User manual:

proCANtool CANbus monitoring tool

Document version: 1.1
Documentation date: May, 18th, 2011

No part of this document or the software described herein may be reproduced in any form without prior written agreement from EMS Dr. Thomas Wünsche.

For technical assistance please contact:
EMS Dr. Thomas Wünsche
Sonnenhang 3
D-85304 Ilmmünster
Tel. +49-8441-490260
Fax +49-8441-81860
support@ems-wuensche.com

Our products are continuously improved. Due to this fact specifications may be changed at any time and without announcement.

WARNING: proCANtool CANbus monitoring tool may not be used in applications where damage to life, health or private property may result from failures
# Table of contents

1. **Introduction** ......................................................................................................................... 6
   1.1 **Description** ......................................................................................................................... 6
   1.2 **Requirements** ...................................................................................................................... 7

2. **The main window** .................................................................................................................... 7
   2.1 **Main menu bar** .................................................................................................................... 8
       a) **Menu Program** ................................................................................................................ 8
       b) **Menu Trace** .................................................................................................................... 8
       c) **Menu View** .................................................................................................................... 9
       d) **Menu Messages** ............................................................................................................ 10
       e) **Menu Signals** ................................................................................................................ 10
       f) **Menu Controls** ............................................................................................................... 10
       g) **Menu Extras** ................................................................................................................ 11
       h) **Menu ? or Help/Info** ...................................................................................................... 11

2.2 **Working with Views** .......................................................................................................... 12
       a) **Object view** .................................................................................................................... 12
       b) **Sequential view** ............................................................................................................. 12
       c) **Historical view** ............................................................................................................. 12
       d) **Filtered View** .............................................................................................................. 12

3. **Open dialog (Opening a CAN channel)** ............................................................................... 13

4. **Extended Settings dialog** .................................................................................................. 13

5. **Channel editor window** .................................................................................................... 14

6. **Trace dialog (Setting up a Trace)** .................................................................................... 15

7. **TracePlayer dialog (Replaying a Trace)** ......................................................................... 17

8. **Send message dialog** ....................................................................................................... 18
   8.1 **Send custom defined message** ...................................................................................... 18
   8.2 **Send messages from a script** ......................................................................................... 18
   8.3 **Send messages from a trace file** .................................................................................... 18

9. **Resolving Errors / Debugging** ............................................................................................ 19
   9.1 **Logging Window** ........................................................................................................... 19
   9.2 **Information for resolving Bugs** ..................................................................................... 19

10. **Working with Signals** ...................................................................................................... 20
    10.1 **Global settings** ........................................................................................................... 20
    10.2 **Adding a normal message** ......................................................................................... 21
    10.3 **Adding a multiplexed message** ................................................................................. 21
    10.4 **Adding a signal** ......................................................................................................... 21
        a) **Common settings** ...................................................................................................... 22
        b) **Bit and Type settings** .............................................................................................. 22
        c) **Scaling and offsets** ................................................................................................. 22
        d) **Default and boundary values (only Tx-Signals)** ......................................................... 22
        e) **Data types** ............................................................................................................... 22
        f) **Conversion** .............................................................................................................. 23
11. Creating a graphical Signal Layout

11.1 Static layout elements
   a) Row or Column
   b) Spacer
   c) Legend
   d) Bitmap
   e) Text

11.2 Dynamic layout elements
   a) Angular meter
   b) Level meter
   c) State led display
   d) Value display
   e) State bitmap display

11.3 User input elements
   a) Button control
   b) Combo box control
   c) Slider control
   d) Toggle button control

11.4 Example layouts
   a) Example row/column layout
   b) Use all displays – example layout
   c) Different types of Angularmeter displays
   d) Different types of Levelmeter displays

Image references

The main window with its areas
Open channel dialog
Extended settings
CPC-Conf Editor to edit the current system configuration
Traceplayer with active trace file
Minimal tracefile example with the containing data fields
Logging Window of canSuite with LogLevel on DEBUG
Signal editor with Message and multiple signals
Byte order - little or big endian
Example layout with rows and columns
Tree view of the row/column signal layout
Use all elements - example layout tree
Use all elements - example layout view
Different types of angular meter displays
Different types of level meter displays
1. Introduction

This user manual describes the functionality and usability of proCANtool CANbus monitoring suite.

1.1 Description

This GUI application is a versatile monitoring suite to visualize and manipulate CAN messages on a CANbus system in a comfortable way. EMS Dr. Thomas Wünsche offers this software for free to its customers, but only in connection with purchases of EMS CAN PC Interfaces.

This application will be available for Windows and Linux platforms and supports all kinds of EMS CAN PC Interfaces running with predefined baudrates or if applicable also with custom baudrates.

The main features of proCANtool CANbus monitoring suite are:

- View and analyze the messages on the bus including a history of the most recent messages.
- Send custom messages or use message scripts or trace files.
- Show the available messages as a continued list or as object view.
- Configure the object view to listen for changes on the identifier, the length, frame format, frame type and direction or on single or multiple data bytes of the message.
- Message filters are on a „per view“-basis. This means there can be different filters for each view, but also global ones for every view.
- Beside the filter mechanisms implemented in software the hardware filters of SJA1000 can also be set separately.
- Information area for current busload, message counters and error counters.
- Write the entire CAN traffic to a log file including the ability to start and stop the trace by a definable trigger condition and to save an amount of preceding or subsequent messages.
- Tracefiles can be replayed in almost realtime.
- Define messages and signals and show their values in a configurable signal overview with displays like AngularMeter, LevelMeter, State-Items, Static text widgets, Enumeration-Controls and simple Value-Displays and even Sliders, Buttons and Dropdowns to modify and resend the values on the bus.
1.2 Requirements

proCANtool CANbus monitoring suite is a rather complex software and therefore needs adequate hardware to perform properly. A real minimum requirement cannot be set – it's all a performance issue.

As mentioned the suite will run also on older machines (e.g. single core with 1GHz and 256MB of RAM), but for a satisfactory result we recommend at least 1.6 Ghz and 1GB of RAM to use all features also on high busloads.

proCANtool CANbus monitoring suite will run on Windows XP and later and on Linux 2.6 with an installed wxWidgets 2.8 or later.

For some Linux distributions we will provide software packages.

2. The main window

![Image 1: The main window with its areas](image)

(1) The information area for interface, driver and library information such as version or serial.

(2) The status overview with some statistical information about the channel, baudrate, message and error counters and busload information. There is also the button „Message History“ to switch from live to history message view. The live CAN traffic will be processed in the background for the meantime.

(3) The main message overview for received and sent messages and also for the different error information messages on the bus. It also holds the different other available views such as the object view, signal view or the message history.
2.1 Main menu bar

The menu items of the main menu bar are marked with 3 dots (...) if the command needs more input in an extra window.

The menu structure is as follows:

a) Menu Program

- **Open Channel ...** *(Ctrl + O)* *(page 13)*
  
  Opens the dialog to specify some channel settings for the channel, which shall finally be opened.

- **Close Channel** *(Ctrl + C)*
  
  Closes the actually opened channel and resets the device.

- **Edit Channels ...**
  
  Start the editor for channels on EMS interfaces.

- **Load Project**
  
  - **Load projectfile ...**
    
    Load a project file, selected with a file chooser.
  
  - **Load last project**
    
    Load the last auto-saved project.

- **Save project as ...**
  
  Save the project data to a project file on filesystem.

- **Quit program** *(Ctrl + Q)*
  
  Quit the program - Channel is closed and interface is resetted.

b) Menu Trace

- **Configure ...** *(page 15)*
  
  Choose the settings for a Trace (Start-/Stop-Trigger, prepending and trailing message count, Tracefile).

- **Deactivate trace**
  
  Deactivate the actually set triggers and stop trace.

- **Reactivate trace**
  
  Reactivate the actually set triggers or start trace.

- **Replay Trace ...** *(page 17)*
  
  Open the control window for replaying a trace from a loadable trace file.
c) Menu View

➢ Add View (Ctrl + Shift + A) (page 12)
Add a new tab with standard settings (hex, display msgs, sequential view, no filters) for an extra view within the main message view. (See: Image 1, #3 for the location of this area)

➢ Duplicate View
Create a duplicate of the active view incl. all settings in a new tab.

➢ Rename View
Change the name of the currently active view.

➢ Clear View (Ctrl + Shift + C)
Delete the contents of the message buffer of the active view.

➢ (*) Display Decimal
Switch the identifier and the data bytes to decimal representation.

➢ (*) Display Hexadecimal
Switch the identifier and the data bytes to hexadecimal representation.

➢ [*] Display Messages
Toggle the visibility of the messages within the message view.

➢ [*] Object View (page 12)
Toggle between sequential message view and object view.

➢ [*] Signal View (currently disabled)
Toggle a viewer area for Rx and Tx-Signals on or off.

  currently disabled – will be available in future versions

➢ Object view settings
Configure which parts of a message trigger object view updates. (Identifier, length, format, type, direction and/or each data byte)

  IMPORTANT: These settings do not apply to already shown messages, so - better clear the message history

➢ Filter ... (page 12)
Define message filters for the active view, based on their Identifier.

➢ Close View
Close the currently active view and its settings.

➢ Close others
Close all views and their settings except the currently active view.
d) Menu Messages

➢ **Send message ...** *(Ctrl + S)* *(page 18)*

Opens an extra window to define messages, load message scripts and send messages from multiple tabs.

➢ *[•]* **Enable receiving**

Toggle the handler for receiving messages on or off.

➢ **Global Filter options ...**

Opens an extra window to set filter options on the Identifier.

➢ *[•]* **Enable global filters**

Toggle global filter options on or off.

e) Menu Signals

➢ **Edit signals ...** *(page 20)*

Opens an extra window to edit messages and their contained signals.

➢ **Edit layout ...**

Opens an extra window to edit the layout to analyze the values defined in the different messages and their signals.

➢ **Show layout ...**

Opens an extra window to show the configured layout with the different signal displays and controls.

➢ **Send generatable messages**

Send all messages as defined in the signal editor (content, default values, repetition time etc.).

f) Menu Controls

➢ **Reset counter**

Set the different counters shown in the status overview back to zero. *(See: Image 1 #2)*

➢ **Reinitialize controller** *(Ctrl + I)*

Clear error states and error message queues of the controller.
g) Menu Extras

- **Inquire interface version**
  Do a request for the Interface version.

- **Inquire interface serial**
  Do a request for the Interface serial.

- **Inquire CAN state**
  Do a request for the Interface CANstate.

- **Inquire CAN params**
  Do a request for the Interface CAN params.

- **Inquire driver version**
  Do a request for the driver version.

- **Inquire driver serial**
  Do a request for the driver serial.

- **Inquire CPC library version**
  Do a request for the library version.

- **Change CAN params ...**
  Opens a new window to change the CAN parameters.

- **Display log window ...**
  Opens a new window to view debug outputs and errors.

h) Menu ? or Help/Info

- **About**
  Opens a new window with information about the company and the developers of the proCANtool CANbus monitoring suite.
2.2 Working with Views

All actions available from Menu View refer to the active tab and are also available as a popup menu for the currently active tab by a right click on the tab flag. Beside adding new views, duplicate a view and its settings, rename it or clean up the contents, there are options to close the active tab or close all other tabs.

The settings, such as the display format (dec/hex), toggle message visibility, toggle between sequential view and object view are only valid for the active tab. The settings within object view settings are only active if object view is activated.

a) Object view

This view shows only the last sent/received unique messages and discards the other ones. The fields on which the uniqueness is calculated can be set from the object view settings-submenu.

b) Sequential view

All messages are shown as sequence in a list, the newest one is shown on the bottom and the oldest one at the top of the list. Messages not fitting into the view area are discarded.

c) Historical view

Accessible by pressing a button located in the status area. (See: Image 1, #2) The last 500 messages in the buffer at the time of pressing the button are displayed as list. Receiving and sending of messages will proceed further in the background without altering the displayed historical messages.

d) Filtered View

Each view can have their independent filter settings. At the moment it is only possible to filter on message type (standard,extended) and a single identifier or on identifier ranges. Filters can be configured to display all identifiers NOT covered by the filters or to display all identifiers that are covered by the filter settings.

The local per view filters can be extended by global filter settings accessible by Menu Messages -> Global filter options ... which has to be enabled separately by a checkbox right below the mentioned menu entry (Enable global filters).
3. Open dialog (Opening a CAN channel)

(1) List of available channels
(2) Settings of the currently chosen channel
(3) Type of controller (only SJA1000 is supported)
(4) Baudrate settings
(5) Open the extended channel settings
(6) Edit channels or reload channel configuration

If the „Edit channels“ button isn’t active the configuration file is not editable.

4. Extended Settings dialog

Changing the default settings (such as baudrate, acceptance filter and listen only mode) is possible altering the extended settings. (Image 3)

The extended settings dialog allows to change the BTR0, BTR1 and acceptance code/mask registers of the NXP SJA1000 CAN controller. Even if you have a device of EMS not equipped with the SJA1000, thus not directly supporting these settings, the EMS device drivers will have added support for it.

The extended settings are also available from Menu Extras -> Change CAN params. If settings has been changed the controller will be reinitialized with these settings.

The CAN channel is closed by Menu Program -> Close channel or by unplugging the Interface (e.g. a CPC-USB Interface).
5. Channel editor window

![CPC-Conf Editor](image.png)

**Image 4: CPC-Conf Editor to edit the current system configuration**

**IMPORTANT:**

This window is only accessible if the cpcconf.ini on your system can be read and written by the user running the program.

On the left hand tree you see a overview of the current **active** configuration with expandable details and on the right hand tree there is a list of the current **available** and active interfaces in your system and a list of **templates** for all available EMS PC interfaces.

The button „**Refresh**“ refreshes the tree showing the available interfaces in your system by rereading the information from system internals.

The „**Edit**“ button on the left, opens an editor window for the selected entry from the left tree to edit the current settings for this entry.

The „**Add**“ button creates a copy of the selected available interface from the right tree to the left tree and opens an editor to edit the copied entries before finally adding it to the configuration tree.

The „**Del**“ button removes the selected entry from the current configuration on the left.

After you finished editing choose **File -> Save INI File** to save the settings permanently.
6. Trace dialog (Setting up a trace)

It is supported to record the CAN communication into a file for further analysis or for later replay. The setup is done from **Menu Trace -> Configure**

**Global Settings**

- File on the filesystem where the trace data should be saved to.
- Format (hexadecimal/decimal) of the identifier and data bytes.

**Single mode tracing**

If enabled the triggers will only fire once, according to the defined start and/or stop trigger events and deactivate these triggers afterwards.

**Overwrite/Append tracefile**

If file should be overwritten every existing content will be deleted immediately within the chosen file. By appending all content will be safe.

The „Start Trigger“ and „Stop trigger“ tabs are used to define settings to start and/or stop a trace automatically on different kind of events.

- **EventType**
  - *Message Count*: Amount of received messages
  - *Identifier*: Specified identifier (also only some match bits can be masked), message type or length
  - *Bus Error*: Amount of occurred bus errors

- **Prepending / Trailing Messages**

The amount of messages that should be saved to the file, before the start trigger event got fired and after the stop trigger event got fired.

**IMPORTANT:** You have to enable each trigger condition by activating the check box bottom right of the corresponding tab otherwise no trigger events will be fired! By clicking **Activate** the trace is started if no start trigger is given. Otherwise the triggers are activated.

**Manual Start/Stop**

If no start and/or stop conditions are given you are able to manually start the trace by selecting **Menu Trace -> Reactivate Trace** and manually stop the trace by selecting **Menu Trace -> Deactivate Trace**.
7. TracePlayer dialog (Replaying a Trace)

By selecting **Menu Trace** -> **Replay trace** ... a window is opened to load and replay previously recorded communication on the CAN bus.

If the trace file specifies a date and time, when the communication was originally recorded, it will be taken as the base time while replaying.

![Image 5: Traceplayer with active trace file](image)

The main elements of the TracePlayer dialogue are a **menu bar**, the **file selector control**, a **time line** with **handles** for the start time, the current time and the stop time and 3 buttons. The first button is the **start/pause**-button, the second is the **stop**-button. To the right there is a checkbox-button for **continuous replay/loop**.

The handles can be dragged to define the start, current and stop position if the trace is in stopped state. The **gray areas** are not played, the **green area** has already been played and the **blue area** is remaining.

The **extended options** are located in an expandable pane activated by the little arrow-button to the left. Here the start time, current time and stop time is visible in the choosen format. The available formats are 24h-format and duration format.

The two different formats can be choosen from the menu item „**View**“ from the TracePlayer window.

The **24h-format** will show the actual times based on the original start time of the trace.

The **duration-format** will show the time in seconds, and microseconds starting at zero.

The **status bar** shows the time that's left for the current trace and different other information, depending on the actions taken within the window.

For a syntax example, see the section: „**Send messages from a trace file**“,
8. Send message dialog

To open the send message dialog select **Menu Messages -> Send message ...** from the main menu. The send message dialog can hold multiple tabs with a message pane, which can be loaded with a different type of message or script/trace.

8.1 Send custom defined message

If the radio button „user defined message“ is activated each relevant part of the message can be altered. There are settings for the message type (Extended, RTR), the identifier, length of the data part and the values of the single data bytes.

8.2 Send messages from a script

If the radio button „script/trace“ is activated a script file can be loaded from filesystem. The script can generate sequences of messages with defined pause periods. The syntax is as follows

- **D**: Standard data message
- **R**: Standard RTR message
- **XD**: Extended data message
- **XR**: Extended RTR message

These are followed by the identifier (if hex with preceding „x“) the length and the data bytes (if hex with preceding „x“) each delimited with a space.

e.g.: D x7e5 8 x01 x2 x04 5 6 7 x8 x09
- **C**: Comment (also ; is allowed)

Comments are delimited by a space and can not be multilined (each line has to be started with a C or a Semicolon)
- **W**: Wait time in microseconds

The wait item is followed by a number delimited by a space.

8.3 Send messages from a trace file

The contents of a previously traced communication can be sent again. This can be done from **Menu Trace -> Replay Trace** too.

<table>
<thead>
<tr>
<th>direction</th>
<th>type</th>
<th>seconds</th>
<th>microseconds</th>
<th>identifier</th>
<th>length</th>
<th>data bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start trace on 2010-May-11 14:12:11</td>
<td>Tx SD 000000018 999981002 0x000007FF 3 0xFF 0xFF 0xFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop trace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Image 6: Minimal tracefile example with the containing data fields*
9. Resolving Errors / Debugging

If unknown or strange behavior occurs someone may want to debug the program. Because a direct debugging of source code is not feasible for a user of the program we've added a logging window.

9.1 Logging Window

The dropdown on the upper left sets a LogLevel for the running program. This loglevel is not persistent and resets to default for the next run.

The checkbox „Raise window on event“ is stored within settings and is restored from the last active or a saved configuration.

The button „Save“ stores all currently available debug messages (1000 messages maximum) into the file selected from the file chooser.

The button „Clear“, clears the logging messages list.

9.2 Information for resolving Bugs

If you want to send us bug information please consider adding the following informations:

- Operating system with name (XP/Vista/7) and version (32/64bit)
- The version of EMS driver kit you've installed, with version number
  - WRK = Windows runtime kit
  - WDKg = Windows development kit – general license
  - WDKs = Windows development kit – single license
- A brief description of the error and how to reproduce it (debug log, message trace, CAN environment description, screenshots)
- The license key of the canSuite you are running ( ? -> About )
10. Working with Signals

By activating **Menu Signals -> Edit Signals** the Signal editor window opens. Herein the layout of the different messages on the bus can be defined to be reused as a value reference to the different signals and signal displays.

The settings may be loaded from or saved to a file from **File** menu of the current window. The menu **Settings** gives control over the floating point display format (scientific or fixed).

![Image 8: Signal editor with Message and multiple signals](image)

10.1 Global settings

The global settings are mounted to the root of the hierarchical tree on the left. These settings are mandatory for the resulting XML files.

- **Name**: The internal name of the signal database
- **Company**: The creator's company name
- **Author**: The author's name
- **Version**: The document's version
- **Date**: The creation date
- **Comment**: A descriptive comment
10.2 Adding a normal message

A normal message is represented by a single CAN-Frame on the bus and assigned by its unique identifier and length. The optional name should be descriptive for the message.

The repetition period is used to identify messages, which fail to appear or are received too late. It is also used to repeat message generation. Enable it for each message you want to generate and activate sending from within menu:

Menu Signals -> Send generatable messages ...

10.3 Adding a multiplexed message

Multiplexed messages are an extension to normal messages wanting entries for name, identifier and length.

Multiplexing a message means that a specified section of the data bytes (in most cases the first byte) controls the interpretation of the other contained values. Multiplexers are mostly implemented as simple counters, counting up for the different types for this unique identifier.

These additional settings are:

- **Multiplexer Start byte**: The byte where the multiplex value starts
- **Multiplexer Start bit**: The first bit holding the value within the defined start byte
- **Multiplexer Length**: The length of the value measured in bits
- **Generate mode**: Generate all variants, user defined ones or do nothing – user-defined means controlled by each single muxer item with „Generate message“.

The settings for a muxer entry are only the muxer name, the above mentioned muxer value (the counter value) and the repetition period.

10.4 Adding a signal

Signals may be added to normal or multiplex entries of multiplexed messages and define a section within the data part of a message, holding a defined value.

On the bottom of the signal settings panel a overview of the data bytes with the configured signals is given. These signal entries can also be altered by dragging the signal or moving its borders with the mouse.

To enable dragging, the signal has to be selected first.
There are the following settings to define a signal:

a) Common settings
   Specify a descriptive name and the unit of the contained value.

b) Bit and Type settings
   Specify the start byte and the related start bit and define the length of the value in bits. Bytes are numbered from left to right starting at 1. Bits are counted from right to left beginning with 0. The length, given as the number of bits, is expanded from right to left.
   e.g. Byte 4, Bit 1 with a length of 8:
   
   ![Byte order diagram]

   c) Scaling and offsets
   The value extracted from the CAN message will be converted by the given formula to the visualized signal value.

d) Default and boundary values (only Tx-Signals)
   Signals to be generated have to respect value borders and also have to be initialized to a default value if no other value is available. So if you want to generate a signal give the minimum and maximum value to be sent and a default value for signal initialization.

e) Data types
   Define the endianness (byte order) of the signal value (Intel or Motorola) and define the base (10 - dec or 16 - hex) which the values should be displayed for. Intel is default.

   ![Data types table]

As an example, lets see the value 0x0A0B0C0D filled into the first 4 bytes of the CAN data part of a message.

See also: [http://en.wikipedia.org/wiki/Byte_order](http://en.wikipedia.org/wiki/Byte_order)
f) Conversion
Specify the source format of the signal value and the destination format it should be converted to.

**Source Formats** (see: [Signed_number_representations](#))
- **Unsigned** = all bits are the relevant value
- **Sign and Magnitude** = MSBit is the sign (0=pos; 1=neg)
- **One's complement** = 0 – 127 are positive values, so bitwise NOT of the numbers are the corresponding negative values.
- **Two's complement** = 0 – 127 are positive values so bitwise NOT and adding 1 to the result is the negative value.

**Destination Formats** (see: the corresponding links)
- **Boolean** = true or false (0 or 1)
- **Double** = floating point 64 bits (1 sign, 11 exp, 52 fraction)
- **Integer** = 32 bits (sign and magnitude)
- **Unsigned Integer** = 32 bits; only positive values

### 11. Creating a graphical Signal Layout
This is available from **Menu Signals -> Edit layout / Show layout**

#### 11.1 Static layout elements

a) **Row or Column**
   Can contain and align elements horizontal or vertical, add a border with a descriptive title (empty name hides border)

b) **Spacer**
   Add a Spacer with a specific width and height to align the other elements with spaces.

c) **Legend**
   A multiline text display with available formatting (bold, italics, underline, color, font) and width/height attribute.

d) **Bitmap**
   A scalable (width/height) image with optional displayed name (leave empty to avoid the displaying of the name)

e) **Text**
   A static text with a choosable font size (single line) and optional displayed name. (leave empty to avoid this).
11.2 Dynamic layout elements

These elements are bound to a signal and display the value extracted from the matching messages (rx and tx).

a) Angular meter

Highly configurable display with radius, start and end angle, start and end value, definable colors for the needle and the different definable sections. Clockwise and anticlockwise direction and configurable ticks with hideable values.

b) Level meter

A vertical or horizontal bar with upper and lower limits, invertible fill direction, definable sections with a unique color (if the value reaches this section), border color, number of separations and a descriptive name with definable position within the bar.

c) State led display

Display a colored circle defined by the corresponding color-value pair. The visibility of the textual value is configurable and an optional name can also be given (empty = hidden). If the signal times out a special color can be shown.

d) Value display

Display a textual value with definable font size and optional name (empty = hidden). The value can be substituted by a defined text per value. This is especially suitable for states that are not clearly represented by colors or images. e.g. show different numerical states of a state machine with their textual representation.

e) State bitmap display

Working exactly as a normal state display this display uses bitmaps instead of colors to show if a specific value is hit. It also supports a special image for timed out signals and scales the given images to a specific width and height.
11.3 User input elements

a) Button control
If this button is pressed and released a message with the signal bound to the button is sent to the bus, containing the last actual value or the value defined as default value.

b) Combo box control
These defined name/value pairs are combined to a drop down list with the name as choosable element. If choosen a message with the corresponding value is filled in the configured signal part of the messages data field and finally sent on the bus.

c) Slider control
A draggable vertical or horizontal slider with upper and lower limit and a choosable dimension in pixels. The area between the limits is divided by the available pixels.

So, each pixel dragged will increase or decrease the value by the calculated amount per pixel. This value is filled into the messages signal part and as before sent on the bus.

d) Toggle button control
The toggle control has two available values, one value for the ON-state and the other value for the OFF-State. On each change of the button state a message with the corresponding signal value is sent on the bus.
11.4 Example layouts

a) Example row/column layout

This example shows a layout with a horizontal row containing two vertical rows with two multiline text elements each.

The next row is vertical aligned and contains two horizontal rows containing two multiline text elements again.

The image to the right shows the corresponding tree from layout editor with all elements listed above.
b) Use all displays – example layout

Using all available elements arranged in horizontal rows.

**Image 12:** Use all elements - example layout tree

**Image 13:** Use all elements - example layout view
c) Different types of Angularmeter displays

Image 14: Different types of angular meter displays

- Define start and end angle and direction (clockwise/anticlockwise)
- Define start and end of sections with a color
- Define needle color
- Show or hide tick values, name and unit
- For a textual representation of the value define an additional text-control
d) Different types of Levelmeter displays

- Use different colors for background, foreground, border, ticks and text.
- Define different sections for status display. The value-bar will be colored with the section color and the section borders are shown by small lines. Beginning with a full stroke and ending with a dotted stroke.
- Define the position of the Signal name bound to this display
- Define the orientation of the display -> vertical or horizontal
- Define the fill orientation of the display -> bottom to top or top to bottom/ left to right or right to left
- Show or Hide border line, ticks, name and value

Image 15: Different types of level meter displays