GRAF-SYTECO

Manual

GDSDesigner

Language: english

Operate

Visualization

Control

Systems
Technical
Communication

H301A1

1

Chapter 1
Symbols

1.1 Warning notices

Classification of warning notices

The warning notices differentiate between three types of dangers indicated
by the following signal words:
*   **Caution** warns of material damages.
*   **Warning** warns of bodily injuries.
*   **Danger** warns of a danger to life.

Layout of the warning notices

Signal words

Type and source of danger!
▶ Action to avoid the danger.

1.2   Other symbols

Instructions

Layout of instructions:

▸ Instruction to do something.

Result of the action, if necessary.

Lists

Layout of bulleted lists:
• List level 1
    • List level 2

Layout of numbered lists:
1. List level 1
2. List level 1
    2.1 List level 2
    2.1 List level 2

Markings

Menu items, buttons and references to on-screen text are marked as follows in the text:
• *Menu item*
• *Button*
• *Reference to on-screen text*

2 Chapter 2
Function

2.1 Intended Use

▸ Use GDSDesigner exclusively for creating visualizations for control sets of the series listed below:
  - MCQ
  - ARGOS
  - OEM control sets authorized by GRAF-SYTECO

2.2 Function

GDSDesigner is a component of GrafDesignStudio (GDS) and is installed along with it.

GDSDesigner is a graphic design tool for creating visualizations to display on control sets.

The display of the visualization in GDSDesigner is identical to the display on the control set.

Visualization and programming

⚠️ **Caution**	**Data loss or unforeseeable effects!** Parallel programming of control processes in GDSLogic and C will cause data loss and unforeseeable effects due to asynchronous processing of program flows. ▶ Program control processes exclusively in GDSLogic or C.

Control with C

If the control task is programmed in C

- the programming is done from GDSDesigner in C Editor GDCSEdit.
- the visualization is created in GDSDesigner.

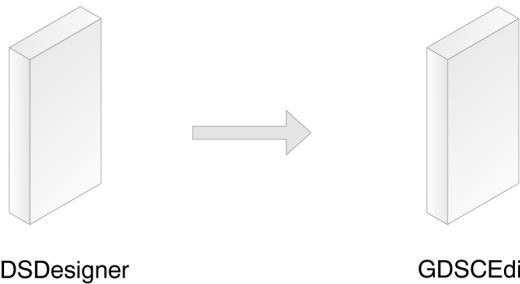GDSDesigner   GDSCEdit

Fig. 1   Visualization and control with GDSLogic

Control with GDSLogic

If the control task is programmed with GDSLogic

- the programming is done in the GDSLogic program.
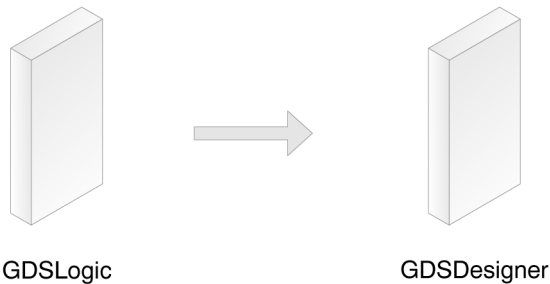- the visualization is created from GDSLogic in the GDSDesigner program.

GDSLogic   GDSDesigner

Fig. 2   Visualization and control with GDSLogic

3 Chapter 3
Installation

3.1 Requirements

▸ Observe the following minimum requirements for the installation:

- • operating system: Windows® 2000/XP/Vista/7
- • hard disk storage unit minimum 700 MB
- • processor minimum Pentium
- • RAM: minimum 512 MB
- • screen resolution: minimum 1024 x 768

3.2 Installation

▸ Put installation CD into CD-ROM drive of computer.

If autostart is activated:

Installation program will start automatically.

If autostart is deactivated:

▸ Open CD-ROM drive in Windows®-Explorer.
▸ Double-click on file *setup.exe*.

Installation program starts.

▸ Select folder of desired installation language in directory tree.
▸ Open folder *Software*.
▸ Select *GrafDesignStudio (GDS)*.
▸ Click on button *Install*.

The installation program for *GrafDesignStudio* starts.

Fig. 3   Installation

▶ Follow instructions of installation program.

4

Chapter 4
Program surface
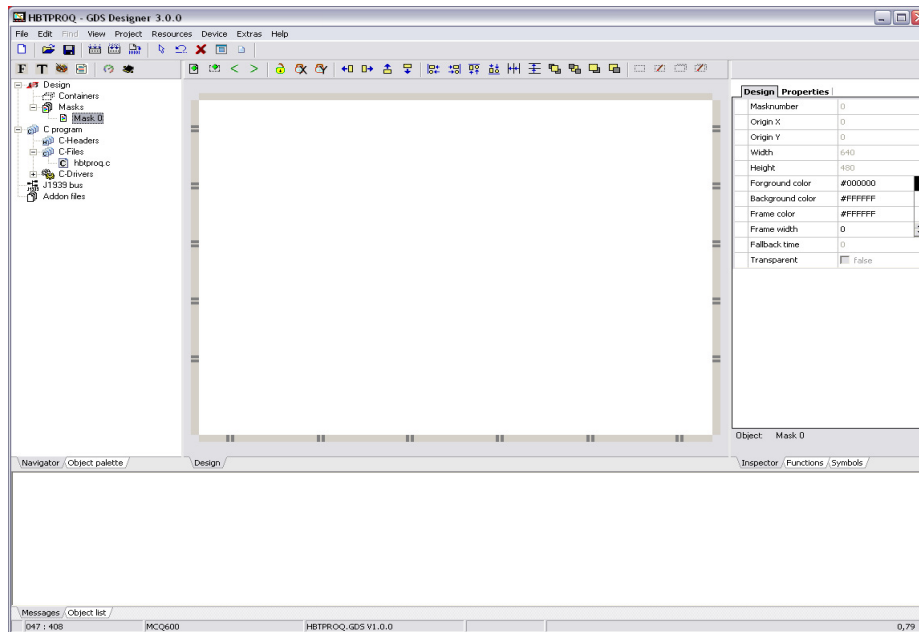
Fig. 4   Program surface

(1) Menu bar
(2) Button bar
(3) Project navigator/object palette
(4) Display
(5) Object list/output
(6) Object inspector

4.1 Menu bar

Contains menus for project management and manipulation of highlighted objects.
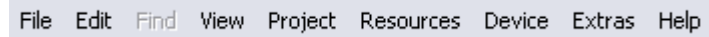
Fig. 5   Menu bar

4.2 Button bar

The button bar contains buttons for project management and manipulation of highlighted objects.
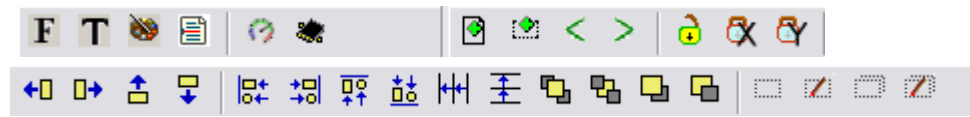
Fig. 6   Button bar

4.3 Display

Shows how the visualization is displayed.
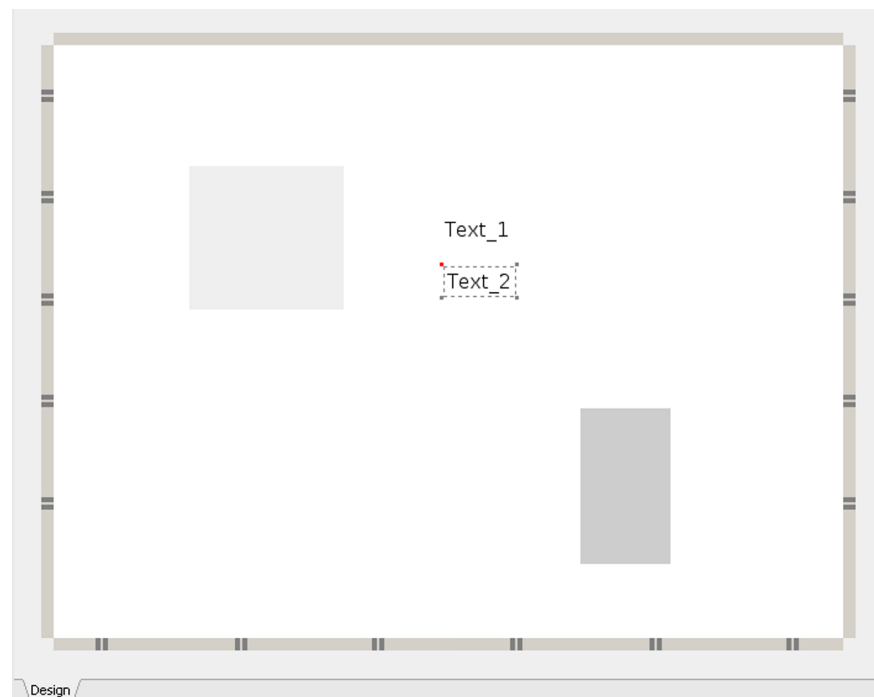
Fig. 7   Menu bar

The edited mask or the edited container is displayed in the same way as on the control set.

Editing highlighted objects is possible using the buttons in the button bar (*see 4.2 Button bar*) or via the menus (*see 5 Menu structure*).

4.4   Project navigator

The project navigator manages the project in a tree structure consisting of containers, masks and their visual objects, files of the C program, drivers and supplementary files.
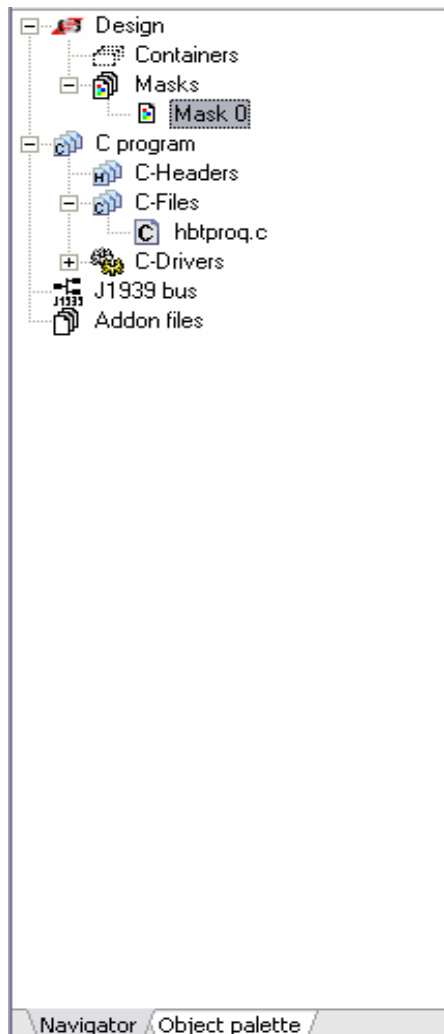
Fig. 8   Project navigator

Objects highlighted in the tree structure are marked by a frame on the display. The object inspector shows the object properties and allows modifying them.

The following actions are possible via the context menu:

• Deleting masks, containers and objects
• Aligning masks, containers and objects (*see 5.3 Find*)
• Renaming masks, containers and objects
• Changing the object ID of masks, containers and objects

The object order within the tree structure is identical to the objects' selection order in the control set.

4.5   Object inspector

The object inspector shows the properties of the selected object.

Fig. 9   Object inspector

The object inspector allows configuring the properties of an object.

Which properties are configurable depends on the object highlighted (*see 10 Object library*).

4.6   Object list/output

The object list shows a list of the project's objects.

The message window shows error messages of the C compiler and messages of the program.

Fig. 10 Object list

Note
Changing or setting is not possible in the object list.

For information on objects, *see 10 Object library*.

4.7 Object library

Allows selecting and pasting objects into the project.
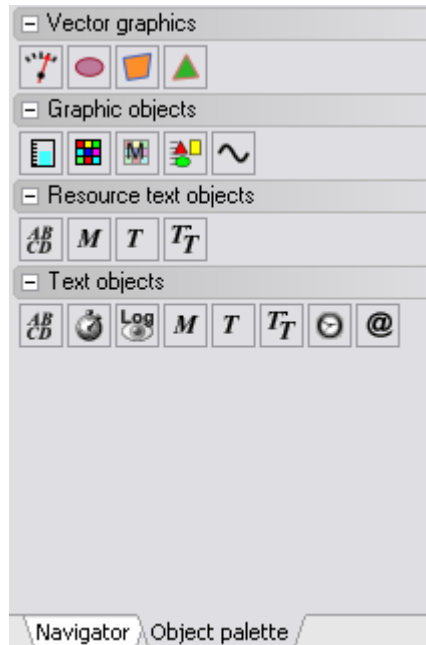
Fig. 11   Object palette

For descriptions of individual objects, *see 10 Object library*.

5

Menu structure

5.1 File

New

New mask	Create new mask.
New container	Create new container.
C file	Create new C file.
Header file	Create new header file.
New project	Create new project.
Open project	Open existing project.
Recent projects	Shows list of projects last opened.
Open	Open C editor.
	C editor allows writing and compiling programs running on the control set.
Save	Save project.
Save as	Save project with a different name or in a different directory.
Quit	Quit GDSDesigner.

Note

The menu items Open project, Recent projects and Save as are only available if GDSDesigner has been started using the Windows® start menu or a desktop shortcut.

5.2   Edit

Align

Move to left	Highlighted objects are moved to the left by one pixel
Move to right	Highlighted objects are moved to the right by one pixel
Move up	Highlighted objects are moved up by one pixel
Move down	Highlighted objects are moved down by one pixel
Edges left	Align left edges of several highlighted objects.
Edges right	Align right edges of several highlighted objects.
Edges top	Align top edges of several highlighted objects.
Edges bottom	Align bottom edges of several highlighted objects.
Space evenly horizontal	Place several highlighted objects in such a way that the spaces between them are identical in horizontal direction.
Space evenly vertical	Place several highlighted objects in such a way that the spaces between them are identical in vertical direction.
To background	Put highlighted object into background.
To foreground	Put highlighted object into foreground.
Toward foreground by one level	Move highlighted object toward foreground by one level.
Toward background by one level	Move highlighted object toward background by one level.

Undo	Undo most recent change in project.

Copy

Copy	Copy properties of highlighted object to clipboard.
Copy as image	Copy current display to clipboard as an image.

Paste	Paste object from clipboard to mask.
Delete	Delete highlighted object, mask or container.
Automatic Container	Show/Hide available container in mask.
Rename	Change name of highlighted object.
Change object ID	Change ID of highlighted object.

5.3   Find

Find	Find a word or command.
Replace	Replace a word or command.
Grep	Find a word or command in more than one file.

Note
The Design tab does not provide the Find option.

5.4   View

Zoom	Zoom in on or out of display contents as desired.
Selected	Display highlighted object in operating mode *Selected*.
All selected	Display all objects in operating mode *Selected*.
In input mode	Display highlighted object in operating mode *Edited*.
All in input mode	Display all objects in operating mode *Edited*.

Note
The C code tab does not provide this view.

5.5   Project

Make	Translate changed C files.
Make clean	Delete dependencies of C files.
Make rebuild	Delete dependencies of C files and translate changed C files.
Import	Import earlier project with filename extension .XML.
Binary export	Export project to control set or USB stick.
Project information	Show or enter project information.
Pack project	Compress project.

Note
Project information is transferred to the control set, where it is available e. g. for an external control.

5.6   Resources

Every loaded resource has its own configuration dialog and adds a menu item in the *Resources* menu to open the configuration dialog.

For information on resources, *see 6 Resources*.

5.7   Device

Device parameters   Open configuration dialog of loaded device.

For information on device parameters, *see 7 Device settings*.

5.8   Extras

Presettings	Open presettings dialog of GDSDesigner (*see 8 Presettings*).
Scale generator	Open scale generator (*see 11 Scale generator*).
MCM module configuration	Open MCM module configuration (*see Manual MCM Utility Software*).

5.9   Help

Help	Show help file of GDSDesigner.
Documentation	
GDS	Selection of help documents
ITE	Selection of help documents
Kommunikation.pdf	
Version information	Show information on program version and versions of GDSDesigner components.
Online update	Update GDSDesigner.
	▸ Make sure that an internet connection is established.

6

Resources

Resources allow a central management of the following project contents:

- Variables (*see 6.1 Variable resources*)
- Bitmaps (*see 6.2 Bitmap resource*)
- Texts (*see 6.3 Text resource*)
- Standard fonts (*see 6.4 True Type Font Resource*)

6.1 Variable resources

Allows managing project variables.

To call up variable resource:

▸ Select *Variable resource* in menu *Resources*.
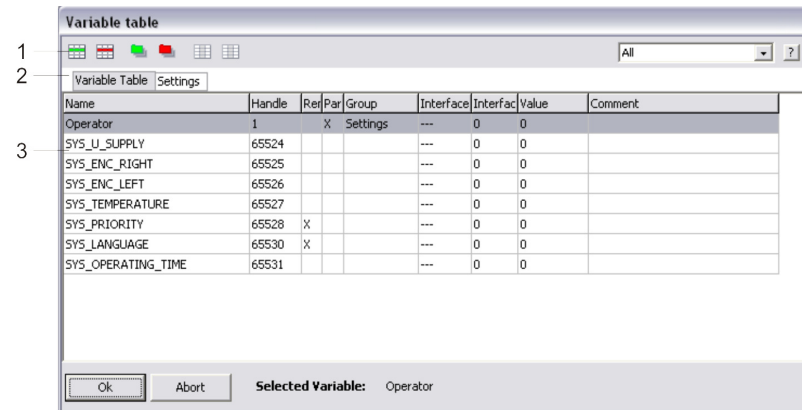
Fig. 12   Variable resource

(1) Button bar
(2) Parameter groups
(3) Variables

For information on managing variables and using the variable resource, *see 9.5 Variables*.

6.2 Bitmap resource

The bitmap resource allows managing all bitmap graphics of a project.

To call up bitmap resource:

Select *Bitmap resource* in menu *Resources*.
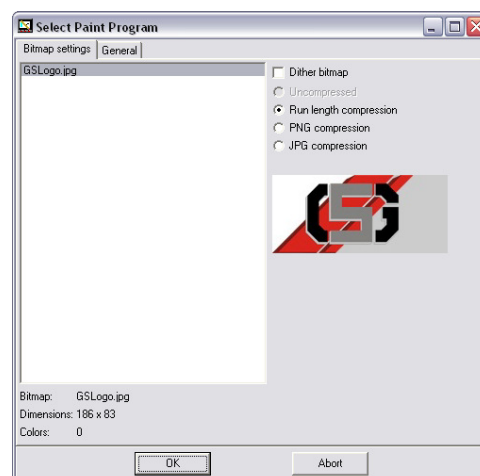
Fig. 13   Bitmap resource, tab *Bitmap settings*

The *Bitmap settings* tab shows the graphics used in the project.
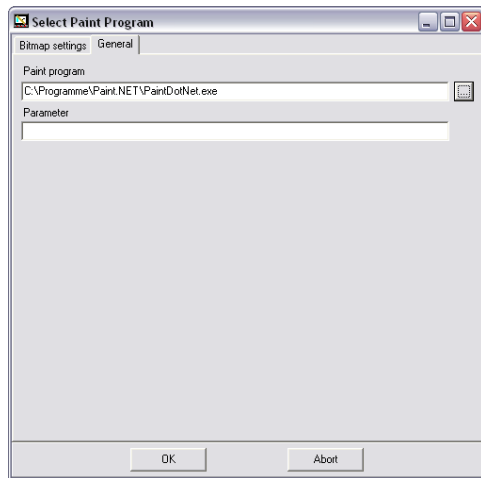
Fig. 14   Bitmap resource, tab *General*

The *General* tab allows indicating a graphics software for editing objects (e.g. for single-picture graphics).

Note
If no graphics software is indicated, GDSDesigner will use Windows® Paint for editing objects.

6.3   Text resource

The text resource allows managing all texts of a project.

It is possible to save texts in several languages and switch between these while the control set is running.

To call up text resource:

▸ Select *Text resource* in menu *Resources*.
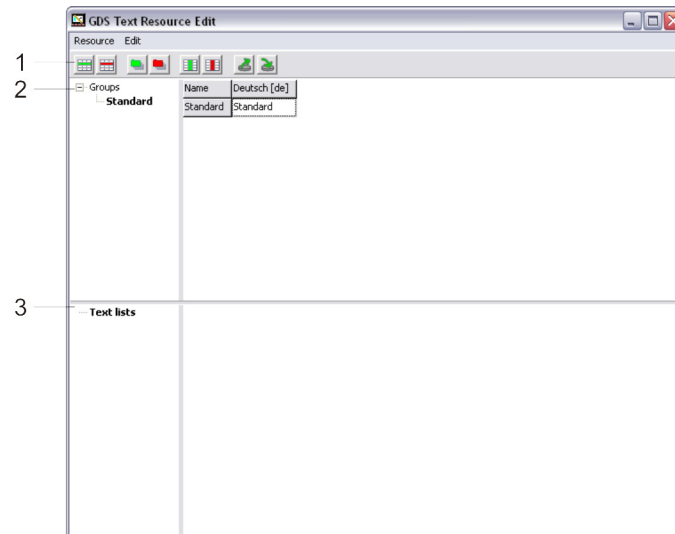
Fig. 15 Text resource

    (1) Button bar
    (2) Text and text groups
    (3) Text lists

The text resource codes characters in Unicode (UTF8) and supports non-Latin fonts (e. g. Greek, Cyrillic, Chinese).

It is possible to edit resources using the button bar or the context menu.

Text groups

Text groups allow dividing texts into subject areas and thus make it possible to arrange projects more clearly. For simple projects, one text group is sufficient.

Text lists

Text lists consist of several texts defined in *Texts and text groups*.

To add texts to a text list:

▸ Pull respective item from *Texts and text groups* to *Text lists* using drag & drop.

Button bar

The button bar contains buttons for editing resources.

Whether buttons are available or not depends on the resource highlighted.

Allows adding a language.   **Add language**

To add a language:
▸ Click on 🈳 in button bar.
▸ Enter name for language.
▸ Enter language code in lower case.

Allows deleting a language.   **Delete language**

⚠️ **Caution**	**Data loss due to deleting languages!** The respective language will be deleted from all text groups and text lists. Before deleting a language: ▸ Make sure that you really want to remove the language highlighted.

To delete a language:
▸ Highlight entire column of language to be deleted by clicking column header.
▸ Click on 🈳 in button bar.

Allows adding a group or text list.   **Add group**

To add a new text list:
▸ Highlight *Groups* in tree view.
▸ Click on 📂 in button bar.
▸ Enter group name.
  Make sure to use a group name that does not exist yet.

To add a new text list:
▸ Highlight *Text lists* in tree view.
▸ Click on 🈳 in button bar.
▸ Enter text list name.
  Make sure to use a text list name that does not exist yet.

Delete group   Allows deleting a group or text list.

Note
Deleting a group or text list is exclusively possible if no object of the project is currently using the group or text list.

To delete group:
▸ Highlight respective group in tree view.
▸ Click on ▪ in button bar.

To delete text list:
▸ Highlight respective text list in tree view.
▸ Click on ▦ in button bar.

New text   Allows entering a new text.

To enter new text:
▸ Highlight respective group.
▸ Click on ▦ in button bar.
▸ Enter text name.
  Make sure to use a text name that is not already in use in the group.

To change text:
▸ Double-click respective item in list.

Delete Text/text list element   Allows deleting a text or text list element.

Note
Deleting a text or text list element is exclusively possible if no object of the project is currently using the text or text list element.

To delete a text:
▸ Highlight respective text in list.
▸ Click on ▦ in button bar.

To delete a text list element:
▸ Highlight respective text list element in text list.
▸ Click on ▦ in button bar.

Allows exporting all texts (groups and text lists) to an XLS file.
The XLS file codes characters in Unicode (UTF8).

Export to XLS file

To export from XLS:
▸ Click on 🖳 in button bar.
▸ Select respective directory and file name.

Allows importing texts from an XML file.

Import from XLS file

⚠	**Data loss due to overwriting existing texts!**
	When importing from an XML file, all texts of the text resource will be overwritten.
Caution	▸ Check whether texts from XML file may overwrite texts in text resource.

⚠	**Data loss due to faulty XLS file!**
	When importing from an XML file, all texts of the text resource will be overwritten.
Caution	▸ Make sure that XLS file is error-free and correctly coded.

Note
Graf-Syteco recommends making a backup copy of the project before importing an XLS file.

To import an XLS file:
▸ Click on 🖳 in button bar.
▸ Select respective XLS file.

6.4 True Type Font Resource

Allows managing all standard fonts for text objects.
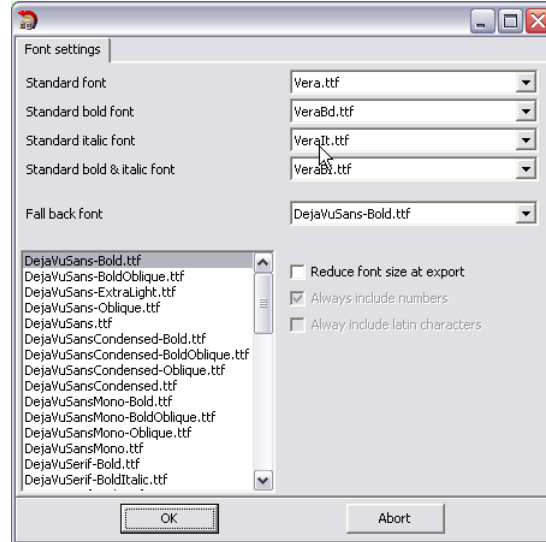
Fig. 16 True Type Font Resource

True Type Font Resource shows the fonts available in the project and allows assigning fonts to certain formats.

The fallback font is used if characters, e.g. Chinese, are not available, e.g. in the Standard italic font.

When exporting the project, it is possible to remove unused characters from the font, so that the font requires less storage space. Setting which characters are to be maintained when exporting is possible.

To remove unused characters from a font when exporting:

▸ Activate *Reduce font size at export* (☑).

To maintain unused numbers when exporting:

▸ Activate *Always include numbers* (☑).

To maintain unused Latin characters when exporting:

▸ Activate *Always include Latin characters* (☑).

7

Chapter 7

Device settings

7.1 Device settings for MCQ, ARGOS and OEM-Projects

Tab *System settings* allows configuring   **System**

- the operating system.
- USB memories.
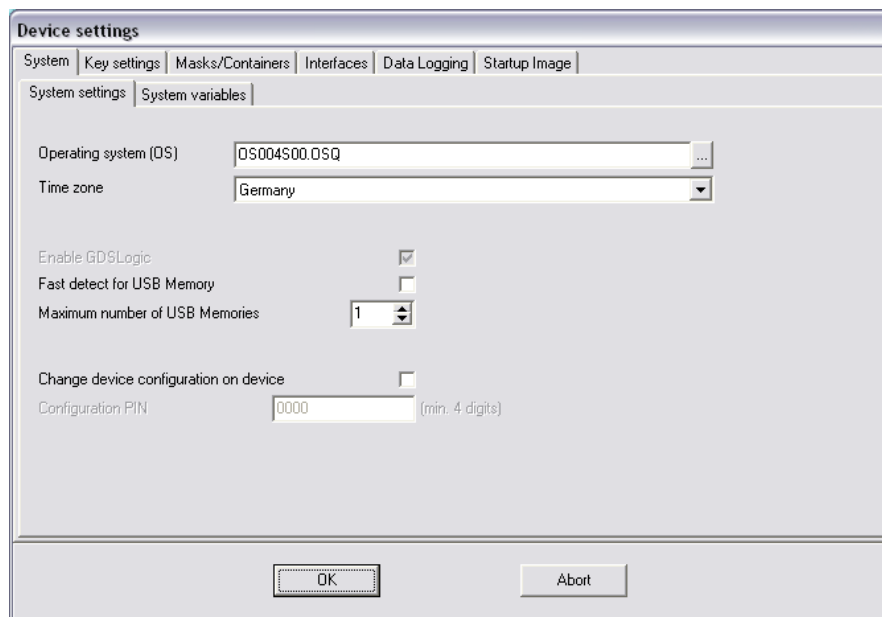
Fig. 17 Device settings for MCQ, ARGOS and OEM-Projects; System; System settings

Tab *System variables* allows configuring the system variables.
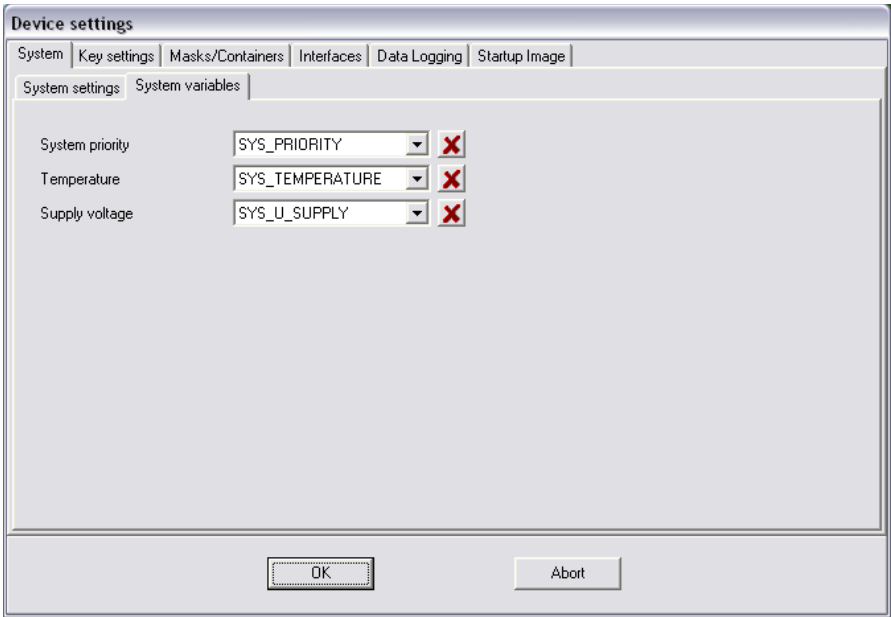
Fig. 18   Device settings for MCQ, ARGOS and OEM-Projects; System variables

Key settings

Tab *Key settings* allows assigning certain keys on the control set individual functions.

Tab *Navigation* allows configuring the key protocols.
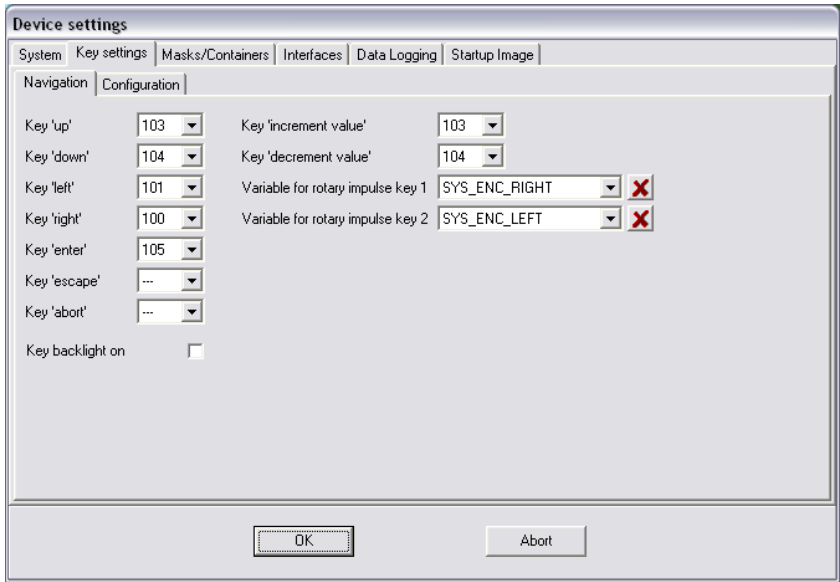
Fig. 19   Device settings for MCQ, ARGOS and OEM-Projects; Key settings

Tab *Configuration* allows

- an overview about which key events are transferred to which code interfaces.
- configuring which key events are transferred to which code interfaces.
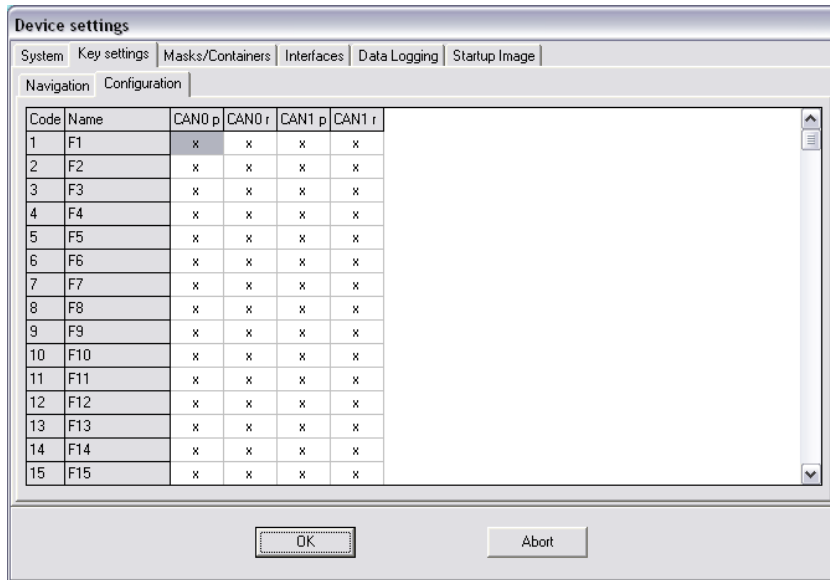
Fig. 20 Device settings for MCQ, ARGOS and OEM-Projects; Key settings

Tab *Masks/containers* allows configuring **Masks/containers**

▸ the mask and container rollup time for automatic mask change.
▸ the blink rate.
▸ the object behavior.
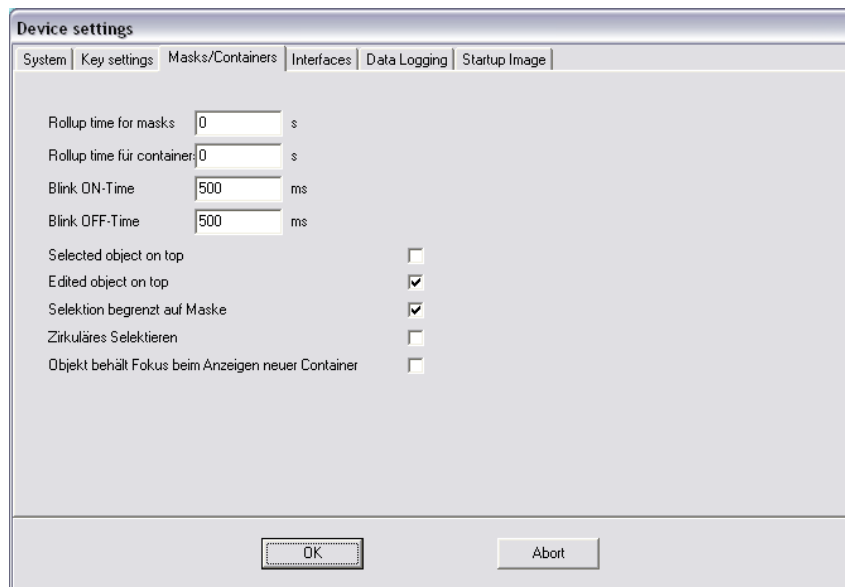
Fig. 21 Device settings for MCQ, ARGOS and OEM-Projects; Masks/containers

Interfaces

Tab *Interfaces* allows configuring
- the CAN interfaces.
- the COM interfaces.
- the ethernet interface.
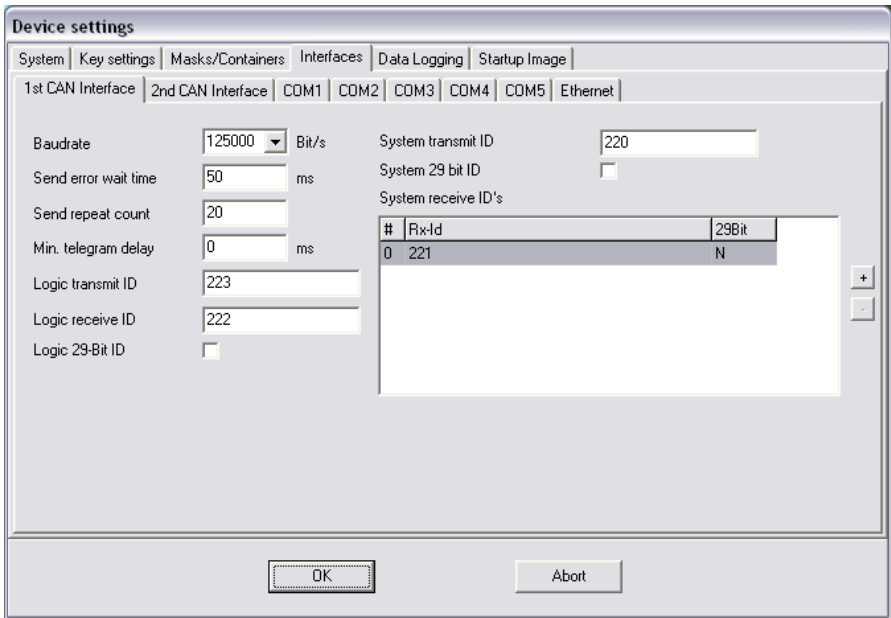
Fig. 22   Device settings for MCQ, ARGOS and OEM projects; Interfaces

Data logging

Tab *Data logging* allows configuring the error message logs.
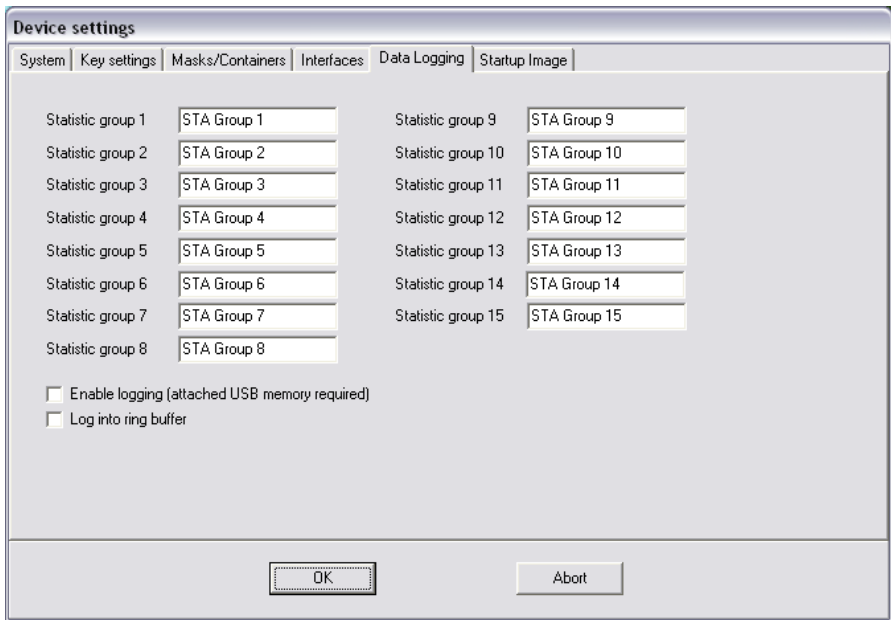
Fig. 23   Device settings for MCQ, ARGOS and OEM projects; Data logging

Tab *Startup Image* allows configuring the startup image on the control set during the startup process.

Startup Image
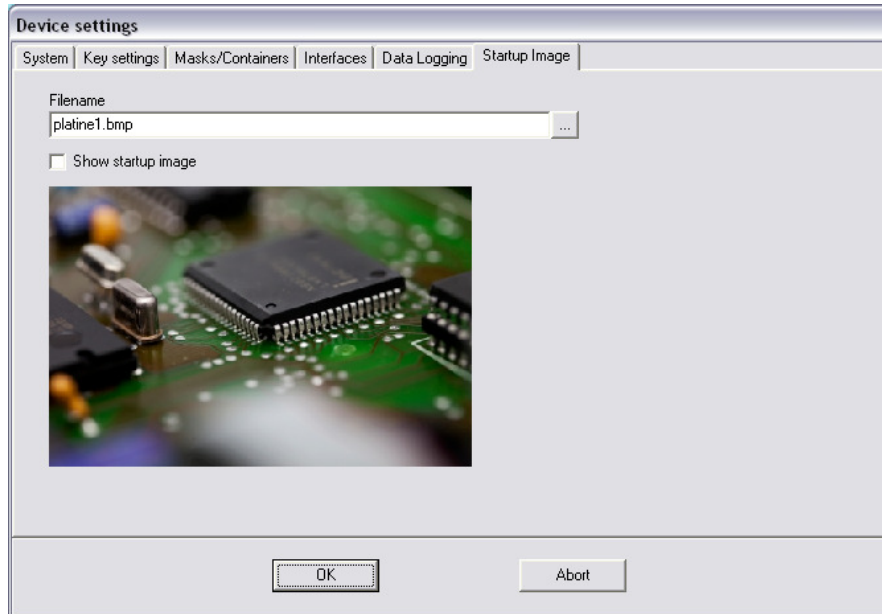
Fig. 24   Device settings for MCQ, ARGOS and OEM projects; Startup image

8

Chapter 8
Presettings

In presettings, it is possible to adjust different settings and paths of the GDSDesigner.

To call up presettings:

▶ Select *Presettings* in menu *Extras*.
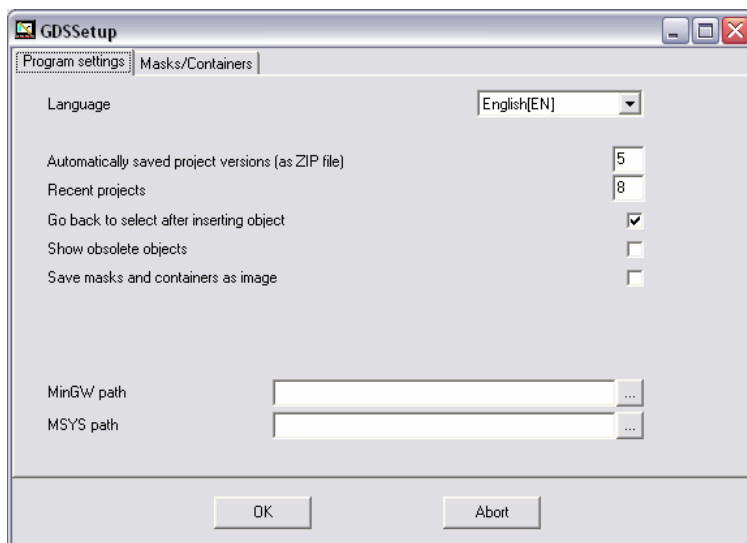
Fig. 25   Program settings

Language	Language of the GDSDesigner's program surface
Automatically saved project versions (as ZIP file)	The five last saves are packed in a ZIP archive (0 = no archive will be generated)
Recent projects	Number of items in menu *File* / *Recent projects*

Go back to select after inserting object	After inserting an object it is necessary to select an object from the object palette (even if it is the same) before inserting another object.
Show obsolete objects	Show objects in object palette that will no longer be supported.
Save masks and containers as image	Additionally save masks and containers as bitmaps in folder *<Project>/Screens*

Masks/Containers
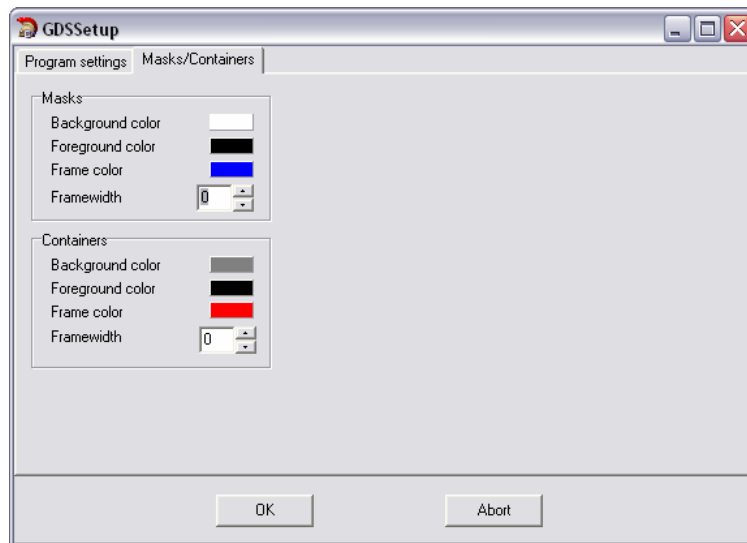
Fig. 26   Color setting for masks and containers

Masks Background color	Standard background color for masks
Masks Foreground color	Standard foreground color for masks
Masks Frame color	Standard frame color for masks
Masks Framewidth	Standard framewidth for masks
Containers Background color	Standard background color for containers
Containers Foreground color	Standard foreground color for containers
Containers Frame color	Standard frame color for containers
Containers Framewidth	Standard framewidth for containers

9

Creating control interfaces

Data loss or unforeseeable effects!

Parallel programming of control processes in GDSLogic and C will cause data loss and unpredictable effects due to asynchronous processing of program flows.

Caution

▶ Program control processes exclusively in GDSLogic or C.

There are several ways to create control interfaces for control sets with control tasks, depending on the method applied for programming the control task (*see 2 Function*).

There are two ways to create control interfaces for MCQ control sets:

- Using the graphic programming system GDSLogic for programming the control task and the project development.
- Using the programming language C for programming the control task and GDSDesigner for the project development.

Note
The creation of the control interface is identical in both these cases.

9.1   Project development with GDSLogic

To start GDSDesigner and create a control interface:

 ‣ Start GDSLogic using the item in the Windows® start menu or the desktop shortcut.
 ‣ Select *New* in menu *Project*.
 ‣ Select desired control device.
 ‣ Enter project name.
 ‣ Adapt project directory if necessary.
 ‣ Click *OK*.

GDSLogic creates new GDSLogic project.

 ‣ Select *GDSDesigner* in menu Tools.

GDSDesigner starts and shows empty visualization project.
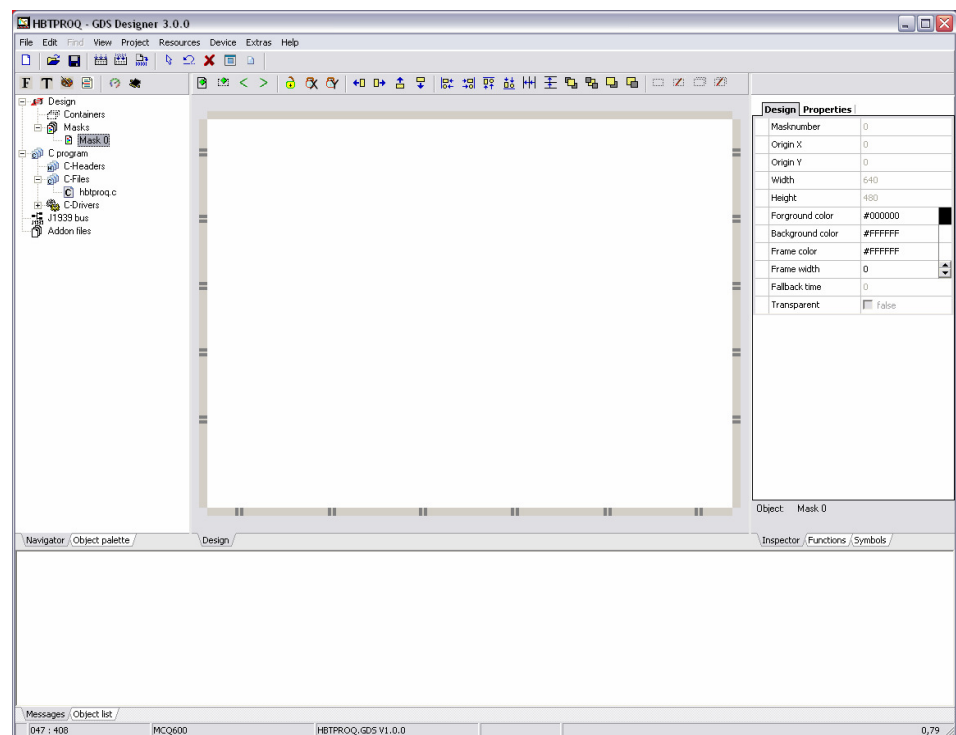
Fig. 27   GDSDesigner

Note
For information on the development of the control task in GDSLogic, see Manual GDSLogic.

9.2 Project development with GDSDesigner

To start GDSDesigner and create a control interface:
‣ Start GDSDesigner using the item in the Windows® start menu or the desktop shortcut.

GDSDesigner starts.

‣ Select *New* and *Project* in menu *File*.
‣ Select desired control device.
‣ Enter project name.
‣ Adapt project directory if necessary.
‣ Click *OK*.

GDSDesigner shows empty visualization project.
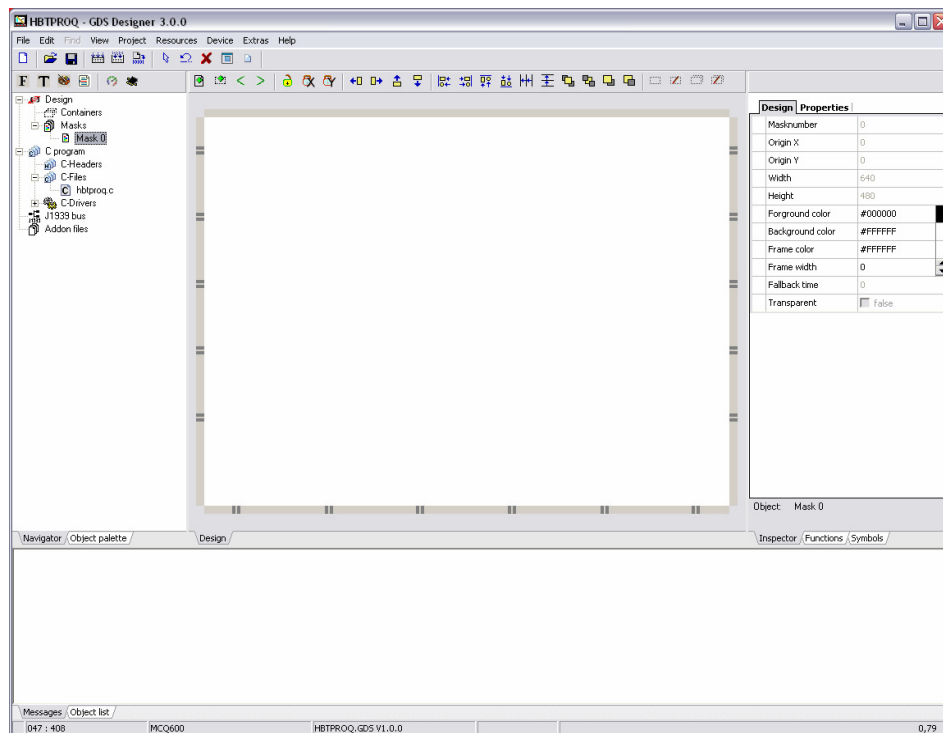
Fig. 28   GDSDesigner

9.3   Masks and Containers

Masks and containers allow a functional design of the control surface.

Show and hide make it possible to adapt the control surface to the control set's respective operating situation.

To add an new mask:
- ▸ Click on button ⊞ in button bar.
- ▸ Enter mask name.

To add a new container:
- ▸ Click on button ⬚ in button bar.
- ▸ Draw container size using the mouse.
- ▸ Enter container name.

9.4   Visualization

Adding objects

To add an object to the control surface:
- ▸ Click on desired object in object library.
- ▸ Use the mouse to draw a rectangle of the desired object size on the display.

Fig. 29   Adding an object

Note
GDSDesigner adopts the size of the rectangle as the object size. For certain objects, GDSDesigner will not use the size of the rectangle, but a set size.

For information on the individual objects, see 10 Object library.

Aligning/positioning objects

To evenly align or position several objects:

▸ Highlight respective objects.
▸ Click button for desired alignment in button bar.

	Move highlighted objects to the left by one pixel
	Move highlighted objects to the right by one pixel
	Move highlighted objects up by one pixel
	Move highlighted objects down by one pixel
	Align left edges of several highlighted objects.
	Align right edges of several highlighted objects.
	Align top edges of several highlighted objects.
	Align bottom edges of several highlighted objects.
	Place several highlighted objects in such a way that the spaces between them are identical in horizontal direction.
	Place several highlighted objects in such a way that the spaces between them are identical in vertical direction.

Text_1   Text_1

          Text_2   Text_2

     Text_3   Text_3
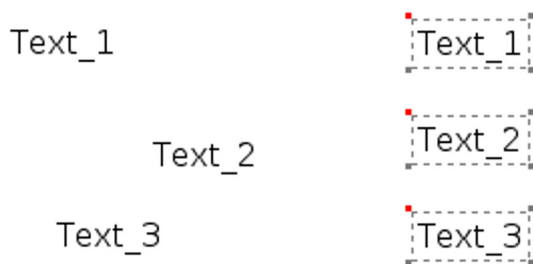
Fig. 30   Aligning/positioning objects

Note
Shortcuts for moving objects

ALT + left/right/up or down: move by one pixel
ALT + SHIFT + left/right/up or down: move by 5 pixels

Aligning object in direction Z

GDSDesigner draws all objects in the order in which they were added to the mask or container.

To change the order:

▶ Highlight respective object.
▶ Click button for desired alignment in button bar.

Put highlighted object into foreground.

Put highlighted object into background.

Move highlighted object toward foreground by one level.

Move highlighted object toward background by one level.

Note

Every mask has only one object list. When moving an object by one level, it may be necessary to click the button several times before the change becomes visible.

Positioning the objects is also possible using the shortcuts Ctrl + PageUp and Ctrl + PageDown.
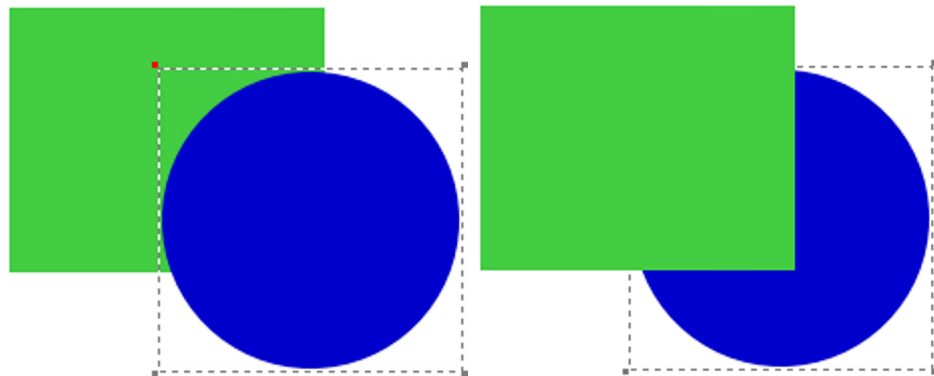
Fig. 31   Aligning object in direction Z

Display lock

When highlighting an object, it is possible to inadvertently move the object. In order to prevent inadvertent moving, GDSDesigner provides the possibility to lock objects.

To lock all objects in direction x and prevent moving them in direction x:
▸ Click on ![icon] in button bar.

Locking in direction x

To unlock an object in direction x:
▸ Click on ![icon] in button bar.

To lock all objects in direction y and prevent moving them in direction y:
▸ Click on ![icon] in button bar.

Locking in direction y

To unlock an object in direction y:
▸ Click on ![icon] in button bar.

To lock an object in directions x and y and prevent any moving:
▸ Highlight respective object.
▸ Click on ![icon] in button bar.

Locking in direction x and in direction y

To unlock an object in directions x and y:
▸ Highlight respective object.
▸ Click on ![icon] in button bar.

Note
If mask object is selected in the navigator, pressing ![icon] *locks or unlocks all objects of this mask.*

Display mode for objects

In GDSDesigner it is possible to display objects in selected and in operating mode. The display in GDSDesigner is identical to the actual one in the control set.

To display an individual object in selected mode:

▸ Highlight respective object(s).
▸ Click button for desired display in button bar.

Display highlighted object in operating mode *Selected*.

Display all objects in operating mode *Selected*.

Display highlighted object in operating mode *Edited*.

Display all objects in operating mode *Edited*.

9.5   Variables

A variable is a management structure consisting of several elements.

Name	Name of the variable
	For identification of variables during project development.
	Not relevant for the control set.
Handle	Clearly defined number in the visualization project under which the variable in the target system is stored in a table.
	Allows adressing when writing and reading the variable using the C or LOGIC program or via the control set's interfaces (e.g. CAN).
	For the C code, the variable table generates symbolic names as *#define* that directly match the handle.
Remanent	If the remanent flag is set, the variable will be buffered power failure-proof (remanently).
	After switching the control set off and on again, the value of the variable will be maintained.
Parameters	If the parameter flag is set, a list of set values is transferred to the control set as parameters for the variable.
	It is possible to set all variables thus marked to set values at the same time using a C function.
Group	It is possible to pool variables and manage them in groups.
Interface	Interface via which an external device requests the value of a variable.
Interface/interval	Time interval of access to interface (value * 10 ms)
Value	Set value of variable
	Exclusively for display in GDSDesigner.
	Nominal value is not transferred to control set.
Comment	Comment on documentation in GDSDesigner.

Variables

- require 4 bytes of memory during runtime.
- are arranged in order in the memory of the control set.
- are addressed in the control set using their handle or index.
- are assigned nominal values in the memory using the CAN interface or the C user program.

The control program reads and writes data contents of the variables.

Data values are displayed in the visualization using a suitable object.

The representation of values in a mask or container does not depend on the internal data type.

The only limit for the number of variables is the working memory space.

GDSDesigner provides the variable resource for variable management (*see 6.1 Variable resources*).

To call up variable resource:

‣ Select *Variable resource* in menu *Resources*.
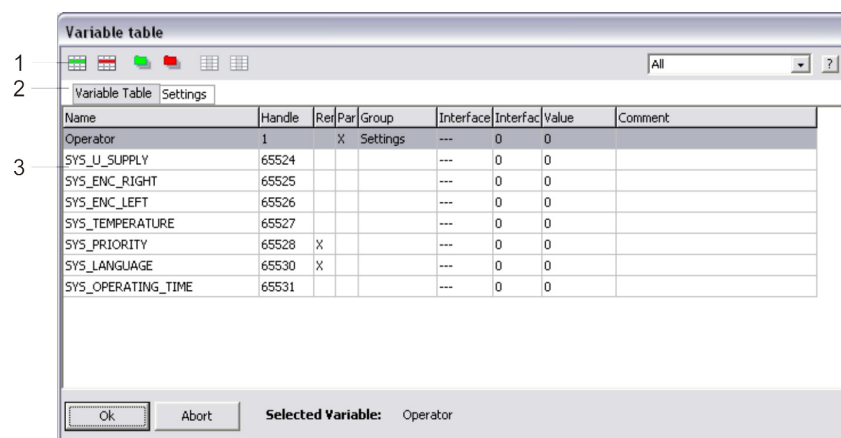
Fig. 32   Variable resource

    (1) Button bar
    (2) Variable groups
    (3) Variables

Creating variables

To insert variable:

‣ Click [icon] in button bar or press key N.
‣ Define variable properties in dialog.

Deleting variables

To delete variable:

‣ Highlight respective variable in list.
‣ Click [icon] in button bar or press key D.

Note
Deleting system variables is not possible.

Editing variables

To edit variable properties:
 ‣ Double-click respective variable.
 ‣ Define variable properties in dialog.

Grouping variables

Variable groups allow dividing variables into subject areas and thus make it possible to arrange projects more clearly.

Creating variable groups

Allows adding a parameter group.

To add a variable group:
 ‣ Click on ▣ in button bar.
 ‣ Enter name for variable group.

Variable resource shows variable group as new tab.

Deleting variable groups

⚠ **Caution**	**Data loss due to deleting a variable group!**
	When deleting a variable group, the parameter data sets contained therein will be deleted.
	All variables of the variable group will lose their group affiliation and parameter flags.
	Before deleting a variable group:
	‣ Check whether you really want to delete the variable group.

To delete a variable group:
 ‣ Click on ▣ in button bar.
 ‣ Highlight respective variable group in list.
 ‣ Click *OK*.

Creating parameters for variable groups

If variables are assigned to groups and the parameter flag is set for at least one of the variables, it is possible to create parameter sets for these variables.

It is possible to store more than one parameter value for one variable.

During runtime, it is possible to set all variables of a group for which the parameter flag is set to the values of a parameter group. For this purpose, the function *LoadParameterSet(...)* is available in the C code.

Variable access from C user program

In order to facilitate access to variables from the C program, GDSDesigner generates the file *VARTAB.H*.

File *VARTAB.H*

- contains a table of all GDSDesigner variables in ascending order, sorted according to their handle.
- represents the binary variable table that is transferred to the control set.

Access to variables in C-API is possible via the functions

- *SetVar(<Handle>,<Value>)*
- *GetVar(<Handle>)*

The file *VARTAB.H* contains *#define* commands which assign the variable names to the corresponding handles.

Example:

- Variable name = water level
- Handle = 24
- #define = water level 24

A second *#define* command (*#define = water level_idx <index>*), using the functions *SetVarIndexed(<Index>,<Value>)* and *GetVarIndexed(<Index>)* allows quick access to the variable.

9.6 Attributes

All objects support attributes that allow determining object properties.

Determining properties is possible using
- an entry in the inspector during the creation of the object.
- a variable.

A combination of objects with fixed and variable attributes in a mask or container is possible.

To change properties during runtime:
▸ Enter variable for controlling the property in inspector in field *Attribute*.
▸ Change value of variable in C code using function *SetVar(...)*.
  Every attribute corresponds to one bit in the variable.

For objects whose attributes are variable-controlled to be visible when starting the control set:
▸ Initialize variable in *Function KOP_Init(...)* or *UserCInit(...)* with a value of at least 0x08 (bit 3 set).

Note
For information on objects, see 10 Object library.

General attributes

Bit no.	Name	Description
Bit 0	Inverted	1 = representation inverted (fore- and background color inverted)
Bit 1	Blinking	1 = object blinking
Bit 2	Underline	1 = object is underlined (exclusively text objects)
Bit 3	Visible	1 = object is visible
Bit 4	Transparent	1 = object background is transparent
Bit 5	Antialiasing	1 = antialiasing is active (exclusively true-type text objects)
Bit 6	Select next object automatically	1 = press <Enter> for next object.
Bit 7	Reserved	Do not use. Always set to 0.

Set value and variable attributes

Bit no.	Name	Description
Bit 8	Set value	1 = object is input object and reacts to key events.
Bit 9	LowerBlinkFg	1 = text of object flashes when falling below limit.
Bit 10	UpperBlinkFg	1 = text of object flashes when exceeding limit.
Bit 11	LowerBlinkBk	1 = background of object flashes when falling below a bottom limit.
Bit 12	UpperBlinkBk	1 = background of object flashes when exceeding a top limit.
Bit 13	Selecting	1 = object tries to maintain focus when showing mask
Bit 14	UpdateOnEdit	1 = update corresponding variable when changing
Bit 15	Animated	1 = animation active (exclusively trend writer, single-picture graphics)
Bit 16 to bit 32	Reserved	Do not use. Always set to 0.

10 Chapter 10
Object library

The object library is a collection of all available visual objects.

The object library is divided into the following sections:
- TrueType text objects
- Resource text objects
- Graphic objects
- Vector graphics objects

Note
The modular concept of GDSDesigner allows including new visual objects without changing the basic software. For information on adding customer-specific objects, contact GRAF-SYTECO.

When using TrueType objects, observe licensing conditions of the TrueType fonts used.

GRAF-SYTECO delivers GDSDesigner exclusively with TrueType fonts that are subject to the free BSD license. All TrueType text objects use the Freetype2 library, which is under Freetype license.

Copyright 1996-2002, 2006 by David Turner, Robert Wilhelm and Werner Lemberg; see freetype-license.txt in installation path.

Setting color values

To set color values in object properties:
▶ Click color field behind input line.
▶ Select desired color in dialog.

- or -

▶ Enter color value in HTML format (#RRGGBB) in input line.

Setting common object properties

To set properties of several objects at the same time:

▸ Highlight objects

▸ Change respective property
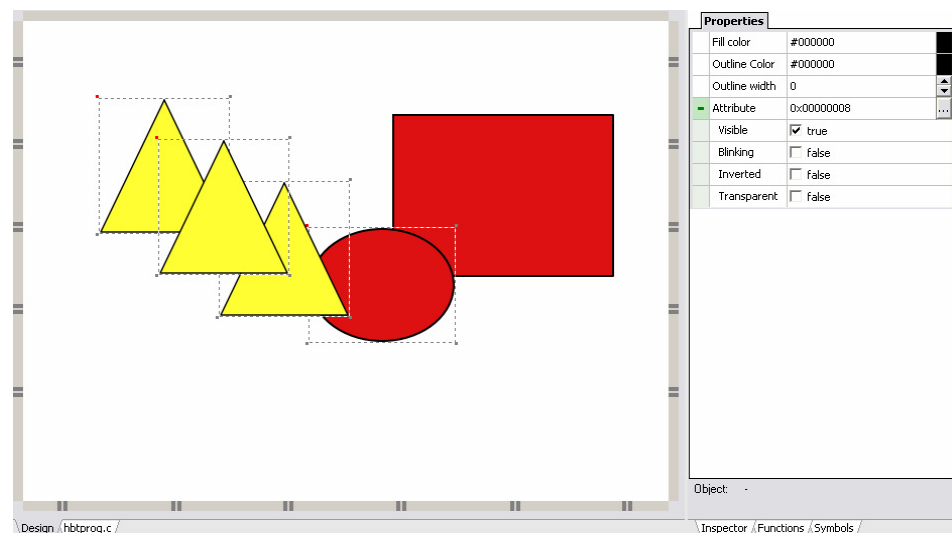
Fig. 33   Inspector shows common properties of ellipse and triangle

Note
This function simplifies updating more than one object. The more different objects are highlighted, the less common properties remain.
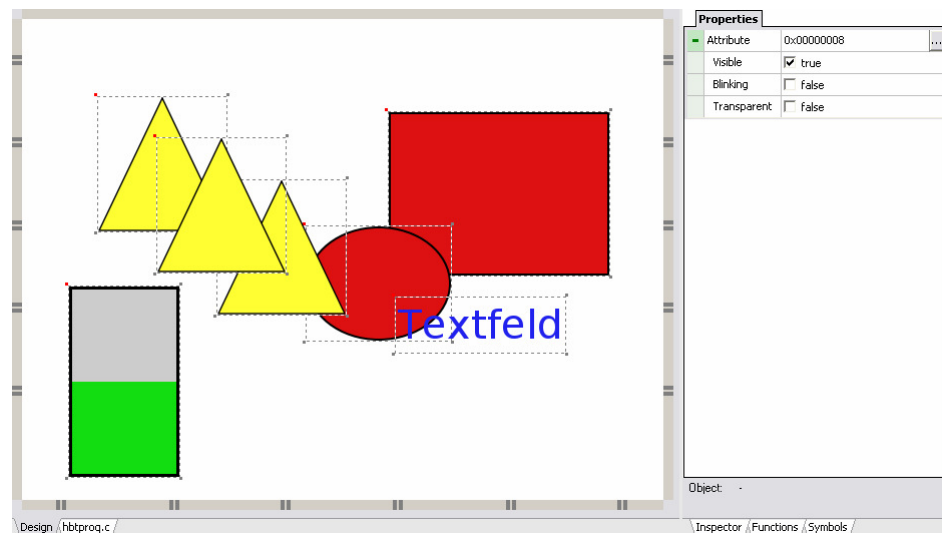
Fig. 34   All objects highligted

10.1 TrueType text objects

Unlike pixel text objects, TrueType text objects are created like vector graphics and feature a smoother typeface thanks to aliasing.
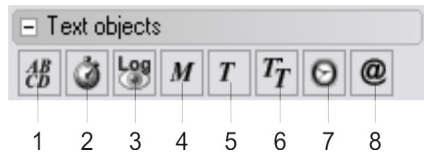
Fig. 35   TrueType text objects

    (1) Editable TrueType text
    (2) TrueType time
    (3) TrueType log viewer
    (4) TrueType menu text
    (5) TrueType text
    (6) TrueType text list
    (7) TrueType time and date
    (8) TrueType variable

Editable TrueType text

Editable TrueType text is a variable text element and suitable for entering texts in masks and containers.

Properties	Colors
Available characters	abcdefghijklmnopqrstuvwx
Length	16
Origin X	11
Origin Y	314
Font	Standard
Font size	16
Text color	#000000
Background color	#FFFFFF
▬ Attribute	0x00000028
Visible	☑ true
Blinking	☐ false
Underline	☐ false
Inverted	☐ false
Transparent	☐ false
Antialiasing	☑ true
Change text immediately	☐ false
Grab focus	☐ false
Auto-select next on \<enter\>	☐ false

Fig. 36   Properties window

Properties	Colors
Selected text color	#FFFFFF
Selected background color	#000000
Selected font	Standard
Selected font size	16
Selected frame color	#000000
Selected frame width	0
Edit text color	#FFFFFF
Edit background color	#000000
Edit cursor color	#FFFFFF
Edit font	Standard
Edit font size	16
Edit frame color	#000000
Edit frame width	0

Fig. 37   Color selection

Note

Changing the position in the mask or container is not possible during runtime.

The TrueType resource defines the font.

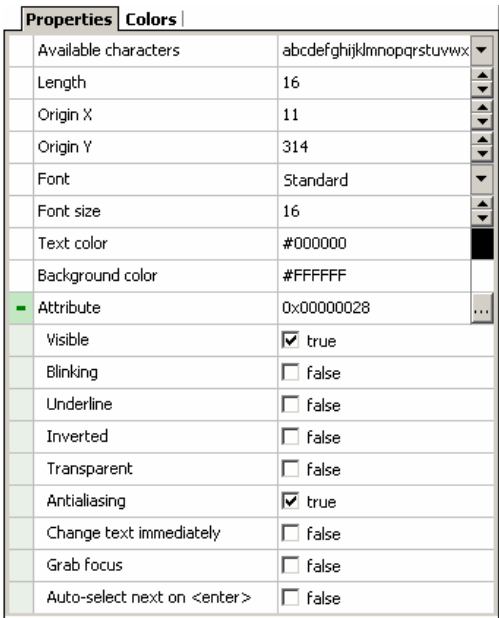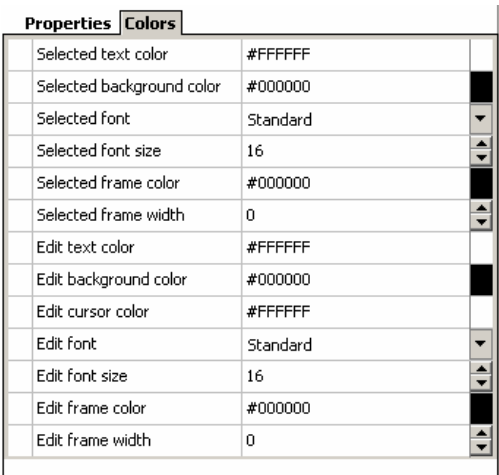|	
Available characters	Type of available characters for text input
Length	Maximum length for text input
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Font	TrueType font of text element It is possible to replace the list of fonts displayed by a personal list. Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display If the attribute *Transparent* is active, the background is not displayed.
Attribute	Field with individual flags which influences the display of the variables. Entering HEX values in the input field is possible to create a static attribute. To change the attribute in the control set during runtime: ▸ Select the variable controlling the attribute from list. Attributes supported: • Visible • Inverted • Blinking • Transparent • Underline • Antialiasing • Selecting • Change text immediately • Auto-select next on <Enter>

Colors

Selected text color	Color of text when variable value is within limits
Selected background color	Background color of variable
	Background will not be drawn in case of transparent display. Allows positioning object e.g. over a bitmap.
Selected font	Font used for displaying object in selected mode.
Selected font size	Font size used for displaying object in selected mode.
Selected frame color	Frame color used for displaying object in selected mode.
Selected frame width	Frame width used for displaying object in selected mode.
Edit text color	Color of text for object in editing mode.
Edit background color	Background color of text for object in editing mode when displaying non-transparent.
Edit cursor color	Color of cursor for variable object in editing mode.
Edit font	Font used for displaying object in editing mode.
Edit font size	Font size used for displaying object in editing mode.
Edit frame color	Color of frame around object for selected object in editing mode.
Edit frame width	Frame width used for displaying object in editing mode.

TrueType time

TrueType time displays a variable's value as a time value in independence of the set format.

Properties	
Format	hh:mm:ss
Origin X	167
Origin Y	336
Font	Standard
Font size	16
Text color	#000000
Background color	#FFFFFF
Variable	SYS_U_SUPPLY
– Attribute	0x00000028
Visible	☑ true
Blinking	☐ false
Underline	☐ false
Inverted	☐ false
Transparent	☐ false
Antialiasing	☑ true

Fig. 38   Properties window

Note
Changing the position in the mask or container is not possible during runtime.

The TrueType resource defines the font.

Properties

Format	Display format of time or date Possible formats: • hh:mm:ss • hhh:mm:ss • hhhh:mm:ss • hhhhh:mm:ss • hh:mm • hhh:mm • hhhh:mm • hhhhh:mm • hh • hhh • hhhh • hhhhh • mm:ss Display of hours: • up to five digits Display of minutes and seconds: • two digits
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Font	TrueType font of text element It is possible to replace the list of fonts displayed by a personal list. Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Background color of object
Variable	Variable displayed as second value

Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Inverted
	• Underline
	• Antialiasing
	• Blinking
	• Transparent

TrueType log viewer

TrueType log viewer allows displaying the log data on the control set.

The control set displays a multiline text.

The following selections are possible using the navigation keys of the control set:
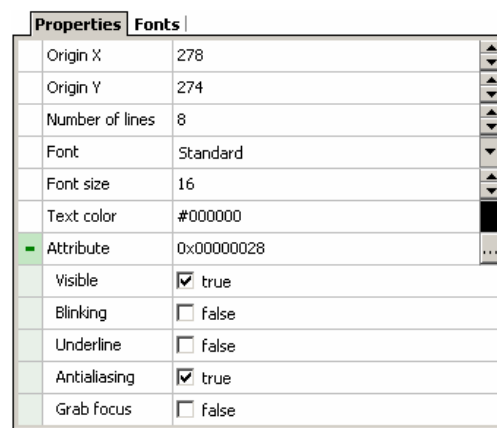
- year
- month
- day
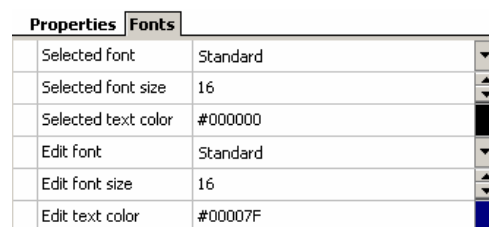- line in log file

Fig. 39   Properties window

Fig. 40   Selected font

Note
Displaying the log files is exclusively possible with the log files organized in directories.

Changing the position in the mask or container is not possible during runtime.

The TrueType resource defines the font.

Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Number of lines	Number of lines for displaying log data.
Font	TrueType font of text element
	It is possible to replace the list of fonts displayed by a personal list.
	Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Underline
	• Antialiasing
	• Blinking
	• Selecting

Selected font	Font used for displaying object in selected mode.
Selected font size	Font size used for displaying object in selected mode.
Selected text color	Font color used for displaying object in editing mode.
Edit font	Font used for displaying object in editing mode.
Edit font size	Font size used for displaying object in editing mode.
Edit text color	Font color used for displaying object in editing mode.

TrueType menu text M

TrueType menu text creates menu structures for

*   calling up masks.
*   showing/hiding containers.

All masks are managed via a stack. When showing a mask, it is put on top of the stack.

All message containers are managed via a stack. When showing a message container, it is put on top of the stack.

The element on top of the stack will be shown on the display.

Properties	Colors		
Text	MenuText_1	
Origin X	396	
Origin Y	306	
Font	Standard	
Font size	16	
Text color	#000000	
Background color	#FFFFFF	
− Attribute	0x00000028	...
Visible	☑ true	
Blinking	☐ false	
Underline	☐ false	
Inverted	☐ false	
Transparent	☐ false	
Antialiasing	☑ true	
Grab focus	☐ false	
Auto-select next on <enter>	☐ false	
Function	Call Mask	
Function number	0	

Fig. 41   Properties window

Properties	Colors		
Selected text color	#FFFFFF	
Selected background color	#000000	
Selected font	Standard	
Selected font size	16	
Selected frame color	#000000	
Selected frame width	0	

Fig. 42   Color selection

Note
Changing the position in the mask or container is not possible during runtime.

The TrueType resource defines the font.

Text	Text displayed in mask or container (maximum 255 characters)
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Font	TrueType font of text element It is possible to replace the list of fonts displayed by a personal list. Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display If the attribute *Transparent* is active, the background is not displayed.
Attribute	Field with individual flags which influences the display of the variables. Entering HEX values in the input field is possible to create a static attribute. To change the attribute in the control set during runtime: ▸ Select the variable controlling the attribute from list. Attributes supported: • Visible • Inverted • Blinking • Transparent • Underline • Antialiasing • Selecting • Auto-select next on \<Enter\>
Function	*see Function list*
Function number	Number of set function

Function list

Call up mask	Show mask with the number indicated in field *Number*. If this mask is already on the stack with another mask covering it, the display remains unchanged.
Call up prio mask	Show mask with the number indicated in field *Number*. If this mask is already on the stack with another mask covering it, the mask called up is put on top of the stack and displayed.
Return to previous mask	Hide mask on top and show mask below.
Show info container	Show info container with the number indicated in field *Number*.
Hide info container	Hide info container with the number indicated in field *Number*.
Show message container	Show message container with the number indicated in field *Number*.
Hide message container	Hide message container with the number indicated in field *Number*. If message container is on top of stack: • Message container below is shown. If message container is anywhere but on top of stack: • Message container is removed from stack. If no message container is on top of stack anymore: • Currently shown mask is displayed.
Transfer menu index to C	Transfer current menu index to user C program.

Colors

Selected text color	Color of text when variable value is within limits
Selected Background color	Background color of variable
	Background will not be drawn in case of transparent display. Allows positioning object e.g. over a bitmap.
Selected font	Font used for displaying object in selected mode.
Selected font size	Font size used for displaying object in selected mode.
Selected Frame color	Frame color used for displaying object in selected mode.
Selected frame width	Frame width used for displaying object in selected mode.

TrueType text T

TrueType text displays static text.

Note
Changing the position in the mask or container is not possible during runtime.
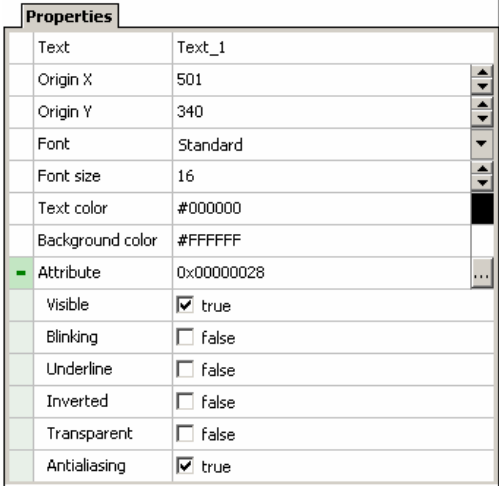
The TrueType resource defines the font.

Fig. 43   Properties window

Properties

Text	Text displayed in mask or container (maximum 255 characters)
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Font	TrueType font of text element
	It is possible to replace the list of fonts displayed by a personal list.
	Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display.
	If the attribute *Transparent* is active, the background is not displayed.

Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Inverted
	• Blinking
	• Transparent
	• Underline
	• Antialiasing

TrueType text list T_T

	Inadvertent changes!
⚠ **Caution**	It is possible for more than one text list element to access a shared text list file. Changes in the text list file will affect all text list elements.
	▸ Exclusively change shared text list files if the respective change is intended for all text list elements.

TrueType text list displays a text list.

It is possible to open an existing text list or to create a new text list.

To open an existing text list:
- ▸ Go to tab *Properties* and click button 📂.
- ▸ Open desired text list.

Note
Import from AT projects is possible (.txl-file).

To open a new text list:
- ▸ Enter text in tab *Textlist*.

Note
It is possible to change the name of the text list file in project management.

Changing the position in the mask or container is not possible during runtime.

The TrueType resource defines the font.

Fig. 44   Properties window

Fig. 45   Color selection

Fig. 46   Textlist

Properties

File name	File name of linked text list.
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Variable	Variable that defines which text list entry is displayed.
Font	TrueType font of text element It is possible to replace the list of fonts displayed by a personal list. Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display If the attribute *Transparent* is active, the background is not displayed.
Attribute	Field with individual flags which influences the display of the variables. Entering HEX values in the input field is possible to create a static attribute. To change the attribute in the control set during runtime: ▸ Select the variable controlling the attribute from list. Attributes supported: • Visible • Inverted • Blinking • Transparent • Antialiasing • Nominal value • Underline • Selecting • Change variable immediately • Auto-select next on <Enter>

Colors

Selected text color	Color of text when variable value is within limits.
Selected background color	Background color of variable
	Background will not be drawn in case of transparent display. Allows positioning object e.g. over a bitmap.
Selected font	Font used for displaying object in selected mode.
Selected font size	Font size used for displaying object in selected mode.
Selected frame color	Frame color used for displaying object in selected mode.
Selected frame width	Frame width used for displaying object in selected mode.
Edit frame color	Color of text for object in editing mode.
Edit background color	Background color of object for object in editing mode when displaying non-transparent.
Edit font	Font used for displaying object in editing mode.
Edit font size	Font size used for displaying object in editing mode.
Edit frame color	Color of frame around object for selected object in editing mode.
Edit frame width	Frame width used for displaying object in editing mode.

TrueType time and date

TrueType time and date displays time or date.

Note
Changing the position in the mask or container is not possible during runtime.

The TrueType resource defines the font.

Fig. 47   Properties window

Fig. 48   Color selection

Time/Date format	Display format of time or date
	Possible formats:
	• hh:mm:ss (24h)
	• hh:mm (24h)
	• hh:mm:ss (am/pm)
	• hh:mm (am/pm)
	• DD.MM.YYYY
	• MM/DD/YY
	• MM/DD/YYYY
	• DD.MM.YY
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Font	TrueType font of text element
	It is possible to replace the list of fonts displayed by a personal list.
	Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display. If the attribute *Transparent* is active, the background is not displayed.

Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▶ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Inverted
	• Blinking
	• Transparent
	• Underline
	• Adjustable
	• Selecting
	• Antialiasing
	• Auto-select next on \<Enter\>

Colors

Selected text color	Color of text when variable value is within limits.
Selected background color	Background color of variable
	Background will not be drawn in case of transparent display. Allows positioning object e.g. over a bitmap.
Selected font	Font used for displaying object in selected mode.
Selected font size	Font size used for displaying object in selected mode.
Selected frame color	Frame color used for displaying object in selected mode.
Selected frame width	Frame width used for displaying object in selected mode.
Edit frame color	Color of text for object in editing mode.
Edit background color	Background color of object for object in editing mode when displaying non-transparent.
Edit cursor color	Color of cursor for variable object in editing mode.
Edit font	Font used for displaying object in editing mode.
Edit font size	Font size used for displaying object in editing mode.
Edit frame color	Color of frame around object for selected object in editing mode.
Edit frame width	Frame width used for displaying object in editing mode.

TrueType variable @

TrueType variable displays the value of a variable, depending on

• the set format.
• the data type.

Note
Changing the position in the mask or container is not possible during runtime.

Properties	Colors	Limits	
Format	@@@@@	
Data type	32-bit unsigned integer	▼
Origin X	73	▲▼
Origin Y	377	▲▼
Font	Standard	▼
Font size	16	▲▼
Text color	#000000	■
Background color	#FFFFFF	
Variable	Var_1_1	...
Attribute	0x00000028	...
Visible	☑ true	
Blinking	☐ false	
Underline	☐ false	
Inverted	☐ false	
Transparent	☐ false	
Nominal value	☐ false	
Antialiasing	☑ true	
Change variable immediately	☐ false	
Grab focus	☐ false	
Auto-select next on <enter>	☐ false	

Fig. 49   Properties window

Properties	Colors	Limits	
Selected text color	#FFFFFF	
Selected background color	#000000	■
Selected font	Standard	▼
Selected font size	16	▲▼
Selected frame color	#000000	■
Selected frame width	0	▲▼
Edit text color	#FFFFFF	
Edit background color	#000000	■
Edit cursor color	#FFFFFF	
Edit font	Standard	▼
Edit font size	16	▲▼
Edit frame color	#000000	■
Edit frame width	0	▲▼

Fig. 50   Selected color

Fig. 51   Limits

Properties

Format	Display format of variable
	Every @ character stands for a digit in the variable value displayed.
	Number of digits displayed for the variable value is limited to the number of @ characters.
	Set decimal places by entering a point.
Data type	Data type of variable
	Variables in the variable table are always 32-bit variables in the memory of the control set.
	It is possible to display only a part of the variable value (LO word).
	Possible data types:
	• 16-bit binary
	• 16-bit signed integer
	• 16-bit unsigned integer
	• 16-bit hexadecimal
	• 32-bit binary
	• 32-bit signed integer
	• 32-bit unsigned integer
	• 32-bit hexadecimal
	• 32-bit float
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)

Font	TrueType font of text element
	It is possible to replace the list of fonts displayed by a personal list.
	Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display
	If the attribute *Transparent* is active, the background is not displayed.
Variable	Variable whose value is to be visualized.
Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Inverted
	• Blinking
	• Transparent
	• Nominal value
	• Underline
	• Selecting
	• Antialiasing
	• Change variable immediately
	• Auto-select next on <Enter>

Colors

Selected text color	Color of text when variable value is within limits
Selected background color	Background color of variable
	Background will not be drawn in case of transparent display. Allows positioning object e.g. over a bitmap.
Selected font	Font used for displaying object in selected mode.
Selected font size	Font size used for displaying object in selected mode.
Selected frame color	Frame color used for displaying object in selected mode.

Selected frame width	Frame width used for displaying object in selected mode.
Edit frame width	Color of text for object in editing mode.
Edit background color	Background color of object for object in editing mode when displaying non-transparent.
Edit cursor color	Color of cursor for variable object in editing mode.
Edit font	Font used for displaying object in editing mode.
Edit font size	Font size used for displaying object in editing mode.
Edit frame color	Color of frame around object for selected object in editing mode.
Edit frame width	Frame width used for displaying object in editing mode.

Limits

Upper limit	Upper limit for nominal value input ▸ Indicate upper limit as numerical value. 　- or - ▸ Select variable from list. If the value of the variable exceeds the upper limit, color and behavior of the display will change to set values.
Upper text color	Text color when exceeding upper limit.
Upper background color	Background color when exceeding upper limit.
Blinking text	Text blinks when exceeding upper limit.
Blinking background	Background blinks when exceeding upper limit.
Lower limit	Lower limit for nominal value input ▸ Indicate lower limit as numerical value. 　- or - ▸ Select variable from list. If the value of the variable falls below the lower limit, color and behavior of the display will change to set values.
Lower text color	Text color when falling below the lower limit.
Lower background color	Background color when falling below the lower limit.
Blinking text	Text blinks when falling below lower limit.
Blinking background	Background blinks when falling below lower limit.

Step value	Step value of nominal value input
	‣ Select step value from list.
	- or -
	‣ Select variable from list.
	Step value other than 0:
	• Variable will be incremented or decremented by the entered step value when editing.
	Step value 1 or no entry
	• Variable will be incremented or decremented by 1 when editing.

10.2 Resource text objects

Resource text objects obtain the text contents from texts or text lists defined in the text resource (*see 5.6 Resources*).

Like TrueType text objects, resource text objects are created like vector graphics and feature a smoother typeface thanks to aliasing.
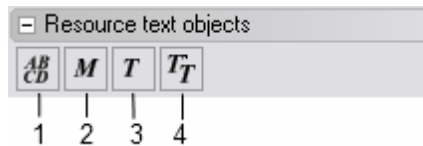

Fig. 52 Resource text objects

    (1) Resource TrueType edit text
    (2) Resource TrueType menu text
    (3) Resource TrueType text
    (4) Resource TrueType text list

Resource TrueType edit text

Resource-TrueType editable text displays text defined in the text resource.

Via the system variable SYS_LANGUAGE it is possible to change the language during runtime if the respective languages are available in the text resource (*see 5.6 Resources*).

Resource TrueType editable text creates menu structures for

- calling up masks
- showing / hiding containers.

It is possible to align resource TrueType menu text horizontally and vertically.

The basis for calculating the text dimensions is the longest text in all defined languages.

When adding the object on the display:

- Select respective text in resource dialog.

Note
Changing the position in the mask or container is not pos-sible during runtime.

Fig. 53 Properties window

Fig. 54 Color selection

Available characters	Name of text in text resource To edit selected text or select a different text list: Click button ... .
Length	Maximum length for text input
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Font	TrueType font of text element It is possible to replace the list of fonts displayed by a personal list. Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display If the attribute *Transparent* is active, the background is not displayed.
Attribute	Field with individual flags which influences the display of the variables. Entering HEX values in the input field is possible to create a static attribute. To change the attribute in the control set during runtime: ▸ Select the variable controlling the attribute from list. Attributes supported: • Visible • Inverted • Blinking • Transparent • Underline • Antialiasing • Selecting • Change text immediately • Auto-select next on \<Enter\>

Colors

Selected text color	Color of text when variable value is within limits
Selected background color	Background color of variable
	Background will not be drawn in case of transparent display. Allows positioning object e.g. over a bitmap.
Selected font	Font used for displaying object in selected mode.
Selected font size	Font size used for displaying object in selected mode.
Selected frame color	Frame color used for displaying object in selected mode.
Selected frame width	Frame width used for displaying object in selected mode.
Edit text color	Color of text for object in editing mode.
Edit background color	Background color of text for object in editing mode when displaying non-transparent.
Edit cursor color	Color of cursor for variable object in editing mode.
Edit font	Font used for displaying object in editing mode.
Edit font size	Font size used for displaying object in editing mode.
Edit frame color	Color of frame around object for selected object in editing mode.
Edit frame width	Frame width used for displaying object in editing mode.

Resource TrueType menu text M

Resource TrueType text displays text defined in the text resource.

Via the system variable *SYS_LANGUAGE* it is possible to change the language during runtime if the respective languages are available in the text resource (*see 5.6 Resources*).

Resource TrueType menu text creates menu structures to

- calling up masks.
- showing/hiding containers.

All masks are managed via a stack. When showing a mask, it is put on top of the stack.

All message containers are managed via a stack. When showing a message container, it is put on top of the stack.

It is possible to align resource TrueType menu text horizontally and vertically.

The basis for calculating the text dimensions is the longest text in all defined languages.

When adding the object on the display:

▸ Select respective text in resource dialog.

Note
Changing the position in the mask or container is not possible during runtime.

The TrueType resource defines the font.
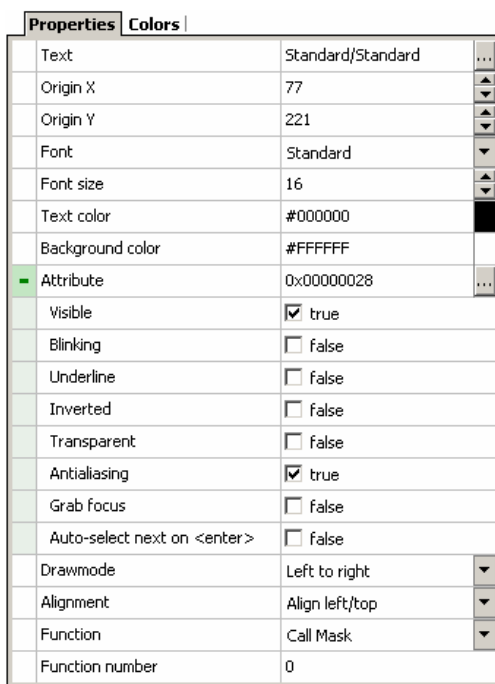
Fig. 55   Properties window

Fig. 56   Color selection
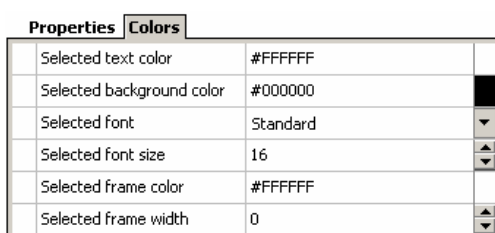
Properties

Text	Name of text in text resource To edit selected text or select a different text list: ▸ Click button […].
Origin X	Position of left upper corner of object (x coordinate)

Origin Y	Position of left upper corner of object (y coordinate)
Font	TrueType font of text element
	It is possible to replace the list of fonts displayed by a personal list.
	Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display
	If the attribute *Transparent* is active, the background is not displayed.
Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Inverted
	• Blinking
	• Transparent
	• Underline
	• Antialiasing
	• Selecting
	• Auto-select next on \<Enter\>
Text alignment	Horizontal alignment of text in rectangle
	Possible text alignments:
	• Left to right
	• Right to left
	• Top to bottom
Alignment	Vertical alignment of text in rectangle
	Possible alignments:
	• Left/top
	• Centered
	• Right/bottom
Function	*see Function list*

Function number	Number for set function

Call up mask	Show mask with the number indicated in field *Number*.
	If this mask is already on the stack with another mask covering it, the display remains unchanged.
Call up prio mask	Show mask with the number indicated in field *Number*.
	If this mask is already on the stack with another mask covering it, the mask called up is put on top of the stack and displayed.
Return to previous mask	Hide mask on top and show mask below.
Show info container	Show info container with the number indicated in field *Number*.
Hide info container	Hide info container with the number indicated in field *Number*.
Show message container	Show message container with the number indicated in field *Number*.
Hide message container	Hide message container with the number indicated in field *Number*.
	If message container is on top of stack:
	• Message container below is shown.
	If message container is anywhere but on top of stack:
	• Message container is removed from stack.
	If no message container is on top of stack anymore:
	• Currently shown mask is displayed.
Transfer menu index to C	Transfer current menu index to user C program.

Colors

Selected text color	Color of text when variable value is within limits
Selected background color	Background color of variable
	Background will not be drawn in case of transparent display. Allows positioning object e.g. over a bitmap.
Selected font	Font used for displaying object in selected mode.
Selected font size	Font size used for displaying object in selected mode.
Selected frame color	Frame color used for displaying object in selected mode.
Selected frame width	Frame width used for displaying object in selected mode.

Resource TrueType text T

Resource TrueType text displays text defined in the text resource.

Via the system variable *SYS_LANGUAGE* it is possible to change the language during runtime if the respective languages are available in the text resource (*see 5.6 Resources*).

It is possible to align the text list horizontally and vertically.

The basis for calculating the text dimensions is the longest text in all defined languages.

When adding the object on the display:
‣ Select respective text in resource dialog.

Note
Changing the position in the mask or container is not possible during runtime.
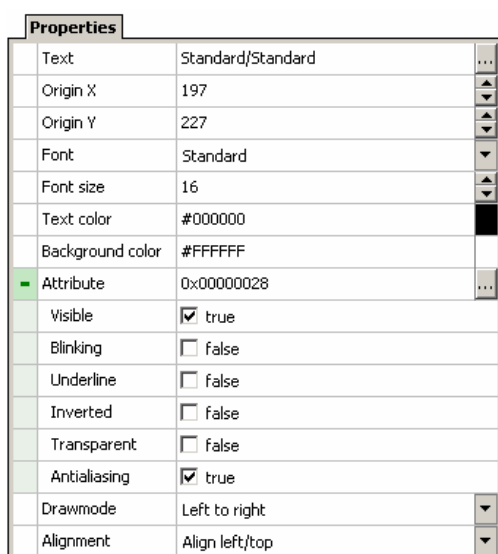
The TrueType resource defines the font.

Fig. 57 Properties window

Properties

Text	Name of text in text resource
	Text length is limited to 255 characters
	To edit selected text or select a different text list:
	‣ Click button ... .
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)

Font	TrueType font of text element
	It is possible to replace the list of fonts displayed by a personal list.
	Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display
	If the attribute *Transparent* is active, the background is not displayed.
Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Inverted
	• Blinking
	• Transparent
	• Underline
	• Antialiasing
Text alignment	Horizontal alignment of text in rectangle
	Possible text alignments:
	• Left to right
	• Right to left
	• Top to bottom
Alignment	Vertical alignment of text in rectangle
	Possible alignments:
	• Left/top
	• Centered
	• Right/bottom

Resource TrueType text list T_T

Resource TrueType text list displays a text list. A text list is a collection of several texts in the text resource.

Via the system variable $SYS\_LANGUAGE$ it is possible to change the language during runtime if the respective languages are available in the text resource (*see 5.6 Resources*).

It is possible to align the text list horizontally and vertically.

The basis for calculating the text list dimensions is the longest text in all defined languages.

When adding the object on the display:

‣ Select respective text list in resource dialog.

Note
Changing the position in the mask or container is not possible during runtime.

The TrueType resource defines the font.

Properties	Colors
Text	@L1
Origin X	324
Origin Y	225
Font	Standard
Font size	16
Text color	#000000
Background color	#FFFFFF
Variable	SYS_U_SUPPLY
Attribute	0x00000028
Visible	☑ true
Blinking	☐ false
Underline	☐ false
Inverted	☐ false
Transparent	☐ false
Antialiasing	☑ true
Nominal value	☐ false
Change variable immediately	☐ false
Grab focus	☐ false
Auto-select next on <enter>	☐ false
Drawmode	Left to right
Alignment	Align left/top

Fig. 58 Properties window

Fig. 59   Color selection

Properties

Text	Name of text list in text resource
	@ character signals distinction between text lists and texts.
	To edit selected text list or select a different text list:
	▸ Click button ... .
Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Font	TrueType font of text element
	It is possible to replace the list of fonts displayed by a personal list.
	Every font requires additional memory capacity in the project memory.
Font size	Font size in pixels
Text color	Text color for normal display
Background color	Color of text background for normal display
	If the attribute *Transparent* is active, the background is not displayed.
Variable	Variable that defines which text list entry is displayed.

Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Inverted
	• Blinking
	• Transparent
	• Underline
	• Antialiasing
	• Nominal value
	• Selecting
	• Change variable immediately
	• Auto-select next on \<Enter>
Text alignment	Horizontal alignment of text list in rectangle
	Possible text alignments:
	• Left to right
	• Right to left
	• Top to bottom
Alignment	Vertical alignment of text list in rectangle
	Possible alignments:
	• Left/top
	• Centered
	• Right/bottom

Colors

Selected text color	Color of text when variable value is within limits
Selected background color	Background color of variable
	Background will not be drawn in case of transparent display. Allows positioning object e.g. over a bitmap.
Selected font	Font used for displaying object in selected mode.
Selected font size	Font size used for displaying object in selected mode.
Selected frame color	Frame color used for displaying object in selected mode.
Selected frame width	Frame width used for displaying object in selected mode.
Edit text color	Color of text for object in editing mode.
Edit background color	Background color of object for object in editing mode when displaying non-transparent.
Edit font	Font used for displaying object in editing mode.
Edit font size	Font size used for displaying object in editing mode.
Edit frame color	Color of frame around object for selected object in editing mode.
Edit frame width	Frame width used for displaying object in editing mode.

10.3 Graphic objects

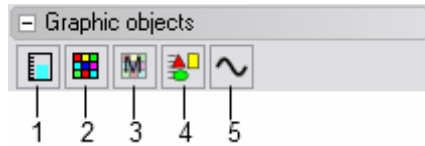GDSDesigner provides tools for simple, static and animateable graphical objects.

Fig. 60   Graphic objects

      (1) Bar chart
      (2) Single-picture graphic
      (3) Menu bitmap
      (4) Simple bitmap
      (5) Trend writer

Bar graphic

A bar graphic displays the visualization of a variable value as a bar.

The variable value is visualized as a colored rectangle whose size can be changed during runtime.

For displaying a scale, it is possible to show a bitmap object behind the bar chart.

Note
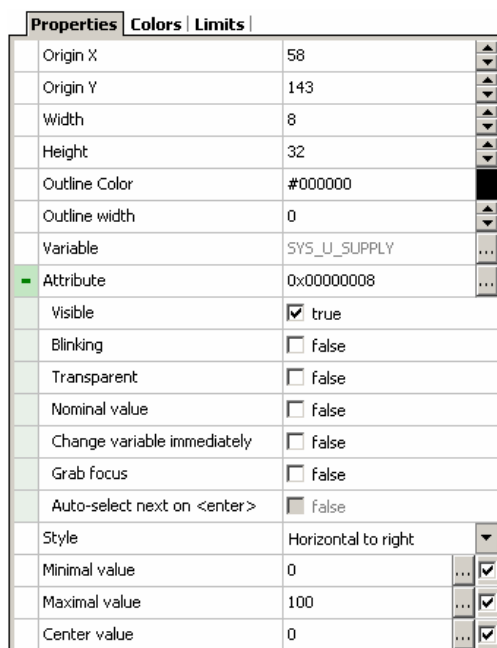*Changing the position in the mask or container is not possible during runtime.*

Properties	Colors	Limits
Origin X	58	
Origin Y	143	
Width	8	
Height	32	
Outline Color	#000000	
Outline width	0	
Variable	SYS_U_SUPPLY	
Attribute	0x00000008	
Visible	☑ true	
Blinking	☐ false	
Transparent	☐ false	
Nominal value	☐ false	
Change variable immediately	☐ false	
Grab focus	☐ false	
Auto-select next on \<enter\>	☐ false	
Style	Horizontal to right	
Minimal value	0	☑
Maximal value	100	☑
Center value	0	☑

Fig. 61   Properties window

Fig. 62   Color selection

Fig. 63   Limits

Properties

Origin X	Position of left upper corner of object (x coordinate) Assigning a variable is possible.
Origin Y	Position of left upper corner of object (y coordinate) Assigning a variable is possible.
Width	Maximum width of bar chart in pixels
Height	Maximum height of bar chart in pixels
Outline color	Color of frame around object for selected object in editing mode.
Outline width	Frame width used for displaying object in editing mode.
Variable	Variable whose value is to be visualized.

Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Blinking
	• Transparent
	• Selecting
	• Nominal value
	• Change variable immediately
	• Auto-select next on <Enter>
Bar style	Bar style of bar chart
	Possible bar styles:
	• Horizontal bar to the right
	• positive values increasing to the right
	• negative values decreasing to the left
	• Vertical bar upwards
	• positive values increasing upwards
	• negative values decreasing downwards
	• Horizontal bar to left
	• positive values increasing to the left
	• negative values decreasing to the right
	• Vertical bar downwards
	• positive values increasing downwards
	• negative values decreasing upwards
	For every bar style, it is possible to select it with the segmented option, i.e. the bars will be divided into normal, lower and upper area if values exceed or fall below limits (*see Limits*).
Minimum value	Minimum value of variables for full bar
Maximum value	Maximum value of variables for full bar
Center	Offset of variable value for asymmetrical positioning of positive and negative values Scaling unchanged

Colors

Bar color	Color of bar when variable value is within limits.
Background color	Background color of object If the attribute *Transparent* is active, the background is not displayed.
Gradient color	Color of bar if option *Gradient* is active.
Use color gradient	Display bar with horizontal color gradient for 3D effect.
Selected frame color	Frame color used to display object in selected mode.
Selected frame width	Frame width used to display object in selected mode.
Edit Frame color	Color of frame around object for selected object in editing mode.
Edit frame width	Frame width used to display object in selected mode.

Limits

Upper limit	Upper limit ▸ Indicate upper limit as numerical value. - or - ▸ Select variable from list. If the value of the variable exceeds the upper limit, color and behavior of the display will change to set values.
Upper bar color	Bar color when exceeding upper limit.
Upper background color	Background color when exceeding upper limit.
Upper gradient color	Gradient color when exceeding upper limit.
Blinking bar	Bar blinks when exceeding upper limit.
Blinking background	Background blinks when exceeding upper limit.
Lower limit	Lower limit ▸ Indicate lower limit as numerical value. - or - ▸ Select variable from list. If the value of the variable falls below the lower limit, color and behavior of the display will change to set values.
Lower bar color	Bar color when falling below the lower limit.
Lower background color	Background color when falling below the lower limit.

Lower gradient color	Gradient color when falling below the lower limit.
Blinking bar	Bar blinks when falling below lower limit.
Blinking background	Background blinks when falling below lower limit.
Step value	Used exclusively for entering Nominal value. Step value other than 0: • Variable will be incremented or decremented by the entered step value when editing. Step value 1 or no entry: • Variable will be incremented or decremented by 1 when editing.

Single-picture graphic

A single-picture graphic displays the part of a bitmap graphic. The individual pictures in the bitmap graphic are positioned in rows and columns.
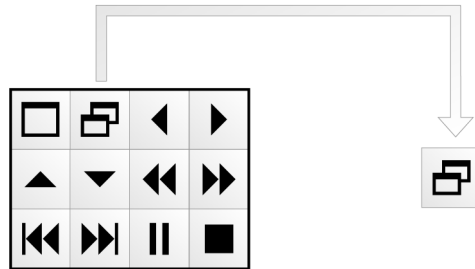
Fig. 64   Basic bitmap graphic and single-picture graphic

By configuring the properties *Variable* and *Frequency*, the single picture graphic allows simple animations.

When adding object:
▸ Select respective file.
▸ If necessary, adjust height and width of graphic.
▸ Indicate number of images per line.
▸ Indicate overall number of images.

It is possible to change the graphic using the configured drawing program (*see 8 Presettings*).

To edit graphic:
▸ Double-click graphic.

Note
Changing the position in the mask or container is not possible during runtime.

Fig. 65   Properties window

Fig. 66   Color selection

Properties

File name	File name of graphic
Origin X	Position of left upper corner of object (x coordinate) Assigning a variable is possible. If *Origin X* is assigned to a variable, it is no longer possible to move the graphic in direction x on the display. To adjust position: ▸ Change value of variable in variable table.
Origin Y	Assigning a variable is possible. If *Origin Y* is assigned to a variable, it is no longer possible to move the graphic in direction y on the display. To adjust position: ▸ Change value of variable in variable table.

Image width	Width of single-picture graphic in pixels
Image height	Height of single-picture graphic in pixels
Images per line	Number of images per line.
No. of images	Overall number of images
Variable	Index of individual image displayed of basic bitmap graphic. Index 0 = image in left top corner of basic bitmap graphic If index > number of individual pictures in basic bitmap graphic, the last individual picture is displayed.
Frequency	Defines the intervals (in sec) between individual images when being shown in succession. Interval range: • 0 ... 48 images/s
Transparency color	*Transparency color* indicates the color not displayed when drawing. Objects below are visible. It is possible to switch transparency on or off during runtime via the attribute *Transparency.* Transparent areas in a graphic slow down the display of the graphic.
Attribute	Field with individual flags which influences the display of the variables. Entering HEX values in the input field is possible to create a static attribute. To change the attribute in the control set during runtime: ▶ Select the variable controlling the attribute from list. Attributes supported: • Visible • Blinking • Transparent • Selecting • Nominal value • Self Animated • Change variable immediately • Auto-select next variable on <Enter>

Colors

Selected frame color	Frame color used for displaying object in selected mode.
Selected frame width	Frame width used for displaying object in selected mode.
Edit frame color	Color of frame around object for selected object in editing mode
Edit frame width	Frame width used to display object in selected mode.

Menu bitmap

Menu bitmap creates menu structures for

- calling up masks.
- showing/hiding containers.

All masks are managed via a stack. When showing a mask, it is put on top of the stack.

All message containers are managed via a stack. When showing a message container, it is put on top of the stack.

When adding object:

▸ Select respective file.

▸ If necessary, adjust height and width of graphic.

It is possible to change the graphic using the configured drawing program (*see 8 Presettings*).

To edit graphic:

▸ Double-click graphic.

Note
Changing the position in the mask or container is not possible during runtime.

Properties	Colors	
Filename	TSTMENUBMP1_40X40.BMP	
Origin X	166	
Origin Y	148	
Image width	20	
Image height	40	
Transparency color	#FFFFFF	
Attribute	0x00000008	
Visible	☑ true	
Blinking	☐ false	
Transparent	☐ false	
Grab focus	☐ false	
Auto-select next on <enter>	☐ false	
Function	Call Mask	
Function number	0	

Fig. 67   Properties window

Properties	Colors	
Selected frame color	#000000	
Selected frame width	1	

Fig. 68   Color selection

Properties

File name	File name of graphic
Origin X	Position of left upper corner of object (x coordinate) Assigning a variable is possible. If *Origin X* is assigned to a variable, it is no longer possible to move the graphic in direction x on the display. To adjust position: ▸ Change value of variable in variable table.
Origin Y	Position of left upper corner of object (y coordinate) Assigning a variable is possible. If *Origin Y* is assigned to a variable, it is no longer possible to move the graphic in direction y on the display. To adjust position: ▸ Change value of variable in variable table.
Image width	Width of graphic in pixels
Image height	Height of graphic in pixels
Transparency color	*Transparency color* indicates the color not displayed when drawing. Objects below are visible. It is possible to switch transparency on or off during runtime via the attribute *Transparency*. Transparent areas in a graphic slow down the display of the graphic.
Attribute	Field with individual flags which influences the display of the variables. Entering HEX values in the input field is possible to create a static attribute. To change the attribute in the control set during runtime: ▸ Select the variable controlling the attribute from list. Attributes supported: • Visible • Blinking • Transparent • Selecting • Auto-select next object on \<Enter\>

Function	*see Function list*
Function number	Number for set function

Function list

Call up mask	Show mask with the number indicated in field *Number*. If this mask is already on the stack with another mask covering it, the display remains unchanged.
Call up prio mask	Show mask with the number indicated in field *Number*. If this mask is already on the stack with another mask covering it, the mask called up is put on top of the stack and displayed.
Return to previous mask	Hide mask on top and show mask below.
Show info container	Show info container with the number indicated in field *Number*.
Hide info container	Hide info container with the number indicated in field *Number*.
Show message container	Show message container with the number indicated in field *Number*.
Hide message container	Hide message container with the number indicated in field *Number*. If message container is on top of stack: • Message container below is shown. If message container is anywhere but on top of stack: • Message container is removed from stack. If no message container is on top of stack anymore: • Currently shown mask is displayed.
Transfer menu index to C	Transfer current menu index to user C program.

Colors

Selected frame color	Frame color used for displaying object in selected mode.
Selected frame width	Frame width used for displaying object in selected mode.

Simple bitmap

Simple bitmap displays bitmap graphic in defined place within a mask or container.

When adding object:
▸ Select respective file.

It is possible to
• change the graphic's position during runtime.
• change the graphic using the configured drawing program (*see 8 Presettings*).
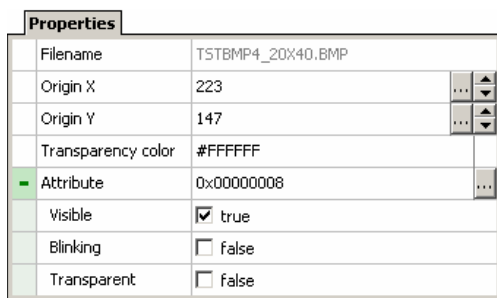
To edit graphic:
▸ Double-click graphic.

Properties	
Filename	TSTBMP4_20X40.BMP
Origin X	223
Origin Y	147
Transparency color	#FFFFFF
Attribute	0x00000008
Visible	☑ true
Blinking	☐ false
Transparent	☐ false

Fig. 69 Properties window

Properties

File name	File name of graphic
Origin X	Position of left upper corner of object (x coordinate) Assigning a variable is possible. If *Origin X* is assigned to a variable, it is no longer possible to move the graphic in direction x on the display. To adjust position: ▸ Change value of variable in variable table.
Origin Y	Position of left upper corner of object (y coordinate) Assigning a variable is possible. If *Origin Y* is assigned to a variable, it is no longer possible to move the graphic in direction y on the display. To adjust position: ▸ Change value of variable in variable table.

Transparency color	*Transparency color* indicates the color not displayed when drawing. Objects below are visible.
	It is possible to switch transparency on or off during runtime via the attribute *Transparency*.
	Transparent areas in a graphic slow down the display of the graphic.
Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Blinking
	• Transparent

Trend writer ∿

The trend writer graphically represents the value of a variable within a set period of time.

For displaying a scale, it is possible to show a bitmap object behind the trend writer.

Note
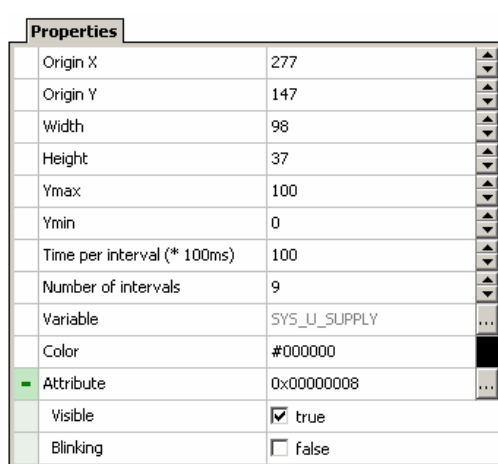*Changing the position in the mask or container is not possible during runtime.*

Fig. 70 Properties window

Properties

Origin X	Position of left upper corner of object (x coordinate)
Origin Y	Position of left upper corner of object (y coordinate)
Height	Object height
Width	Object width
Ymax, Ymin	Area the object is scaled to fit. If variable value ≥ *Ymax*: • Graph is shown on top edge. If variable value ≤ *Ymin*: • Graph is shown on bottom edge.
Time per interval (100 ms)*	The variable value is recorded in a defined interval. Setting defines time between two recordings in steps of 100 ms.
Number of intervals	Number of intervals in objects.

Variable	Variable whose value is to be visualized.
Color	Graph color
Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	‣ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Blinking

10.4 Vector graphic object

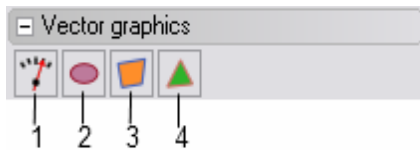Fig. 71   Vector graphic object

    (1) Pointer instrument
    (2) Ellipse
    (3) Quadrangle
    (4) Triangle

Pointer instrument

Pointer instrument displays a pointer.

For displaying a scale, it is possible to show a bitmap object behind the bar chart.

Note
Changing the position in the mask or container is not possible during runtime.

Fig. 72   Properties window

Fig. 73 Properties window

Fig. 74 Properties window

Properties

Origin X	Origin of pointer (x-coordinate)
Origin Y	Origin of pointer (y-coordinate)
Radius	Pointer radius
Origin offset	Offset of pointer origin
Start angle	Start angle of pointer
End angle	End angle of pointer
Start value	Variable value at which pointer shows minimum deflection.
End value	Variable value at which pointer shows maximum deflection.
Graphic	Pointer graphic used
Variable	Variable whose value is to be visualized.

Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Blinking
	• Nominal value

Colors

Pointer color	Color of pointer when variable value is within limits.
Pointer outline color	Pointer outline color of object
Pointer outline width	Pointer outline width of object
Selected pointer clor	Color used for displaying object in selected mode
Selected outline color	Outline color used to display object in selected mode.
Selected outline width	Outline width used to display object in selected mode.
Edit pointer color	Color of selected pointer in editing mode
Edit outline color	Color of outline around object for selected object in editing mode.
Edit outline width	Outline width used to display object in selected mode.

Limits

Upper limit	Upper limit
	▸ Indicate upper limit as numerical value.
	- or -
	▸ Select variable from list.
	If the value of the variable exceeds the upper limit, color and behavior of the display will change to set values.
Upper pointer color	Pointer color when exceeding upper limit.
Upper outline color	Outline color when exceeding upper limit.
Blinking pointer	Pointer blinks when exceeding upper limit.

Blinking outline	Outline blinks when exceeding upper limit.
Lower limit	Lower limit ▸ Indicate lower limit as numerical value. - or - ▸ Select variable from list. If the value of the variable falls below the lower limit, color and behavior of the display will change to set values.
Lower pointer color	Pointer color when falling below the lower limit.
Lower outline color	Outline color when falling below the lower limit.
Blinking pointer	Pointer blinks when falling below lower limit.
Blinking outline	Outline blinks when falling below lower limit.
Step value	Used exclusively for entering Nominal value. Step value other than 0: • Variable will be incremented or decremented by the entered step value when editing. Step value 1 or no entry: • Variable will be incremented or decremented by 1 when editing.

Ellipse

Ellipse is a geometrical object for displaying an ellipse or a circle.

Note
GRAF-SYTECO recommends using the object exclusively when one or more corner points are to be changed during runtime. In order to not slow down the display speed, GRAF-SYTECO recommends using a bitmap object for unmodifiable objects.

Fig. 75   Properties window

Properties

Fill color	Fill color of ellipse If the attribute *Transparent* is active, the filling is not displayed.
Outline color	Color of ellipse outline
Outline width	Width of ellipse outline in 1/10 pixels
Origin X, *Origin Y*	Center of ellipse (x or y-coordinate) ▸ Indicate center as numerical value. - or - ▸ Select variable from list. If the coordinate of a center is assigned to a variable, moving the ellipse in the coordinate direction on the display is no longer possible.
Radius X, *Radius Y*	radius of ellipse (x or y-coordinate) ▸ Indicate radius as numerical value. - or - ▸ Select variable from list.

Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Inverted
	• Blinking
	• Transparent

Quadrangle

Quadrangle is a geometrical object for displaying a quadrangle.

Note
GRAF-SYTECO recommends using the object exclusively when one or more corner points are to be changed during runtime. In order to not slow down the display speed, GRAF-SYTECO recommends using a bitmap object for unmodifiable objects.

Properties		
Fill color	#000000	■
Outline Color	#FFFFFF	
Outline width	0	
P1 X	247	
P1 Y	40	
P2 X	289	
P2 Y	40	
P3 X	289	
P3 Y	75	
P4 X	247	
P4 Y	75	
Attribute	0x00000008	
Visible	☑ true	
Blinking	☐ false	
Inverted	☐ false	
Transparent	☐ false	

Fig. 76 Properties window

Properties

Fill color	Fill color of quadrangle If the attribute *Transparent* is active, the filling is not displayed.
Outline Color	Color of quadrangle
Outline width	Width of quadrangle outline in 1/10 pixels
P1 X, P1 Y, *P1 X, P2 Y,* *P1 X, P3 Y,* *P1 X, P4 Y,*	Corner points of quadrangle on x or y-coordinates ▸ Indicate corner points as numerical values. - or - ▸ Select variable from list. If the coordinate of a corner point is assigned to a variable, moving the quadrangle in the coordinate direction on the display is no longer possible.

Attribute	Field with individual flags which influences the display of the variables.
	Entering HEX values in the input field is possible to create a static attribute.
	To change the attribute in the control set during runtime:
	▸ Select the variable controlling the attribute from list.
	Attributes supported:
	• Visible
	• Inverted
	• Blinking
	• Transparent

Triangle

Triangle is a geometrical object for displaying a triangle.

Note
GRAF-SYTECO recommends using the object exclusively when one or more corner points are to be changed during runtime. In order to not slow down the display speed, GRAF-SYTECO recommends using a bitmap object for unmodifiable objects.

Properties	
Fill color	#000000
Outline Color	#FFFFFF
Outline width	0
P1 X	363
P1 Y	42
P2 X	339
P2 Y	78
P3 X	388
P3 Y	78
Attribute	0x00000008
Visible	☑ true
Blinking	☐ false
Inverted	☐ false
Transparent	☐ false

Fig. 77   Properties window

Properties

Fill color	Fill color of triangle If the attribute *Transparent* is active, the filling is not displayed.
Outline Color	Color of triangle
Outline width	Width of triangle outline in 1/10 pixels
P1 X, P1 Y, *P1 X, P2 Y,* *P3 X, P3 Y*	Corner points of triangle on x or y-coordinates ▸ Indicate corner points as numerical values. - or - ▸ Select variable from list. If the coordinate of a corner point is assigned to a variable, moving the triangle in the coordinate direction on the display is no longer possible.

Attribute	Field with individual flags which influences the display of the variables. Entering HEX values in the input field is possible to create a static attribute. To change the attribute in the control set during runtime: ▸ Select the variable controlling the attribute from list. Attributes supported: • Visible • Inverted • Blinking • Transparent

11 Chapter 11
Scale generator

The scale generator is a tool for creating and adjusting scales, e.g. of a tachometer.

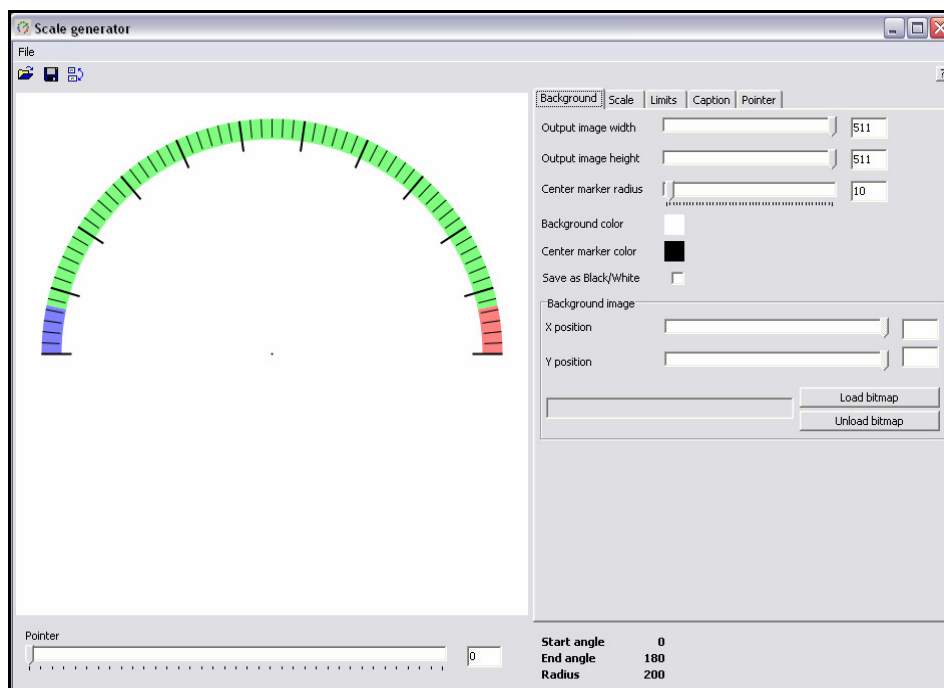To start scale generator:

▸ Select *Scale generator* in menu *Extras*.

Fig. 78   Scale generator

To open a scale:

▸ Click on 📂 in button bar.

To save a scale:

▸ Click on 💾 in button bar.

To change into widescreen mode:

▸ Click on ⊞ in button bar.

Background Tab *Background* allows setting size and base colors of scale.
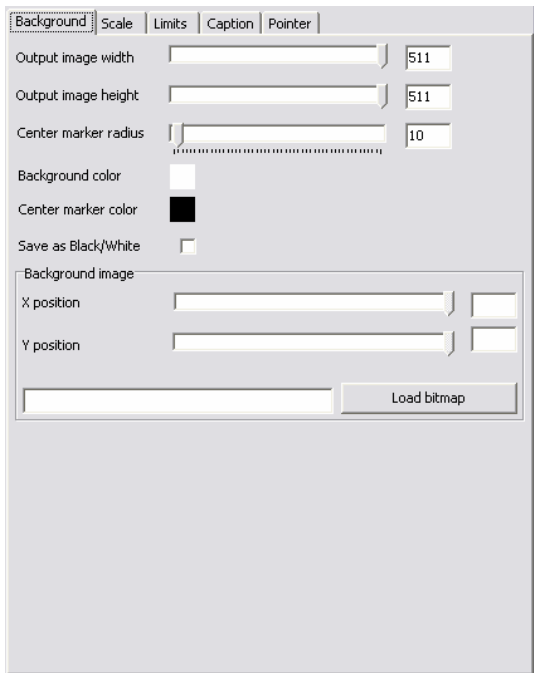

Fig. 79   Scale generator; Background

Scale Tab *Scale* allows setting scaling, scale colors and scale angle.
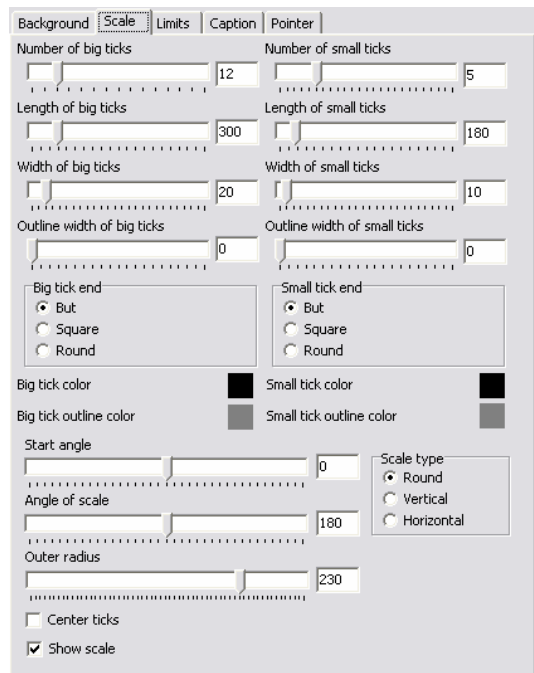

Fig. 80   Scale generator, Scale

Tab *Limits* allows setting upper and lower limits and scale color for limit width.

Limits

Fig. 81   Scale generator; Limits

Tab *Caption* allows setting the scale caption.

Caption

Fig. 82   Scale generator; Caption

Pointer   Tab *Pointer* allows setting pointer characteristics and importing a pointer.

Fig. 83   Scale generator, Pointer

12 Chapter 12
Shortcuts

Ctrl N	New mask
Ctrl S	Save project
Ctrl A	Select all objects
Escape	Deselect all objects
Ctrl C	Copy
Ctrl V	Paste
Ctrl Z	Undo
Ctrl Alt C	Copy image in clipboard
Ctrl left MouseDown	Object multiselection +/-
Ctrl Shift Mouse	Move/enlarge/reduce container
Ctrl +	Zoom +
Ctrl -	Zoom -
Ctrl MouseWheel	Zoom +/-
Ctrl 0..9	Select tab (0 = Design, 1 = Code, ...)

Selected objects

Ctrl O	Change object ID
Ctrl R	Rename object/mask
Ctrl Left/Right/Up/Down	Align objects with left/right/top/bottom edge
Alt PageUp	Horiz. even spacing of objects
Alt PageDown	Vert. even spacing of objects
Alt Left/Right/Up/Down	Move objects to right/left/top/bottom (1 pixel)
Alt Shift Left/Right/Up/ Down	Move objects to right/left/top/bottom (5 pixel)

Delete		Delete object/mask
Alt G		Open device settings
Space		Check/uncheck in inspector
F1		Call up help menu
F9		Make
Shift F9		Make clean
Ctrl F9		Make rebuild
F10		Binary export
F12		Tab change: design < > code

Navigator

Ctrl Pos1	Object/mask order to front
Ctrl End	Object/mask order to back
Ctrl PageUp	Object/mask order toward front by one level
Ctrl PageDown	Object/mask order toward back by one level
Alt Pos1	To navigator start
Alt End	To navigator end
Space	Check/uncheck driver in navigator

CodeTab

Ctrl N	New file
Ctrl O	Open file
Ctrl S	Save file
Ctrl F	Search
Ctrl R	Replace
Ctrl H	Replace
Ctrl G	Grep search
Ctrl J	C templates
Ctrl C	Copy
Ctrl V	Paste
Ctrl Z	Undo
Ctrl Y	Restore undo
Ctrl L	Delete line
Ctrl T	Exchange line
Ctrl Pos1	Go to first code line
Ctrl End	Go to last code line

Ctrl MouseWheel	Zoom +/-
Ctrl 0..9	Select tab (0 = Design, 1 = Code, ...)
Alt G	Open device parameters
Alt J	Select function tab (Inspector tab 1)
Alt V	Select symbol tab (Inspector tab 2)
F1	Call up help menu
F3	Search again (if a search has been carried out before)
Shift F3	Search backwards again (if a search has been carried out before)
F9	Make
Shift F9	Make clean
Ctrl F9	Make rebuild
F10	Binary export
F12	Tab change: design < > code

General

Ctrl Shift L	Open most recent project
N	Variable resource dialog new variable
D	Variable resource dialog Delete variable