

Sprecher : Andreas Lipowsky /CEO
Chat Moderator: Leonard Immel (Creator of GUI Editor)

Agenda

- Begrüßung
- GUI Editor Konzept
- Wesentliche Schritte zur Erstellung einer Baby-LIN GUI
- Verfügbare Widget und deren Features
- Kontrolle von GUI Elementen zur Laufzeit
- Fragen, Antworten und Feedback

Ausgangspunkt

- SDF Lösungen für LIN und CAN-Applikationen
- Nutzung von Bus-Signalen, virtuellen Signalen, Protokollen, Macro und Events.
- SDF Ausführung mit SimpleMenu oder Standalone (z.B. Baby-LIN-RC)

Je nach Anwendung ist auch eine GUI über Simple Menu vorhanden.

SessionConf v2.36.1 - [C:/documents/presentations/Gui-Editor-Webinar/Demo-Gui-Elements-EKB00xx.sdf]

File Edit View Tools Help

Hide expert settings Required SDF version: v3.20 FID:

SDF Version 3	Type	Name	Target	Comment
0	Monitored signal	SDF-Project	SDF-Project	
1	Monitored signal	SDF-Version	SDF-Version	
2	Edit signal	Cfg-PositionTolerance	Cfg-PositionTolerance	
3	Macro	DutConfig-670100-EKB0001	DutConfig-670100-EKB0001	
4	Macro	DutConfig-670200-EKB0001	DutConfig-670200-EKB0001	
5	Macro	DutConfig-670100-EKB0012	DutConfig-670100-EKB0012	
6	Macro	DutConfig-670200-EKB0012	DutConfig-670200-EKB0012	
7	Macro	ConfigAccess	ConfigAccess	
8	Macro	ConfigParams	ConfigParams	
9	Macro	ConfigAux	ConfigAux	
10	Macro	Reference	Reference	
11	Macro	RunToPos	RunToPos	
12	Macro	WritePartnum	WritePartNum	
13	Macro	WriteFunctionId	WriteFunctionId	
14	Macro	WriteHwVersion	WriteHwVersion	
15	Macro	SavePosition	SavePosition	
16	Macro	VerifyAll	VerifyAll	

Signals Macros Macroselections

Add signal by drag and drop or double click

Filter:

SignalNr	Signalname	Frame	
0	VA_3W_InfVar_Posn_Rq	VA_3W_Rq_Frm	LIN_Master (master)
1	VA_3W_InfVar_RefDrive_Enbl_Rq	VA_3W_Rq_Frm	LIN_Master (master)
2	VA_3W_InfVar_PosnActSav_Rq	VA_3W_Rq_Frm	LIN_Master (master)
3	VA_3W_Posn_Stat	Stat_Frm_0x1F	Valve
4	VA_3W_Move_Stat	Stat_Frm_0x1F	Valve
5	VA_3W_Err_Stat	Stat_Frm_0x1F	Valve
6	VA_3W_Err_ElectricFault_Stat	Stat_Frm_0x1F	Valve
7	VA_3W_Err_UnderVolt_Stat	Stat_Frm_0x1F	Valve
8	VA_3W_Err_OverVolt_Stat	Stat_Frm_0x1F	Valve
9	VA_3W_Err_OverTemp_Stat	Stat_Frm_0x1F	Valve
10	VA_3W_Err_MechBreak_Stat	Stat_Frm_0x1F	Valve
11	VA_3W_Err_UnexpBlock_Stat	Stat_Frm_0x1F	Valve
12	VA_3W_Err_RefFault_Stat	Stat_Frm_0x1F	Valve
13	VA_3W_Err_Group_ERR_Stat	Stat_Frm_0x1F	Valve
14	VA_3W_Err_Group_SNA_Stat	Stat_Frm_0x1F	Valve

Section properties

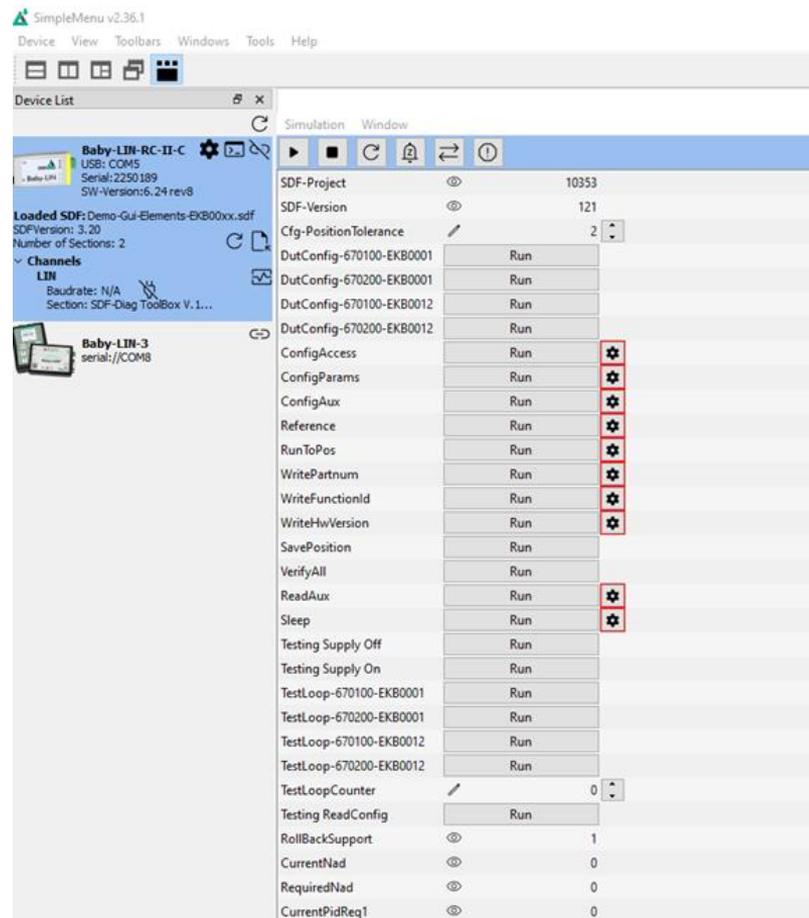
1-LIN: SDF-Diag Toolbox V.1.10

Section properties

- Bus description
- Emulation
- Tables
- Virtual signals
- Signalfunctions
- Protocols
- GUI-Elements (SimpleMenu/HARP etc)**
- Macros
 - [0] DutConfig--670100-EKB0001
 - [1] DutConfig--670200-EKB0001
 - [2] DutConfig-670100-EKB0012
 - [3] DutConfig-670200-EKB0012
 - [4] ConfigAccess(Nad, PidReq, Pid...
 - [5] ConfigParams(FsTime, FsPositi...
 - [6] ConfigAux(Rollback, SafeDista...
 - [7] _resetCycleFlags
 - [8] Reference(PosReq, TimeoutSec)
 - [9] RunToPos(TargetPos, TimeoutS...

Nachteile der bisherigen GUI-Elemente

- Anzeige nur im Simple Menu (Ausnahme HARP), deshalb keine Lösung für StandAlone Betrieb ohne Computer.
- Die Anzeige bietet keine Gestaltungsmöglichkeiten.
- Es ist immer
 - alles untereinander in einer Liste angeordnet
 - alles sichtbar
 - für alles die gleiche Textfarbe und Grösse verwendet.
- War eigentlich nur als Inbetriebnahmewerkzeug gedacht, wurde aber von etlichen Anwendern als Werkerfrontend z.B an Handarbeitsplätzen verwendet.
- Eine benutzerfreundliche Bedienoberfläche war damit nicht wirklich möglich.



Entwicklungsziele für den neuen GUI-Editor

1. Sollte Teil des LINWorks SessionConf werden
2. GUI soll mit einem erweiterten SimpleMenu auch für Generation 2 Geräte nutzbar sein.
3. Verwendung als stand-alone GUI auf den neuen Baby-LIN-3 Geräten mit eigenem Display.
Baby-LIN-3-RC und Baby-LIN-3-RCplus
4. Verwendung als Stand-alone GUI auf dem Baby-LIN-3-MB über dessen Web-Interface.

Konzept der neuen GUI Lösung

- Erstellung der GUI mit LINWorks SessionConf
- Speicherung der GUI-Definitionen im SDF
- Funktionaler Ersatz für die bisherigen GUI-Elemente (Signal, Makro, etc.)
- Zusätzliche Designelemente (Widgets) wie Textlabel, Bild, Linie, Rahmen, Meter und mehr.
- Freie Positionierung der Widgets auf der Anzeige
- Umfangreiche Einstellmöglichkeiten für die Widget-Eigenschaften
- Unterschiedliche Layouts für die verschiedenen Abspielpfade (SimpleMenu, DeviceDisplay)
- Verteilung der Informationen auf mehrere Panels
- Sichtbarkeit und andere Eigenschaften der Widgets dynamisch aus dem SDF steuern.

GUI Anzeigepfade

- **Standalone Solution**
Gui-Definition wird mit SDF in das Gerät geladen und dort lokal ausgeführt.
- **SimpleMenu Solution**
Die Gui-Definitionen werden nur in die DLL geladen (nicht in das Gerät) und auf dem PC ausgeführt, so dass auch Geräte der Generation 2 diesen Abspielpfad mit allen Features nutzen können.
- **BL-3-MB Web interface solution**
Das SDF läuft auf dem Gerät und eine darin enthaltene GUI-Definition wird über den integrierten Webserver an einen angeschlossenen WebBrowser ausgeliefert und dort ausgeführt.

Technischer Hintergrund:

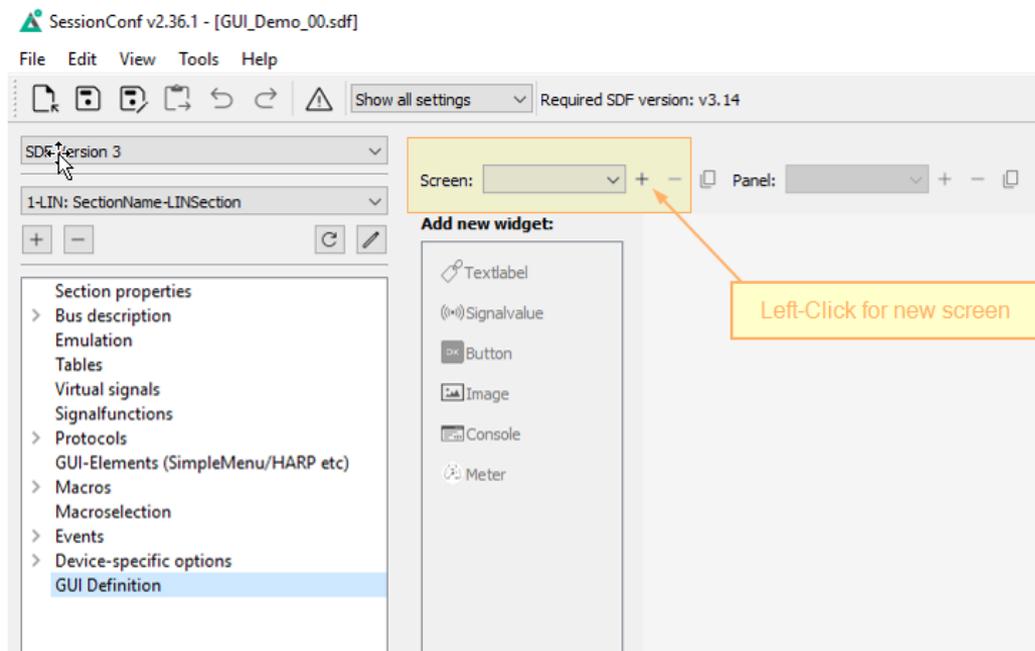
Da alle drei Auslieferungsmethoden die gleiche Codebasis verwenden, ist eine einfache Weiterentwicklung und Pflege dieser Softwarekomponenten gewährleistet.

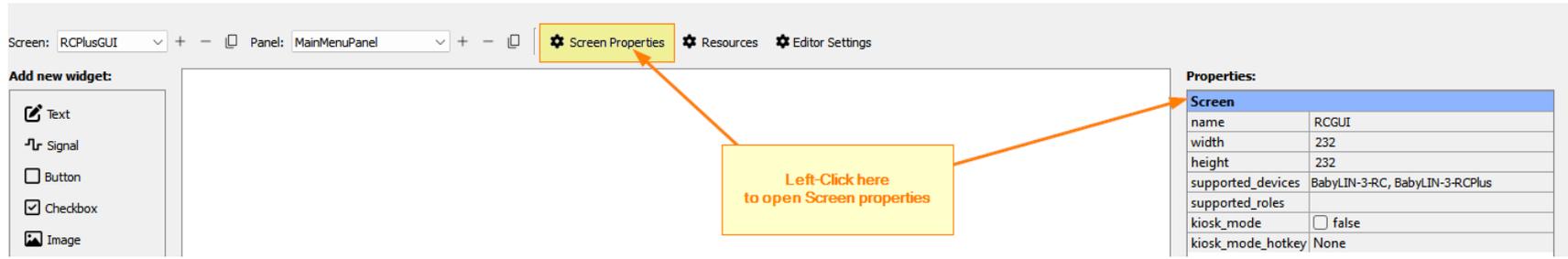
Grundelemente des GUI-Editors

- **Screen**
Ist das Basiselement und definiert die Auflösung des Anzeigegeräts und die möglichen Abspielpfade
Mehrere Bildschirme für verschiedene Geräte in einem SDF möglich.
- **Panel**
Ein oder mehrere Panels pro Bildschirm
Gruppierung von visuellen Informationen auf einem Panel
Panel Auswahl/Umschaltung zur Laufzeit möglich
- **Resources**
In den Ressourcen sind Schriftfonts und Bilder zusammengefasst, und können von allen Panel genutzt werden.
- **Widgets**
Einzelne Anzeigeelemente, die auf einem Panel angeordnet werden.

Screen

- Erstes Element beim Erstellen einer GUI.
- Ist das Basiselement und definiert die Auflösung der Anzeige.
- Mehrere Screens für verschiedene Geräte können in eine SDF untergebracht werden.
- Simple Menu Gui Screens erfordern keine zusätzlichen Speicher in den Baby-LIN Geräten, deshalb ist das auch auf Generation2 Geräten ausführbar.





supported_devices

Legt fest, auf welchem Gerät dieser Screen angezeigt werden soll.

Hinweis:

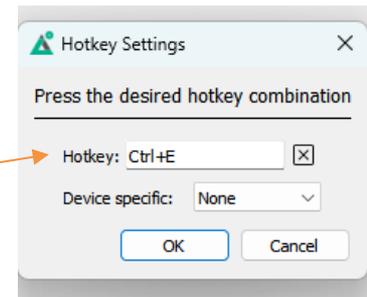
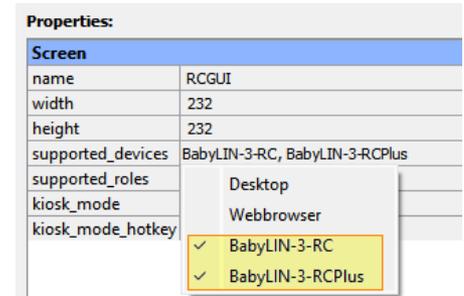
Um einen Screen zur Anzeige auf Baby-LIN-3-RC und Baby-LIN-3-RCplus Geräten zu definieren, beide anwählen.

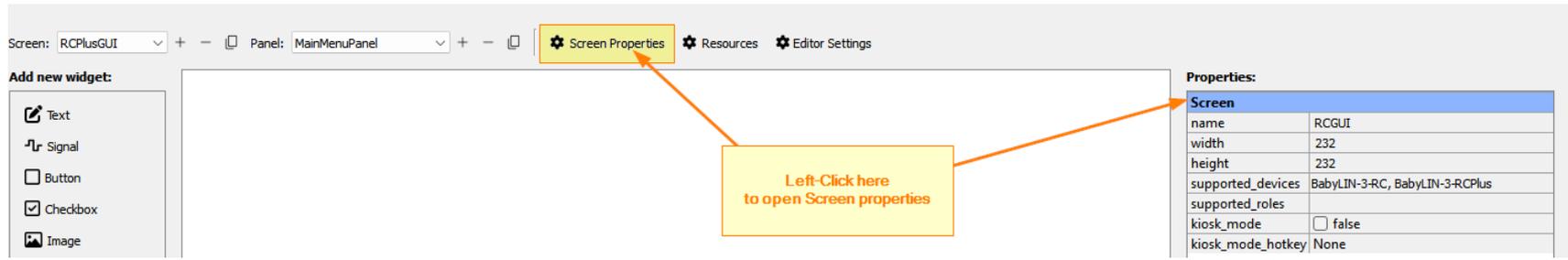
Kiosk Mode

Wenn aktiviert, wird beim Laden der SDF im Simple-Menu alle anderen Anzeigeinhalte (device List, GUI Definitions etc) ausgeblendet.

Um den KioskMode zu beenden, kann man einen Hotkey zuweisen.

Um einen Hotkey für das SimpleMenu zuzuweisen, einfach die entsprechende Taste bzw. Tastenkombination auf der PC-Tastatur drücken.





Screen: RCPPlusGUI + - Panel: MainMenuPanel + - Screen Properties Resources Editor Settings

Add new widget:

- Text
- Signal
- Button
- Checkbox
- Image

Properties:

Screen	
name	RCGUI
width	232
height	232
supported_devices	BabyLIN-3-RC, BabyLIN-3-RCPlus
supported_roles	
kiosk_mode	<input type="checkbox"/> false
kiosk_mode_hotkey	None

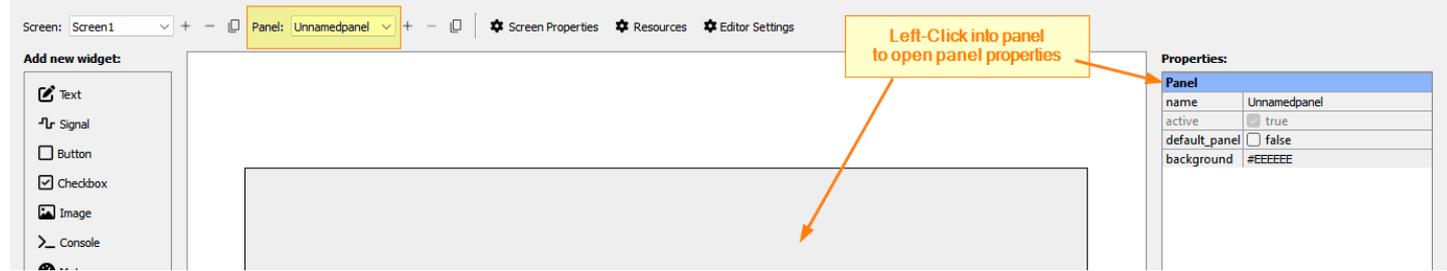
Left-Click here to open Screen properties

supported_roles

- Auf Baby-LIN-3-MB, mit aktivierter Benutzerverwaltung und Anmeldung, gehört jeder registrierte Benutzer zu einer bestimmten Gruppe/Rolle.
- Wenn er sich anmeldet, wird der Screen, der zu seiner Gruppe/Rolle passt, angezeigt.
- Wird keine passende Rolle in supported_role gefunden oder hat sich der Benutzer gar nicht angemeldet, wird der Screen für die Rolle nobody ausgeliefert.
- Ist supported_roles leer, wird dieser Screen immer ausgeliefert.
- Gibt es nur Screen mit nicht leeren support_roles Definitionen, aber keinen bei dem die Rolle Nobody enthalten ist, bleibt das Display leer.

Panel

- Ein oder mehrere Panels pro Screen
- Definition von visuellen Informationen durch Anordnung entsprechender Widgets auf dem Panel
- Ein erstes Panel wird automatisch erstellt, wenn ein Screen definiert wird.
- Der Name des Panels, die Hintergrundfarbe und die Eigenschaft `default_panel` können in den Panel-Eigenschaften (Properties) festgelegt werden.
- Solange es nur ein Panel gibt, spielt die Eigenschaft `default_panel` keine Rolle.
- Gibt es mehrere Panels, steuert die Eigenschaft `default_panel`, welches Panel zu Beginn angezeigt wird (Eigenschaft "active"). Wenn die Eigenschaft `default_panel` für kein Panel gesetzt ist, wird das erste Panel auf "active"="true" gesetzt.
- Die Eigenschaft "active" ist in SessionConf nicht editierbar, da sie zur Laufzeit definiert wird, um das tatsächlich angezeigte Panel zu markieren. Es kann ein Panel gleichzeitig aktiv sein.
- Im GUI-Editor wird die Eigenschaft "active" für jedes aktuell angezeigte Panel gesetzt.



Resources

Das sind für alle Panels
gemeinsam nutzbare Ressourcen.

- 4 fest definierte Schriftfonts
- Import von PNG Bilder
- Import von Custom Schriftfonts (TTF)

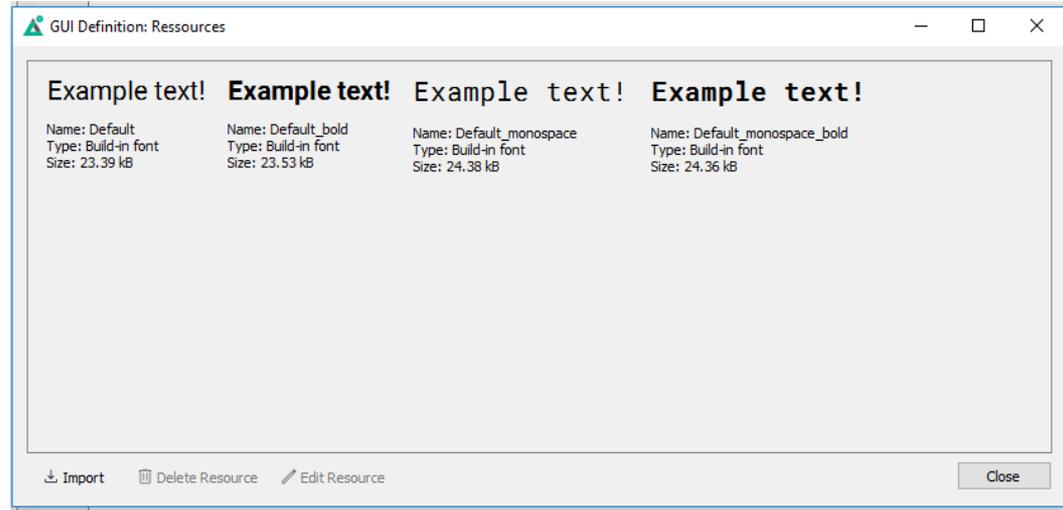


Image Import

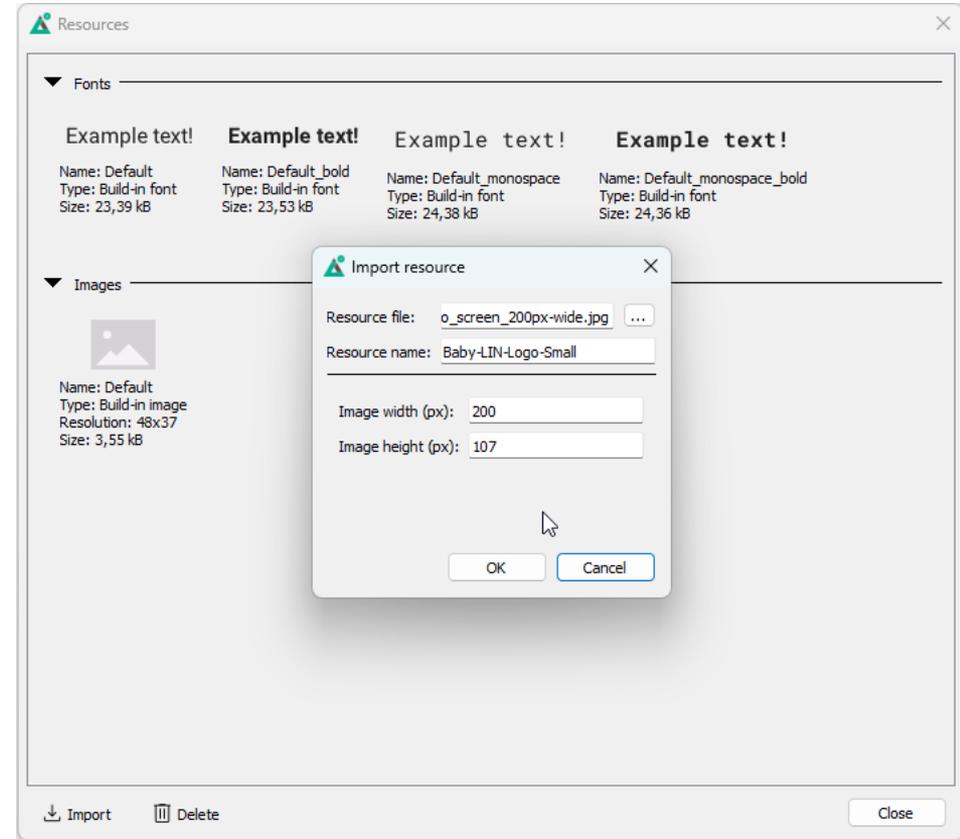
Bilder müssen in der Auflösung importiert werden, die für die gewählte Bildschirmauflösung geeignet ist.

Eine Größenänderung während der Positionierung eines Bild-Widgets auf einem Panel wird noch nicht unterstützt.

Ändern Sie daher die Größe Ihres Bildes mit einem externen Bildbearbeitungsprogramm, um die richtige Größe zu erhalten.

Beachten Sie, dass das Baby-LIN-3-RC/RCplus Display eine Auflösung von 232 x 232 Pixel hat.

Es kann also sinnvoll sein, 2 Versionen eines Bildes in Ihren Ressourcen zu haben, falls Ihre GUI Screens für Desktop- und Geräteanzeigen bereitstellt.



Widgets

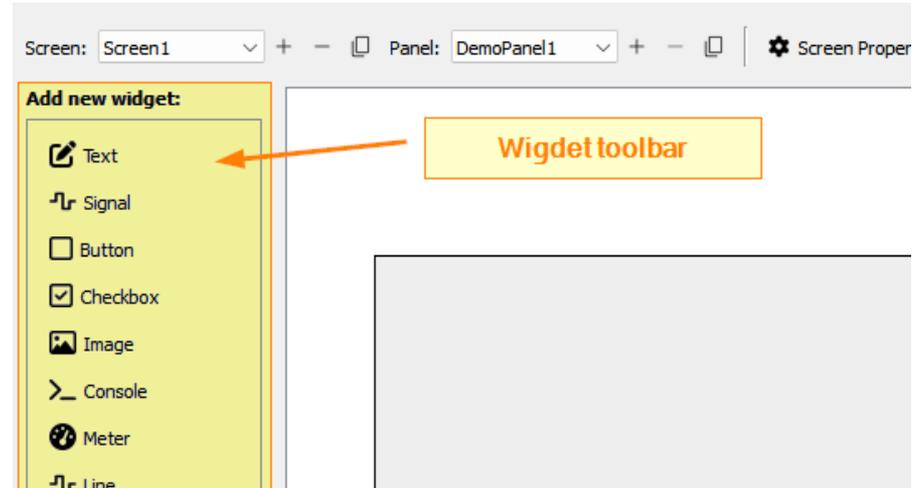
Nach Definition von Screen und Panel werden die Widgets auf dem Panel platziert.

Durch Anklicken eines Widgets in der Widget-Symbolleiste auf der linken Seite wird es auf dem Panel platziert und kann anschließend an die gewünschte Position verschoben werden.

Durch mehrmaliges Anklicken desselben Widgets in der Symbolleiste wird dem Widget ein "Stacked Widget indicator" hinzugefügt, der die Anzahl der durch das oberste Widget verdeckten Elemente anzeigt.

Hier wurden 5 Textbeschriftungen hinzugefügt.

Auf der rechten Seite wird für jedes ausgewählte Widget ein Eigenschaftsmenü mit den spezifischen Einstellungsmöglichkeiten angezeigt.



Back

Font type test:

Default Font <>

Default Font Bold

Default Mono Font

Default Mono Font Bold

CUSTOM FONT RESOURCE

Text color test:

Default Color (Black)

White Text

Red Text

Green Text

Blue Text

Text alignment test:

Default Alignment (Left)

Left Aligned

Centered

Right Aligned

Text wrap test:

Wrap Mode: Clip (XXXXXXXXXX)

Wrap Mode: Dots (XXXXXXXXXX...)

rap Mode: Scroll (XXXXXXXXXX)

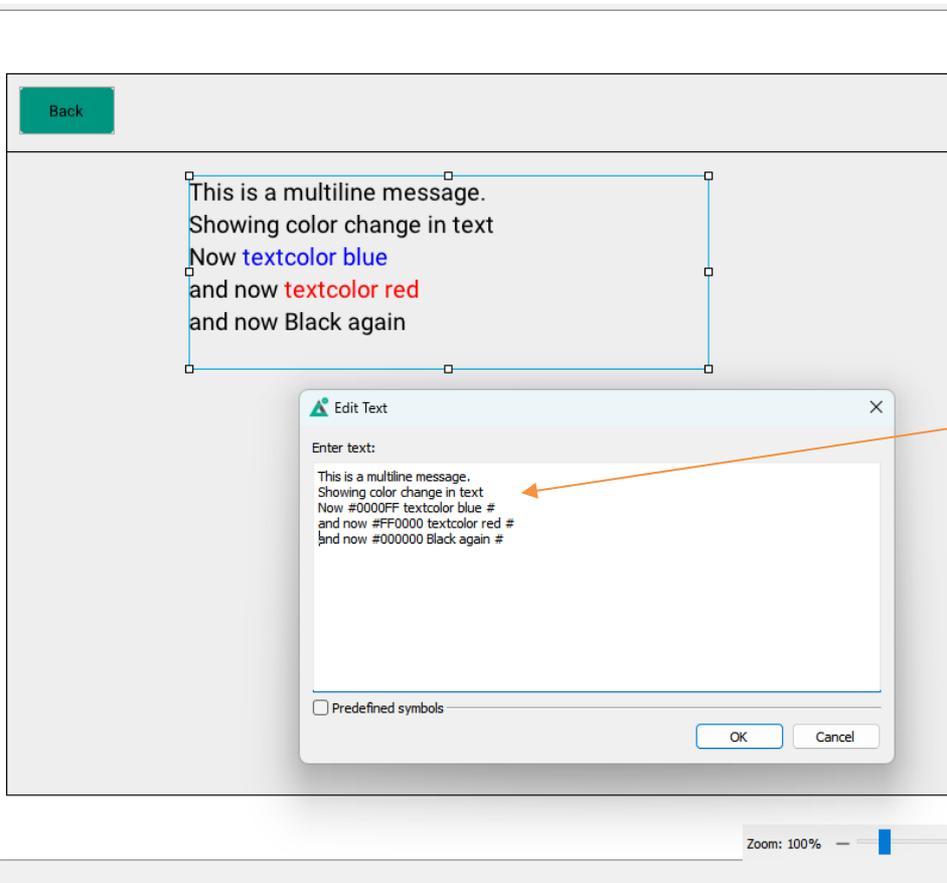
(XXXXXXXXXX) Wrap Mode: Scro

Wrap Mode:Expand (XXXXXXXXXX)

Properties:

Widget	
name	label
x	356
y	238
width	99
height	27
visible	<input checked="" type="checkbox"/> true
z_index	13
opacity	100%
bg_color	#FFFFFF
bg_opacity	0%
Text properties	
font	Default
font_size	24
text_color	#00FF00
text_alignment	Default
text_wrap_mode	Clip
Label	
text	Green Text
text_recolor	<input checked="" type="checkbox"/> true

- Neue Eigenschaften
- Background color
- Background opacity
- Extended label text editor
- text_recolor erlaubt Änderung der Textfarbe im Text



The screenshot shows a GUI editor with a 'Back' button and a text label. The label contains the text: "This is a multiline message. Showing color change in text. Now **textcolor blue** and now **textcolor red** and now Black again". An 'Edit Text' dialog box is open, showing the text with color codes: "This is a multiline message. Showing color change in text. Now #0000FF textcolor blue # and now #FF0000 textcolor red # and now #000000 Black again #".

Properties:

Widget	
name	label
x	152
y	84
width	436
height	162
visible	<input checked="" type="checkbox"/> true
z_index	0
opacity	100%
bg_color	#FFFFFF
bg_opacity	0%
Text properties	
font	Default
font_size	24
text_color	#000000
text_alignment	Default
text_wrap_mode	Clip
Label	
text	row #000000 Black again # ...
text_recolor	<input checked="" type="checkbox"/> true

Description:
Allows to change the color of parts of the text using a in-text formatter.

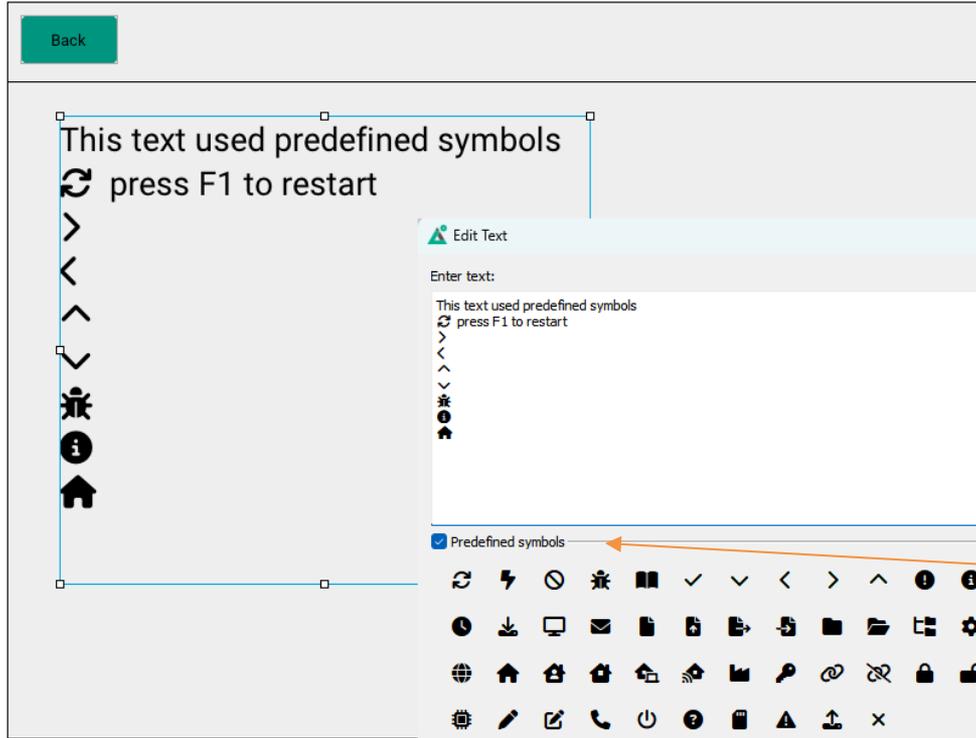
Format: "#RRGGBB #"
Example: "#FF0000 RedText # NormalText "
or when changing text by macro print command from SDF:
[GUI]
setprop widgetname "text"="#FF0000 RedText # NormalText"
If a text must contain a '#' character, just put '##'

Property text_recolor erlaubt Änderung der Textfarbe innerhalb eines Textes.

Die Farbänderungen werden den Text integriert.

Diese Checkbox aktiviert das recolor Feature.

Links Click auf den Property Namen öffnet dieses Hilfe Fenster.



Properties:

Widget	
name	label
x	42
y	92
width	436
height	383
visible	<input checked="" type="checkbox"/> true
z_index	0
opacity	100%
bg_color	#FFFFFF
bg_opacity	0%
Text properties	
font	Default
font_size	32
text_color	#000000
text_alignment	Default
text_wrap_mode	Clip
Label	
text	> \n<\n^<\n>\n#<\n!<\n...
text_recolor	<input checked="" type="checkbox"/> true

Erweiterter TextLabel Editor
Mit vordefinierten Icons

Eigenschaften mehrerer Elemente gleichzeitig ändern

Ändern Sie die Properties mehrerer Elemente gleichzeitig, indem Sie diese Elemente markieren und die Eigenschaft auf den neuen gemeinsamen Wert setzen, den sie alle haben sollen.

Hier wird die Breite aller markierten Textbeschriftungen auf 100 geändert.



The screenshot shows a GUI editor interface. On the left, a vertical column of five 'Textlab' widgets is selected, indicated by a yellow dashed border. On the right, the 'Properties' panel is open, showing a list of properties for the selected widgets. The 'width' property is highlighted in blue, and an orange arrow points from it to the selected widgets. The 'Properties' panel is structured as follows:

Widget	
name	label
x	63
y	[...]
width	100
height	30
visible	<input checked="" type="checkbox"/> true
z_index	[...]
opacity	100%
bg_color	#FFFFFF
bg_opacity	0%
Text properties	
font	Default
font_size	24
text_color	#000000
text_alignment	Default
text_wrap_mode	Clip
Label	
text	Textlabel
text_recolor	<input checked="" type="checkbox"/> true

Screen: Screen1 + - Panel: Unnamedpanel + -  Screen Properties  Resources  Editor Settings

Add new widget:

-  Label / Text
-  Signal
-  Button
-  Checkbox
-  Image
-  Console
-  Meter
-  Line
-  Frame



TestSignal1: 0

TestSignal2: 0

TestSignal3: 0

Properties:

Widget	
name	signal
x	307
y	254
width	127
height	27
visible	<input checked="" type="checkbox"/> true
z_index	2
opacity	100%
bg_color	#FFFFFF
bg_opacity	0%
Text properties	
font	Default
font_size	24
text_color	#000000
text_alignment	Default
text_wrap_mode	Expand
Signal	
signal	[1-LIN] TestSignal2 
default_text	0
display_format	TestSignal2: {VALUE} ...
signal_representation	Integer
Signal - Interact	
interactable	<input type="checkbox"/> false
hotkey	None
dialog_title	
live_update	<input type="checkbox"/> false
live_update_step	1
allowed_input_ranges	

Signal Verknüpfung mit der SDF

Signal representation

Interactive Section

Hinweis: Im Gegensatz zu den bisherigen GUI-Elementen ist die neue GUI global für alle im SDF enthaltenen Busabschnitte, so dass Signale aus verschiedenen Busabschnitten im gleichen Panel angezeigt werden können.

Signalwerte können auf unterschiedliche Weise dargestellt werden.

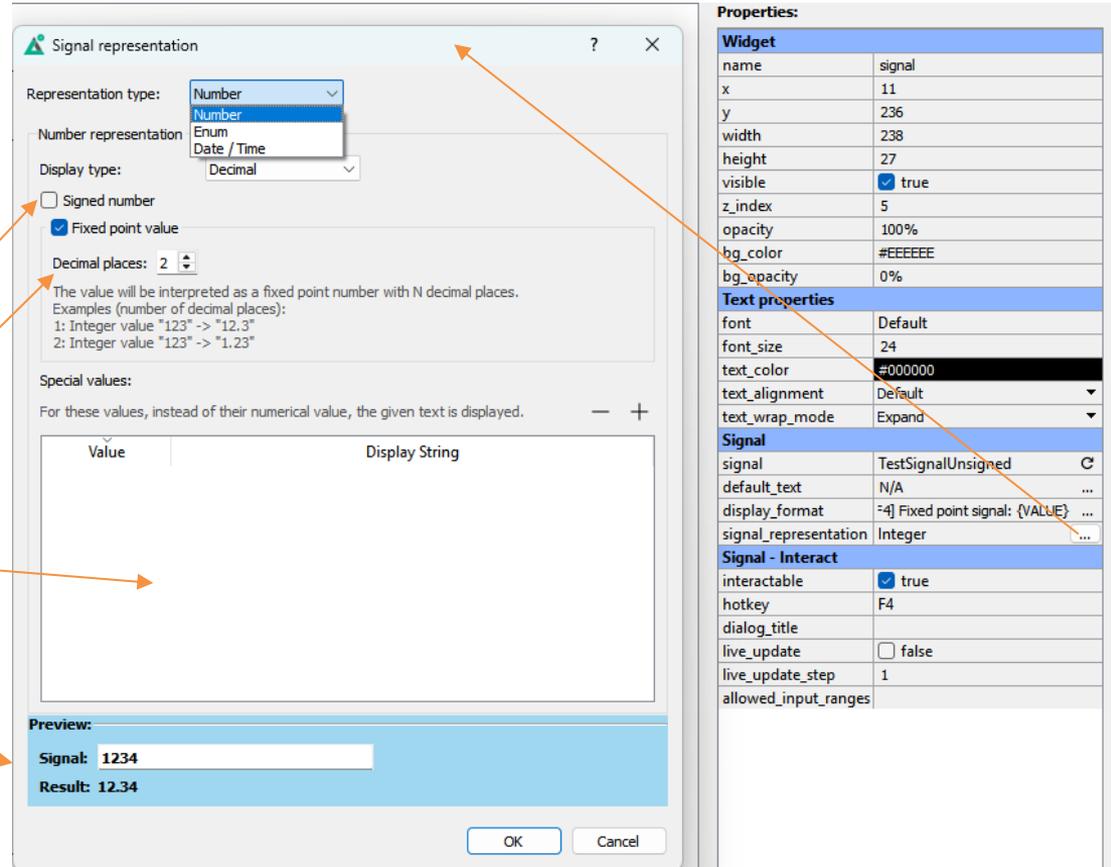
- Nummer
- Enum
- Als konvertierte Zeit/Datumsdarstellung.

Für die Zahlendarstellung kann die Interpretation als vorzeichenbehaftete Ganzzahl und eine Festkommadarstellung gewählt werden.

Bei der Darstellungsart Nummer können Sonderwerte definiert werden, für die dann anstelle des Wertes ein String angezeigt wird.

Z.B. OFF für 0 und ON für 1 usw.

Die Vorschau zeigt den angezeigten Wert für den gegebenen Signalwert an.



Signal representation

Representation type: **Number**

Number representation

Display type: **Decimal**

Signed number

Fixed point value

Decimal places: **2**

The value will be interpreted as a fixed point number with N decimal places.
 Examples (number of decimal places):
 1: Integer value "123" -> "12.3"
 2: Integer value "123" -> "1.23"

Special values:

For these values, instead of their numerical value, the given text is displayed.

Value	Display String

Preview:

Signal: **1234**

Result: **12.34**

Properties:

Widget	
name	signal
x	11
y	236
width	238
height	27
visible	<input checked="" type="checkbox"/> true
z_index	5
opacity	100%
bg_color	#EEEEEE
bg_opacity	0%
Text properties	
font	Default
font_size	24
text_color	#000000
text_alignment	Default
text_wrap_mode	Expand
Signal	
signal	TestSignalUnsigned
default_text	N/A
display_format	[-4] Fixed point signal: {VALUE}
signal_representation	Integer
Signal - Interact	
interactable	<input checked="" type="checkbox"/> true
hotkey	F4
dialog_title	
live_update	<input type="checkbox"/> false
live_update_step	1
allowed_input_ranges	

Back

No Signal: N/A

Default: N/A

[F2] Unsigned signal: N/A

[F3] Signed signal: N/A

[F4] Fixed point signal: N/A

[F5] 16 Bit Hex signal: N/A

[F6] Enum signal: N/A

Time signal: N/A

Date signal: N/A

Date & Time signal: N/A

Live update: N/A

Live update Enum: N/A

Set 0 Set 1 Set 2 Set Invalid

Widget	
name	signal
x	11
y	316
width	191
height	27
visible	<input checked="" type="checkbox"/> true
z_index	8
opacity	100%
bg_color	#FF0000
bg_opacity	0%
Text properties	
font	Default
font_size	24
text_color	#000000
text_alignment	Default
text_wrap_mode	Expand
Signal	
signal	DeviceLabelExpandState
default_text	N/A
display_format	[F6] Enum signal: {VALUE}
signal_representation	Enum
Signal - Interact	
interactable	<input checked="" type="checkbox"/> true
hotkey	F6
dialog_title	
live_update	<input type="checkbox"/> false
live_update_step	1
allowed_input_ranges	

New property section
Interact

Enables value editor in GUI
Value Editor can either be
opened

- By Mouse click in SimpleMenu or WebGui
- By Hotkey
- By focus & select method on device

Note: In contrast to the previous GUI elements, the new GUI is global for all bus sections contained in the SDF, allowing signals from different bus sections to be displayed in same panel.

Standardmäßig geben Werteditoren den geänderten Wert erst beim Schließen des Editors an das SDF weiter. Erst dann wird der neue Wert aktiv und z.B. auf dem LIN-Bus gesendet.

Manchmal eine sofortige Aktualisierung der Signaländerung gewünscht. Dies wird durch die Funktion `live_update` ermöglicht.

In diesem Fall sieht der Signalwert-Editor so aus.

Für jeden Linksklick auf die Auf-/Ab-Pfeile (SimpleMenu oder Web Gui) oder bei jedem Druck auf die Auf-/Ab-Tasten auf der Gerätetastatur, wird der Wert um die angegebene Schrittweite inkrementiert oder dekrementiert.

Die Werte sind durch die Definition des zulässigen Bereichs begrenzt.

Wenn kein Bereich definiert ist, bestimmt die Signalgröße die Minimal- und Maximalwerte.

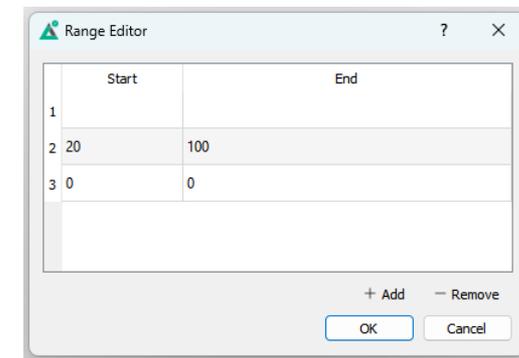
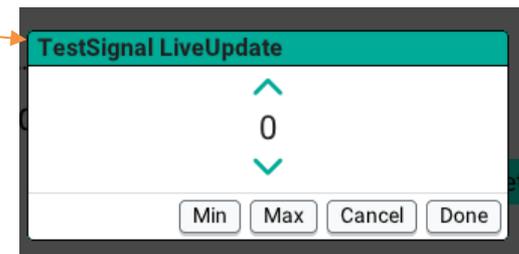
Die Schaltflächen Min und Max ermöglichen eine schnelle Einstellung des minimalen und maximalen Wertes.

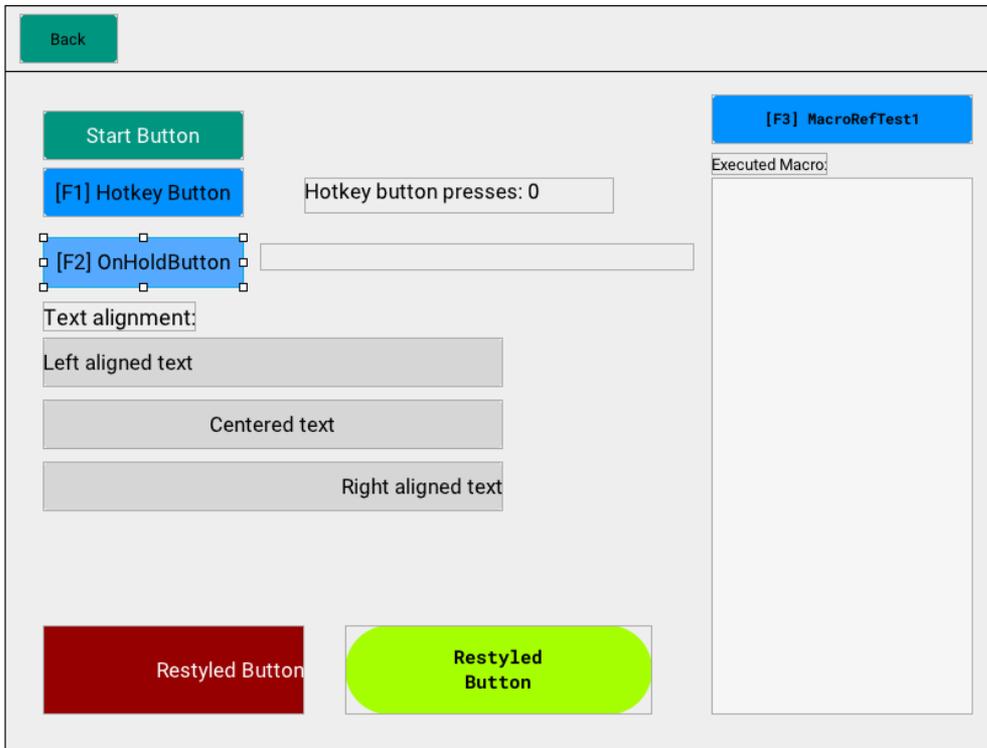
Der zulässige Bereich kann durch Kombination mehrerer Bereiche konfiguriert werden.

Hier ein Beispiel für einen kombinierten Bereich, der Werte für $\{0, 20, 21, 22, \dots, 100\}$ erlaubt.

Für die Feinabstimmung eines Signalwertes bietet `live_update` eine praktikable Lösung.

Signal - Interact	
<code>interactable</code>	<input checked="" type="checkbox"/> true
<code>hotkey</code>	None
<code>dialog_title</code>	TestSignal LiveUpdate
<code>live_update</code>	<input checked="" type="checkbox"/> true
<code>live_update_step</code>	10
<code>allowed_input_ranges</code>	0,20-100





The screenshot displays a GUI editor interface with several interactive elements:

- Back**: A green button in the top left corner.
- Start Button**: A green button.
- [F1] Hotkey Button**: A blue button with a hotkey label.
- [F2] OnHoldButton**: A blue button with a hotkey label and a small square icon next to it.
- Text alignment:** A section showing three text boxes: "Left aligned text", "Centered text", and "Right aligned text".
- Hotkey button presses: 0**: A text label next to a small input field.
- [F3] MacroRefTest1**: A blue button with a hotkey label.
- Executed Macro:**: A large white text area below the [F3] button.
- Restyled Button**: A red rectangular button.
- Restyled Button**: A bright green rounded rectangular button.

Properties:

Widget	
name	button
x	30
y	186
width	163
height	41
visible	<input checked="" type="checkbox"/> true
z_index	5
opacity	100%
Text properties	
font	Default
font_size	20
text_color	#000000
text_alignment	Center
text_wrap_mode	Clip
Button	
background	#55AAFF
text	[F2] OnHoldButton
corner_radius	7
border_width	0
border_color	#000000
Button - Interact	
interactable	<input checked="" type="checkbox"/> true
panel_id	[None]
hotkey	F2
on_press_macro	[1-LIN] OnHoldButtonPressed
on_release_macro	[1-LIN] OnHoldButtonReleased

Position und Grösse

Textfont, Größe, Farbe und Alignment

Interact Section mit 2 Macroverknüpfungen für Press and Release

Um die Simulation von Mausclicks auf dem Gerätedisplay zu ermöglichen, wurde der Focus & Select-Modus eingeführt.

Wenn ein Panel mit mindestens einem interaktiven Widget ohne Hotkey-Zuweisung angezeigt wird, lässt sich der Focus & Select Modus auf 2 Weg aktivieren:

- Langer Druck auf die Menütaste oder
- Kurzer Druck auf die Menütaste und Auswahl Focus & Select mit der Enter (F3) Taste

Im Modus Fokus & Select werden die Tasten F1 ...F6 auf ihre sekundären Funktionen umgeschaltet (grüne Beschriftung)

F2/F4,F5 und F6 werden verwendet, um das nächste interaktive Element auf dem Bildschirm zu fokussieren.

Die Taste F3 löst das Element aus, öffnet einen Werteditor (Signal) oder führt ein Makro (Button) aus.

Der Modus Fokus & Select wird mit der Taste ESC(F1) beendet.

Während der Focus & Select-Modus aktiviert ist, werden die Tastenaktionen nicht an SDF zur Verarbeitung weitergeleitet (Systemsignale SYSDIGIN1-6, digitale Eingangsereignisse usw.)

Das Checkbox Widget ist hauptsächlich für die Anzeige oder Eingabe von Binärwerten gedacht.

Back

Unchecked

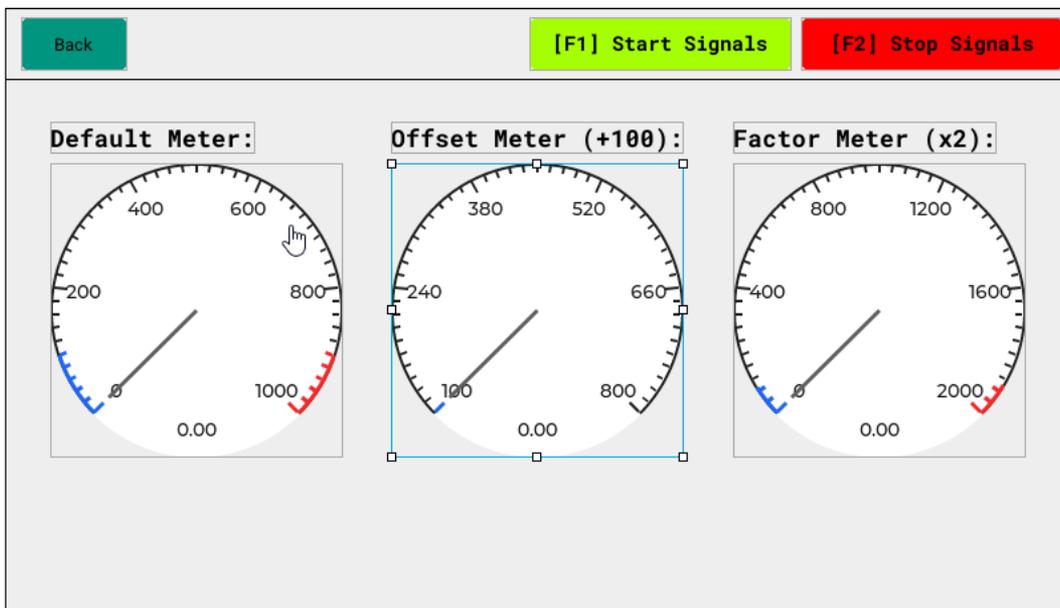
Referenced signal: 0

Styled Checkbox

Properties:

Widget	
name	checkbox
x	49
y	130
width	94
height	22
visible	<input checked="" type="checkbox"/> true
z_index	3
opacity	100%
Text properties	
font	Default
font_size	16
text_color	#000000
text_alignment	Default
text_wrap_mode	Clip
Checkbox	
signal	CheckboxSignal C
checked_text	Checked ...
unchecked_text	Unchecked ...
checked_range_list	1-65535
check_set_value	1
uncheck_set_value	0
Checkbox - Interact	
interactable	<input checked="" type="checkbox"/> true
hotkey	None
Checkbox - Style	
unchecked_bg_color	#FFFFFF
checked_bg_color	#FFFFFF
tick_color	#00957F
border_color	#000000

Das Meter Widget ermöglicht es, einen Signalwert in einer Tachodarstellung anzuzeigen. Einige neue Eigenschaften wurden diesem Widget hinzugefügt.



Properties:

Widget

name	signal_meter
x	289
y	115
width	220
height	220
visible	<input checked="" type="checkbox"/> true
z_index	7
opacity	100%

SignalMeter

signal	MeterTestSignal	C
signal_representation	Integer	...
unit		...
scale_start	100	
scale_end	800	
lower_limit	100	
upper_limit	900	
factor	1.000000	
offset	100.000000	

SignalMeter - Interact

interactable	<input type="checkbox"/> false
hotkey	None
dialog_title	
live_update	<input type="checkbox"/> false
live_update_step	1
allowed_input_ranges	

Das Line-Widget kann zur Dekoration der GUI und zur Abtrennung von Bereichen auf dem Panel verwendet werden.

Es gibt keine Laufzeitfunktion oder SDF-Aktion im Zusammenhang mit diesem Widget.

Back

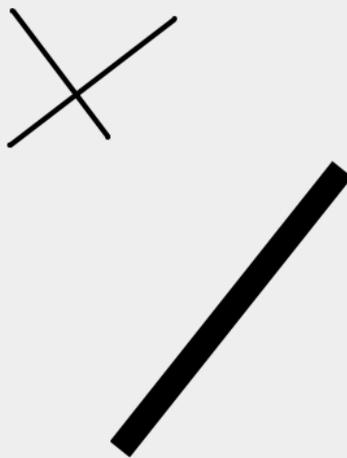
Horizontal line:



Vertical line:



Free line:

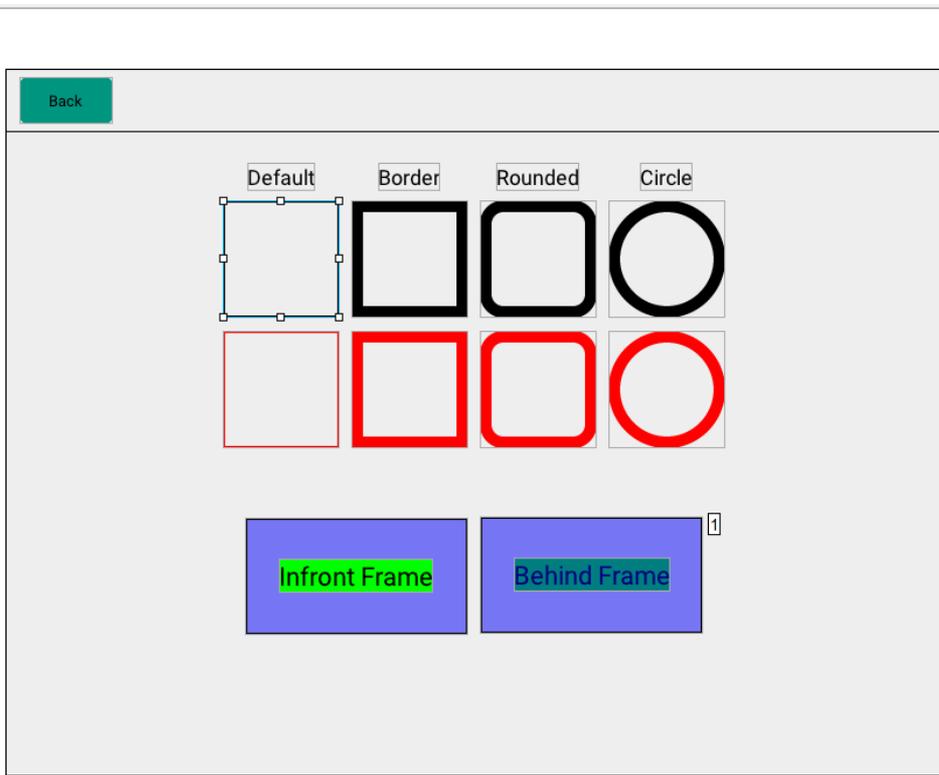


Properties:

Widget	
name	line
x	223
y	135
visible	<input checked="" type="checkbox"/> true
z_index	9
opacity	100%
Line	
orientation	Vertical
start_x	233
start_y	145
end_x	233
end_y	402
line_width	20
line_color	#999999
line_rounded	<input checked="" type="checkbox"/> true

Das Frame-Widget kann zur Dekoration der GUI und zur Trennung von Bereichen auf dem Panel verwendet werden.

Es gibt keine Laufzeitfunktion oder SDF-Aktion im Zusammenhang mit diesem Widget.



Properties:

Widget	
name	frame
x	185
y	111
width	100
height	100
visible	<input checked="" type="checkbox"/> true
z_index	3
opacity	100%
bg_color	#FFFFFFF
bg_opacity	0%
Frame	
border_width	2
border_color	#000000
border_radius	0

GUI Element Kontrolle zur Laufzeit

Es gibt mehrere Wege, wie sich die GUI während der SDF-Ausführung aktualisiert.

Signal-, Meter- und Checkbox-Widgets

Diese Widgets sind typischerweise mit einem SDF Signal verlinkt und zeigen den aktuellen Signalwert an.

Signal- und Button-Widgets

mit `interactable = true` erlauben Signaländerungen oder Makroausführung zur Laufzeit.

Konsolen-Widgets

zeigen die Reportmonitorausgaben aus der SDF in einem scrollbaren Listenfenster an.

Widget-Eigenschaften ändern

Alle Eigenschaften eines Widgets können zur Laufzeit geändert werden: Text, Textfarbe, Sichtbarkeit, etc.

MessageBox

Dieser Mechanismus ermöglicht die Übertragung einer Nachricht vom SDF zur GUI. Außerdem ist optional eine Benutzerrückmeldung möglich.

Änderungen von Widget Properties

Es gibt eine spezielle Syntax für den Befehl Macro Print mit dem Target Debug report, um die Eigenschaften des Widgets zu ändern.

[GUI]

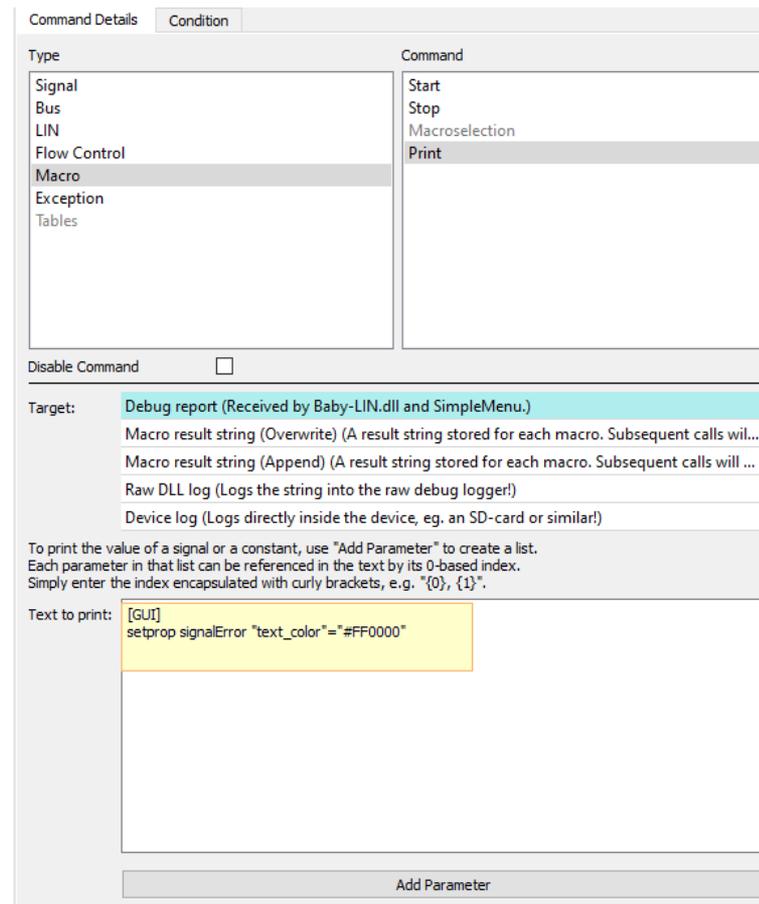
```
setprop signalError "text_color" = "#FF0000"
```

[GUI] und setprop sind reservierte Schlüsselwörter.

signalError ist hier der Name des Widgets. (Groß- und Kleinschreibung beachten!)

Auf den Widget-Namen folgen der Name und der neue Wert der Eigenschaft (Property) , beide in Anführungszeichen eingeschlossen und durch ein '='-Zeichen getrennt.

Wenn der Widget Name Blanks enthält muss dieser auch in Anführungszeichen eingeschlossen werden.



Command Details Condition

Type

- Signal
- Bus
- LIN
- Flow Control
- Macro
- Exception
- Tables

Command

- Start
- Stop
- Macroselection
- Print

Disable Command

Target: Debug report (Received by Baby-LIN.dll and SimpleMenu.)

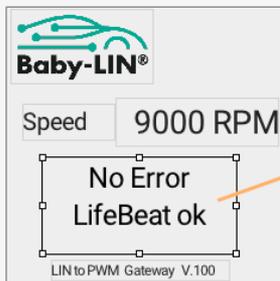
- Macro result string (Overwrite) (A result string stored for each macro. Subsequent calls will...
- Macro result string (Append) (A result string stored for each macro. Subsequent calls will ...
- Raw DLL log (Logs the string into the raw debug logger!)
- Device log (Logs directly inside the device, eg. an SD-card or similar!)

To print the value of a signal or a constant, use "Add Parameter" to create a list. Each parameter in that list can be referenced in the text by its 0-based index. Simply enter the index encapsulated with curly brackets, e.g. "{0}", "{1}".

Text to print: [GUI] setprop signalError "text_color" = "#FF0000"

Add Parameter

Properties Change Beispiel:
 [GUI]
 setprop signalError "text_color"="#FF0000"



Properties:

Widget	
name	signalError
x	30
y	125
width	164
height	82
visible	<input checked="" type="checkbox"/> true
z_index	1
opacity	255
Font	
text_font	Default
text_size	28
text_color	... #000000
text_align	Center
text_wrap	Wrap
Signal	
signal	... [1] HELLA_LIN2PWM_co
default_text	... No Error\nLifeBeat ok
display_format	... {VALUE}
editable	<input type="checkbox"/> false
signal_representation	... Enum

Weitere Beispiele

[GUI]
 setprop signalError "visible" = "0"

Or
 [GUI]
 setprop signalError "visible" = "1"
 setprop signalError "text_color" = "#00FF00"

Mehrere setprop Anweisungen können in einem Macro Print Command zusammengefasst werden.

Für Screens mit mehreren Paneln existieren 2 Methode zur Umschaltung:

- Über das Button-Widget mit zugewiesener Panel-ID.
Am Gerät muss dafür zuerst in den Focus & Select Modus gewechselt werden, um die Schaltfläche im Display auszulösen. Alternativ kann man dem Button auch einen Hotkey auf der Baby-LIN Tastatur zuweisen.

- Via MacroPrint Macro Kommando aus der SDF heraus
[GUI]
setprop Panel1 "active" = "1"
oder
[GUI]
setprop Panel2 "active" = "1"

Es genügt, nur das gewünschte Panel auf aktiv = 1 zu setzen, alle anderen werden automatisch deaktiviert.



Properties:

Widget	
name	buttonSwitchToPanel2
x	602
y	27
width	122
height	40
visible	<input checked="" type="checkbox"/> true
z_index	8
opacity	100%
Text properties	
font	Default
font_size	16
text_color	#FFFFFF
text_alignment	Center
text_wrap_mode	Wrap
Button	
background	#00957F
text	PanelSwitch per Button ...
corner_radius	7
border_width	0
border_color	#000000
Button - Interact	
interactable	<input checked="" type="checkbox"/> true
panel_id	Panel2
hotkey	None
on_press_macro	[None]
on_release_macro	[None]

SessionConf v2.36.1 - [C:/documents/presentations/Gui-Editor-Webinar/GUI_Demo_PanelSwitchFromSDF.sdf]

File Edit View Tools Help

Hide expert settings Required SDF version: v3.24 FID: 0x0 PID: Hex 0x80

SDF Version 3

Macro number 0

Name PanelSwitch

Parameter count 0

Comment

Label	Condition	Command	Comment
0		CurrentPanel = CurrentPanel + 1	
1	If Signal CurrentPanel > 2	CurrentPanel = 1 (0x01)	
2		Print on Debug report: " [GUI] setprop Panel(0) "active"="1" ", Parameter: [0] = CurrentPanel	

Section properties

- Bus description
- Emulation
- Tables
- Virtual signals
- Signalfunctions
- Protocols
- GUI-Elements (SimpleMenu/HARP etc)
- Macros
 - [0] PanelSwitch
 - Macroselection
 - Events
 - Device-specific options
 - GUI Definition

Command Details

Type

- Signal
- Bus
- LIN
- Flow Control
- Macro
- Exception
- Tables

Command

- Start
- Stop
- Macroselection
- Print

Disable Command

Target:

- Debug report (Received by Baby-LIN.dll and SimpleMenu.)
- Macro result string (Overwrite) (A result string stored for each macro. ...)
- Macro result string (Append) (A result string stored for each macro. ...)
- Raw DLL log (Logs the string into the raw debug logger!)
- Device log (Logs directly inside the device, eg. an SD-card or similar!)

To print the value of a signal or a constant, use "Add Parameter" to create a list. Each parameter in that list can be referenced in the text by its 0-based index. Simply enter the index encapsulated with curly brackets, e.g. "{0}", "{1}".

Text to print:

```
[GUI]
setprop Panel(0) "active"="1"
```

Delete Parameter 0 Signal

Signal CurrentPanel

Siehe auch
GUI_Demo_panelSwitch.sdf

Messagebox-Funktion

Für Anzeigen aus der SDF im Sinne einer Messagebox gibt es eine weitere spezielle Syntax für den Macro Print-Befehl

[GUI]

```
msgbox open "Das ist der Anzeigetext" 1:23 " Select Yes" "Select No"
```

1:23 definiert Sektion und Signalindex für ein Rückmeldesignal, in dem der Index der gewählten Option (1...n) gespeichert wird und im SDF ausgewertet werden kann.

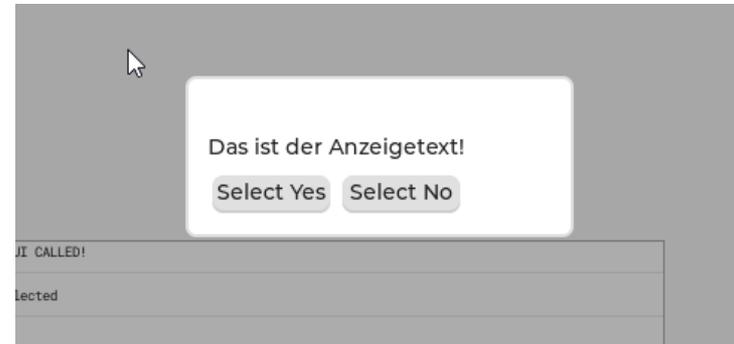
Die beiden Textstrings "Select Yes " und "Select No " sind die Bezeichnungen der beiden Schaltflächen. Sie können derzeit bis zu 10 Schaltflächen definieren.

Target:

- Debug report (Received by Baby-LIN.dll and SimpleMenu.)
- Macro result string (Overwrite) (A result string stored for each macro. Subsequent...
- Macro result string (Append) (A result string stored for each macro. Subsequent ...
- Raw DLL log (Logs the string into the raw debug logger!)
- Device log (Logs directly inside the device, eg. an SD-card or similar!)

To print the value of a signal or a constant, use "Add Parameter" to create a list. Each parameter in that list can be referenced in the text by its 0-based index. Simply enter the index encapsulated with curly brackets, e.g. "{0}", "{1}".

Text to print: [GUI]
msgbox open "Das ist der Anzeigetext!" 1:23 "Select Yes" "Select No"



The screenshot shows a white message box with a grey border on a dark grey background. The text inside the box reads "Das ist der Anzeigetext!". Below the text are two buttons: "Select Yes" and "Select No". Below the message box, there is a grey bar with the text "JI CALLED!" and another grey bar with the text "lected".

Varianten der Messagebox Syntax

Message Box öffnen
ohne Signalindex und Feedbacksignal

[GUI]

msgbox open "Message Box closes in 5s"

Update Message Box Text:

[GUI]

msgbox update "Message Box closes in {0}s"

Message Box schliessen:

[GUI]

msgbox close

Macro number	2		
Name	Msgbox Update		
Parameter count	0		
Comment			
Label	Condition	Command	
0		Print on Debug report: " [GUI] msgbox open "Message Box closes in 5s" "	
1		popupcloseTimer = 5 (0x05)	
2	loop	Delay 1000ms	
3		popupcloseTimer = popupcloseTimer - 1	
4		Print on Debug report: " [GUI] msgbox update " Message Box closes in {0}s" ", Parameter: {0} = popupcloseTimer	
5	If Signal popupcloseTimer > 0	Jump to "loop"	
6		Print on Debug report: " [GUI] msgbox close "	

- Bilder beim Positionieren im Panel resizen
- Support für weitere Bildformate
- Optimierung der Bildspeicherung
- Signalplotter
- Ihre Ideen und Wünsche