Arg	Description
0	Name of the input signal *x*
1	Unit of the output signal ($0 – Empty, $1 – Unit of input signal)
2	Factor
3	Offset
4	Coefficient a_n
...	
$n + 3$	Coefficient a_1
$n + 4$	Coefficient a_0

6.2.21 Calculation Module QUANT

The module *QUANT* calculates the function
$y(x) = m \cdot \lfloor((x - tt)/mm)\rfloor + t$ with $\lfloor x \rfloor = \text{(int) } x$

Arg	Description
0	Name of the input signal *x*
1	Offset *tt*
2	Factor *mm*
3	Factor *m*
4	Offset *t*

6.2.22 Calculation Module REPLACEVAL

The module *REPLACEVAL* replaces certain values of the input signal.

Arg	Description
0	Name of the input signal *x*
1	Value to be replaced
2	New value (or *NEXT*, or *PREV*)
3	Minimum of the visible range
4	Maximum of the visible range

6.2.23 Calculation Module SCRIPT

The module *SCRIPT* executes a *Tellert Script File* to evaluate the resulting signal.

Arg	Description
0	Name of the binary script file without file name extension
1 … *n*	Parameters which are directly passed on to the script

6.2.23.1 Calculation Script Module ALTITUDE.SX

The script module *ALTITUDE* calculates the altitude in meters with the international altitude formula (valid for altitude ≤ 11 km).

Arg	Description
0	Name of the input signal *x* with physical unit hPa

6.2.23.2 Calculation Script Module FIXVAL.SX

The script module *FIXVAL* replaces a value of a given range.

Arg	Description
0	Name of input signal *x*
1	Minimal value of range
2	Maximal value of range
3	Replacement value

6.2.23.3 Calculation Script Module RATIO.SX

The script module *RATIO* calculates the function $y(x_1, x_2) = (m \cdot x_1) / x_2$

Arg	Description
0	Name of input signal x_1
1	Name of 2nd input signal x_2
2	Factor *m*
3	Unit of the output signal
4	Upper ratio limit

6.2.23.4 Calculation Script Module SDIFF.SX

The script module *SDIFF* calculates the difference of two signals $y(x_1, x_2) = x_1 - x_2$.

Arg	Description
0	Name of the input signal x_1
1	Name of the input signal x_2

6.2.23.5 Calculation Script Module SLOPE.SX

The script module *SLOPE* calculates the slope of the road.

Arg	Description
0	Name of the altitude signal in m
1	Name of the speed signal in km/h
2	Factor *m*
3	Unit of the output signal
4	Distance limit in meters
5	Minimum of the visible range
6	Maximum of the visible range

6.2.24 Calculation Module SMOOTH

The module *SMOOTH* smooths an input signal.

Arg	Description
0	Name of the input signal *x*
1	Order *N* (= 3, 5, 7, ...)

6.2.25 Calculation Module SUMT

The module *SUMT* calculates the integral over time of the input signal:

$$y(k) = s \cdot T \cdot \sum_{i=0}^{k} x(i)$$

Arg	Description
0	Name of input signal *x*
1	Factor *s*
2	Physical unit of the output signal
[3]	Bit factor *m* of the output signal quantization
[4]	Offset *t* of the output signal quantization

Example: Calculation of the distance by integrating the speed signal, where the unit of the speed signal is kph.

Name of the input signal *x*: *v*
Factor *s*: 0.00027777778
Unit: km
Bit factor *m*: 0.001 (to force a signal resolution of 1 m per bit)

6.2.26 Calculation Module TABLE

The module *TABLE* performs a table look-up.

Arg	Description
0	Name of the table (without the *.td*/*.tab* file extension)
1	Name of the input signal

6.2.27 Error Numbers

If TEMES could not perform a calculation, it displays one of the following error numbers:

No	Description
1	Unknown error
2	Measurement block is invalid
3	Invalid calculation module
4	Invalid calculation module version
5	Unknown input signal
6	Sample not found
7	Out of memory
8	Out of disk space
≥ 1000	Calculation module dependent error number

6.2.28 Directories

TEMES supports two calculation module directories. The first directory is the directory for user defined CDFs, DLLs, binary script files and resources like characteristic curves. By default this directory can be found at *<Temes Data Directory>\Calc*. The second directory is used for files which are shared by all TEMES users. By default it is found at *<TemesProgramDirectory>\Calc*. All modules and resource files are assumed to be found in the first directory. If a module or a resource file is not found, it is searched in the second directory.
The location of the user defined files can be modified by changing the Windows registry entry *HKEY_CURRENT_USER\Software\Tellert\TEMES\1.0\Directories\Calc*, and the location of shared files can be modified by changing the Windows registry entry *HKEY_CURRENT_USER\Software\Tellert\TEMES\Directories\CalcBin*.

6.3 Look-Up Tables

TEMES uses two ASCII files to define a characteristic curve. The first file describes the physical meaning of the data, and the latter defines the floating point values of the characteristic curve.

6.3.1 File <Table>.TAB

The file *<TABLE>.TAB* is a profile file 186 which describes the physical meaning of the floating point values. The description is given by four entries within the section *[Table]*. The entries are defined as follows:

```
[Table]
ChanCount = 02
```

```
Chan00 = <Data for input signal>
Chan01 = <Data for output signal>
Ext = .TD
```

The data for input and output signal is given by a comma separated string:

Position	**Description**
0	Signal name
1	Physical unit
2 … 7	Parameter for 8 bit quantization
8 … 13	Parameters for 16 bit quantization
14 … 19	Parameters for 32 bit quantization

Position	**Description of quantization parameters**
0	First bit value
1	Second bit value
2	First physical value
3	Second physical value
4	Default start of working range
5	Default end of working range

6.3.2 File <Table>.TD

The file *<TABLE>.TD* contains only floating point values with the restriction that each line of *<TABLE>.TD* contains the same number of floating point values.

Two column format: Each line describes a point of the characteristic curve. The first number is the value of the input signal and the second number is the value of the output signal. The numbers of the first column must increase and may appear only once within the first column. Output values for those input values which cannot be found within the first column, are received by linear interpolation (or extrapolation respectively).

Three column format: This format is supported to allow direct usage of *x-y Recorder* generated characteristic curves. The first column is redundant and contains no information. The second column corresponds to the first column the of two column format and the third column corresponds to the second column of the two column format. By definition, the first value of the first column is one, and the following values of the first column are always zero.

6.4 F6 Protocol

The f6 protocol is a very simple byte oriented protocol where the master unit sends a *request byte* and receives a byte as an answer from the slave unit. The serial default settings are 9600 bd, 8 data bits, 1 stop bit, no parity bit.

Expression	Description
Request byte	8-bit value; a slave answers (within a max. responding time) with an 8-bit value;
Control byte	8-bit value; a slave reacts device specific;
Address	An *address* is an 8-bit query value which is not used as a reserved query value within the corresponding protocol. The slave unit remembers the address value and responds with the least significant 8-bit value of the address value.
NEXT	*NEXT* is the query byte with the value $f6_{16}$.
INFO	*INFO* is the query byte with the value $f7_{16}$.
NEXT_DATA	*NEXT_DATA* is the query byte with the value $f8_{16}$.
IGNORE	*IGNORE* is the query byte with the value $f9_{16}$.
CHECKSUM	*CHECKSUM* is the query byte with the value fa_{16}.
NEXT_SAMPLE	*NEXT_SAMPLE* is the query byte with the value fc_{16}.
CR	*CR* is the abbreviation for carriage return (value $0d_{16}$).

6.4.1 F6A Protocol

Only NEXT is a reserved query byte. A slave unit must never answer on the sequence { *INFO*, *NEXT_DATA*, *NEXT_DATA* } or { *INFO*, *NEXT*, *NEXT* } with the ASCII values { 'F', '6', 'B' }. The value size is 16 bits. The slave responds with the high byte of the last remembered address value when the master sends *NEXT*.

6.4.2 F6B Protocol

Only the query bytes *NEXT*, *INFO*, *NEXT_DATA* and *IGNORE* are reserved. NEXT provides successively the next higher byte of the lastly remembered address value. *INFO* provides the least significant byte of the information block. All other bytes of the information block can be requested after *INFO* with *NEXT* (or *NEXT_DATA* for version 1.0).

The information block is defined as follows:

Data Type	**Name**	**Description**
BYTE	id[3]	{ 'F', '6', 'B' }
BYTE	size	Number of following bytes incl. checksum; if *size* equals 0 then *value_size* equals 4;
BYTE	value_size	Size of the address value in bytes (≥ 4)
DWORD	number	Device number (0: no valid device number)
BYTE	type_size	Size of the following device type description
BYTE	type[type_size]	Device type description
WORD	version	Device version
DWORD	built	Manufacturing date in the format: YYYYMMDD
DWORD	modified	Date of last modification (YYYYMMDD)
DWORD	verified	Date of last verification (YYYYMMDD)
DWORD	good_thru	Date of next verification (YYYYMMDD)
BYTE	info_size	Size of the following info
BYTE	info[info_size]	Information
BYTE	checksum	Exclusive-Or over all bytes from *value_size* to *info*[*info_size*-1]
BYTE	eob	0 (end marker)

6.5 Profile File

Some items require a profile file in TEMES or at least the contents of a profile file.

A profile file is a text file which is separated in sections. Sections are defined with an opening bracket at the line beginning, followed by the section name, and followed by a closing bracket (e. g. "[section name]"). The following lines are the contents of the section. And are usually key definitions. A key is defined by the key name at the line beginning, optionally followed by whitespace, followed by an equality sign, optionally followed by white space, followed by the key value (e. g. "key=value", or "key = value").

A list in TEMES is defined by two or more keys, the first key defines the list size and is named by concatenating the list name with the text "Count" (e. g. "ListCount"). The following keys are the list items which are named by concatenating the list name with the zero based item position (e. g. "List0", or "List1"). Some lists are required to use the same number of digits for its numbers. This can be achieved by using preceding zeros.

Some key values consist of arrays. Array items are separated by a comma and optional whitespace (e. g. "Array=item0,item1,item2,item3"). If a white-space/comma is required for the array item, it can be embedded within two double quotation marks as an escape sequence.

The following text is translated inside an escape sequence:

Text used inside an escape sequence	**Expanded text**
\a	ASCII character 07_{16} (alert)
\b	ASCII character 08_{16} (backspace)
\f	ASCII character $0c_{16}$ (form feed)
\n	ASCII character $0a_{16}$ (new line)
\r	ASCII character $0d_{16}$ (carriage return)
\t	ASCII character 09_{16} (horizontal tab)
\v	ASCII character $0b_{16}$ (vertical tab)
\\	\
\'	'
\"	"
\?	?
\x*XX*	8-bit character with hexadecimal digit *X*
\XXX	8-bit character with hexadecimal digit X
OOO	8-bit character with octal digit *O*

6.6 TEMES View

Only a preview version of *TEMES View* is available at the moment. *TEMES View* is a performant partial-replacement for *TEMES Chart*. It loads the entire TEMES file into the memory and keeps all signals as 64-bit-floating-point values into the memory. This memory demanding process also yields a better performance.

6.6.1 Requirements (TEMES View)

In order to run *TEMES View* correctly, following items are required:

- Operating System: Windows 10/8.1/8/7 (x86 or x64 Edition)
- Microsoft .NET Framework 4.5 (for Windows 7)
- Optional: Microsoft .NET Framework 4.5 Full Language Pack (for Windows 7)
- Recommended: an x64 Windows Edition
- Recommended: at least 32 GiB RAM
- Recommended: a DirectX10/11 (or later) capable video card
- Recommended: monitor resolution at least 800 x 600

6.6.2 Version History (TEMES View)

V0.5.6

- **New:** Library update to SciChart v5.4.0.12119 and to Math.NET Numerics V4.8.1

V0.5.4

- **New:** Extrema and statistics display.

V0.5.3

- **Modified:** *Google Earth* interface – 1-digit-difference allowed for *Latitude* and *Longitude* signals.

V0.5.2

- **New:** Integration of *Google Earth* interface.

V0.5.1

- **New:** FFT for the visible range of the currently selected slot/curve.
- **New:** Draw curves in standard view (by applying the user defined default factor, offset, signal delay, signal name, signal unit)

V0.5.0

- **New:** Draw curves only with the raw signal definition.

6.6.3 Running TEMES View

TEMES View is run by executing "*TEMES View.exe*". It is a .NET 4.5 framework WPF application based on several libraries including *TIL_TMS.DLL*.

Following mouse events are available:

Mouse left button (chart area)	Zooms the chart in x direction. *Shift* – Zooms the chart in both direction. *Ctrl* – Moves the chart.
Mouse wheel	Scrolls the chart in x direction. *Shift* – Zooms/Unzooms the chart in x direction. *Ctrl* – Zooms/Unzooms the chart in both directions.
Mouse right button (chart area)	Moves the chart.
Mouse left button (axis area)	Scrolls an axis. *Ctrl* – Zooms an axis (The clicked axis half is zoomed while the other half's end point is fixed).
Mouse left button (curve)	Selects/Deselects a curve in cursor mode. *Ctrl* – Selects/Deselects an additional curve in cursor mode.
Mouse left button (double click in chart area)	Extends the chart to the maximal values.

Following keystrokes are available:

Left, Right, Up, Down	Scolls the chart by half in the desired direction.
TAB, Shift+TAB	Select the next/previous slot.
z, Z	*z* – Zooms the chart in x direction. *Z* – Zooms the chart in y direction.
u, U	*u* – Unzooms the chart in x direction. *U* – Unzooms the chart in y direction.
v	Toggles the visibility state of the corresponding curve. When the time slot is selected, all curves are affected.
a	Toggles the visibility state of the corresponding axis. When the time slot is selected, all curves are effected.
c	Centers the chart around the selected x value in cursor mode, or selects the difference cursor as new visible range.
n	(Re)sets the difference cursor in cursor mode.
d, D	Sets the axis to the default value(s). *D* – When the time slot is selected, all y axes are affected.
t, T	Sets the axis to the maximal value(s). *T* – When the time slot is selected, all y axes are affected.
G	If °N/°E signals exist: Make a KML file of the visible image section and call *Google Earth*. If *Google Earth* is installed but does not start, then the *KML-Fixer* (https://www.tellert.de/?product=temes) can be helpful.
e, E, Ctrl+e	*e* – Exports the diagram as a *.png*/*.xps* file. *E* – Marks the maximal value of the selected slot/curve for the visible range. *Ctrl+e* – marks the minimal value of the selected slot/curve for the visible range.
r	Resets the extrema markers.
s	Displays statistics of the current slot/curve for the visible range: Min – minimal value; Max – maximal value; Avg – average

	value; Avg0 – average value without counting signal value equals 0; N – sample count; N0 – sample count without counting signal value equals 0; SD – sample standard deviation; AD – absolute deviation;

6.6.3.1 FFT

An FFT window is opened for the visible range of the currently selected slot/curve. The FFT is done according to *Matlab*, and the magnitude is normalized by dividing with the half of the number of samples for frequencies greater than 0 Hz, and by dividing with the number of samples for frequency equals 0 Hz.

Following keystrokes are available:

d, D	*d* – Sets the *x* axis to the default value. *D* – Sets the *y* axes to the default values.
t, T	*t* – Sets the *x* axis to the maximal visible range. *T* – Sets the *y* axes to the maximal visible ranges.
m	Toggles the visibility state of the magnitude.
p	Toggles the visibility state of the phase.
l, L	Toggles the logarithmic/linear axis. *l* – affects the *x* axis (= frequency). *L* – affects the left *y* axis (= magnitude).
F9	Toggles the visibility state of the cursor.
e	Exports the diagram as a *.png*/*.xps* file

6.6.4 Licenses

6.6.4.1 SciChart

You are not allowed to debug and to develop against: SciChart.Charting.dll, SciChart.Core.dll, SciChart.Data.dll, SciChart.Drawing.DirectX.dll, SciChart.Drawing.dll, SharpDX.D3DCompiler.dll, SharpDX.Direct3D9.dll, SharpDX.Direct3D11.dll, SharpDX.Direct3D11.Effects.dll, SharpDX.dll, SharpDX.DXGI.dll, SharpDX.Mathematics.dll, sharpdx_direct3d11_1_effects_x64.dll, sharpdx_direct3d11_1_effects_x86.dll, TEMES View.exe, TEMES View.resources.dll.

If you want to debug and to develop against the above listed files, you will need at least a developer-license from https://www.scichart.com (See also http://support.scichart.com/index.php?/Knowledgebase/Article/View/7/7/licensing-faq).

6.6.4.2 SharpDX

Copyright (c) 2010-2014 SharpDX - Alexandre Mutel

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

6.6.4.3 Math.NET Numerics

Math.NET Numerics License (MIT/X11)
Copyright (c) 2002-2019 Math.NET

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

6.6.4.4 TIL_TMS.DLL

Permission is hereby granted, free of charge, to any person obtaining a copy of these TIL_TMS.DLL files, to deal in these TIL_TMS.DLL files without restriction, including without limitation the rights to use, copy, compress, rename, embed, merge, publish, distribute, sublicense, and/or sell copies of these TIL_TMS.DLL files. However, further modifications of the TIL_TMS.DLL files, or a reverse engineering of these TIL_TMS.DLL files is not allowed.

THE "TIL_TMS.DLL" FILES ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

6.6.4.5 TIL_TMA.DLL

This software (= ZLIBWAPI.DLL) is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

(Further information is available at http://www.winimage.com/zLibDll/minizip.html and http://zlib.net)

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- I -

- K -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- X -