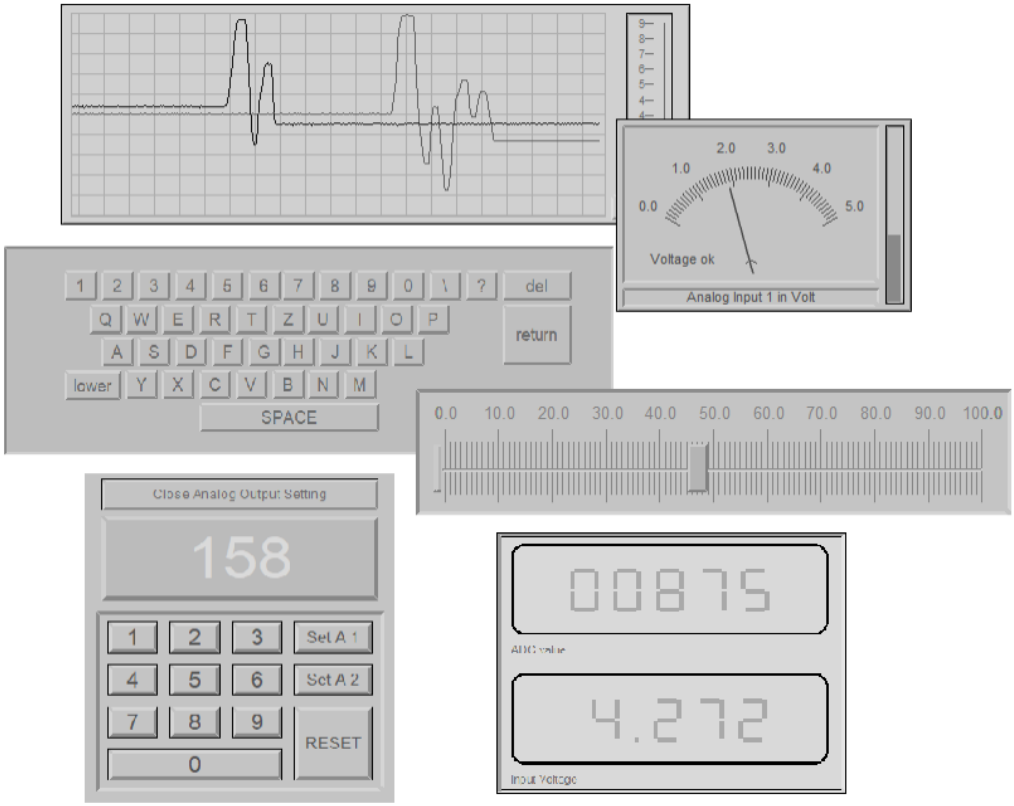
RPT-One

Benutzer Handbuch



Rapid Prototyping für graphische Oberflächen in technischen Anwendungen

wilfried schude WiSoft

RPT-One

Software zur Erstellung graphischer Oberflächen ohne Programmierung.

Handbuch für Version 2

Vorwort:

Im Bereich der Maschinensteuerung hat sich in den letzten Jahren der Einsatz von Bildschirmen mit Touch Eingabe durchgesetzt. Das Human Interface wird dabei durch entsprechende Graphikdarstellungen realisiert.

Die Erstellung solcher graphischer Oberflächen erfordert im allgemeinen Kenntnisse in Objektorientierten Programmiersprachen als auch den Umgang mit oftmals Betriebssystem spezifischen Klassenbibliotheken.

Der Umgang mit diesen ist meist mit einem hohen Aufwand an Einarbeitung verbunden.

RPT-One bietet hier die Möglichkeit ein graphischen User Interfaces (Tool basierend) zu entwerfen und die Funktionalität zu testen.

Inhaltsverzeichnis

1 Programmeinführung

RPT-One erlaubt die Erstellung graphischer Interfaces (technische Anwendungen) ohne die Einarbeitung in hochkomplexe Programmiersprachen wie 'C++' und den entsprechenden Klassenbibliotheken.

Der Vorteil von RPT-One ist, das im Gegensatz zu gängigen Tools, wie beispielsweise Microsoft Foundation Classes, Borland Builder und Qt Classenbibliotheken die auf C++ basieren um individuelle Designs zu ermöglichen – die graphische Bedienoberfläche wie mit einem CAD Tool erzeugt werden kann. Zudem wird ein Interface zum Datenaustausch mit in 'C' geschriebenen Programmen zur Verfügung gestellt, einer für Programmierer aus den Bereich der Micro Controller Anwendungen gängigen Programmiersprache. Somit entfällt ein erheblicher Aufwand an Einarbeitung in die Programmiersprache C++ und die entsprechenden Klassenbibliotheken.

RPT-One deckt die gängigen Anwendungen im Steuerungsbereich oder der Simulation von technischen Abläufen ab, da in der Regel nur eine begrenzte Anzahl von graphischen Elementen benötigt wird.

Das Programm ist als Microsoft Windows als auch als Linux Version verfügbar. Die erstellten Applikationen können mit kleineren betriebssystemspezifischen Modifikationen transferiert werden.

Ein Compilieren des Programms ist nicht erforderlich (what you see is what you get). Zusätzlich wird neben der graphischen Gestaltung auch die Steuerung und das Verhalten der Benutzeroberfläche von RPT-One unterstützt.

Um individuelle User Funktionen an das Programminterface von RPT-One anzubinden, reichen Programmierkenntnisse in C aus. Das Programminterface ermöglicht zugleich ein Zusammenspiel mehrerer RPT-One Applikationen.

2 Programmaufbau

Eine mit RPT-One erstellte Applikation besteht im wesentlichen aus den in Abbildung 2.1 dargestellten drei Hauptteilen:

Dem Graphik Interpreter

Dem User Programm

und der Externen Hardware

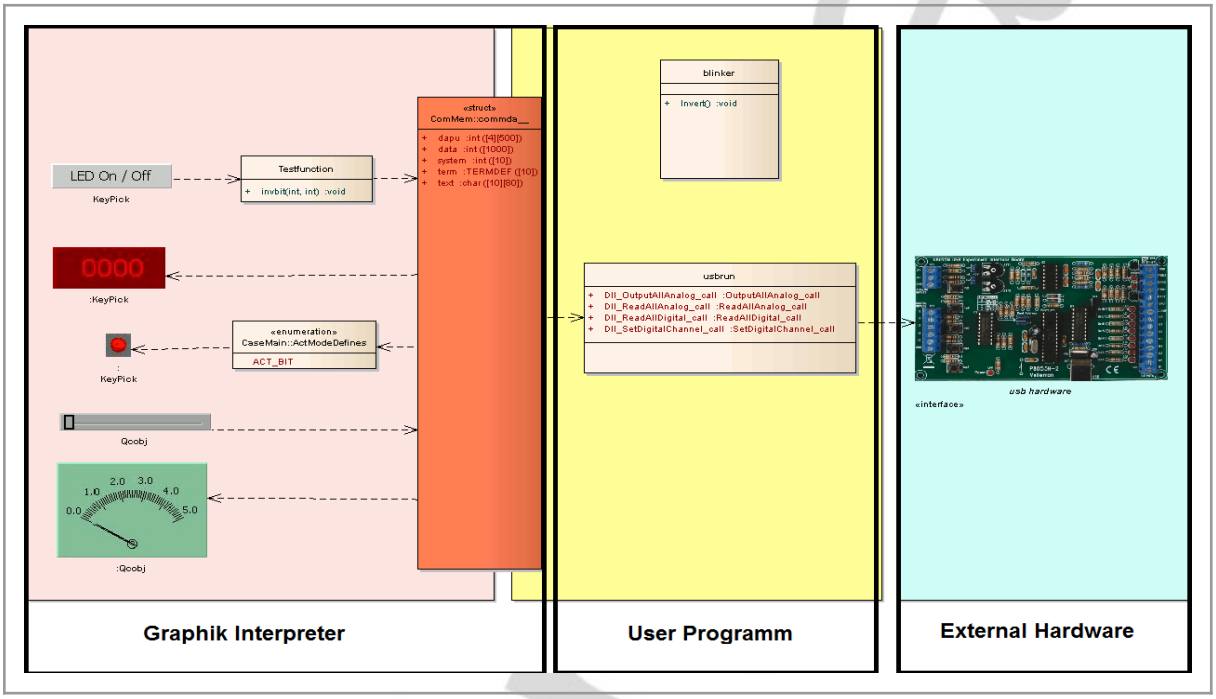


Abbildung 2.1: Programm Komponenten

Der ***Graphik Interpreter*** stellt eine graphische Benutzer Oberfläche zur Anzeige und Eingabe von Applikationsdaten dar, welche über eine Maus oder einen Touchscreen bedient werden kann. Alle Ein und Ausgaben werden in dem RPT-One Interface abgelegt und stehen somit externen Programmen zur Bearbeitung zur Verfügung. Die Elemente der Benutzeroberfläche werden mit dem hier beschriebenen Tool erstellt und durch den Interpreter ausgeführt.

Das ***User Programm*** wird, wie der Name schon sagt, durch den Anwender erstellt. Dieser Teil kann aus einzelnen Funktionen bestehen welche durch den Interpreter aufgerufen werden können, als auch aus einer Laufschleife zur Steuerung einer Applikation oder einer ***externen Hardware (siehe hierzu Anlage xy).***

Dieser Programmaufbau sorgt für eine hohe Modularität einer Applikation, und somit auch einem hohen Grad an wiederverwendbarkeit der erstelltem Software Module, da im Gegensatz zu einem herkömmlich erstelltem Programm die einzelnen Teile immer wieder zu neuen Applikationen zusammengestellt werden können.

Die erstellte Applikation kann in Textform als auch in Binärform abgespeichert werden.

Ist eine Applikation fertiggestellt und ausgetestet, können alle Teile der Applikation in einem Dateiordner zusammengeführt und durch einen Runtime Interpreter ausgeführt werden.

2.1 Elemente in RPT-One

Wie bereits erwähnt, ist die Anzahl der im Interpreter verfügbaren Elemente auf ein Minimum reduziert worden, decken aber durch die setzbaren Optionen fast alle für eine Steuerung erforderlichen Anforderungen ab.

Die in RPT-One verfügbaren graphischen Elemente sind Windows, Keypicks und Objekte. Daneben gibt es noch die 2 Kontrollelement. Kommandos und Aktionen.

Alle diese Elemente werden in indizierten Feldern abgelegt, und in der Reihenfolge ihrer Indizes dargestellt. Das heißt, ein Element mit einem höheren Index überdeckt die Elemente mit niedrigerem Index. Zu erst werden alle Windows gezeichnet, dann die Objekte und danach die Keypicks. Durch die Zuordnung eines Elementes zu einem Layer kann die Reihenfolge der Darstellung beeinflusst werden.

Ein niedriger Layer wird durch einen höheren Layer überdeckt.

2.1.1 Windows

Windows stellen Zeichenflächen dar, die hauptsächlich dem Design dienen. Windows werden mit den durch 'Mode' festgesetzten Eigenschaften dargestellt. Die detaillierten Eigenschaften werden in Kapitel 3.4.2Windows Eigenschaftenbehandelt.

Des weiteren können Windows folgende Inhalte als graphischen Hintergrund darstellen. Windows und Linux Bitmaps, Image und Muster Backgrounds als auch eine stark vereinfachte Version eines HpGl Plottfiles.

Des weiteren kann ein Window auch als aktive Terminal Ausgabe definiert werden. Über ein zugeordnetes FIFO (Fist in First Out) werden die darzustellenden Zeichen von einer externen Funktion eingespeist.

Dem ersten Window kommt hier eine besondere Funktion zu, da es die Darstellungsgröße der Applikation bestimmt.

2.1.2 Keypicks

Keypicks sind graphische Elemente welche ähnliche Eigenschaften wie Windows haben, weisen jedoch weitere Merkmale auf. Die wesentliche Eigenschaft ist, das ein Keypick auf einen Mausclick hin ein Kommando auslösen oder seine Darstellung ändern kann. Dazu hat jedes Keypick eine normale und eine alternative Darstellung welche wie bei den Windows durch einen zugewiesenen Mode bestimmt wird. Des weiteren können Keypicks auch folgende Inhalte darstellen:

Bitmap Icons, Pixmmap Icons, mage Icons, Text, Daten, indizierte Texte als auch 7 Segment Symbole

2.1.3 Objekte

Objekte stellen komplexe Elemente dar wie:

Slider

Instrumente

Diagrammfelder

Grafikfelder

2.1.4 Kommandos

Kommandos sind Funktionen welche durch folgende Aktivitäten ausgelöst werden können.

Maus key down

Maus key up

Mouse key pressed

at Applikation start

at Applikation end

in Applikation loop

Kommandos sind in folgende Kategorien eingeteilt:

System Kommandos

Externe Kommandos

Interne Kommandos

conditions

2.1.5 Aktionen

Aktionen sind Reaktionen von Keypicks auf Zustände des Programm Interfaces. Eine Aktion erzwingt die alternative Darstellung eines Keypicks.

Die Auslösung kann durch folgende Ereignisse ausgelöst werden:

Bit gesetzt

Wert kleines als Vorgabe

Wert gleich der Vorgabe

Wert größer als Vorgabe

Des weiteren können folgende Eigenschaften eines Keypicks durch externen Zugriff verändert werden.

Farbe des Keypick

X Position des Keypick

Y Position des Keypick

Breite des Keypick (absolut oder Prozent)

Höhe des Keypick (absolut oder Prozent)

Text Helligkeit

Feld Helligkeit

2.1.6 RPT-One Interfaces

Das Interface des Programms wird durch ein Speicher Segment gebildet, auf das intern von den Kommandos und den Aktionen zugegriffen wird. Über diese Schnittstelle erfolgt der Datenaustausch mehrerer Applikationen als auch der Datenaustausch mit User Programmen.

2.1.7 Installation von RPT-One

Wurde RPT-One als Zip File heruntergeladen, empfiehlt es sich nach dem öffnen des Zip Files

den Ordner Rptg in einen Ordner C:Progs zu entpacken. Abbildung 2.2 zeigt die sich daraus ergebene Ordner Struktur.

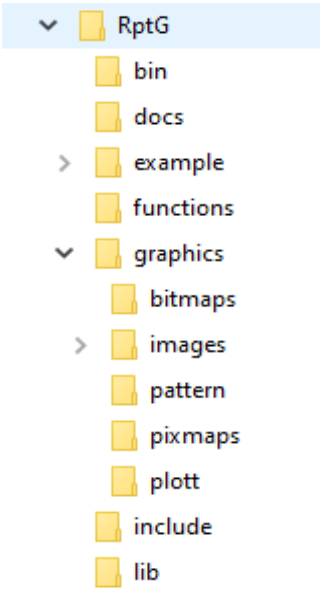


Abbildung 2.2: default folder structure

Im Ordner ‚bin' befinden sich die ausführbaren Dateien ‚rptg_edit.exe' und ‚rptg_run.exe'.

‚docs' enthält die Programmdokumentation.

‚examples' enthält einige mit dem Programm erstellte Beispiele welche im Programm ‚rptg_edit' geöffnet und angeschaut werden können. Alternativ kann man auch die Beispiele mit ‚rptg_run' direkt ausführen.

‚functions' enthält eine Sammlung externer Funktionen welche in den Beispielen benutzt werden.

‚graphics' ist der Sammelort für die verschiedenen in RPT-One benutzten Graphik Formate.

‚include' beherbergt die C Include Files welche zur Erstellung eigener externer Funktionen erforderlich sind, um den Zugriff auf das RPT-One Interface zu ermöglichen.

In ‚lib' sind die Library Link Files für den Zugriff auf das Inderface gespeichert.

Um die für eine einwandfreie Funktion der Software erforderlichen INI Dateien zu erstellen sollte zuerst das Programm ‚rptg_edit.exe' im Ordner bin gestartet werden, worauf hin die in Abbildung 2.3: First Start dargestellte Meldung angezeigt wird.

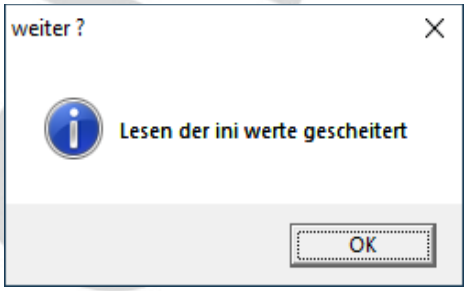


Abbildung 2.3: First Start

Nach dem Bestätigen des ‚OK' Button werden die voreingestellten ‚INI' Files installiert, und der RPT Editor gestartet. Die Einstellungen können dann wie in Kapitel 3.2.3.2 Konfiguration Menü. angepasst werden. Wenn im ‚bin' Ordner kein gültiges ‚rpt.lic' File vorhanden ist, wird die Meldung aus Abbildung 2.4: License Message angezeigt.

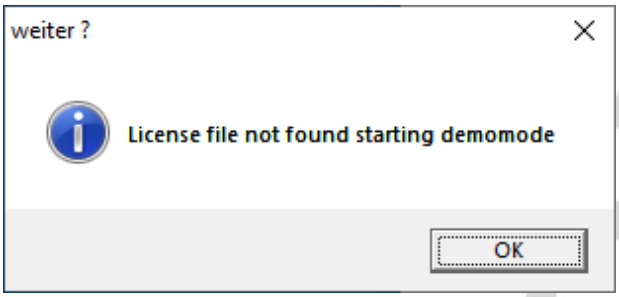


Abbildung 2.4: License Message

Durch das Bestätigen des ‚OK' Button wird das Programm im Demo Mode gestartet. Das bedeutet, ein speichern der Applikation ist nur im binary format möglich und die Testzeit ist begrenzt und wird mit der Meldung aus Abbildung 2.6: Demo Message beendet. Außerdem wird die Darstellung im Applikationsfenster durch den Schriftzug ‚Demo Version' gekennzeichnet.

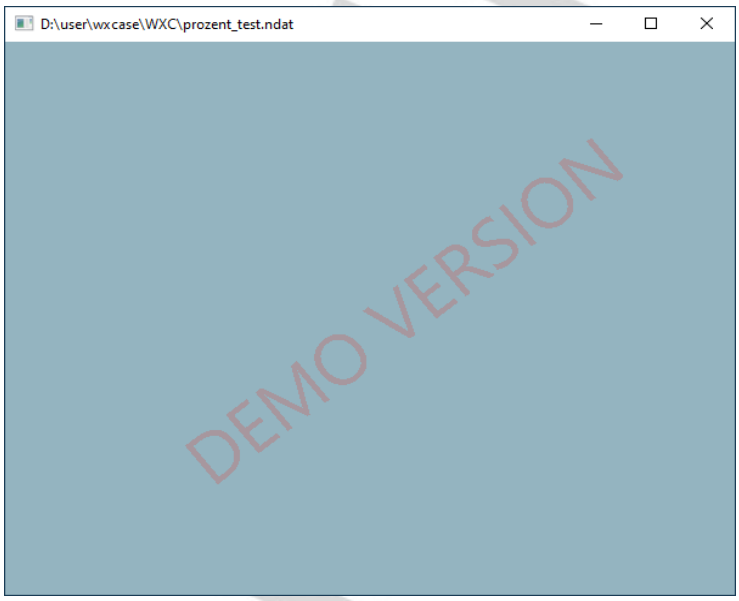


Abbildung 2.5: Darstellung Demo Mode

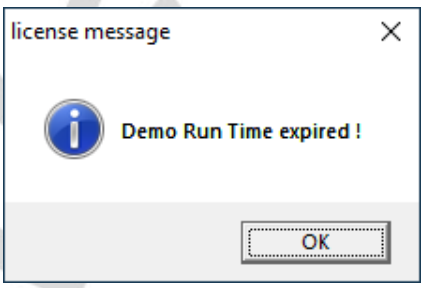


Abbildung 2.6: Demo Message

3 Programmbedienung

In diesem Kapitel wird die Bedienung der einzelnen Menüs im Kommando Fenster zur Erstellung der RPT-One Elemente beschrieben.

3.1 Programmstart

Der Start des Programms erfolgt unter Windows durch den Aufruf 'RPT-One.exe'. Nach diesem Aufruf werden auf dem Bildschirm zwei Fenster dargestellt (Abbildung 3.1).

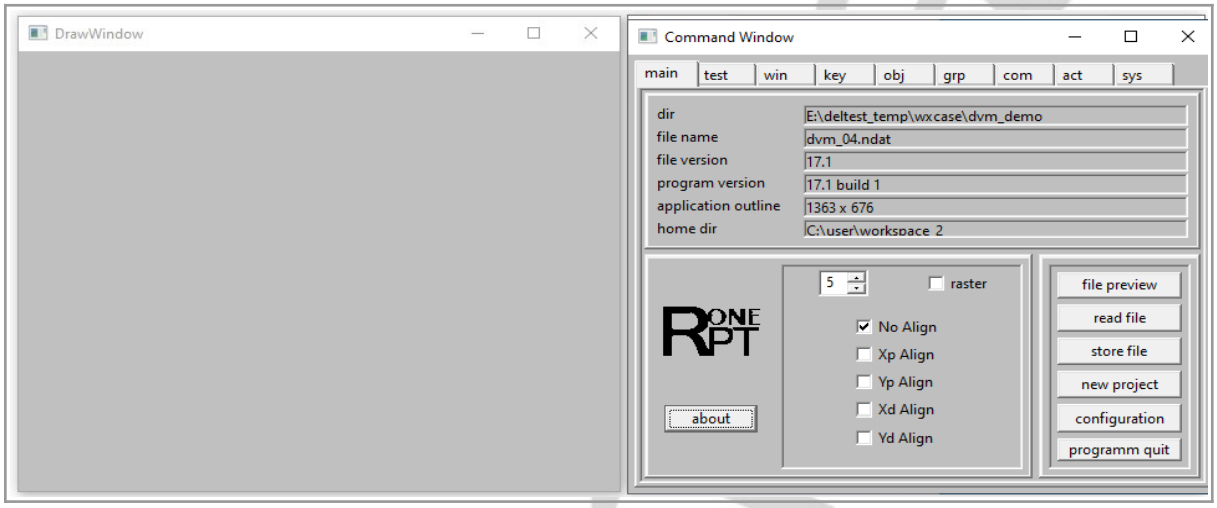


Abbildung 3.1: Startbildschirm

Das Applikation Fenster (DrawWindow) dient als Zeichenfläche in der die Applikation erstellt wird.

Das Kommando Fenster (Command Window) dient der Eingabe aller Kommandos um eine Applikation zu Erstellen.

Im folgenden werden alle Funktionen beschrieben die vom Kommandofenster zur Verfügung gestellt werden.

Über die im Programmkopf aufgelisteten Reiter können die zu bearbeitenden Funktionen aufgerufen werden (Abbildung 3.2).

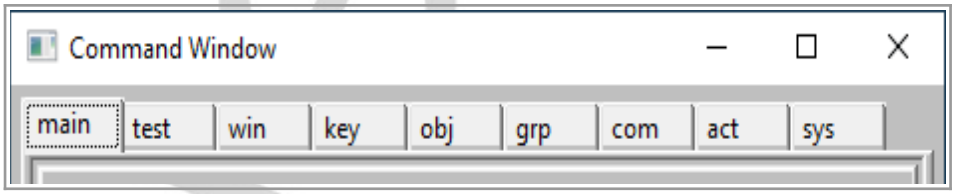
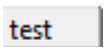
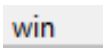
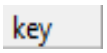
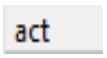
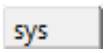


Abbildung 3.2: Coms

Funktion der Reiter

	Main Menü	Funktionen zur Programmausführung
	Programmtest	Aufruf der Test Funktion
	Window Menü	Erstellung von Windows und deren Eigenschaften
	Keypick Menü	Erstellung von Keypicks und deren Eigenschaften
	Objekt Menü	Erstellung von Objekten und deren Eigenschaften
	Gruppen Menü	Bearbeitung von Elemente Gruppen
	Kommando Menü	Erstellung von Kommandos und deren Eigenschaften
	Aktions Menü	Erstellung von Aktionen und deren Eigenschaften
	System Menü	Aufruf von System Funktionen

Nach dem Programmstart ist automatisch der Reiter 'main' aktiv. Durch einen Mausclick auf den entsprechenden Reiter wird auf die dazugehörige Funktionskarte umgeschaltet. Die Beschreibung der einzelnen Funktionskarten und deren Menüeinträge erfolgt in den folgenden Kapiteln.

3.2 Main Menü

Die Registerkarte Main ist in drei Bereiche aufgeteilt (Abbildung 3.3).

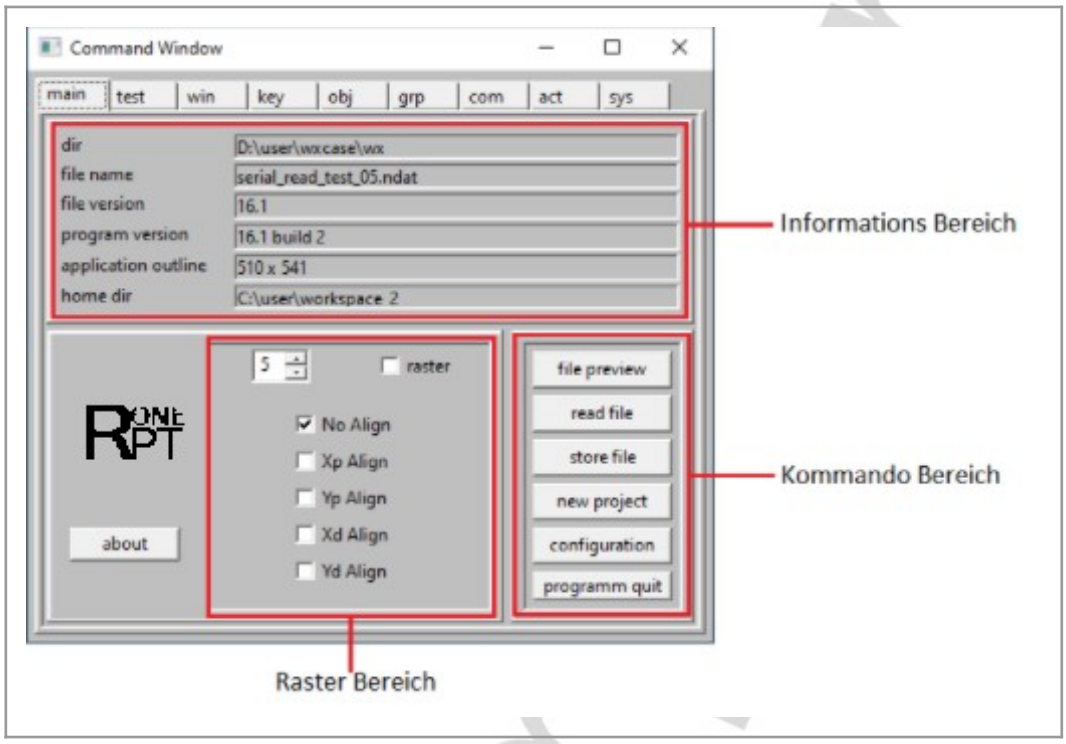


Abbildung 3.3: Aufbau der Register Karte 'Main'

Der Button about öffnet das in Abbildung 3.4 dargestellte Copyright Fenster.

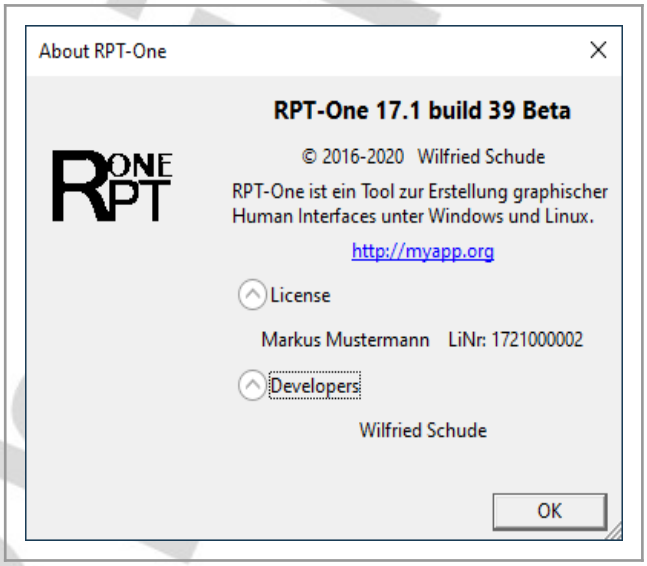


Abbildung 3.4: Copyright Fenster

3.2.1 Informations Bereich

Dieser Bereich zeigt die aktuellen Programm Einstellungen an.

dir	Directory Pfad der aktuell geöffneten Datei
file name	Name der aktuell geöffneten Datei
file version	Version der aktuell geöffneten Datei
program version	Version von RPT-One
application outline	Größe des Applikationsfensters
home dir	Directory Pfad von RPT-One

3.2.2 Raster Bereich

Der Raster Bereich erlaubt das Ein – Ausschalten der Rasteranzeige durch die Checkbox 'raster'. In der Scrol Box werden die Punkt Abstände der Rasteranzeige eingestellt.

Abbildung 3.5: Raster Bereich

Das eingeblendete Raster dient als Orientierungshilfe bei der Applikationserstellung, als auch als Fangpunkte bei einem ebenfalls aktiviertem Alignment.

Das Beispiel Abbildung 3.6 zeigt die Darstellung mit eingeschaltetem Raster und einem Abstand von 10 Bildpunkten.

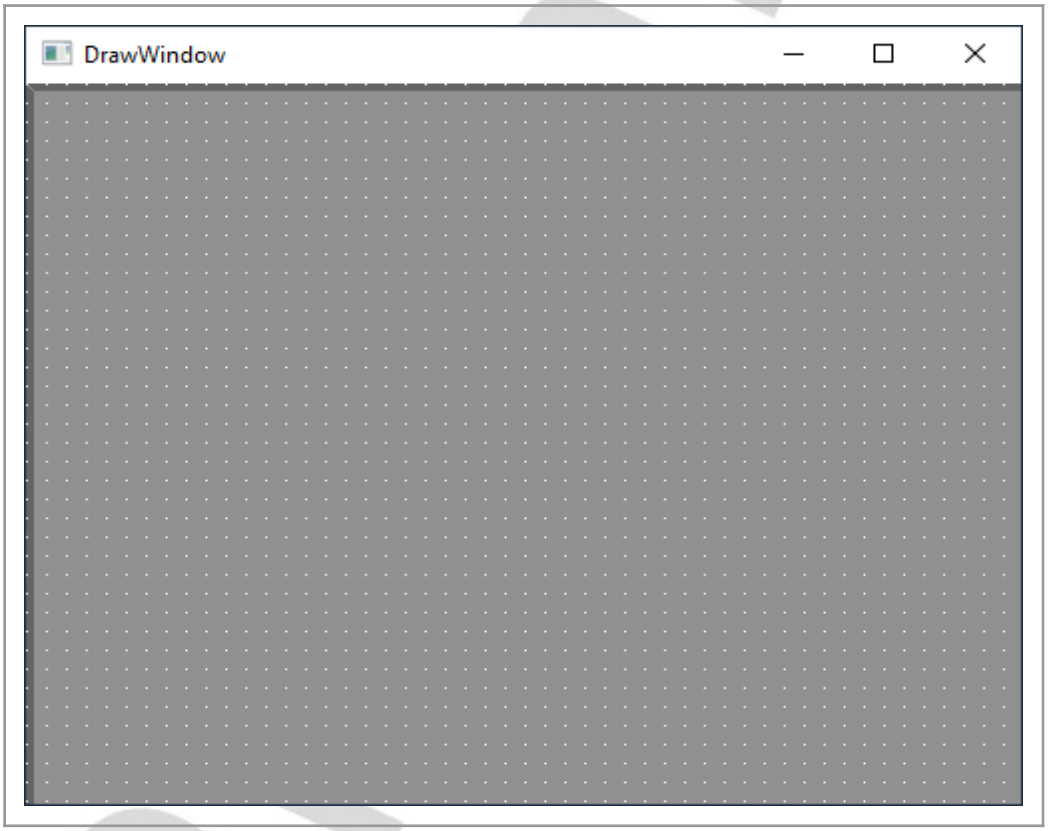


Abbildung 3.6: Application window mit eingeschaltetem Raster

3.2.2.1 Fangraster

In Abbildung 3.7 ist die Fangraster Einstellung 'No Align' dargestellt. Es werden Rasterpunkte mit dem Abstand 20 eingeblendet. Das Graphik Element ist in diesem Fall unabhängig vom Raster positioniert.

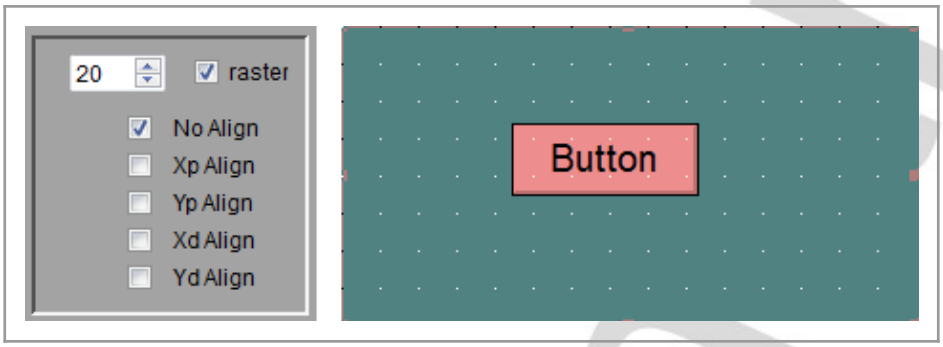


Abbildung 3.7: No Align

In Abbildung 3.8 ist die Fangraster Einstellung für 'X-Position' und 'Y-Positon' Alignment dargestellt. Das Graphik Element wird beim Verschieben automatisch auf den nächst gelegenen Rasterpunkt positioniert.

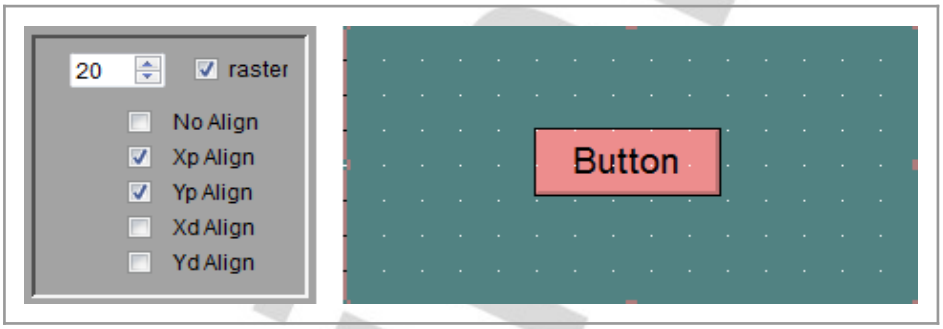


Abbildung 3.8: Xp Align / Yp Align

In Abbildung 3.9 ist zusätzlich die Fangraster Einstellung 'X-Delta' und 'Y-Delta' gesetzt. Die Breite (X-Delta) und die Höhe (Y-Delta) des Graphik Element werden beim Vergrößern auf das Raster positioniert.

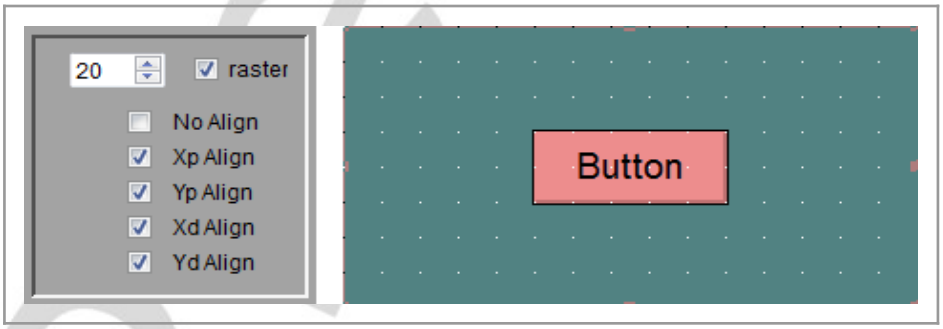


Abbildung 3.9: Xd Align / Yd Align

3.2.3 Kommando Bereich

Der Kommando Bereich (Abbildung 3.10) enthält mehrere Buttons zum Aufruf von Funktionen oder dem Öffnen von weiteren Submenüs.

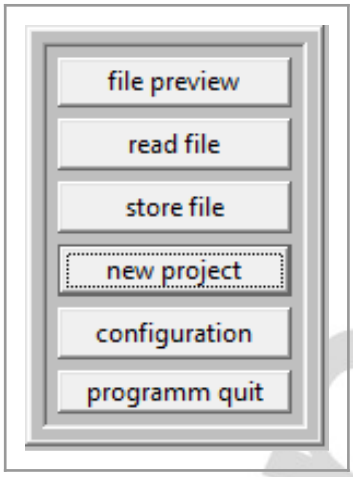
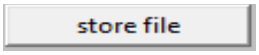


Abbildung 3.10: Kommando Bereich

3.2.3.1 Dateien öffnen und speichern

Die drei folgenden Button 

, 

und 

öffnen ein Datei Vorschau Dialog Fenster (Abbildung 3.12) bzw. Datei Dialog Fenster (Abbildung 3.13) zum Lesen und Schreiben der Applikationsdaten. Voreingestellt ist die Endung .ndat die dem Standartformat von RPT-One entspricht.

Das ‚ndat' Format stellt eine Editierbare Textdatei dar, welche die Eigenschaften der Application beschreibt. Das zweites Format bildet eine binäre Datei mit der Endung ‚qdat', welche nicht editiert werden kann. Des weiteren kann diese Datei durch ein Passwort geschützt werden um eine unbefugtes Auslesen, b.z.w ein verändern zu verhindern.

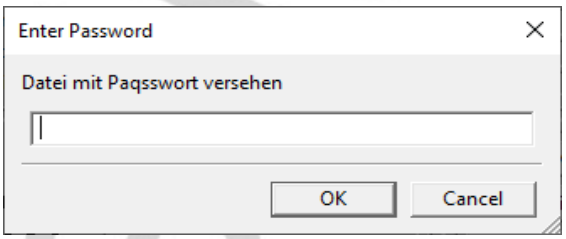
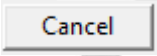


Abbildung 3.11: Passwort Eingabe

Durch den Button 

wird die Datei im Binärformat ohne Passwortschutz gespeichert.

Die Runtime Version ignoriert das Passwort und und führt die Applikation normal aus.

Der Aufruf im Vorschaumodus ist Betriebssystem abhängig. In einer Microsoft Umgebung wird bei der Selektion einer Datei im Browser, diese im Applikationsfenster angezeigt und kann dann mit Betätigen des Load Button geladen werden. In einer Linux Umgebung wird im allgemeinen eine Datei schon mit dem Selektieren derselben geladen.

Der Delete Button erlaubt das Löschen der selektierten Datei.

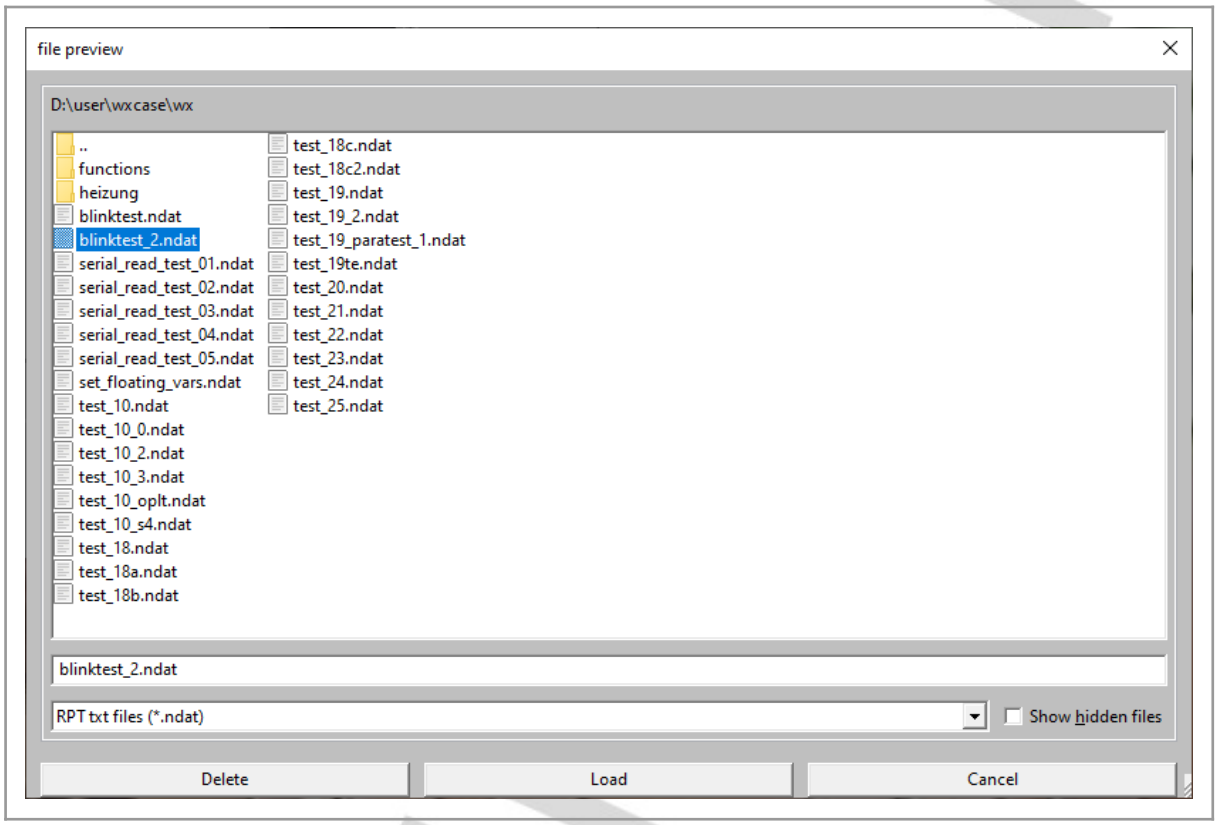


Abbildung 3.12: Datei Vorschau Dialog Fenster

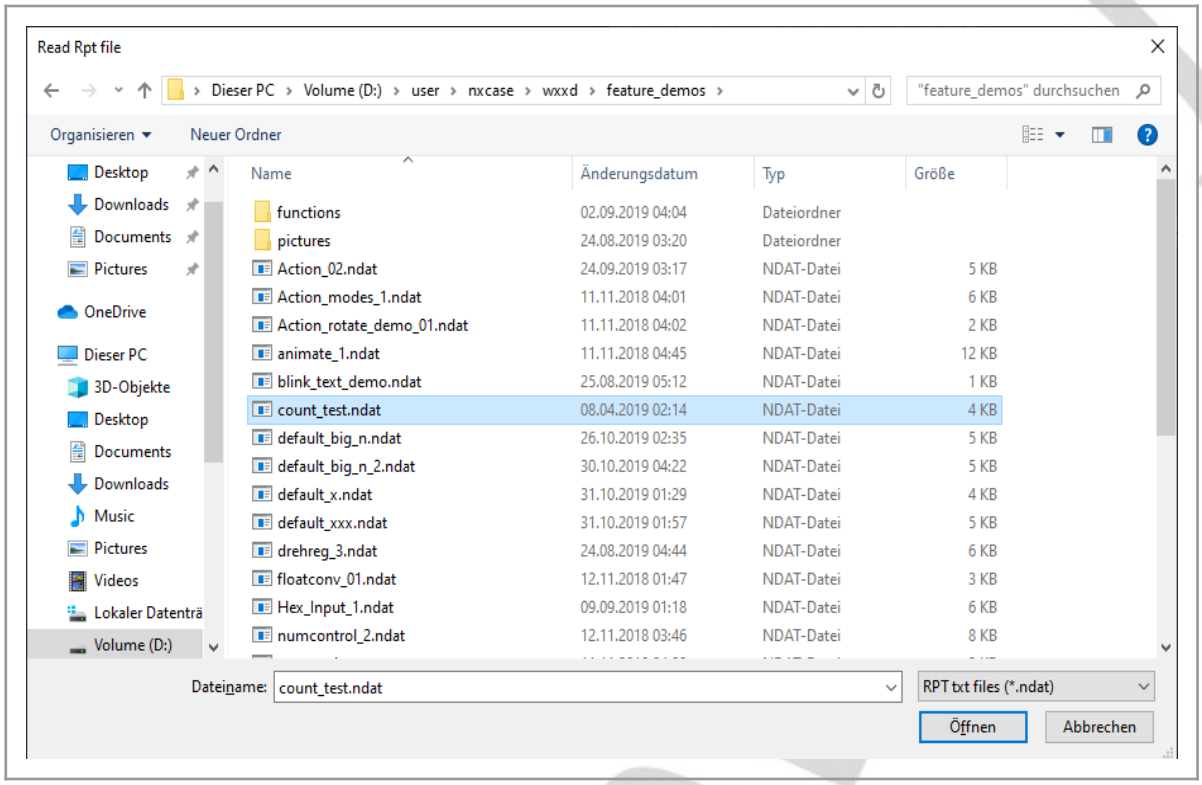


Abbildung 3.13: Datei Dialog Fenster

Um das unbeabsichtigte Überschreiben einer vorhandenen Datei zu verhindern, wird in einem Popup Fenster (Abbildung 3.14: Popup Warnung) nochmal eine Bestätigung des Speicherns angefordert. Zur Sicherheit wird die alte Datei mit der Endung .nbak b.z.w. .qbak versehen. Durch das Ändern der Datei Endung in .ndat b.z.w. qdat ist die alte Version wieder herstellbar.

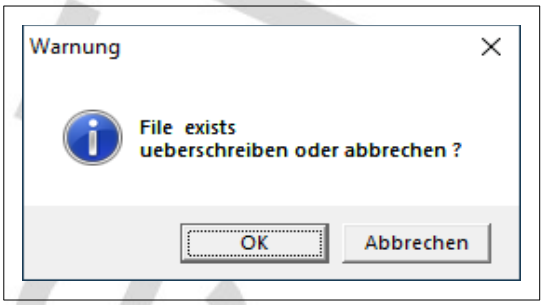


Abbildung 3.14: Popup Warnung

Das Format der zu lesenden b.z.w. zu speichernden Datei wird im File Browser wie in Abbildung 3.15: Datei Format Auswahl gezeigt ausgewählt.

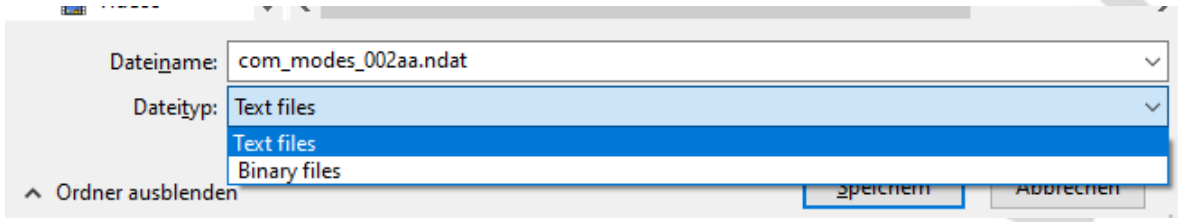


Abbildung 3.15: Datei Format Auswahl

Das Gleiche gilt auch für das Pre View Fenster.

new project Löscht alle Daten und versetzt RPT-One wieder in den Initial Zustand.

programm quit Beendet das Programm.

configuration öffnet ein Untermenü (Abbildung 3.16) in dem die Fenstergröße und die Fonts der Applikation verändert werden können. Des weiteren kann hier der Font des 'Comandwindows' eingestellt werden.

3.2.3.2 Konfiguration Menü.

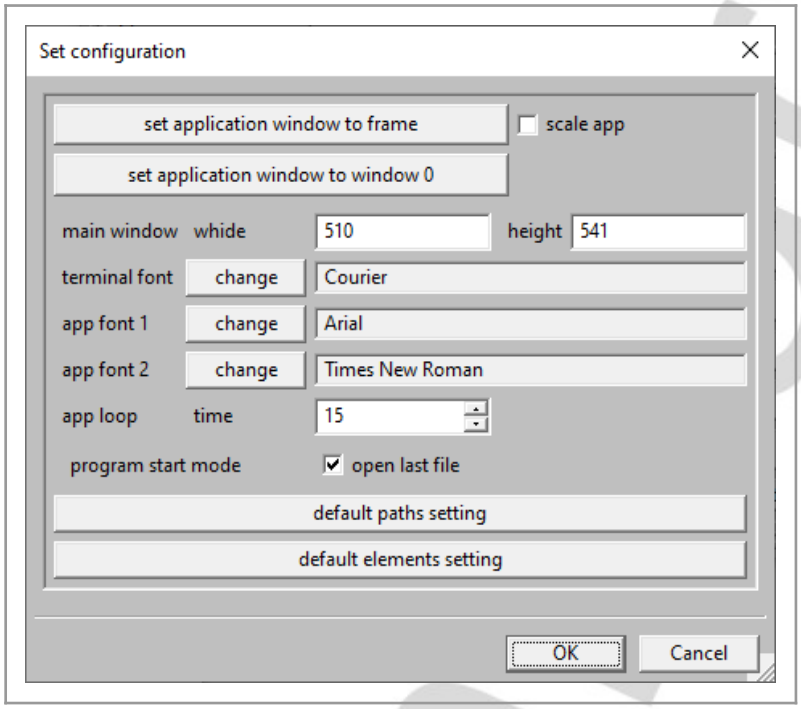


Abbildung 3.16: Konfiguration Untermenü

set application window to frame Setzt das Applikationsfenster auf die Größe des DrawWindows.

Wurde vor dem Wechsel in das Konfiguration Menü die Größe des DrawWindow Frames geändert, so kann hier die Applikation auf die neue Fenstergröße gesetzt werden.

set application window to window 0 Setzt das Applikationsfenster auf die gleiche Größe wie das erste Window (index 1).

Wurde vor dem Wechsel in das Konfiguration Menü die Größe des Initial Windows geändert, so kann hier die Applikation auf die neue Größe des Initial Windows gesetzt werden.

main window whide 510 height 541 zeigt die Größe des Applikationsfensters an. Die Werte können auch durch die direkte Eingabe in den Anzeigefenstern geändert werden. Aktiv wird die Änderung allerdings immer erst beim Schließen des Konfigurationsmenüs durch das Betätigen des Button ok.

Für alle Fälle gilt: Ist die Checkbox scale app gesetzt, werden alle Elemente der Application der neuen Größe angepasst.

cancel schließt das Konfigurationsmenue ohne die Änderungen zu übernehmen.

ACHTUNG ! Diese Änderung kann nicht durch ein Undo zurückgenommen werden.

Es empfiehlt sich daher vor einer Änderung der Größe eine Sicherungskopie anzulegen.

Durch Betätigen einer der in Abbildung 3.17 abgebildeten change Button wird das 'Font Select Menü' geöffnet um den entsprechenden Text Font zu setzen. (Abbildung 3.18)

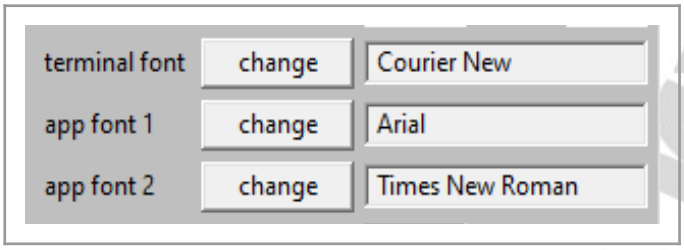


Abbildung 3.17: Font Selectionn

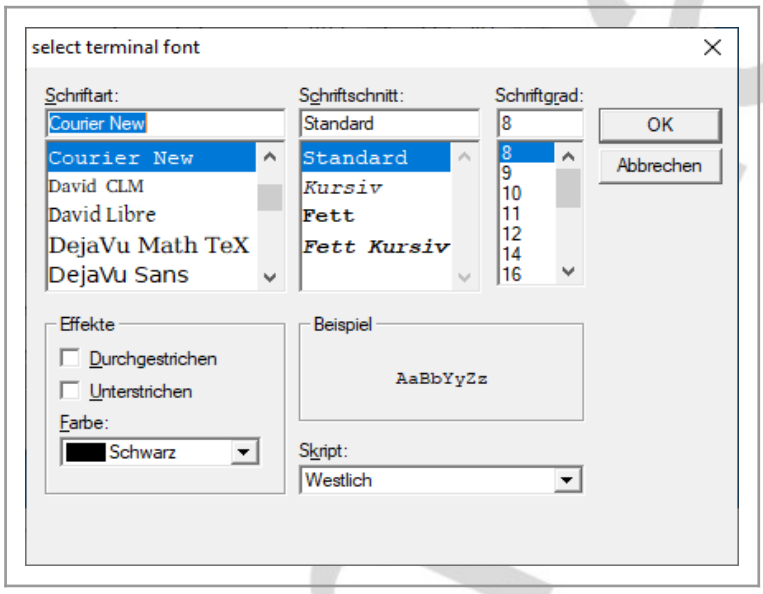
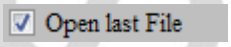


Abbildung 3.18: Fenster zur Font Auswahl

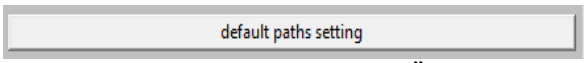
App font 1 entspricht dem default Font für Keypicks und Objekte. App font 2 entspricht einem alternativen Font für Keypicks.

Der Terminal Font wird für die Textanzeige in einem Window benutzt, das als Terminal definiert wurde. Im allgemeinen wählt man hier einen fixed widthe Font aus.

Hier kann man die Zeit in Millisekunden eingeben welche für einen Programmdurchlauf benutzt wird. Diese ist jedoch abhängig vom Betriebssystem. Unter MS Windows ist der kleinste Wert auf rund 16 Millisekunden begrenzt. Unter Linux sind hier auch kleinere Werte möglich.

Ist als „program start mode“ 

gesetzt, startet RptOne mit dem zuletzt geöffnetem File, andernfalls wird RptOne mit einem leeren File gestartet.

Mit 

wird das Fenster aus Abbildung 3.19 geöffnet um die Vorgaben für die Suchpfade zu Ändern.

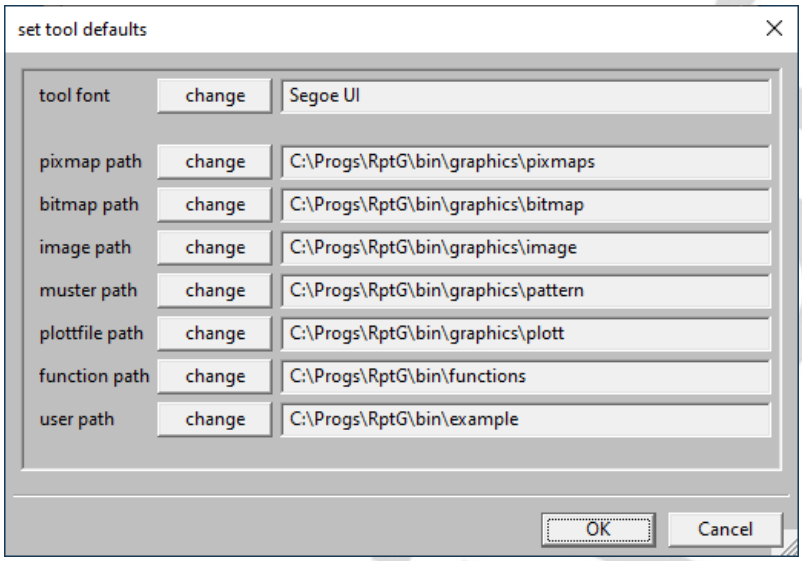
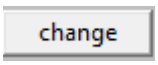


Abbildung 3.19: default path setting

Durch das anklicken des Buttons 

wird das ‚path selektor' menue aufgerufen um den Pfad auf die entsprechende Funktion auszuwählen.

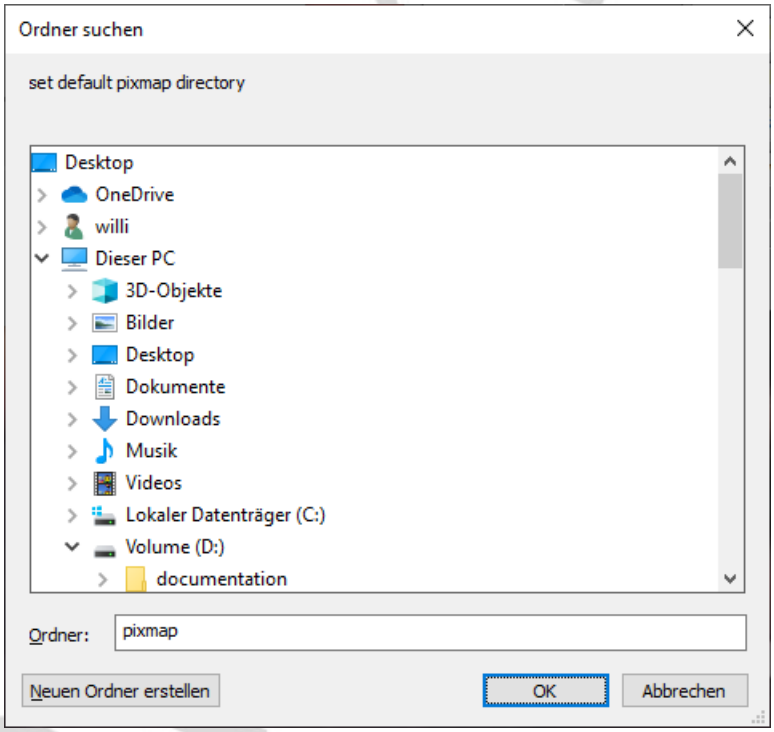
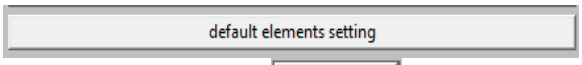
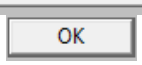


Abbildung 3.20: path selektor

Default Elemente

Mit 

wird das in Abbildung 3.19 default path setting Menü geöffnet. Mit dem 

Button werden die aktuellen Element welche in dem Menü markiert wurden als default Standard abgespeichert. Wird dann ein neues Element erstellt, erscheint dieses mit den abgespeicherten Eigenschaften.

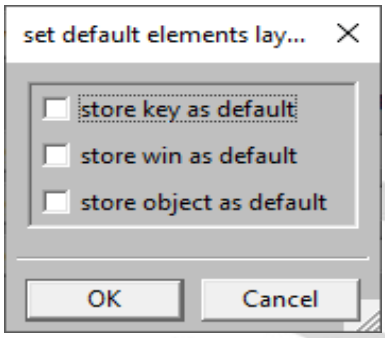


Abbildung 3.21: default element setting

3.3 Programmtest

Der Reiter test schließt das Kommando Fenster und startet die Applikation im Testmode. Der Test Mode kann durch die in Abbildung 3.22 gezeigte Auswahl im Testmode Pop Up (rechte Maustaste / ‘programm quit‘) beendet werden. Das Kommando Fenster wird wieder geöffnet und es kann mit der Bearbeitung fortgefahren werden. Alternativ kann der Testmode auch durch das Betätigen der ‚Esc‘ Taste beendet werden.

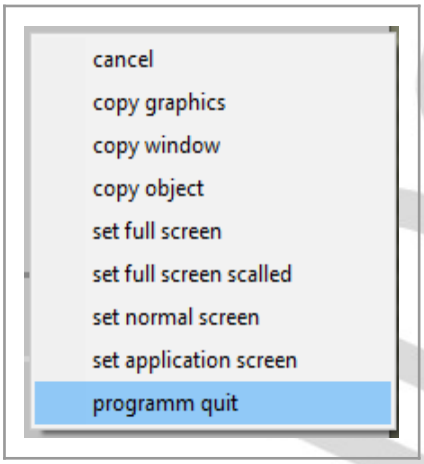


Abbildung 3.22: Testmode Pop Up

cancel	Pop up wird geschlossen
copy graphics	Der Abbild des Applikations Window wird in die Zwischenablage kopiert
Copy window	Das Abbild des Windows unter dem cursor wird in die Zwischenablage kopiert
copy object	Der Inhalt des objekts unter dem cursor wird in die Zwischenablage kopiert
Set full sreen	Schaltet dem ‚Full Sreen Mode‘ ein.
Set full sgreen scalled	Schaltet dem ‚Full Sreen Mode‘ ein. Und scalliert all Inhalte auf das neue Format.
Set normal screen	Schaltet dem ‚Full Sreen Mode‘ aus.
Set application screen	Die letzte Aktion wird wieder hergestellt
Programm quit	Beendet den Test mode Alternativ kann der Testmode auch durch das Betätigen der ‚ESC‘ -Taste Beendet werden.

3.4 Window Menü

Erstellung von Windows und Bearbeitung derer Eigenschaften.

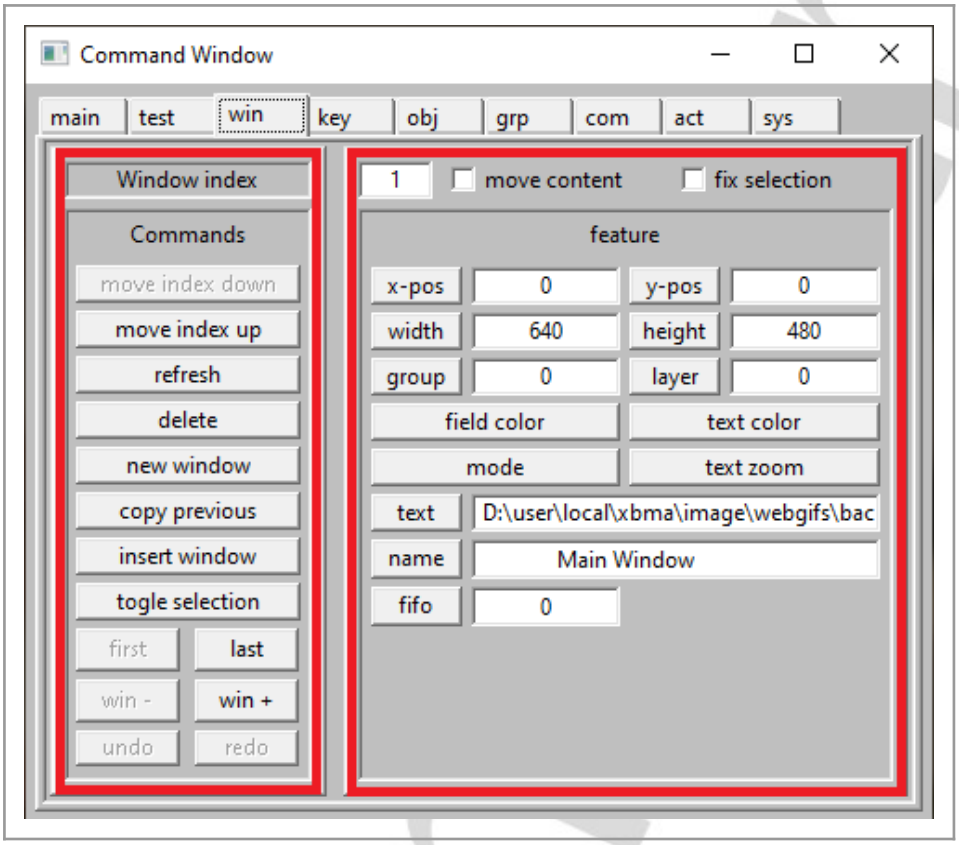


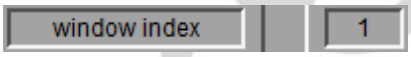
Abbildung 3.23: Window Menü

Wie in Abbildung 3.23 gezeigt, besteht das Window Menü aus zwei Hauptbereichen. Dem 'Commands' Bereich und dem 'Eigenschaften' Bereich.

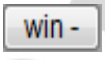
Der 'Commands' Bereich dient der Verwaltung der Windows, während der 'Eigenschaften' Bereich wie der Name schon sagt zur Einstellung der Eigenschaften der Windows vorgesehen ist.

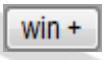
3.4.1 Windows Verwaltung

Wie schon vorab beschrieben sind alle Elemente in Feldern abgelegt. Auf die Windows kann über deren Feldindex zugegriffen werden. Der Index startet mit 1 und reicht bis zum letzten erstelltem Window. Dem Window mit dem Index 1 kommt hier eine besondere Bedeutung zu. Es bestimmt zugleich die Größe des Applikation Fensters.

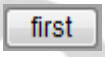
Das Feld 

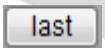
zeigt die Nummer des zur Bearbeitung ausgewählten Windows an.

Mit den Buttons 

und 

kann man sich durch das Feld aufwärts und abwärts bewegen.

Mit den Buttons 

und 

springt man direkt zum Anfang b.z.w zum Ende der Liste.

Alternativ kann ein zu bearbeitendes Window auch im Applikationsfenster durch Positionierung des Cursors in das Window und einem linken Mausclick selektiert werden. Dazu muss jedoch die

Funktion fix selection ausgeschaltet sein. Ist fix selection gesetzt wird eine unbeabsichtigte Selektion eines anderen Windows verhindert.

move index down Verschiebt das aktuelle Window um eine Position in der Liste nach unten.

move index up Verschiebt das aktuelle Window um eine Position in der Liste nach oben.

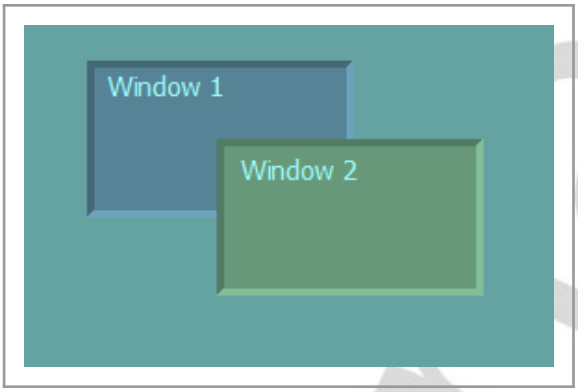


Abbildung 3.24: Window Zeichnungs Reihenfolge

Da die Windows in der Reihenfolge von 1 bis last dargestellt werden, und somit ein Window mit höherem Index alle mit niedrigerem Index überschreibt kann man durch die Indexposition bestimmen welches Window im Vordergrund erscheint. Abbildung 3.24

refresh Erzwingt ein neu zeichnen der Applikation.

delete Löscht das gerade aktuelle Window aus der Liste.

undo Erlaubt das Rückgängig machen der letzten Änderungen im Window Menü.

new window Erstellt ein neues Window ohne Eigenschaften am Ende der Liste.

copy previous Kopiert die Eigenschaften des vorherigen Windows in das aktuelle Window.

Ausnahme: Beim ersten Window wird durch 'copy previous' das Window auf die Größe des Applikationsfensters gesetzt.

insert window Fügt ein neues Window mit den Eigenschaften des im Konfigurationsmenü abgespeicherten default Window hinter dem aktuellem Window ein.

togle selection Schaltet die Markierung des aktuellen Windows im 'Draw Window' aus / ein um zu Kontrollieren ob das Erscheinungsbild des Windows den Vorgaben entspricht.

Alternativ zu der oben aufgeführten Art der Bedienung über die Buttons, können einige Funktionen auch über Popup Menü aufgerufen werden. Befindet sich der Mauscursor im Applikationsfenster und die rechte Maustaste wird betätigt erscheint das Window Pop Up Menü (Abbildung 3.25).

Mit 'copy' wird das aktuelle Window mit allen seinen Eigenschaften in das Clipboard copiert.

Mit 'paste' wird der Inhalt des Clipboards an der neuen Position eingefügt. Die Daten werden im Window Feld hinter dem Index des aktuellen Windows eingefügt.

Mit 'copy graphics' wird der Inhalt des aktuellen Windows als Graphik im Clipboard abgelegt, und kann als Bild in einem Dokument abgerufen werden.

Die Funktionen 'delete' und 'undo' entsprechen den Button delete und undo.

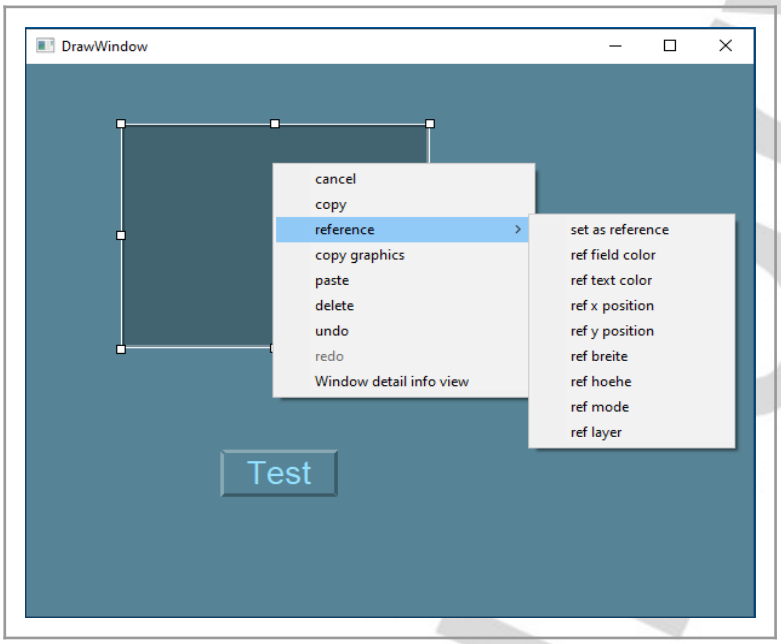


Abbildung 3.25: Window Pop Up Menü

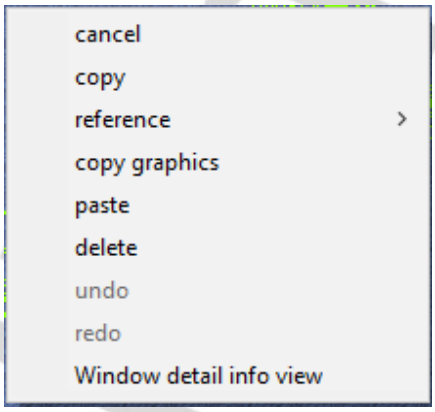


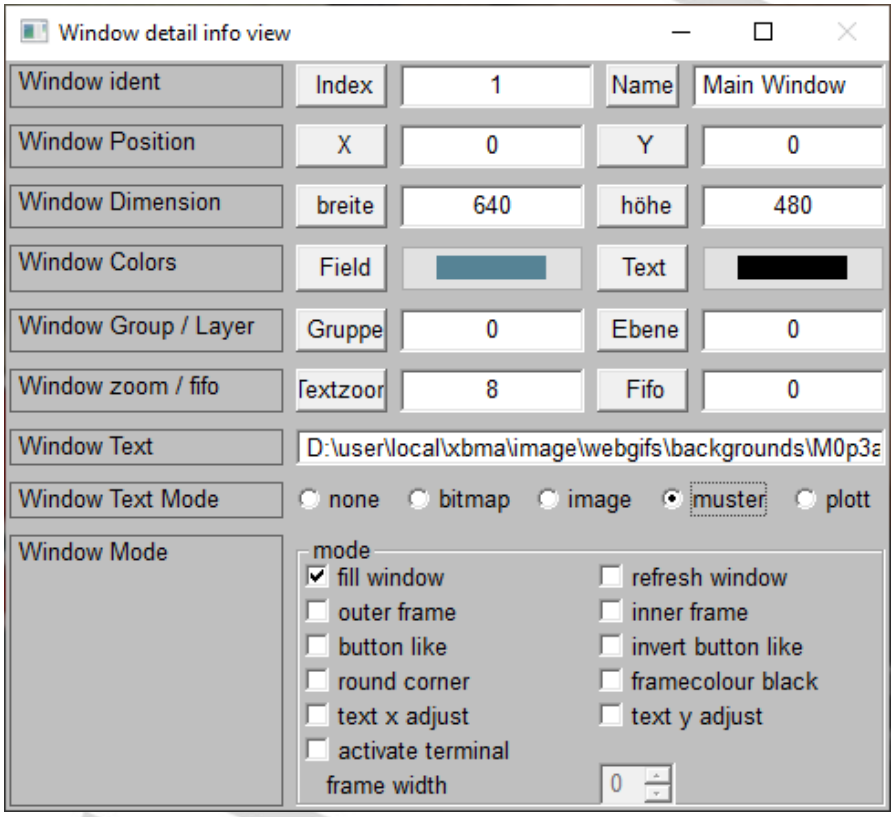
Abbildung 3.26: Window Pop Up Menü Einträge

cancel	Pop up wird geschlossen
copy	Das aktuelle Window wird in die Zwischenablage kopiert
reference	Das Refence Pop Up Menü wird aufgerufen.

Abbildung 3.27: Refence Pop Up Menü

set as reference	Die Einstellungen des aktuellen Windows werden in den Referenzspeicher übernommen
ref field color	Ordnet dem aktuellen Window die abgespeicherte ‚field color , des Referenzspeichers zu.
ref text color	Ordnet dem aktuellen Window die abgespeicherte ‚text color , des Referenzspeichers zu.
ref x position	Ordnet dem aktuellen Window die abgespeicherte ‚x position, des Referenzspeichers zu.
ref y position	Ordnet dem aktuellen Window die abgespeicherte ‚y position, des Referenzspeichers zu.
ref breite	Ordnet dem aktuellen Window die abgespeicherte ‚breite, des Referenzspeichers zu.
ref hoehe	Ordnet dem aktuellen Window die abgespeicherte ‚hoehe, des Referenzspeichers zu.
ref mode	Ordnet dem aktuellen Window den abgespeicherte ‚mode, des Referenzspeichers zu.
ref layer	Ordnet dem aktuellen Window den abgespeicherte ‚layer, des Referenzspeichers zu.

copy graphics	Das Abbild des aktuellen Window wird in die Zwischenablage kopiert
paste	Der Inhalt der Zwischenablage wird als neues Window eingefügt
delete	Das aktuelle Window wird gelöscht
undo	Die letzte Aktion wird rückgängig gemacht

redo	Die letzte Aktion wird wieder hergestellt
window detail info view	Es wird ein Fenster mit allen Window Eigenschaften eingeblendet. Ist bereits ein detail info Fenster geöffnet, so wird das Fenster geschlossen.  *Abbildung 3.28: Window details*

3.4.2 Windows Eigenschaften

Wie Kapitel Fehler: Verweis nicht gefunden Fehler: Verweis nicht gefunden bereits beschrieben, stellen Windows eine rechteckige Zeichenfläche dar. Die Darstellung dieses Bereichs wird durch die im Eigenschafts Fenster festgesetzten Einstellungen festgelegt.

Die Eigenschaften eines Windows sind:

Startposition X und Y im Applikationsfenster

Breite (X-Delta) und Höhe (Y-Delta) des Windows

Die Art der Darstellung (Mode) des Windows

Die Farbe des Windows (Field Color)

Die Farbe eines Textes im Window (Text Color)

Die Größe der Buchstaben des Textes (Text Zoom)

Die Bedeutung des Textes

Der Name eines Windows (nur zur Dokumentation erforderlich)

3.4.2.1 Windows Position und Größe im Applikationsfenster

Der Nullpunkt des Koordinatensystems im Applikationsfenster befindet sich in der linken oberen Ecke. Die Startposition X und Y eines Windows ergibt sich aus dem Abstand zu diesem Punkt. Die Größe des Windows ergibt sich aus den Delta Werten dx und dy zu dem Startpunkt wie in Abbildung 3.29 Window definition dargestellt.

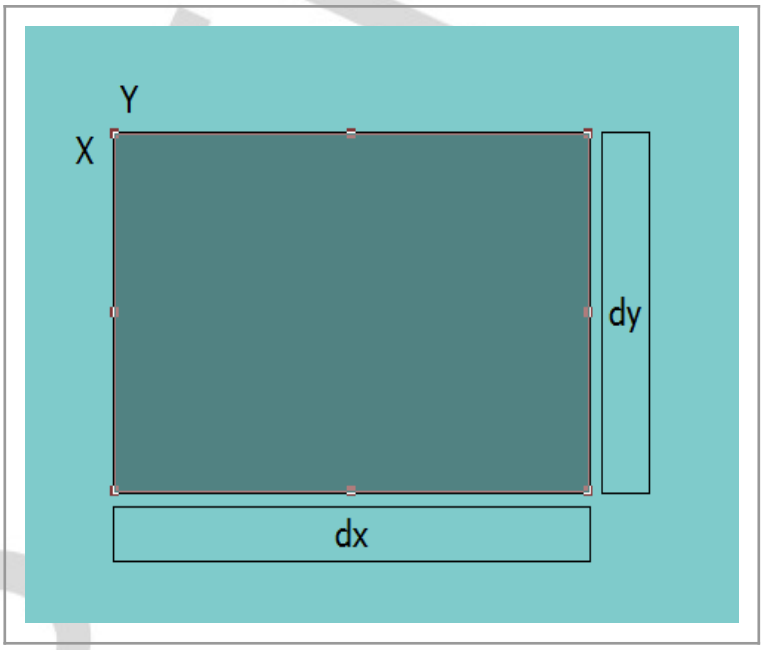


Abbildung 3.29: Window definition

Die aktuellen Werte für Position und Größe werden im Eigenschafts Bereich angezeigt (Abbildung 3.30).

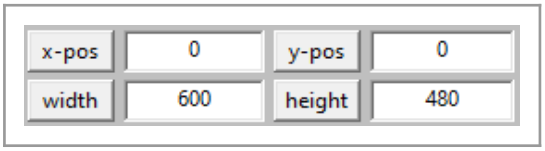
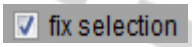


Abbildung 3.30: Winndow Dimension

Die folgenden Beschreibungen der Veränderung der Position und oder Größe eines Windows beziehen sich auf eine Einstellung des Rasters laut Abbildung 3.31. Bei anderen Rastereinstellungen gelten die in Kapitel 3.2.2.1 Fangraster beschriebenen Verhaltensweisen.

Abbildung 3.31: Raster No Allign

3.4.2.1.1 Windows Position und Größe im Applikationsfenster ändern

Die Änderung der Position und Größe eines Windows kann auf zwei Arten erfolgen. Dieses Kapitel beschreibt die Änderung des aktuellen Windows im Applikationsfenster. Um zu Verhindern das beim Betätigen der linken Maustaste unbeabsichtigt ein anders Window selektiert wird empfiehlt es sich 'fix selection' einzuschalten. 

Position im Applikationsfenster ändern: (Abbildung 3.32)

1. Positionieren des Cursors innerhalb des aktuellen Windows.
2. Linke Maustaste gedrückt halten.
3. Window an die gewünschte Position schieben.
4. Maustaste loslassen.

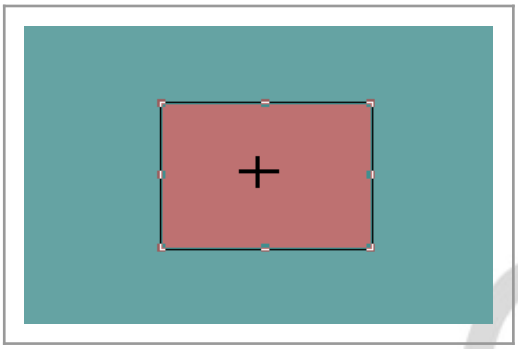
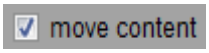


Abbildung 3.32: Window Positionieren

Ist die Funktion 

gesetzt, werden alle Elemente die sich in diesem Window befinden ebenfalls verschoben.

Größe im Applikationsfenster ändern: (Abbildung 3.33)

1. Positionieren des Cursors innerhalb des aktuellen Windows an der Markierung die verschoben werden soll.
2. Linke Maustaste gedrückt halten.
3. Markierung an die gewünschte Position schieben.
4. Maustaste loslassen.

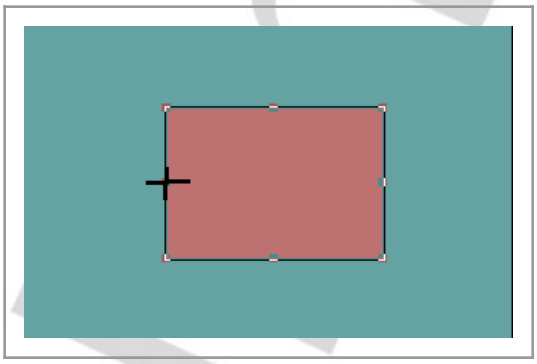


Abbildung 3.33: Window Größe ändern

3.4.2.1.2 Windows Position und Größe im 'Window Menü' ändern.

Ändern der horizontalen Position durch Betätigen von Button 

öffnet Abbildung 3.34: Window X-Start Menü.

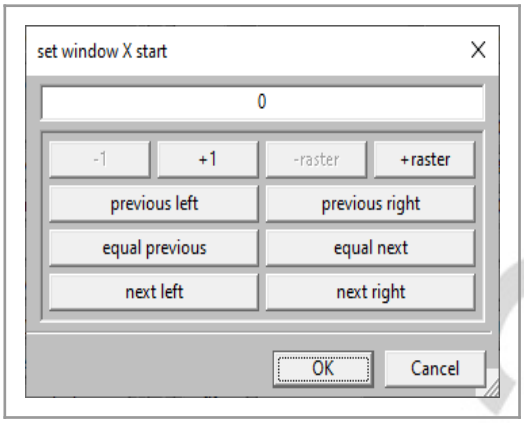


Abbildung 3.34: Window X-Start Menü

Durch das Betätigen des entsprechenden Button ändert sich die horizontale Position des Windows wie im Button Text angegeben.

Ändern der vertikalen Position durch Betätigen von Button y öffnet Abbildung 3.35: Window Y-Start Menü

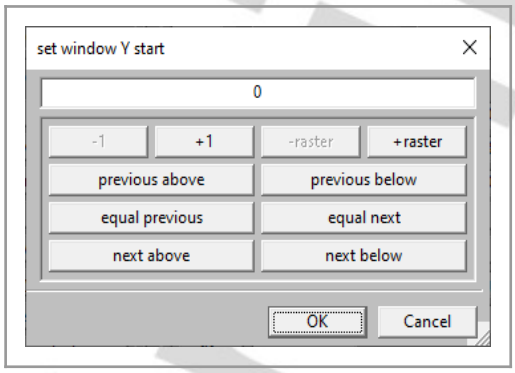


Abbildung 3.35: Window Y-Start Menü

Durch das Betätigen des entsprechenden Button ändert sich die vertikale Position des Windows wie im Button Text angegeben.

Ist die Funktion move content gesetzt, werden alle Elemente die sich in diesem Window befinden ebenfalls verschoben.

Ändern der Breite durch Betätigen von Button dx öffnet Abbildung 3.36: Window Breite Menü

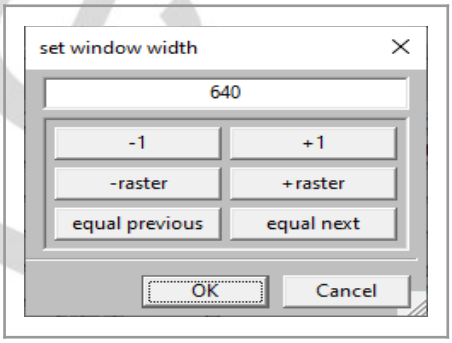


Abbildung 3.36: Window Breite Menü

Durch das Betätigen des entsprechenden Button ändert sich die Breite des Windows wie im Button

Text angegeben.

Ändern der Höhe durch Betätigen von Button dy öffnet Abbildung 3.37: Window Höhe Menü

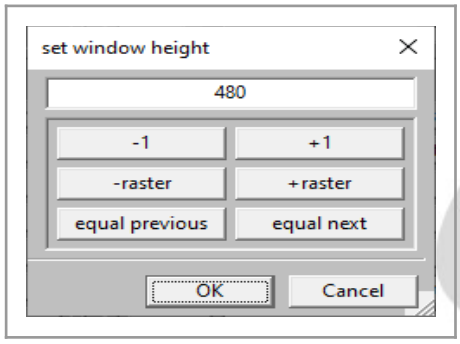


Abbildung 3.37: Window Höhe Menü

Durch das Betätigen des entsprechenden Button ändert sich die Höhe des Windows wie im Button Text angegeben.

3.4.2.2 Das Erscheinungsbild eines Windows ändern.

Die Art der Darstellung eines Windows wird durch den 'Mode' und die gesetzten Farben bestimmt.

mode Öffnet den Window Mode Dialog. Die Abbildung 3.38 zeigt das Menü ohne gesetzten Mode.

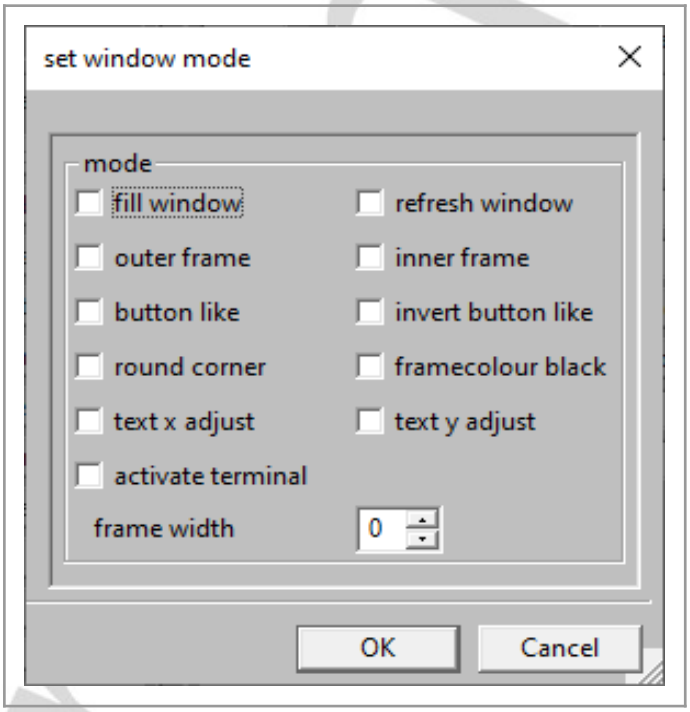


Abbildung 3.38: Window Mode Dialog

Die gesetzten Check Boxen definieren das Erscheinungsbild des Windows.

Wurde zuvor mit field color eine Farbe ausgewählt, lässt sich das Ergebnis direkt im Applikationsfenster verfolgen. In der Tabelle 1 sind Beispiele für verschiedene Mode Kombinationen aufgeführt.

☑ fill window Füllt die Fläche des Windows mit der unter field color gesetzten Farbe.

☑ outer frame erstellt einen Rahmen in der Textfarbe um die Windows Fläche.

☑ inner frame erstellt einen Rahmen in der Textfarbe innerhalb der Windows Fläche. Der Abstand zum Außenrand wird durch frame width 4 bestimmt.

☑ button like lässt das Windows wie einen Button erscheinen. Die Breite des Randes wird durch frame width 4 bestimmt.

☑ invert button like lässt das Windows wie einen gedrückten Button erscheinen. Die Breite des Randes wird durch frame width 4 bestimmt.

☑ framecolour black setzt die Farbe des Randes des Windows auf Schwarz / Weiß.

☑ refresh window hat keinen Einfluss auf das Erscheinungsbild des Windows, sondern erzwingt im Test ein zyklisches neu zeichnen. Diese Funktion wird nur in Sonderfällen benötigt auf welche in späteren Kapiteln näher eingegangen wird.

☑ activate terminal Mit diesem Mode wird das aktuelle Window als ein Terminal Fenster benutzt, in welches über die externe Schnittstelle Texte ausgegeben werden können Abbildung 3.39. Die Farbe und Größe des Ausgabetextes kann mit den Button text color und text zoom gesetzt werden. Der im Terminal verwendete Text Font kann im Konfiguration Menü. (Kapitel 3.2.3.2) ausgewählt werden. Es wird jedoch für alle Terminal Fenster in einer Applikation immer der gleiche Font Typ verwendet. Im Kommando Mode wird im Terminal links unten der Windowname eingeblendet um die Einstellung der Textdarstellung zu erleichtern (Abbildung 3.39).

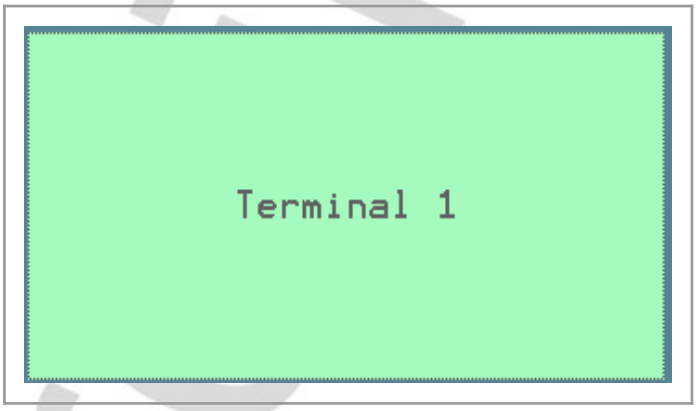


Abbildung 3.39: Terminal Window

Um von extern Daten in ein Terminal Window zu senden, sind im RPT-One Interface 10 Fifo Kanäle implementiert. Der einem Terminal zugeordnete Kanal wird im Feld fifo 0 angezeigt, und kann über den Button geändert werden (Abbildung 3.40: Fifo Dialog). Es darf jeder Fifo Kanal nur einmal einem Terminal Window zugeordnet werden.

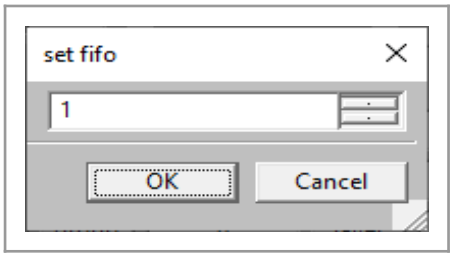
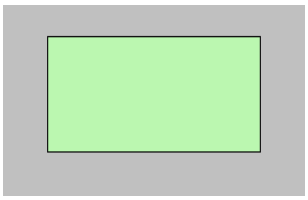
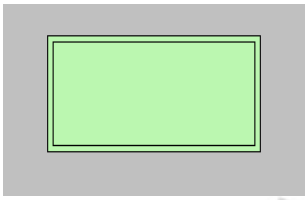
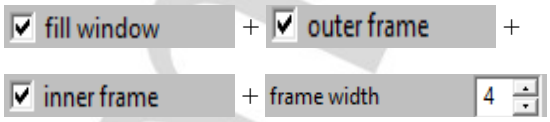
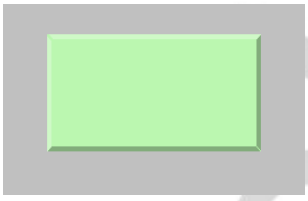


Abbildung 3.40: Fifo Dialog

vergrößert die Randbreite des Windows Rahmen. Der Wert 0 entspricht im Normalfall einem Pixel. Die Wirkung der Rahmen Breite ist abhängig vom anderen Einstellungen des Mode.

In der Tabelle 1 sind einige Beispiele für die Einstellung des Mode dargestellt.

	 Das Window wird mit einem Rahmen in der Textfarbe umgeben. Die Breite des Rahmen ist 0.
	 Das Window wird mit einem Rahmen in der Textfarbe außen und innen mit Abstand 4 Pixel gezeichnet.
	 Das Window wird als Button gezeichnet mit einem 3 Pixel breitem Rand (frame width = 3).
	 Das Window wird als gedrückter Button gezeichnet mit einem 3 Pixel breitem Rand (frame width = 3).

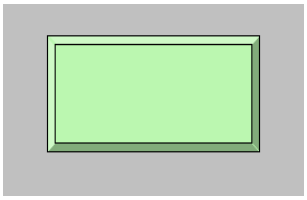
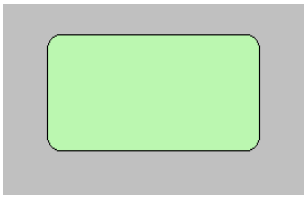
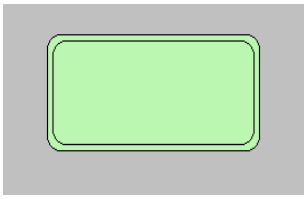
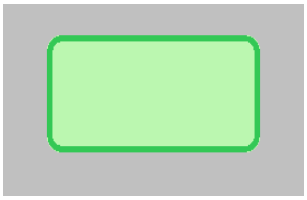
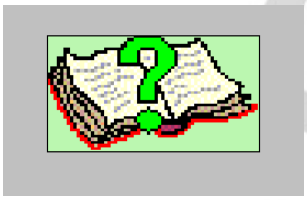
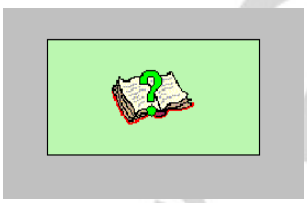
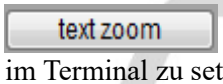
	☑ fill window + ☑ button like + ☑ outer frame + ☑ inner frame Das Window wird als Button gezeichnet mit einem 6 Pixel breitem Rand (frame width = 6). und einer äußeren und inneren Umrandung.
	☑ fill window + ☑ round corner + ☑ outer frame + ☑ framecolour black Hier wird das Fenster mit abgerundeten Ecken dargestellt.
	☑ fill window + ☑ round corner + ☑ outer frame + ☑ inner frame
	☑ fill window + ☑ round corner + ☑ outer frame Hier wird das Fenster mit abgerundeten Ecken dargestellt.
	☑ fill window + ☑ outer frame + ☐ text x adjust + ☐ text y adjust + ☑ framecolour black
	☑ fill window + ☑ outer frame + ☑ text x adjust + ☑ text y adjust + ☑ framecolour black .

Tabelle 1: Beispiele Window Mode



Öffnet das Text Zoom Dialog (Abbildung 3.41) um die Größe der Textdarstellung im Terminal zu setzen.

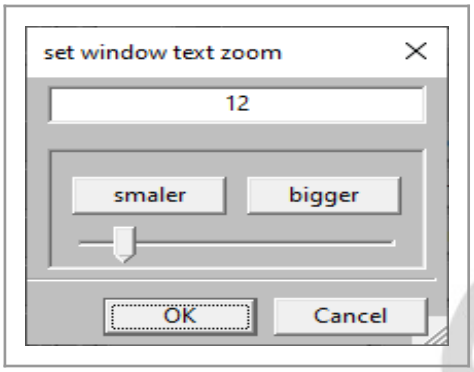


Abbildung 3.41: Text Zoom Dialog

Die Größe der Textanzeige kann über die Buttons als auch über den Slider auf das gewünschte Maß gesetzt werden.

text color und field color öffnen das Color Menü (Abbildung 3.42) in welchem durch Anclicken die gewünschte Farbe ausgewählt wird. Nach dem Öffnen des Color Menü wird die aktuelle Farbe durch eine weißen Rahmen markiert. Nach dem Auswählen einer neue Farbe wird diese ebenfalls umrandet, und beide Farben werden im Kopf des Color Menü angezeigt.

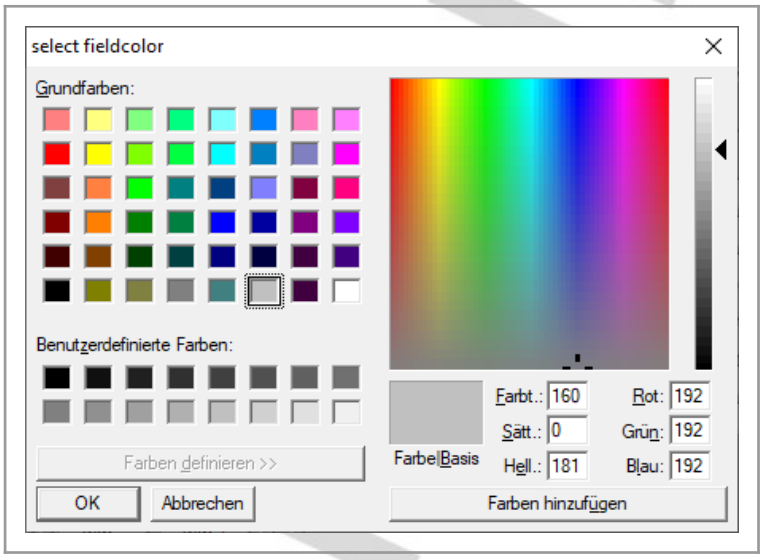


Abbildung 3.42: Color Menü

Mit dem Button text icons/images/webgifs/backgro kann dem Window ein Text zugeordnet werden. Dieser Text ist im allgemeinen der Name einer Datei welche als Graphiken im Window erscheint. Beim Betätigen des Buttons wird der Window Text Dialog geöffnet siehe Abbildung 3.43: Window Text 1. Ist No Background gesetzt, sind alle Bild Optionen deaktiviert, und das Window verhält sich wie vorab beschrieben.

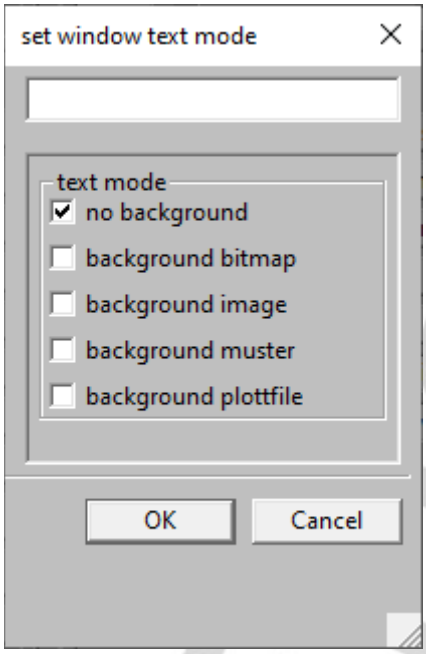


Abbildung 3.43: Window Text 1

Wählt man eine Background Graphik aus, wird der Window Text Dialog wie in Abbildung 3.44 dargestellt, und der Inhalt des Windows wie in Tabelle 2 beschrieben dargestellt.

Die im Window darstellbaren Graphiken müssen in einer Datei abgelegt sein, und können mit dem Button 

über ein File Dialog Fenster ausgewählt werden. Die anderen Eigenschaften des Windows wie in Beispiele Window Mode Tabelle 1 beschrieben, bleiben dabei erhalten. ***Eine Ausnahme bildet hier das 'Terminal Window', in dem keine Bilddarstellung zugelassen wird.***

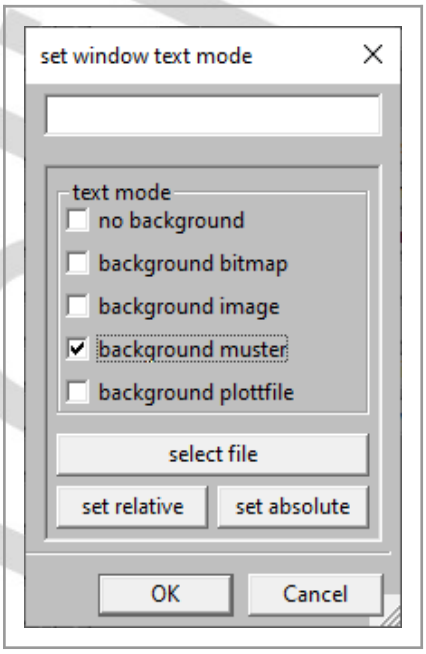


Abbildung 3.44: Window Text Dialog

Die Auswahl einer Graphik erfolgt in dem System eigene File Browser Abbildung 3.45. Da Windows im File Manager keine Vorschau für .xbm, .xpm und .plt Dateien erzeugt, ist das Ergebnis einer Bildauswahl erst nach dem Schließen des Auswahlfensters zu sehen.

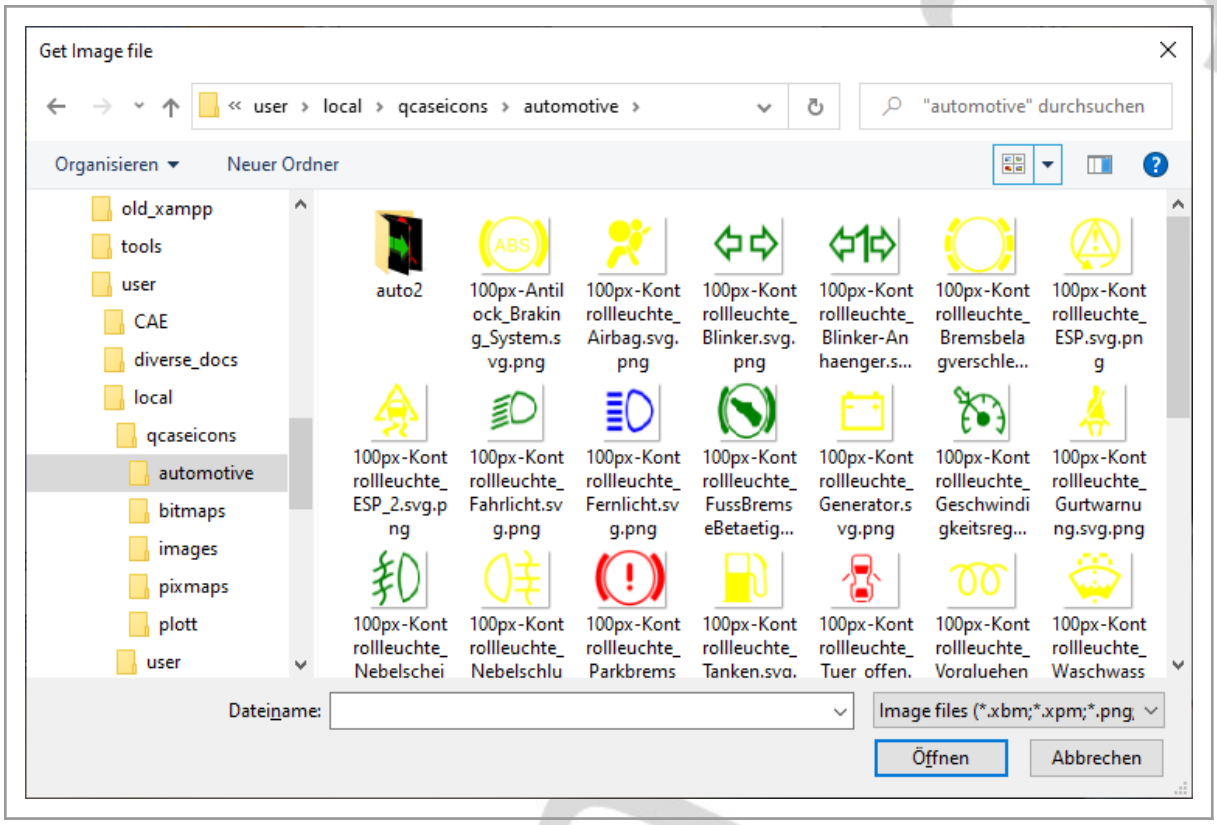


Abbildung 3.45: File Browser

. Für Windows gelten die bereits oben genannten Einschränkungen, während in Linux Systemen alle Formate dargestellt werden.

Beispiele der Graphik Darstellung abhängig vom ausgewählten Bildformat.

Background bitmap

Der Window Hintergrund wird in der 'field color' und die Bitmap in der 'text color' gezeichnet.

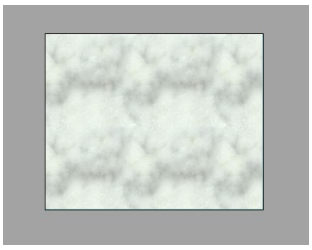
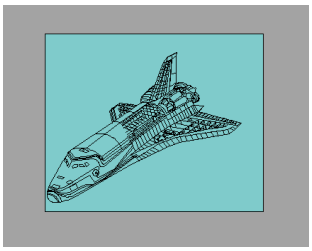
	☑ Background image Das gesamte Window wird als Bild dargestellt.
	☑ Background muster Das Window wird mit dem gekacheltem Image gefüllt.
	☑ Background plottfile Auf dem mit der 'field color' gefülltem Hintergrund wird das 'Hpgl Plottfile mit der 'text color' gezeichnet.

Tabelle 2: Background Graphik

Ist als Window Background ‚bitmap' oder ‚image' gesetzt worden, kann die Breite als auch die Höhe des Bildes durch den ‚text zoom' geändert werden, wenn im ‚mode menue' die Funktion für ‚text x adjust' oder ‚text y adjust' gesetzt wurden.

zeigt die Gruppenzugehörigkeit des Windows an und kann durch das Betätigen des Button geändert werden (Abbildung 3.46). Die Gruppe hat keinen Einfluss auf die Eigenschaften des Windows sondern dient lediglich der internen Kennzeichnung einer Zusammengehörigkeit verschiedener Elemente welche im 'Group Menü' gemeinsam bearbeitet werden können.

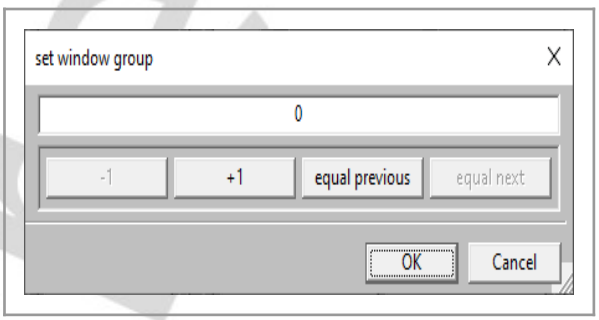
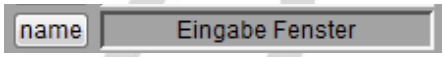


Abbildung 3.46: Group setting



Ebenso wie die Gruppe hat auch der Window Name keinen

Einfluss auf die Eigenschaften. Er dient lediglich zur Dokumentation.

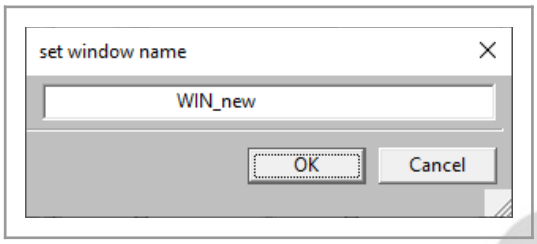


Abbildung 3.47: Window Namens Dialog

3.5 Keypick Menü

Erstellung von Keypicks und Bearbeitung deren Eigenschaften.

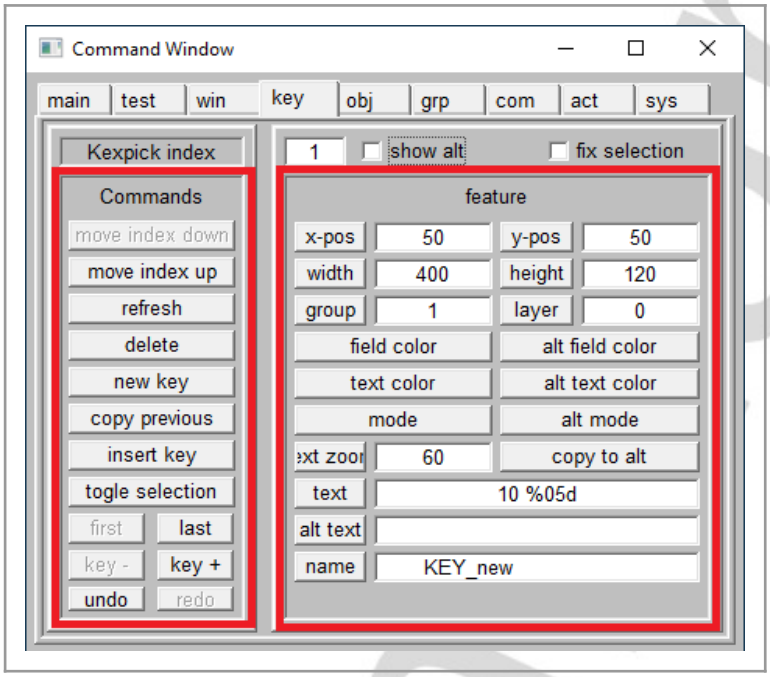


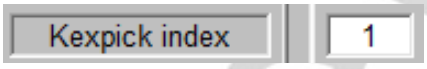
Abbildung 3.48: Keypick Menü

Wie in Abbildung 3.48 gezeigt, besteht das Keypick Menü aus zwei Hauptbereichen. Dem 'Commands' Bereich und dem 'Eigenschaften' Bereich.

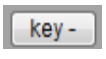
Der 'Commands' Bereich dient der Verwaltung der Keypicks, während der 'Eigenschaften' Bereich wie der Name schon sagt zur Einstellung der Eigenschaften der Keypicks vorgesehen ist.

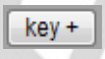
3.5.1 Keypick Verwaltung

Wie schon vorab beschrieben sind alle Elemente in Feldern abgelegt. Auf die Keypicks kann über deren Feldindex zugegriffen werden. Der Index startet mit 1 und reicht bis zum letzten erstelltem Keypicks.

Das Feld 

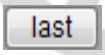
zeigt die Nummer des zur Bearbeitung ausgewählten Keypicks an.

Mit den Buttons 

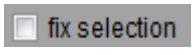
und 

kann man sich durch das Feld aufwärts und abwärts bewegen.

Mit den Buttons 

und 

springt man direkt zum Anfang b.z.w zum Ende der Liste.

Alternativ kann ein zu bearbeitendes Keypick auch im Applikations Fenster durch Positionierung des Cursors in das Keypick und einem linken Mausclick selektiert werden. Dazu muss jedoch die Funktion 'fix selection' ausgeschaltet sein. 

Verschiebt das aktuelle Keypicks um eine Position in der Liste nach unten.

Verschiebt das aktuelle Keypicks um eine Position in der Liste nach oben.

Da die Keypicks in der Reihenfolge von 1 bis last dargestellt werden, und somit ein Keypick mit höherem Index alle mit niedrigerem Index überschreibt kann man durch die Indexposition bestimmen welches Keypick im Vordergrund erscheint. Abbildung 3.49

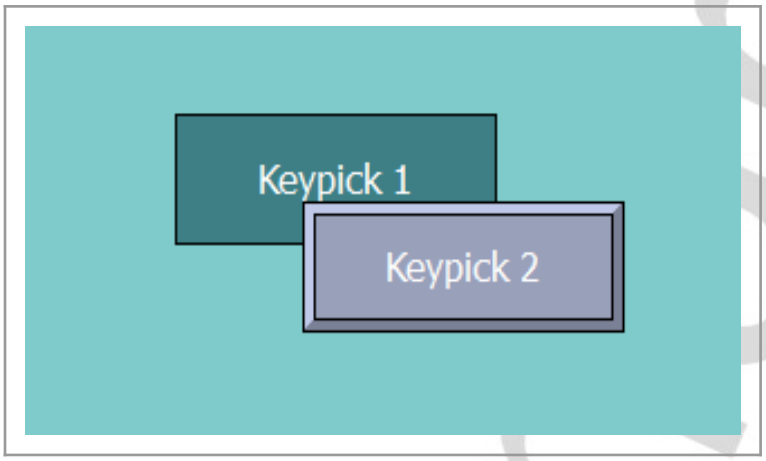


Abbildung 3.49: Keypicks Zeichnungs Reihenfolge

refresh Erzwingt ein neu zeichnen der Applikation.

delete Löscht das gerade aktuelle Keypick aus der Liste.

undo redo Erlaubt das Rückgängig machen , b.z.w. Wiederherstellen der letzten Änderungen..

new key Erstellt ein neues Keypick ohne Eigenschaften am Ende der Liste.

copy previous Kopiert die Eigenschaften des vorherigen Keypicks in das aktuelle Keypick.

insert key Fügt ein neues Keypick ohne Eigenschaften hinter dem aktuellem Keypick ein.

togle selection Schaltet die Markierung des aktuellen Keypicks im Applikationsfenster aus / ein um zu Kontrollieren ob das Erscheinungsbild des Keypicks den Vorgaben entspricht.

Alternativ zu der oben aufgeführten Art der Bedienung über die Buttons, können einige Funktionen auch über Popup Menü aufgerufen werden. Befindet sich der Mauscursor im Applikationsfenster und die rechte Maustaste wird betätigt erscheint das Key Pop Up Menü (Abbildung 3.50).

Mit 'copy' wird das aktuelle Keypick mit allen seinen Eigenschaften in das Clipboard copiert.

Mit 'paste' wird der Inhalt des Clipboards an der neuen Position eingefügt. Die Daten werden im Keypick Feld hinter dem Index des aktuellen Keypicks eingefügt.

Mit 'copy graphics' wird der Inhalt des aktuellen Keypicks als Graphik im Clipboard abgelegt, und kann als Bild in einem Dokument abgerufen werden.

Die Funktionen 'delete' und 'undo' entsprechen den Button delete und undo.

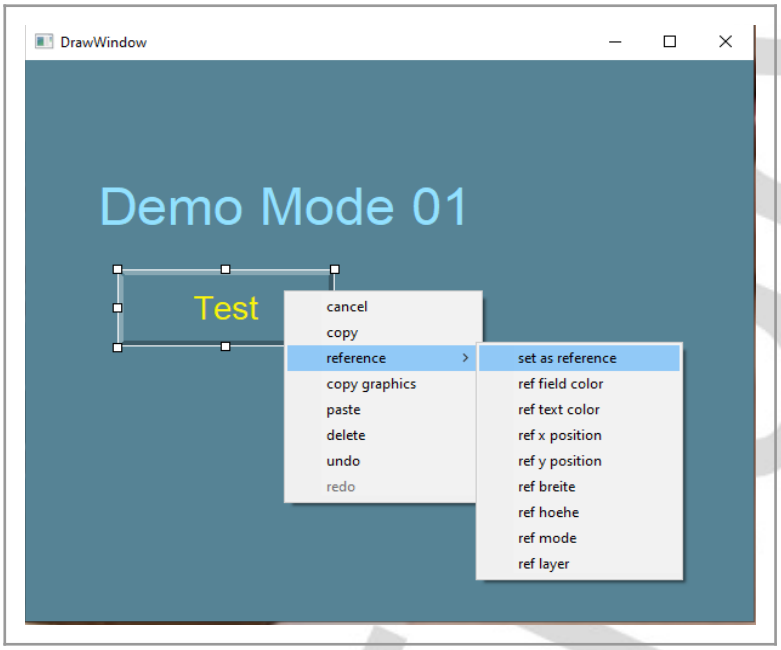


Abbildung 3.50: Key Pop Up Menü

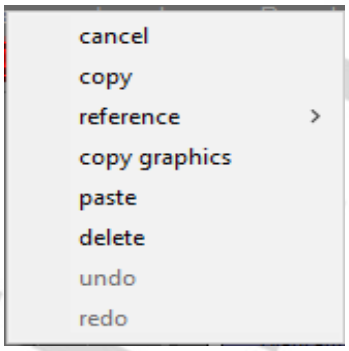


Abbildung 3.51: Keypick Pop Up Menü Einträge

cancel	Pop up wird geschlossen
copy	Das aktuelle Keypick wird in die Zwischenablage kopiert
reference	Das Refence Pop Up Menü wird aufgerufen.

Abbildung 3.52: Refence Pop Up Menü

set as reference	Die Einstellungen des aktuellen Keypicks werden in den Referenzspeicher übernommen
ref field color	Ordnet dem aktuellen Keypicks die abgespeicherte ‚field color , des Referenzspeichers zu.
ref text color	Ordnet dem aktuellen Keypicks die abgespeicherte ‚text color , des Referenzspeichers zu.
ref x position	Ordnet dem aktuellen Keypicks die abgespeicherte ‚x position, des Referenzspeichers zu.
ref y position	Ordnet dem aktuellen Keypicks die abgespeicherte ‚y position, des Referenzspeichers zu.
ref breite	Ordnet dem aktuellen Keypicks die abgespeicherte ‚breite, des Referenzspeichers zu.
ref hoehe	Ordnet dem aktuellen Keypicks die abgespeicherte ‚hoehe, des Referenzspeichers zu.
ref mode	Ordnet dem aktuellen Keypicks den abgespeicherte ‚mode, des Referenzspeichers zu.
ref layer	Ordnet dem aktuellen Keypicks den abgespeicherte ‚layer, des Referenzspeichers zu.

copy graphics	Das Abbild des aktuellen Keypicks wird in die Zwischenablage kopiert
paste	Der Inhalt der Zwischenablage wird als neues Keypicks eingefügt
delete	Das aktuelle Keypicks wird gelöscht
undo	Die letzte Aktion wird rückgängig gemacht

redo	Die letzte Aktion wird wieder hergestellt

3.5.2 Keypick Eigenschaften

Wie Kapitel 2.1.2 Keypicks bereits beschrieben stellen Keypicks eine rechteckige Zeichenfläche dar. Die Darstellung dieses Bereichs wird durch die im Eigenschafts Fenster festgesetzten Einstellungen festgelegt.

Die Eigenschaften eines Keypicks sind:

Startposition X und Y im Applikationsfenster

Breite (X-Delta) und Höhe (Y-Delta) des Keypicks

Die Art der Darstellung (Mode) des Keypicks

Die Farbe des Keypicks (Field Color)

Die Farbe eines Textes im Keypicks (Text Color)

Die Größe der Buchstaben des Textes (Text Zoom)

Die Bedeutung des Textes

Der Name eines Keypicks (nur zur Dokumentation erforderlich)

3.5.2.1 Keypicks Position und Größe im Applikationsfenster

Der Nullpunkt des Koordinatensystems im Applikationsfenster befindet sich in der linken oberen Ecke. Die Startposition X und Y eines Keypicks ergibt sich aus dem Abstand zu diesem Punkt. Die Größe des Keypicks ergibt sich aus den Delta Werten dx und dy zu dem Startpunkt wie in Abbildung 3.53 Keypick definition dargestellt.

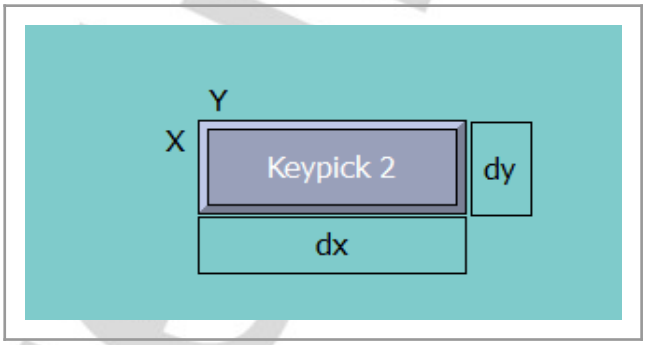


Abbildung 3.53: Keypick definition

Die aktuellen Werte für Position und Größe werden im Eigenschafts Bereich angezeigt (Abbildung 3.54).

Abbildung 3.54: Keypick Dimension

Die folgenden Beschreibungen der Veränderung der Position und oder Größe eines Keypicks beziehen sich auf eine Einstellung des Rasters laut Abbildung 3.55. Bei anderen Rastereinstellungen gelten die in Kapitel 3.2.2.1 Fangraster beschriebenen Verhaltensweisen.

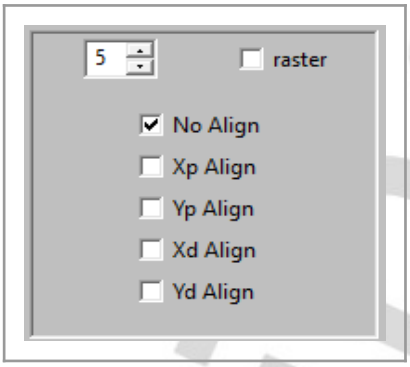
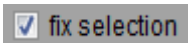


Abbildung 3.55: Raster No Allign

3.5.2.1.1 Keypick Position und Größe im Applikationsfenster ändern

Die Änderung der Position und Größe eines Keypicks kann auf zwei Arten erfolgen. Dieses Kapitel beschreibt die Änderung des aktuellen Keypicks im Applikationsfenster. Um zu Verhindern das beim Betätigen der linken Maustaste unbeabsichtigt ein anders Keypick selektiert wird empfiehlt es sich 'fix selection' einzuschalten. 

Position im Applikationsfenster ändern: (Abbildung 3.56)

5. Positionieren des Cursors innerhalb des aktuellen Keypicks.
6. Linke Maustaste gedrückt halten.
7. Keypick an die gewünschte Position schieben.
8. Maustaste loslassen.

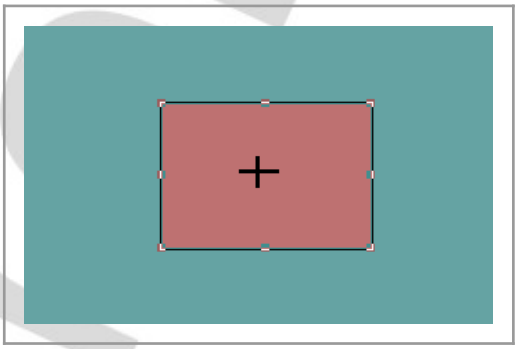


Abbildung 3.56: Keypick Positionieren

Größe im Applikationsfenster ändern: (Abbildung 3.57)

1. Positionieren des Cursors innerhalb des aktuellen Keypicks an der Markierung die verschoben werden soll.
2. Linke Maustaste gedrückt halten.
3. Markierung an die gewünschte Position schieben.
4. Maustaste loslassen.

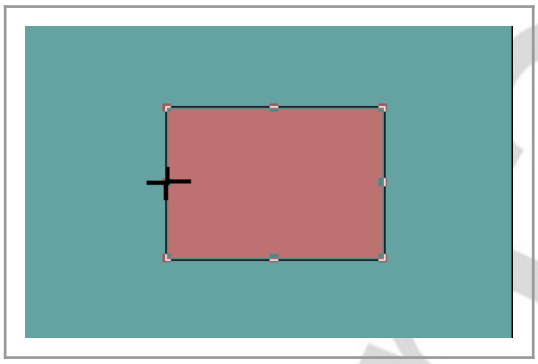


Abbildung 3.57: Keypick Größe ändern

3.5.2.1.2 Keypick Position und Größe im 'Keypick Menü' ändern.

Ändern der horizontalen Position durch Betätigen von Button [X] öffnet Abbildung 3.58: Keypick X-Start Menü.

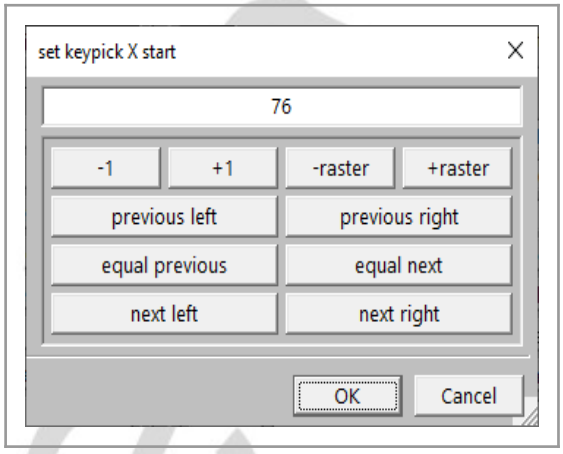


Abbildung 3.58: Keypick X-Start Menü

Durch das Betätigen des entsprechenden Button ändert sich die horizontale Position des Keypicks wie im Button Text angegeben.

Ändern der vertikalen Position durch Betätigen von Button [y] öffnet Abbildung 3.59: Keypick Y-Start Menü

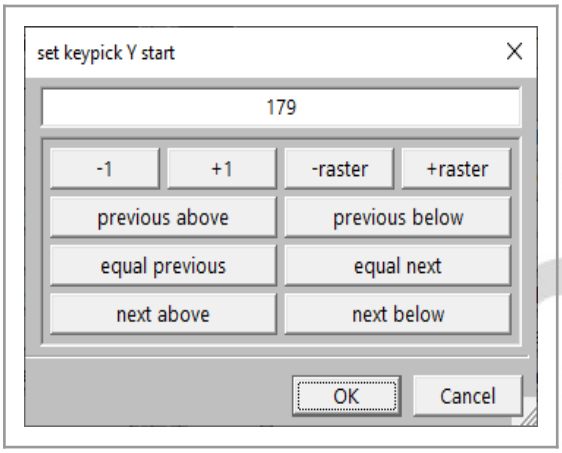


Abbildung 3.59: Keypick Y-Start Menü

Durch das Betätigen des entsprechenden Button ändert sich die vertikale Position des Keypicks wie im Button Text angegeben.

Ändern der Breite durch Betätigen von Button width öffnet Abbildung 3.60: Keypick Breite Menü

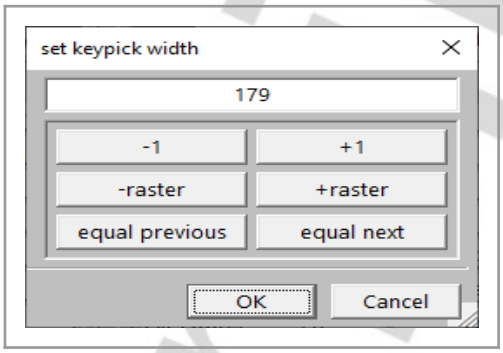


Abbildung 3.60: Keypick Breite Menü

Durch das Betätigen des entsprechenden Button ändert sich die Breite des Keypicks wie im Button Text angegeben.

Ändern der Höhe durch Betätigen von Button height öffnet Abbildung 3.61: Keypick Höhe Menü

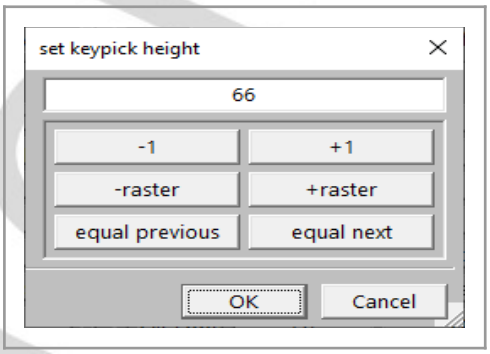


Abbildung 3.61: Keypick Höhe Menü

Durch das Betätigen des entsprechenden Button ändert sich die Höhe des Keypicks wie im Button Text angegeben.

3.5.2.2 Das Erscheinungsbild eines Keypicks ändern.

Die Art der Darstellung eines Keypicks wird durch den 'Mode' und die gesetzten Farben bestimmt. Zusätzlich besitzt jedes Keypick eine alternative Darstellung, die durch einen Mausclick oder eine externe Aktion zur Anzeige gebracht werden kann. Diese Darstellung wird durch den 'alt Mode' und die alternativen Farben bestimmt.

als auch


Öffnet den Keypick Mode Dialog. Die Abbildung 3.62 zeigt das Menü ohne gesetzten Mode.

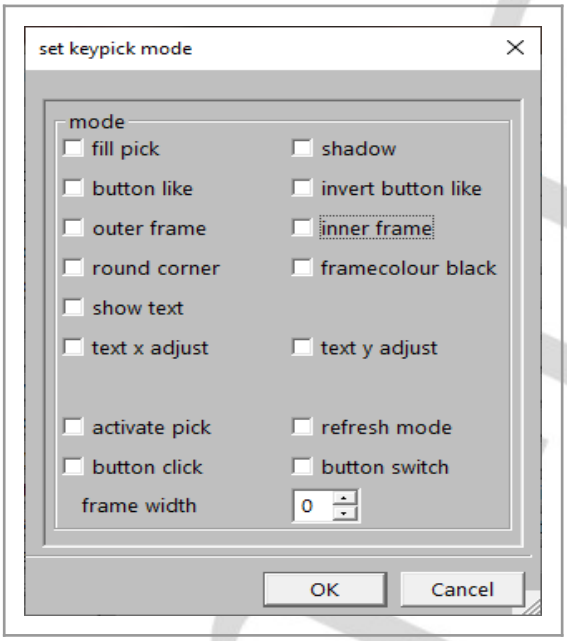
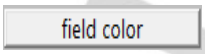


Abbildung 3.62: Keypick Mode Dialog

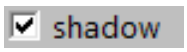
Die gesetzten Check Boxen definieren das Erscheinungsbild des Keypicks. Um die Einstellungen zu Verfolgen wird der Darstellungsmode im Drawwindow direkt angezeigt.

Wurde zuvor mit 

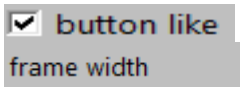
eine Farbe ausgewählt, lässt sich das Ergebnis direkt im Applikationsfenster verfolgen. In der Tabelle 3 sind Beispiele für verschiedene Mode Kombinationen aufgeführt.

Füllt die Fläche des Keypicks mit der unter 

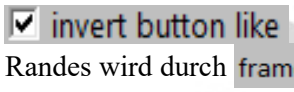
gesetzten Farbe.



Erstellt einen abgedunkelten Schatten hinter dem Keypick.

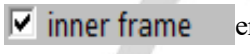


lässt das Keypick wie einen Button erscheinen. Die Breite des Randes wird durch bestimmt.



lässt das Keypick wie einen gedrückten Button erscheinen. Die Breite des Randes wird durch bestimmt.

erstellt einen Rahmen um die Keypick Fläche in der gewählten Textfarbe.



erstellt einen Rahmen innerhalb der Keypick Fläche in der gewählten Textfarbe. Der Abstand zum Außenrand wird durch 

bestimmt.

round corner Stellt das Keypick als Rechteck mit abgerundeten Ecken dar. In der Button Darstellung sind die Ecken des Button nicht gerundet.

framecolour black setzt die Farbe der Rahmen des Keypicks auf Schwarz / Weiß.

show text stellt den im Feld text Voltage ok angezeigten Text mit der in text color gesetzten Farbe und der unter text zoom gesetzten Größe dar.

text x adjust justiert den Text oder das zugeordnete Icon horizontal in die Mitte des Keypicks.

text y adjust justiert den Text oder das zugeordnete Icon vertikal in die Mitte des Keypicks.

Ist im Keypick Text eine Graphik selektiert, und text x adjust oder text y adjust oder auch beide gesetzt, kann mit text zoom die Skalierung der Graphik in Breite oder / und Höhe justiert werden. Andernfalls werden diese Werte der Breite b.z.w. der Höhe des Keypicks angepasst.

activate pick Versetzt das Keypick in einen aktiven Zustand, so daß es auf einen Click der linken Maustaste innerhalb seines Feldes reagiert wenn sich das Programm im Testmode befindet..

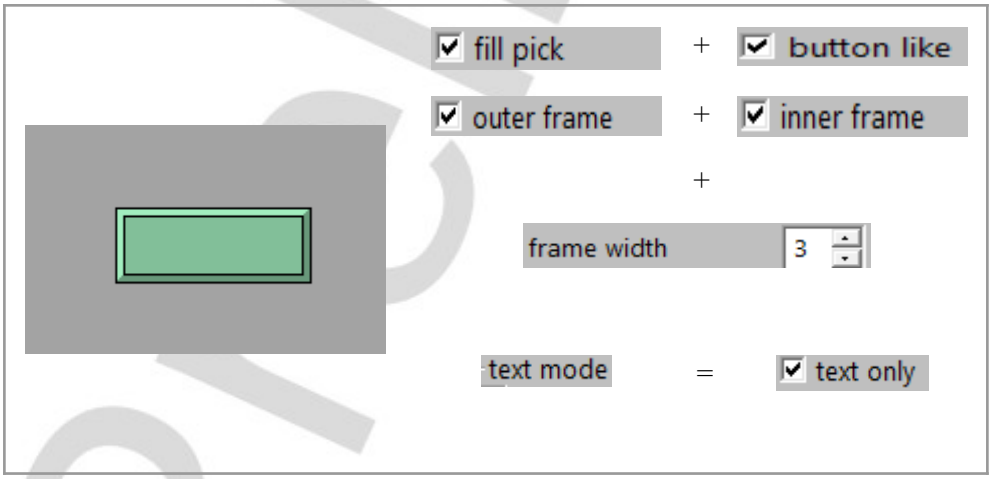
refresh mode aktualisiert das Keypick zyklisch (im Testmode).

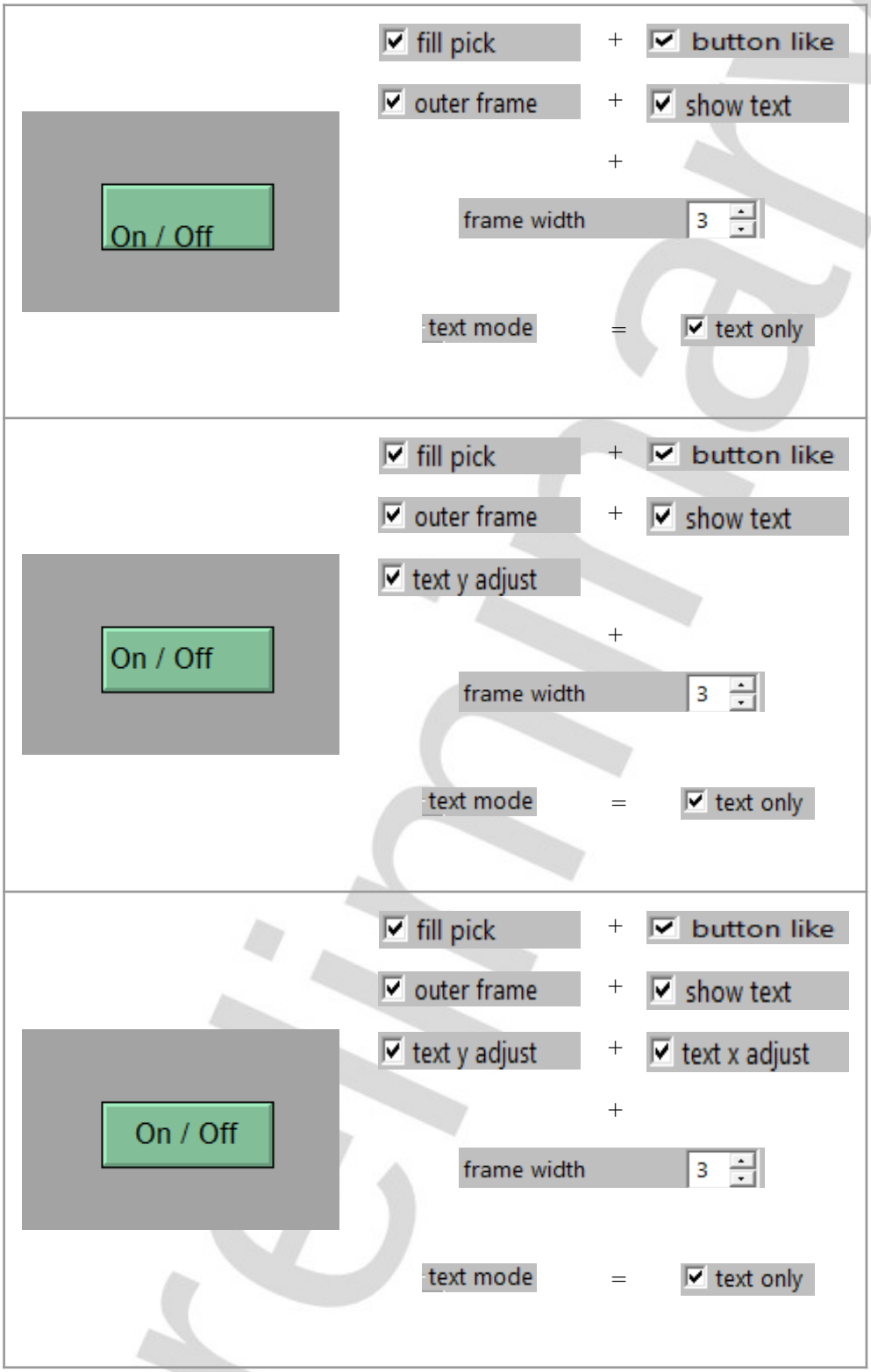
ACHTUNG: Ein Refresh ohne fill pick kann zu unleserlichen Ausgaben führen.

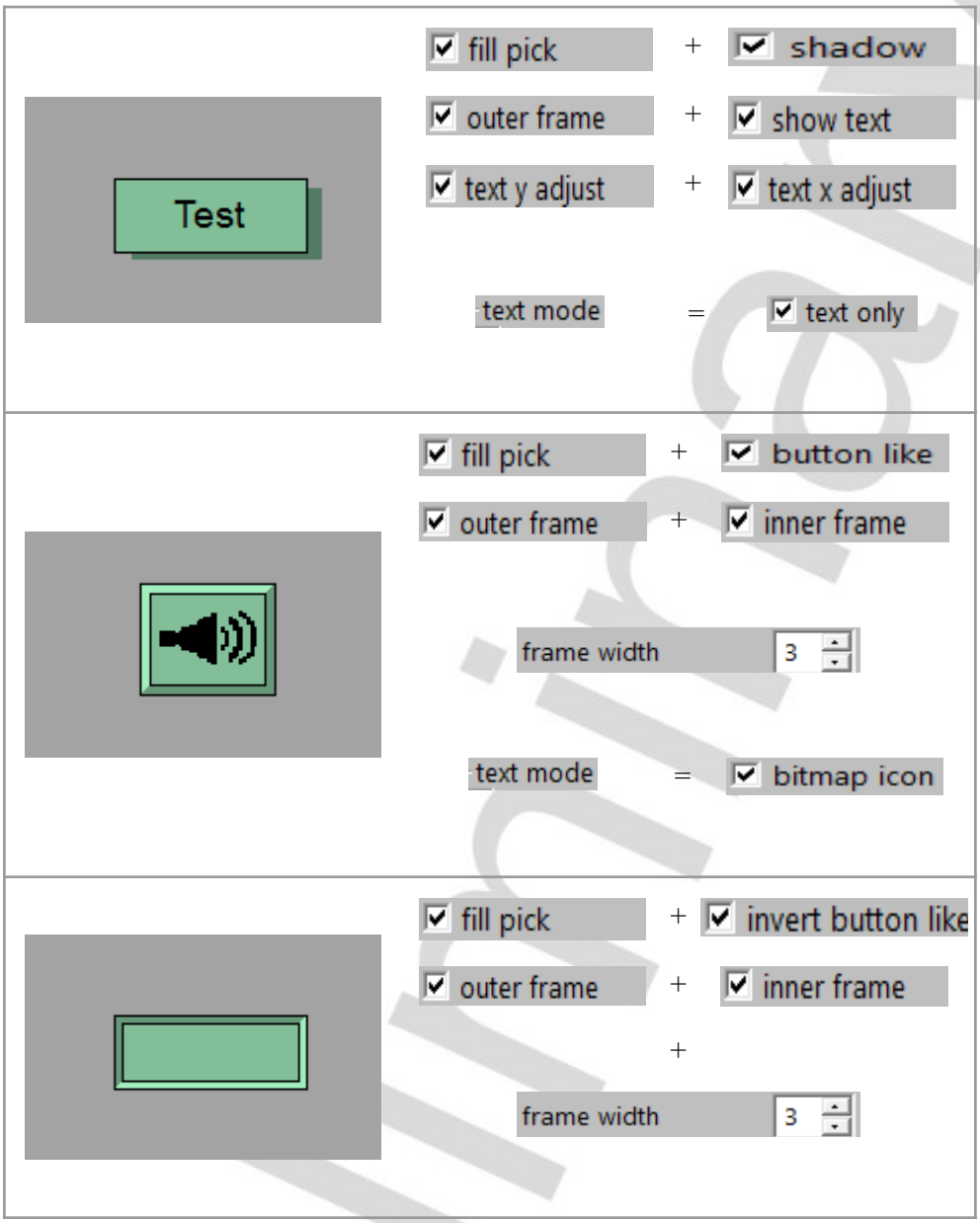
Wurde im Keypick Mode Dialog (Abbildung 3.62) die Option activate pick und button click gesetzt wird, bei einem Betätigen der linken Maustaste innerhalb des Keypicks die alternative Darstellung gezeigt so lange die Maustaste gedrückt ist(im Testmode).

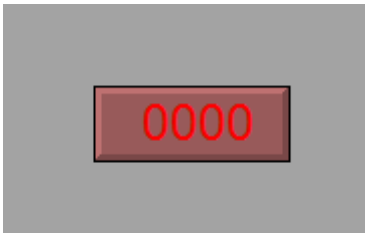
Wurde die Option button switch gewählt, schaltet die Darstellung mit jedem Mausclick um(im Testmode).

frame width 3 vergrößert die Randbreite des Keypicks, der im Normalfall 1 Pixel beträgt, um den angezeigten Wert in Pixel wenn kein inner frame gesetzt ist.









fill pick + button like

outer frame + show text

text y adjust + text x adjust

refresh mode

frame width 3

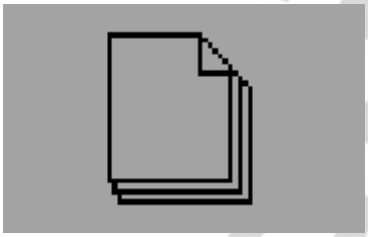
text mode = data

fill pick + button like

outer frame

frame width 3

text mode = pixmap icon



Kein Mode gesetzt. Dem Text wurde eine Bitmap Datei zugeordnet. Da kein Mode gesetzt wurde wird das Icon mit der Textfarbe auf den Hintergrund gezeichnet.

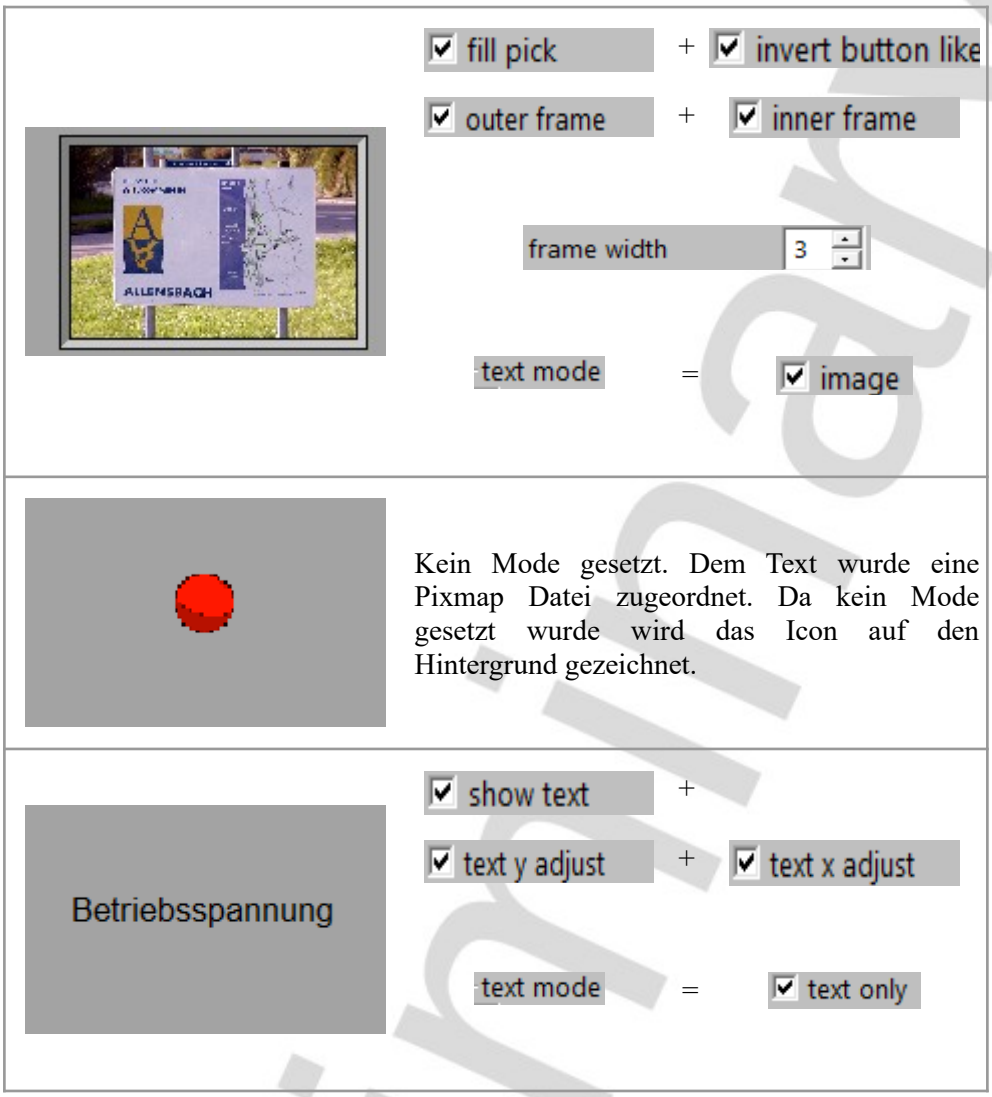


Tabelle 3: Beispiele Keypick

text color , alt text color , field color und alt field color öffnen das Color Menü (Abbildung 3.63) in welchem durch Anclicken die gewünschte Farbe ausgewählt wird.

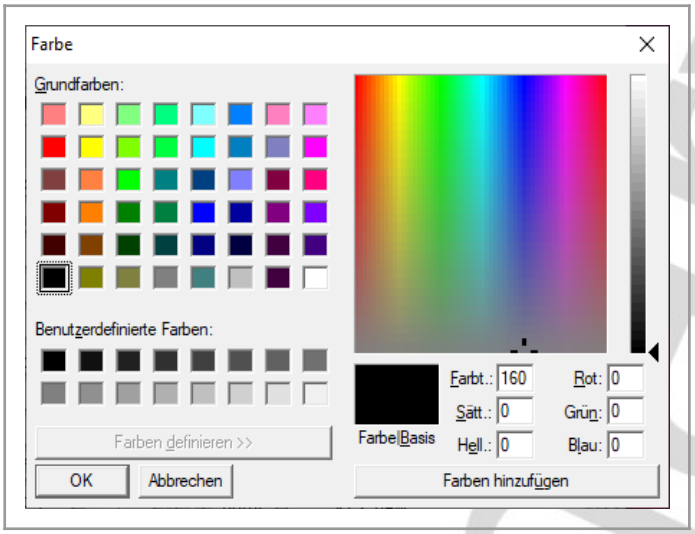


Abbildung 3.63: Color Menü

text Start Test und alt text Stop Test erlauben die Darstellung von Texten und Graphiken im Keypick. Beim Betätigen des Buttons wird der Keypick Text Dialog geöffnet siehe Abbildung 3.64. Wählt man eine Background Graphik aus, wird der Inhalt wie in Tabelle 3 unten gezeigt dargestellt.

copy to alt kopiert den gesetzten Mode des aktuellen Keypicks in den alternativen Mode.

Die Bedeutung der Optionen im Keypick Text Dialog. Alle Angaben gelten auch für den Alternativtext.

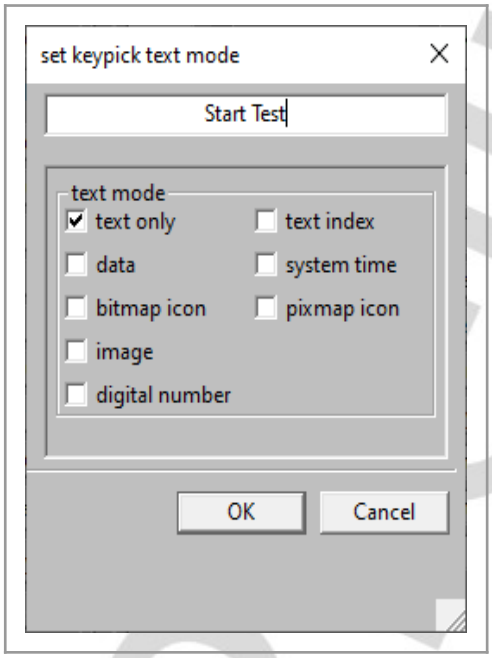


Abbildung 3.64: Keypick Text Dialog

text only stellt den Text aus dem Eingabefeld oben im Keypick Text Dialog (Abbildung 3.64) im Keypick dar. Es sind alle Bild Optionen deaktiviert, und das Keypick verhält sich wie vorab beschrieben.

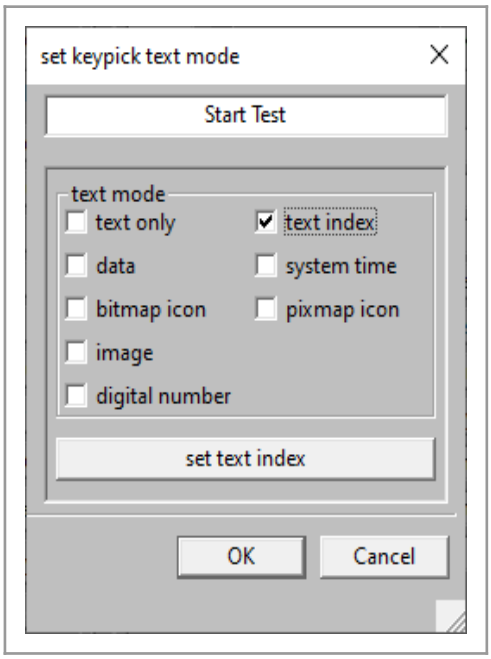


Abbildung 3.65: Keypick Text Dialog

☑ text index stellt statt dem Text aus dem Eingabefeld den Inhalt aus eines Textfeldes aus dem Interface dar., welcher durch den Index des Feldes bestimmt wird.

Mit set text index wird der Text Index Dialog geöffnet, in welchem der Index des anzuzeigenden Textfeldes gesetzt werden kann.

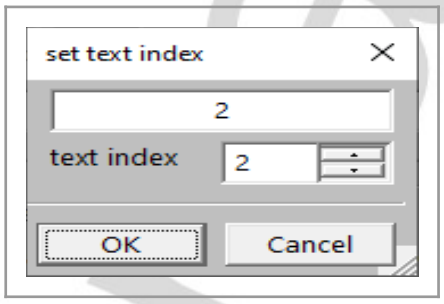


Abbildung 3.66: Text Index Dialog

Der Index kann in der Form %N auch im Eingabefeld direkt eingegeben werden. Wobei N die Nummer des Textfeldes darstellt. Ist im Keypick Mode Dialog (Abbildung 3.62) ebenfalls die Option ☑ refresh mode gesetzt, kann die Textanzeige des Keypicks dynamisch über das Interface geändert werden.

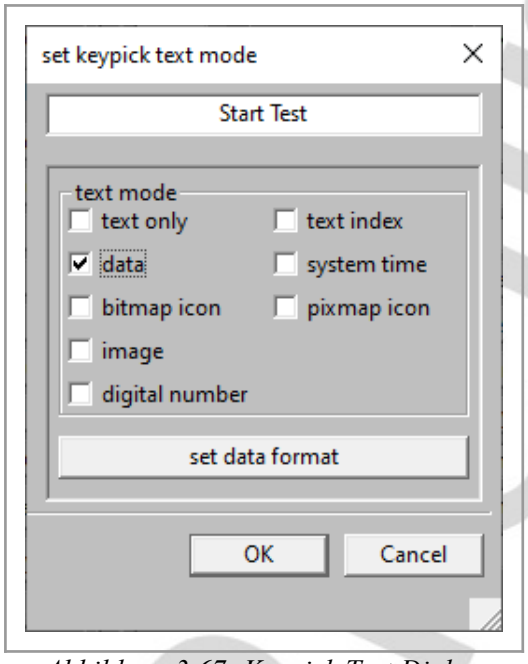


Abbildung 3.67: Keypick Text Dialog

☑ data stellt den Inhalt einer Variablen aus dem Interface formatiert als Text im Keypick dar.

Auch hier kann im Eingabefeld der Index und der Formatstring direkt eingegeben werden. Die Eingabe ist N %F, wobei N den Index der Variablen und %F den Formatstring darstellt. Damit immer der aktuelle Wert der Variablen angezeigt wird, muss auch hier im Keypick Mode Dialog (Abbildung 3.62) die Option ☑ refresh mode gesetzt sein. Für den Formatstring wird die von der Sprache C gebräuchliche Format-definition in einem printf Befehl verwendet.

Ist ☑ digital number gesetzt, werden die Daten als 7 Segment Darstellung ausgegeben.

Mit set data format wird der in Abbildung 3.68 gezeigte Dialog geöffnet, in welchem der Index und Typ der anzuzeigenden Variablen als auch der zu verwendende Formatstring gesetzt werden kann

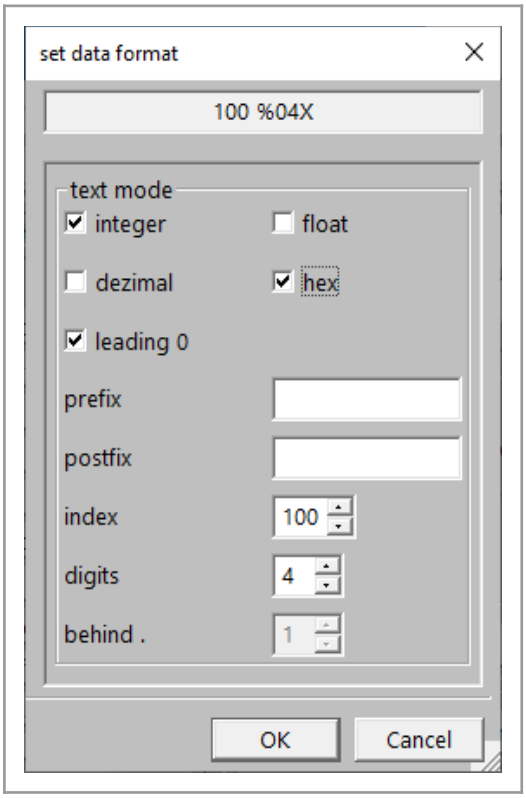


Abbildung 3.68: set data format

In diesem Fenster können die Optionen zur Darstellung der Daten per Mausclick gesetzt werden.

integer float bestimmt ob es sich bei den Daten um Integer oder Floatingpoint Werte handelt.

dezimal hex bestimmt die Ausgabe als Dezimal oder als Hexadezimalwert.

leading 0 wenn diese Option gesetzt ist, werden die führenden Nullen angezeigt.

prefix hier kann ein Text eingefügt werden, der vor dem Datenwert angezeigt werden soll.

postfix hier kann ein Text eingefügt werden, der nach dem Datenwert angezeigt werden soll.

index 100 Index der Variablen im Interface.

digits 4 Anzahl der anzuzeigenden Stelle der Variablen.

behind . 1 gibt bei Floatingpoint Werten die Anzahl der Stellen nach dem Komma an.

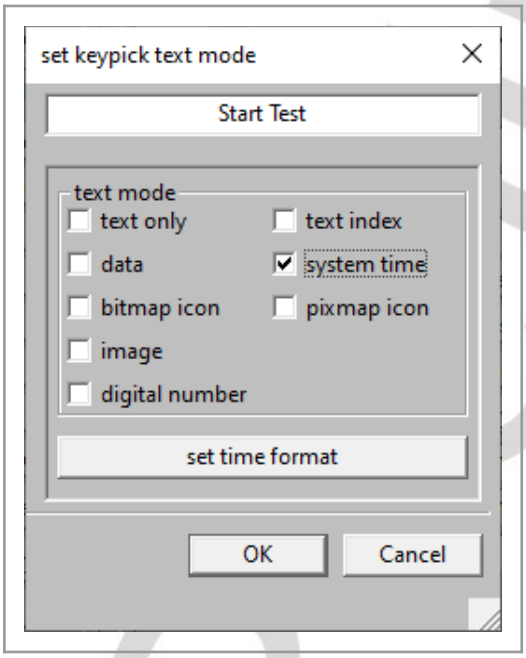


Abbildung 3.69: Keypick Text Dialog

system time Gibt die Systemzeit formatiert als Text im Keypick aus.

Mit set time format wird der set time format Dialog geöffnet, in welchem der Formatstring für die Zeitanzeige gesetzt werden kann. Zur Zeit wird hier defaultmäßig die Uhrzeit in der Form 'Stunde;Minute:Sekunde' gesetzt.

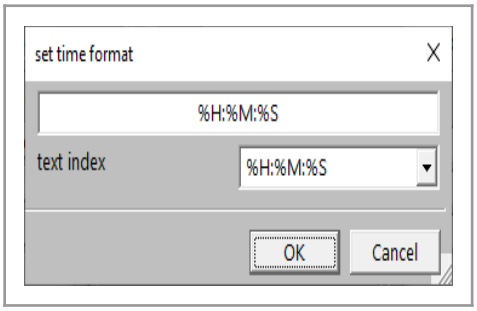


Abbildung 3.70: set time format

Auch hier kann wieder im Eingabefeld der Formatstring direkt eingegeben werden. Der Aufbau des Formtstrings ist in Tabelle 4 beschrieben.

%a	abgekürzter Wochentagsname	Thu
%A	ausgeschriebener Wochentagsname	Thursday
%b	abgekürzter Monatsname	Aug
%B	ausgeschriebener Monatsname	August
%c	volle Datums- und Zeitrepresentation	Thu Aug 23 14:55:02 2001
%d	Tag des Monats (01-31)	23
%H	Stunde im 24-Stunden-Format (00-23)	14
%I	Stunde im 12-Stunden-Format (01-12)	02
%j	Tag des Jahres (001-366)	235
%m	Monat als Dezimalnummer (01-12)	08
%M	Minute (00-59)	55
%p	AM oder PM Angabe	PM
%S	Sekunde (00-61)	02
%U	Wochennummer mit dem ersten Sonntag als ersten Tag der ersten Woche (00-53)	33
%w	Wochentag als Dezimalzahl mit Sonntag als 0 (0-6)	4
%W	Wochennummer mit dem ersten Montag als ersten Tag der ersten Woche (00-53)	34
%x	Datum	08/23/01
%X	Zeit	14:55:02
%y	Jahr, zweistellig (00-99)	01
%Y	Jahr, vierstellig	2001
%Z	Zeitzone oder Abkürzung der Zeitzone	CDT

%% Prozentzeichen	%

Tabelle 4: Systemzeit Formatdefinitionen

```
Vordefiniert sind folgenden time-string-formate.

%H:%M

%H:%M:%S

%d.%m.%Y

%d.%m.%Y %H:%M

%d.%m.%Y %H:%M:%S
```

Eine weitere Beschreibung des Time Ausgabe Formats findet man unter der C++ Funktionsbeschreibung ‚strftime'.

Damit immer die aktuelle Zeit angezeigt wird, muss auch hier im Keypick Mode Dialog (Abbildung 3.62) die Option set refresh mod gesetzt sein.

Die Größe der Textdarstellung in den vorab beschriebenen Optionen kann mit dem Button text zoom im Text Zoom Dialog (Abbildung 3.71) geändert werden.

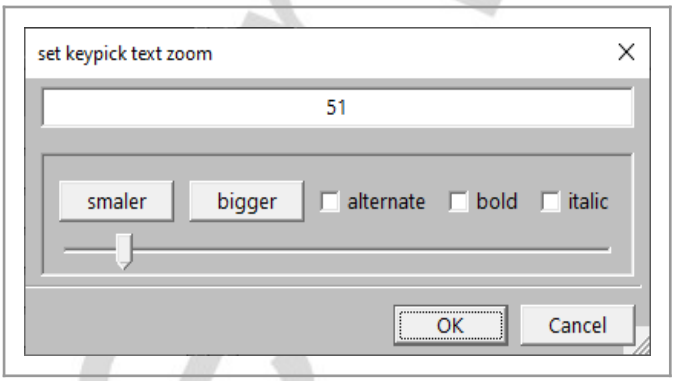


Abbildung 3.71: Text Zoom Dialog

Die Größe der Textanzeige kann über die Buttons als auch über den Slider auf das gewünschte Maß gesetzt werden.

alternate schaltet auf App font 2 um. Siehe Seite 19 Kapitel 3.2.3.2 Konfiguration Menü. Abbildung 3.17: Font Selectionn

bold stellt den Text in Fettschrift dar.

italic stellt den Text in Schrägschrift dar.

Ist eine der Optionen Bitmap Icon , Pixmap Icon oder Image gesetzt, wird der

Textzoom als Scalierungsfaktor der Keypick Graphik interpretiert.

Ist eine der Optionen ☑ Bitmap Icon , ☑ Pixmap Icon oder ☑ Image gesetzt, wird der Keypick Text als Dateiname eines Graphik Files interpretiert welches im Keypick dargestellt wird. Die Auswahl einer Graphik wird hier am Beispiel eines Bitmap Files beschrieben.

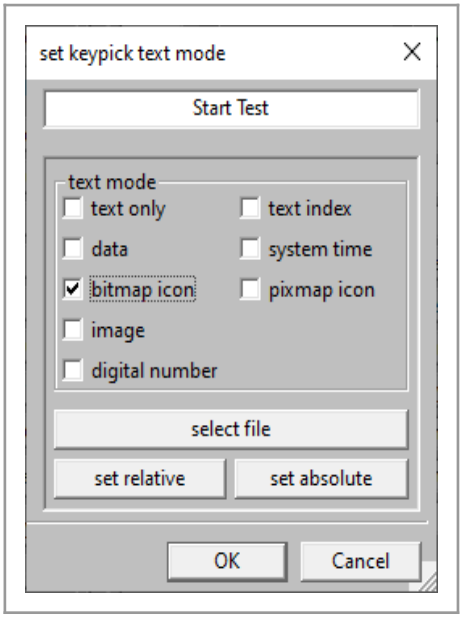


Abbildung 3.72: Keypick Text Dialog

Die im Keypick darzustellende Graphik kann mit dem Button [select file] über ein File Dialog Fenster ausgewählt werden.

Die Auswahl der Graphik erfolgt durch den in Abbildung 3.73 dargestellten File Browser. Da Windows im File Manager keine Vorschau für .xbm, .xpm und .plt Dateien erzeugt muss man diese Dateien zuordnen, um das Ergebnis im DrawWindo zu sehen.

[set relative] wandelt einen absoluten Dateipfad in einen zur Applikation relativen Dateipfad um.

[set absolute] wandelt einen zur Applikation relativen Dateipfad in einen absoluten Dateipfad um.

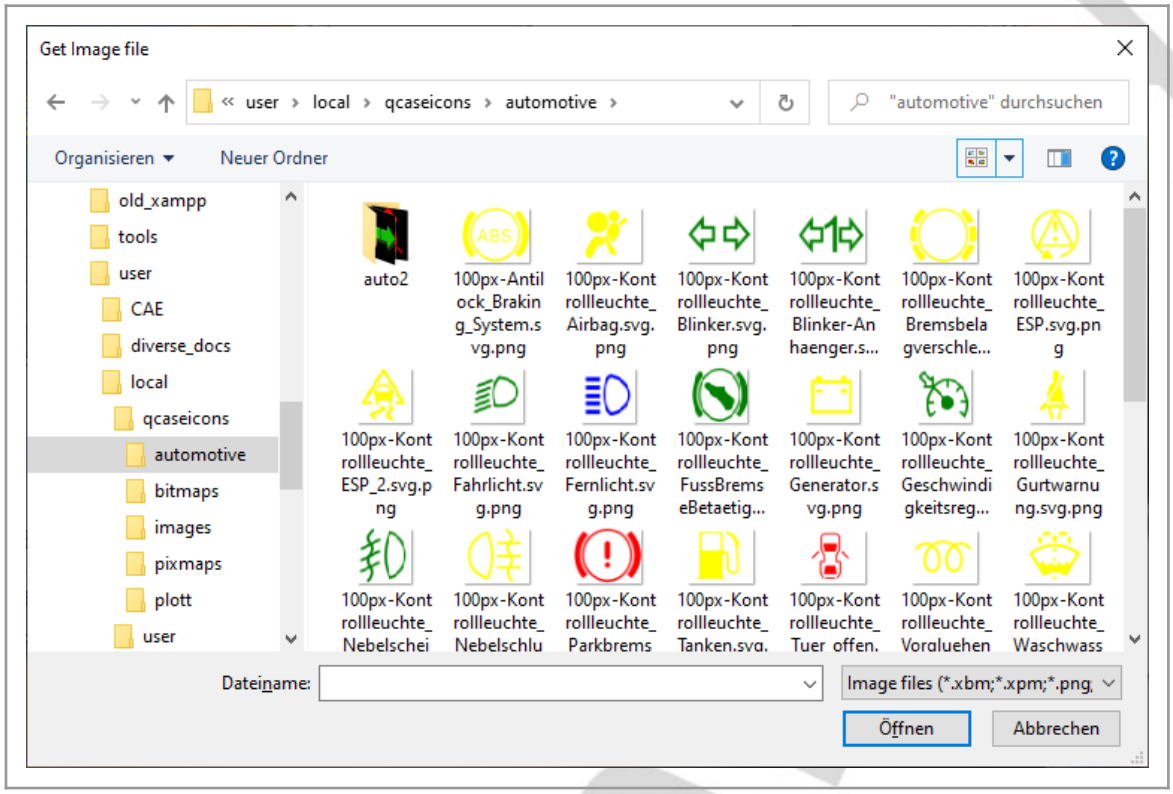


Abbildung 3.73: File Browser

Da der Typ der Grafik in einem Keypick entscheidend ist für die Darstellung und das Zusammenwirken mit Aktionen, folgt hier eine kurze Beschreibung der unterschiedlichen Formate. Die Bezeichnung 'Bitmap' und 'Pixmap' bezieht sich auf die in Linux X11 gebräuchlichen Bildformate. Bedauerlicherweise werden diese Formate im MS Windows Systemen nicht im Filemanager als Grafik angezeigt.

☑ Bitmap Icon Stellt eine Grafik dar die nur den Zustand Pixel gesetzt und Pixel nicht gesetzt kennt. Im Keypick wird ein nicht gesetztes Pixel mit der 'field color' dargestellt und ein gesetztes Pixel mit der 'text color'.

Ist kein Mode gesetzt, werden nur die gesetzten Pixel mit der 'field color' auf den Hintergrund gezeichnet.

Keypick mit 'text color' Schwarz  und 'text color' Gelb 

Über eine 'action' kann die Helligkeit der 'text color' in Abhängigkeit einer Variablen im Interface gesteuert werden um zum Beispiel die Lautstärke anzuzeigen, oder die Helligkeit einer LED zu steuern.

Leise Mittel  Laut  .

LED dunkel  mittel  hell  .

☑ Pixmap Icon Stellt ein mehrfarbiges Icon Format dar bei dem eine Farbe als transparent gesetzt werden kann. Im Keypick werden die transparenten Bereiche mit der 'field color' dargestellt.

Ist kein Mode gesetzt, werden nur die nicht transparenten Pixel auf den Hintergrund gezeichnet.

Keypick mit 'fiel color' Grün 

und 'field color' Rot 

Eine Veränderung der Helligkeit des Icons ist hier nicht möglich.

☑ Image Stellt bekannte Grafikformate wie .jpg .gif u.s.w. dar. Auch hier gibt es in Abhängig vom Format die Möglichkeit eine Farbe als transparent zu setzen. In diesem Fall ergibt sich das gleiche Verhalten wie bei einer Pixmap.

group 12 zeigt die Gruppenzugehörigkeit des Keypick an und kann durch das Betätigen des Button geändert werden (Abbildung 3.74). Die Gruppe hat keinen Einfluss auf die Eigenschaften des Keypicks sondern dient lediglich der internen Kennzeichnung einer Zusammengehörigkeit verschiedener Elemente welche im 'Group Menü' gemeinsam bearbeitet werden können.

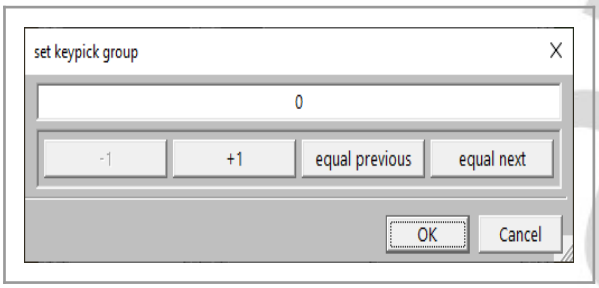


Abbildung 3.74: Group setting

name Pfeilsymbol Gibt dem Keypick einen Nahmen, an dem man es leichter Identifizieren kann als nur an der Indexnummer. Ebenso wie die Gruppe hat auch der Keypick Name keinen Einfluss auf die Eigenschaften. Er dient lediglich zur Dokumentation.

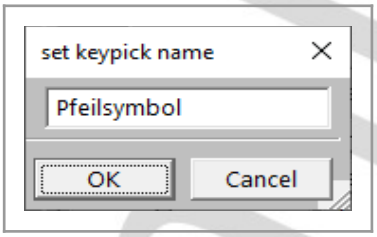


Abbildung 3.75: Keypick Namens Dialog

Wie oben beschrieben besitzen Keypicks im Gegensatz zu Windows eine alternative Darstellung. Hat man die normale Darstellung erstellt, kann man die gesetzten Eigenschaften durch den Button copy to alt auf die alternative Darstellung übertragen. Danach kann der Unterschied zur normalen Ansicht durch alt field color , alt text color , alt mode und a text Voltage to high geändert werden. Wird ein alternate Menü aufgerufen, wird im Applikationsfenster die alternative Darstellung des aktuellen Keypicks angezeigt.

Wenn ☑ show alt gesetzt ist wird im DrawWindow die alternative Darstellung des aktuellen Keypicks angezeigt.

Ist ☑ fix selection gesetzt, kann kein anderes Keypick durch einen Mausclick im DrawWindow selektiert werden.

3.6 Objekte

Erstellung von Objects und Bearbeitung deren Eigenschaften.

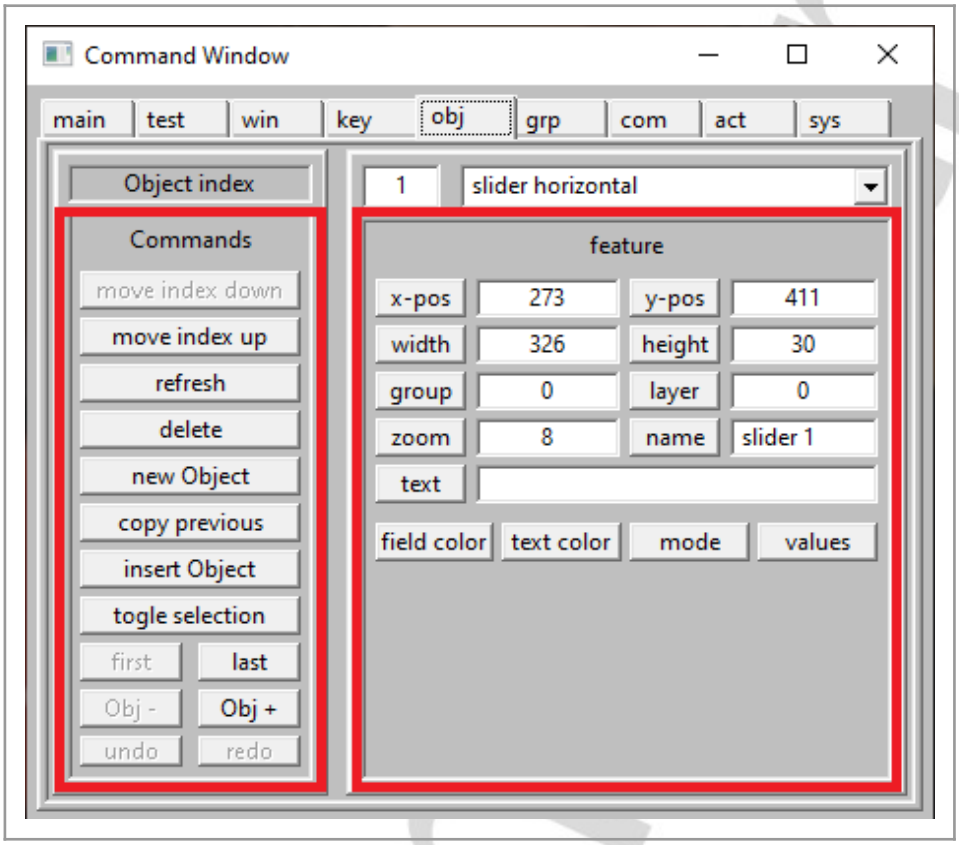


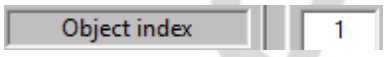
Abbildung 3.76: Objects Menü

Wie in Abbildung 3.76 gezeigt, besteht das Objekt Menü aus zwei Hauptbereichen. Dem 'Commands' Bereich und dem 'Eigenschaften' Bereich.

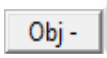
Der 'Commands' Bereich dient der Verwaltung der Objekte, während der 'Eigenschaften' Bereich wie der Name schon sagt zur Einstellung der Eigenschaften der Objekts vorgesehen ist.

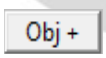
3.6.1 Objekt Verwaltung

Wie schon vorab beschrieben sind alle Elemente in Feldern abgelegt. Auf die Objekte kann über deren Feldindex zugegriffen werden. Der Index startet mit 1 und reicht bis zum letzten erstelltem Objekt.

Das Feld 

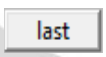
zeigt die Nummer des zur Bearbeitung ausgewählten Objekts an.

Mit den Buttons 

und 

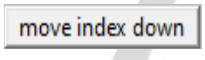
kann man sich durch das Feld aufwärts und abwärts bewegen.

Mit den Buttons 

und 

springt man direkt zum Anfang b.z.w zum Ende der Liste.

Alternativ kann ein zu bearbeitendes Objekt auch im Applikationsfenster durch Positionierung des Cursors in das Objekt und einem linken Mausclick selectiert werden.



Verschiebt das aktuelle Objekt um eine Position in der Liste nach unten.

move index up Verschiebt das aktuelle Objekt um eine Position in der Liste nach oben.

Da die Objekte in der Reihenfolge von 1 bis last dargestellt werden, und somit ein Objekt mit höherem Index alle mit niedrigerem Index überschreibt sollte das Überlappen vermieden werden. Da es sich bei Objekten um komplexe Elemente hantelt kann es sonst zu undefinierten Darstellungen führen.

refresh Erzwingt ein neu zeichnen der Applikation.

delete Löscht das gerade aktuelle Objekt aus der Liste.

undo redo Erlaubt das Rückgängig machen , b.z.w. Wiederherstellen der letzten Änderungen.

new Object Erstellt ein neues Objekt ohne Eigenschaften am Ende der Liste.

copy previous Kopiert die Eigenschaften des vorherigen Objekts in das aktuelle Objekts.

insert Object Fügt ein neues Objekt ohne Eigenschaften hinter dem aktuellem Objekt ein.

togle selection Schaltet die Markierung des aktuellen Objekts im 'Draw Window' aus / ein um zu Kontrollieren ob das Erscheinungsbild des Objekts den Vorgaben entspricht.

undo redo Erlaubt das Rückgängig machen , b.z.w. Wiederherstellen der letzten Änderungen.

Alternativ zu der oben aufgeführten Art der Bedienung über die Buttons, können einige Funktionen auch über Popup Menü aufgerufen werden. Befindet sich der Mauscursor im Applikationsfenster und die rechte Maustaste wird betätigt erscheint das Objekt Pop Up Menue (Abbildung 3.77).

Mit 'copy' wird das aktuelle Objekt mit allen seinen Eigenschaften in das Clipboard copiert.

Mit 'paste' wird der Inhalt des Clipboards an der neuen Position eingefügt. Die Daten werden im Objekt Feld hinter dem Index des aktuellen Objekt eingefügt.

Mit 'copy graphics' wird der Inhalt des aktuellen Objekt als Graphik im Clipboard abgelegt, und kann als Bild in einem Dokument abgerufen werden.

Die Funktionen 'delete' und 'undo' entsprechen den Button delete und undo.

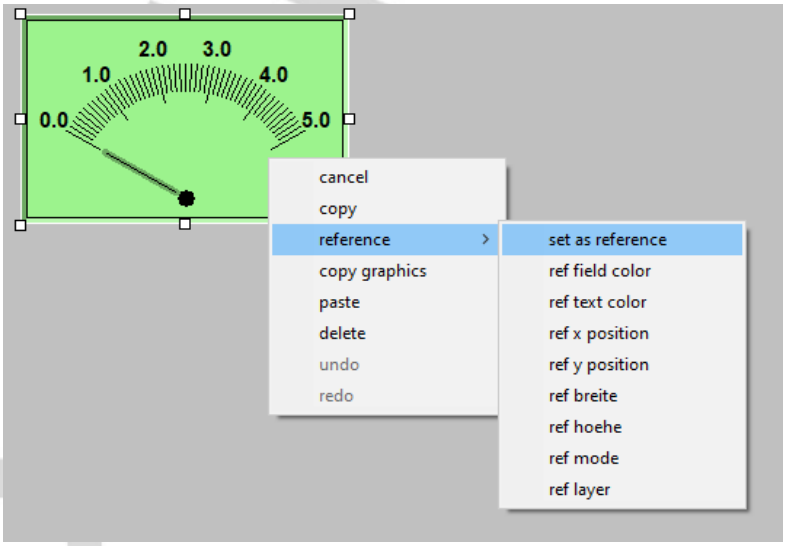


Abbildung 3.77: Objekt Pop Up Menue

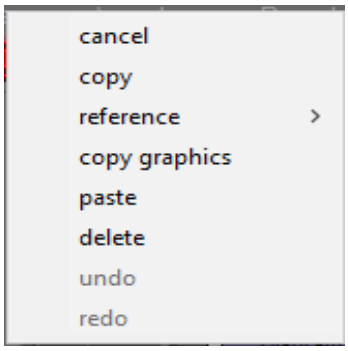


Abbildung 3.78: Keypick Pop Up Menü Einträge

cancel	Pop up wird geschlossen
copy	Das aktuelle Objekt wird in die Zwischenablage kopiert
reference	Das Refence Pop Up Menü wird aufgerufen.

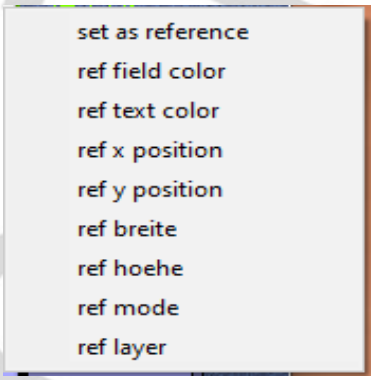


Abbildung 3.79: Refence Pop Up Menü

set as reference	Die Einstellungen des aktuellen Objekt werden in den Referenzspeicher übernommen
ref field color	Ordnet dem aktuellen Objekt die abgespeicherte ,field color , des Referenzspeichers zu.
ref text color	Ordnet dem aktuellen Objekt die abgespeicherte ,text color , des Referenzspeichers zu.
ref x position	Ordnet dem aktuellen Objekt die abgespeicherte ,x position, des Referenzspeichers zu.
ref y position	Ordnet dem aktuellen Objekt die abgespeicherte ,y position, des Referenzspeichers zu.
ref breite	Ordnet dem aktuellen Objekt die abgespeicherte

	,breite, des Referenzspeichers zu.
	ref hoehe — Ordnet dem aktuellen Objekt die abgespeicherte ,hoehe, des Referenzspeichers zu.
	ref mode — Ordnet dem aktuellen Objekt den abgespeicherte ,mode, des Referenzspeichers zu.
	ref layer — Ordnet dem aktuellen Objekt den abgespeicherte ,layer, des Referenzspeichers zu.
copy graphics	Das Abbild des aktuellen Objekt wird in die Zwischenablage kopiert
paste	Der Inhalt der Zwischenablage wird als neues Objekt eingefügt
delete	Das aktuelle Objekt wird gelöscht
undo	Die letzte Aktion wird rückgängig gemacht
redo	Die letzte Aktion wird wieder hergestellt

3.6.2 Objekt Eigenschaften

Wie Kapitel 2.1.3 Objekte bereits beschrieben stellen Objekte eine rechteckige Zeichenfläche dar. Die Darstellung dieses Bereichs wird durch die im Eigenschafts Fenster festgesetzten Einstellungen festgelegt.

Die Eigenschaften eines Objekts sind:

Startposition X und Y im Applikationsfenster

Breite (X-Delta) und Höhe (Y-Delta) des Objekts

Die Art der Darstellung (Mode) des Objekts

Die Farbe des Objekts (Field Color)

Die Farbe eines Textes im Objekt (Text Color)

Die Größe der Buchstaben des Textes (Text Zoom)

Die Bedeutung des Textes

Der Name eines Objekts (nur zur Dokumentation erforderlich)

3.6.2.1 Objekt Position und Größe im Applikationsfenster

Der Nullpunkt des Koordinatensystems im Applikationsfenster befindet sich in der linken oberen Ecke. Die Startposition X und Y eines Objekts ergibt sich aus dem Abstand zu diesem Punkt. Die Größe des Objekts ergibt sich aus den Delta Werten dx und dy zu dem Startpunkt wie in Abbildung 3.80 Objekt Definition dargestellt.

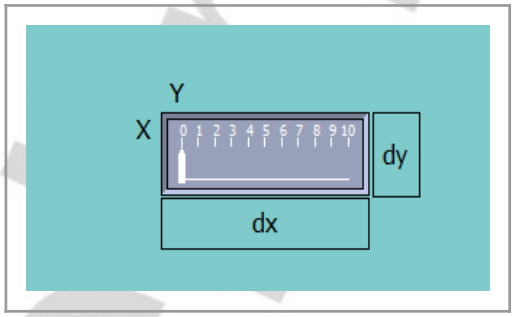


Abbildung 3.80: Objekt Definition

Die aktuellen Werte für Position und Größe werden im Eigenschafts Bereich angezeigt (Abbildung 3.81).

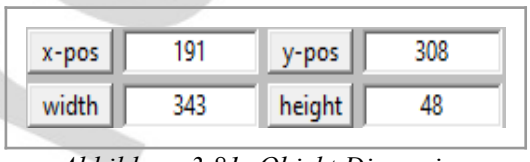


Abbildung 3.81: Objekt Dimension

Die folgenden Beschreibungen der Veränderung der Position und oder Größe eines Objekts beziehen sich auf eine Einstellung des Rasters laut Abbildung 3.82. Bei anderen Rastereinstellungen gelten die

in Kapitel 3.2.2.1 Fangraster beschriebenen Verhaltensweisen.

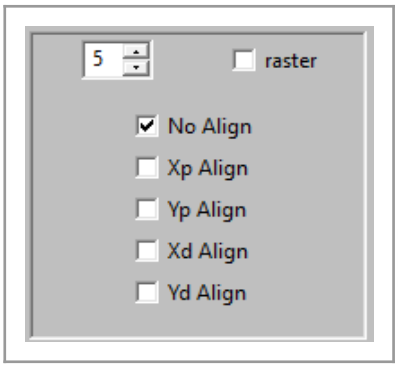


Abbildung 3.82: Raster No Allign

3.6.2.1.1 Objekt Position und Größe im Applikationsfenster ändern

Die Änderung der Position und Größe eines Objekts kann auf zwei Arten erfolgen. Dieses Kapitel beschreibt die Änderung des aktuellen Objekts im Applikationsfenster.

Position im Applikationsfenster ändern: (Abbildung 3.83)

5. Positionieren des Cursors innerhalb des aktuellen Objekts.
6. Linke Maustaste gedrückt halten.
7. Objekt an die gewünschte Position schieben.
8. Maustaste loslassen.

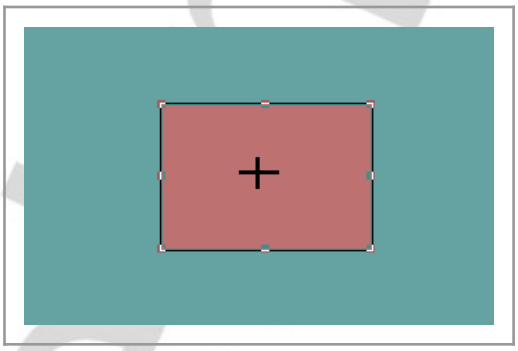


Abbildung 3.83: Objekt Positionieren

Größe im Applikationsfenster ändern: (Abbildung 3.84)

1. Positionieren des Cursors innerhalb des aktuellen Objekts an der Markierung die verschoben werden soll.
2. Linke Maustaste gedrückt halten.
3. Markierung an die gewünschte Position schieben.
4. Maustaste loslassen.

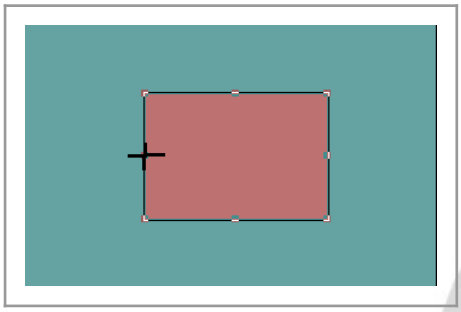


Abbildung 3.84: Objekt Größe ändern

3.6.2.1.2 Objekt Position und Größe im ' Objekts Menü' ändern.

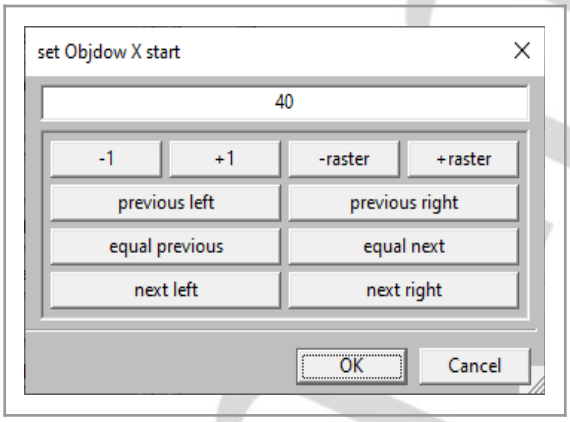


Abbildung 3.85: Objekt X-Start Menü

Ändern der horizontalen Position durch Betätigen von Button x-pos öffnet Abbildung 3.85: Objekt X-Start Menü.

Durch das Betätigen des entsprechenden Button ändert sich die horizontale Position des Objekts wie im Button Text angegeben.

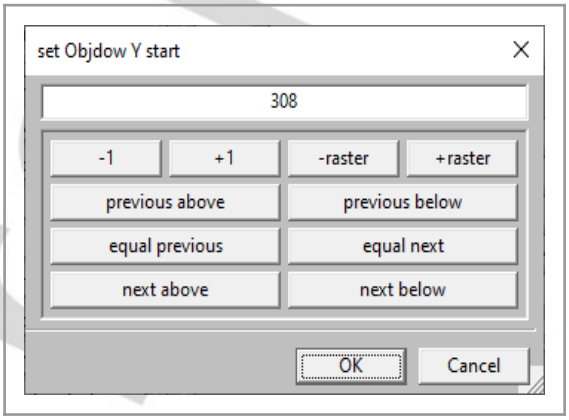


Abbildung 3.86: Objekt Y-Start Menü

Ändern der vertikalen Position durch Betätigen von Button y-pos öffnet Abbildung 3.86: Objekt Y-

Durch das Betätigen des entsprechenden Button ändert sich die vertikale Position des Objekts wie im Button Text angegeben.

Ändern der Breite durch Betätigen von Button width öffnet Abbildung 3.87: Objekt Breite Menü

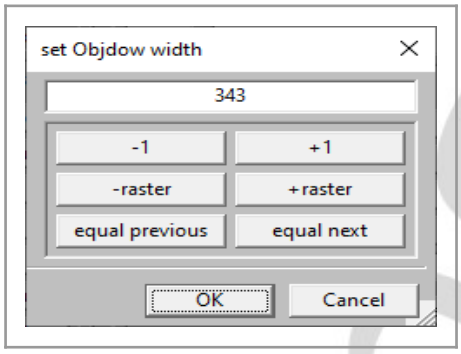


Abbildung 3.87: Objekt Breite Menü

Durch das Betätigen des entsprechenden Button ändert sich die Breite des Objekts wie im Button Text angegeben.

Ändern der Höhe durch Betätigen von Button height öffnet Abbildung 3.88: Objekt Höhe Menü

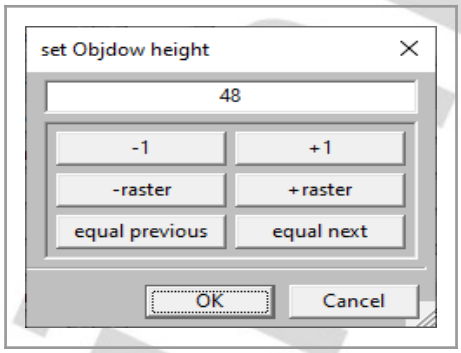


Abbildung 3.88: Objekt Höhe Menü

Durch das Betätigen des entsprechenden Button ändert sich die Höhe des Objekts wie im Button Text angegeben.

3.6.2.2 Das Erscheinungsbild eines Objekts ändern.

Die Art der Darstellung eines Objekts wird durch den 'Mode' und die gesetzten Farben bestimmt.

mode Öffnet den Objekt Mode Dialog. Die Abbildung 3.89 zeigt das Menü ohne gesetzten Mode.

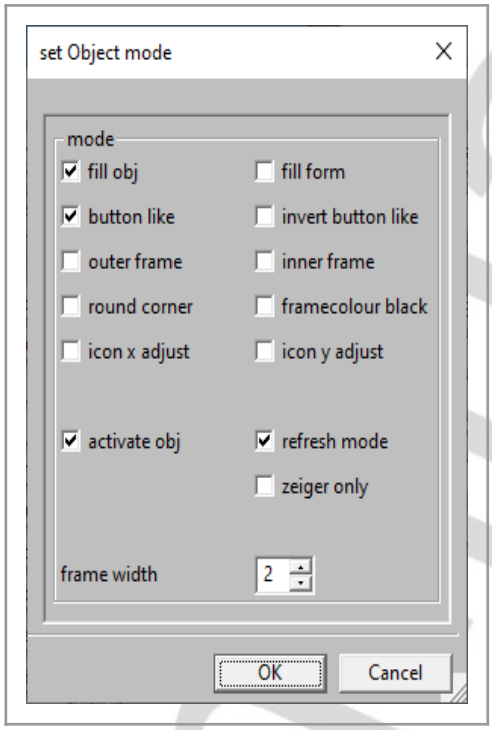


Abbildung 3.89: Objekt Mode Dialog

Die gesetzten Check Boxen definieren das Erscheinungsbild des Objekts. Wurde zuvor mit field color eine Farbe ausgewählt, lässt sich das Ergebnis direkt im Applikationsfenster verfolgen. In der Tabelle 5 sind Beispiele für verschiedene Mode Kombinationen am Beispiel eines slider horizontal aufgeführt.

Auflistung der einzelnen Eigenschaften.

fill obj Füllt die Fläche des Objekts welche mit dx und dy definiert wurde mit der unter field color gesetzten Farbe.

fill form Füllt die Fläche des Objekts abhängig von der Objekt Form und ist nur bei Instrument Objekten wirksam. Ein Beispiel für ein rundes Instrument ist am Ende der Tabelle 5 Beispiele Objekts Modes (slider horizontal) dargestellt.

button like lässt das Objekt wie einen Button erscheinen. Die Breite des Randes wird durch frame width 3 bestimmt.

invert button like lässt das Objekt wie einen gedrückten Button erscheinen. Die Breite des Randes wird durch frame width 3 bestimmt.

outer frame erstellt einen schwarzen Rahmen um die Objekt Fläche.

☑ inner frame erstellt einen schwarzen Rahmen innerhalb der Objekt Fläche. Der Abstand zum Außenrand wird durch frame width 3 bestimmt.

☑ round corner Stellt das Objekt als Rechteck mit abgerundeten Ecken dar. In der Button Darstellung sind die Ecken des Button nicht gerundet.

☑ framecolour black setzt die Farbe des Randes des Objekts auf Schwarz / Weiß.

☐ icon x adjust justiert das zugeordnete Icon horizontal in die Mitte des Objekts.

☐ icon y adjust justiert das zugeordnete Icon vertikal in die Mitte des Objekts.

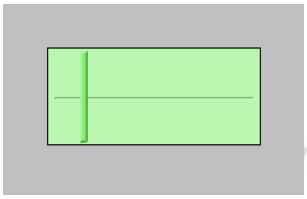
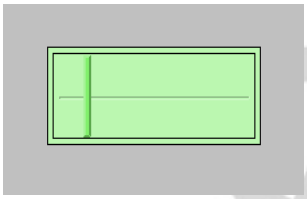
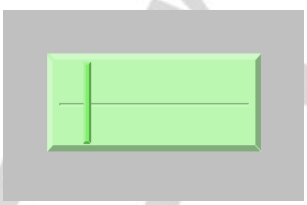
Ist im Objekt Text eine Graphik selektiert, und ☑ text x adjust oder ☑ text y adjust oder auch beide gesetzt, kann mit text zoom die Skalierung der Graphik in Breite oder / und Höhe justiert werden. Andernfalls werden diese Werte der Breite b.z.w. der Höhe des Objekts angepasst.

☑ activate obj Versetzt das Objekt in einen aktiven Zustand so das es als Eingabe Element verwendet werden kann. Das betrifft alle nicht Diagramm Objekte (z.B. Slider, Bars und Instrumente. Um diese Funktion wirksam werden zu lassen muss ebenfalls ☑ refresh mode gesetzt sein.

☑ refresh mode aktualisiert das Objekt zyklisch.

☐ zeiger only unterdrückt den Refresh der Skala bei runden Instrumenten.

frame width 3 vergrößert die Randbreite des Objekts, der im Normalfall 1 Pixel beträgt, um den angezeigten Wert in Pixel wenn kein inner frame gesetzt ist.

	☑ fill obj + ☑ outer frame + ☑ framecolour black Das Objekt wird mit einem schwarzem Rahmen umgeben.
	☑ fill obj ☑ outer frame + ☑ inner frame + ☑ framecolour black + frame width 3 Das Objekt wird mit einem schwarzem Rahmen außen und innen mit Abstand 3 Pixel gezeichnet.
	☑ fill obj ☑ button like + Das Objekt wird als Button gezeichnet mit einem 1 Pixel breitem Rand (rand offset = 0).

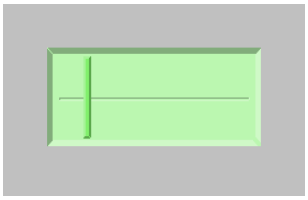
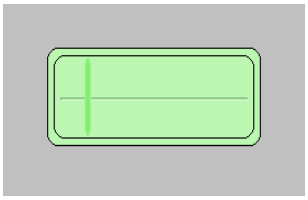
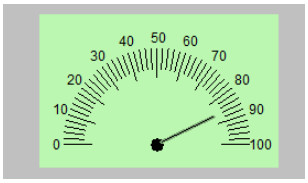
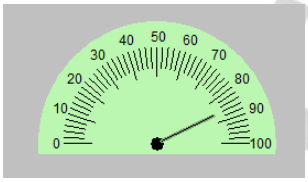
	☑ fill obj + ☑ invert button like Das Objekt wird als gedrückter Button gezeichnet mit einem 1 Pixel breitem Rand (rand offset = 0).
	☑ fill obj + ☑ round corner ☑ outer frame + ☑ inner frame Das Objekt wird mit einem gerundetem schwarzem Rahmen außen und innen mit Abstand 3 Pixel gezeichnet.
	☑ fill obj + ☑ invert button like + ☑ outer frame + ☑ inner frame + ☑ framecolour black + ☑ zeiger only Als Hintergrund wurde hier im Text die Option ‚Background image' gesetzt.
	☑ fill obj Hier wird der gesamte Hintergrund des Objekts mit der Feldfarbe gefüllt.
	☑ fill form Hier wird nur der Hintergrund der durch die Form des Objekts gegeben ist mit der Feldfarbe gefüllt.

Tabelle 5: Beispiele Objekts Modes (slider horizontal)

☑ refresh mode hat keinen Einfluss auf das Erscheinungsbild des Objekts, sondern erzwingt im Test ein zyklisches Aktualisieren der Ausgabe.

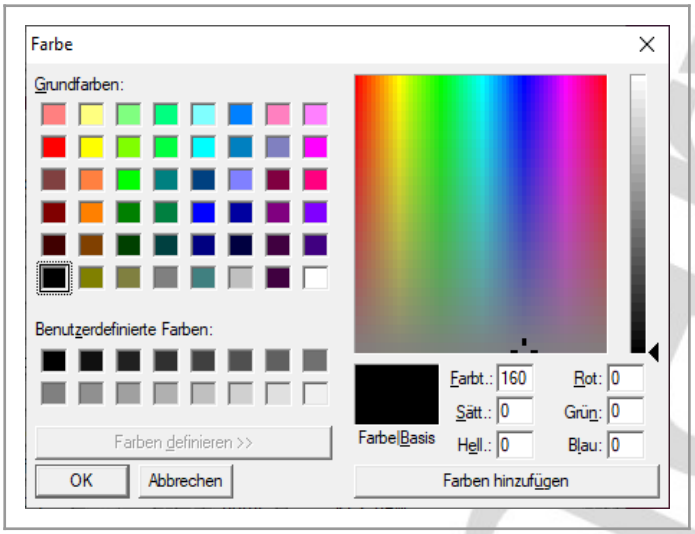


Abbildung 3.90: Color Menü

text color und field color öffnen das Color Menü (Abbildung 3.90) in welchem durch anklicken die gewünschte Farbe ausgewählt wird. text color bestimmt hier die Zeichnungsfarbe des Objekts.Abbildung 3.19

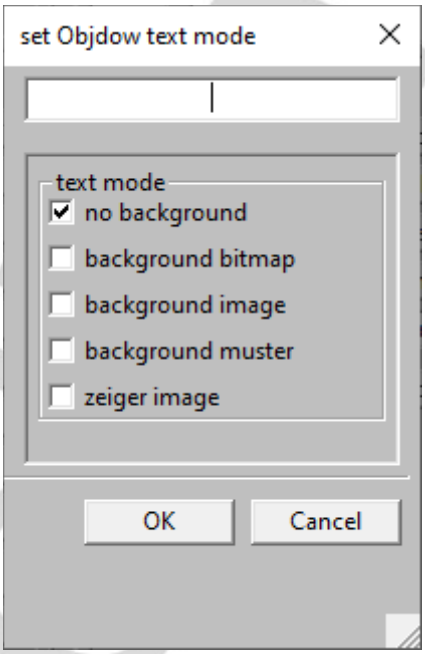
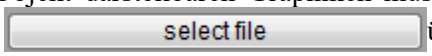


Abbildung 3.91: Objekt Text Dialog

text D:\images\boote_01.jpg erlaubt die Darstellung von Graphiken im Objekt. Beim Betätigen des Buttons wird der Objects Text File Dialog geöffnet siehe Abbildung 3.92. Ist No Background gesetzt, sind alle Bild Optionen deaktiviert, und das Objekts verhält sich wie vorab beschrieben. Wählt man eine Background Graphik aus, wird der Inhalt wie in Tabelle 6 beschrieben dargestellt.

Die im Objekt darstellbaren Graphiken müssen in einer Datei abgelegt sein, und können mit dem Button 

über ein File Dialog Fenster ausgewählt werden. Die anderen Eigenschaften des Objekts wie in Beispiele Objekts Modes (slider horizontal) Tabelle 5 beschrieben, bleiben dabei erhalten.

wandelt einen absoluten Dateipfad in einen zur Applikation relativen Dateipfad um.

wandelt einen zur Applikation relativen Dateipfad in einen absoluten Dateipfad um.

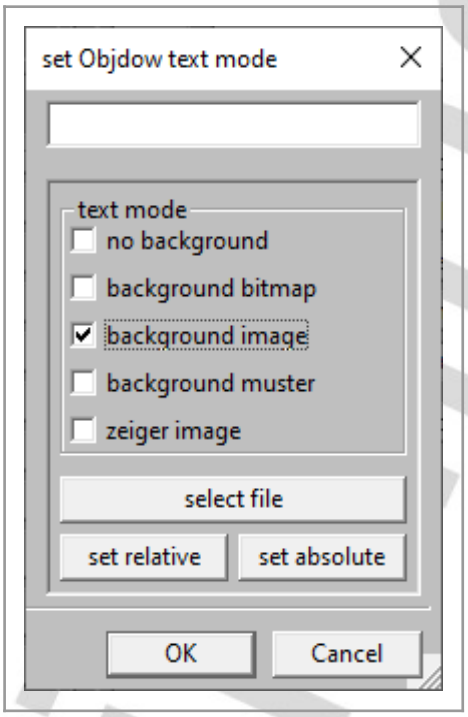


Abbildung 3.92: Objects Text File Dialog

Die Auswahl einer Graphik erfolgt in dem System eigene File Browser Abbildung 3.93.Da Windows im File Manager keine Vorschau für .xbm, .xpm und .plt Dateien erzeugt, ist das Ergebnis einer Bildauswahl erst nach dem Schließen des Auswahlfensters zu sehen.

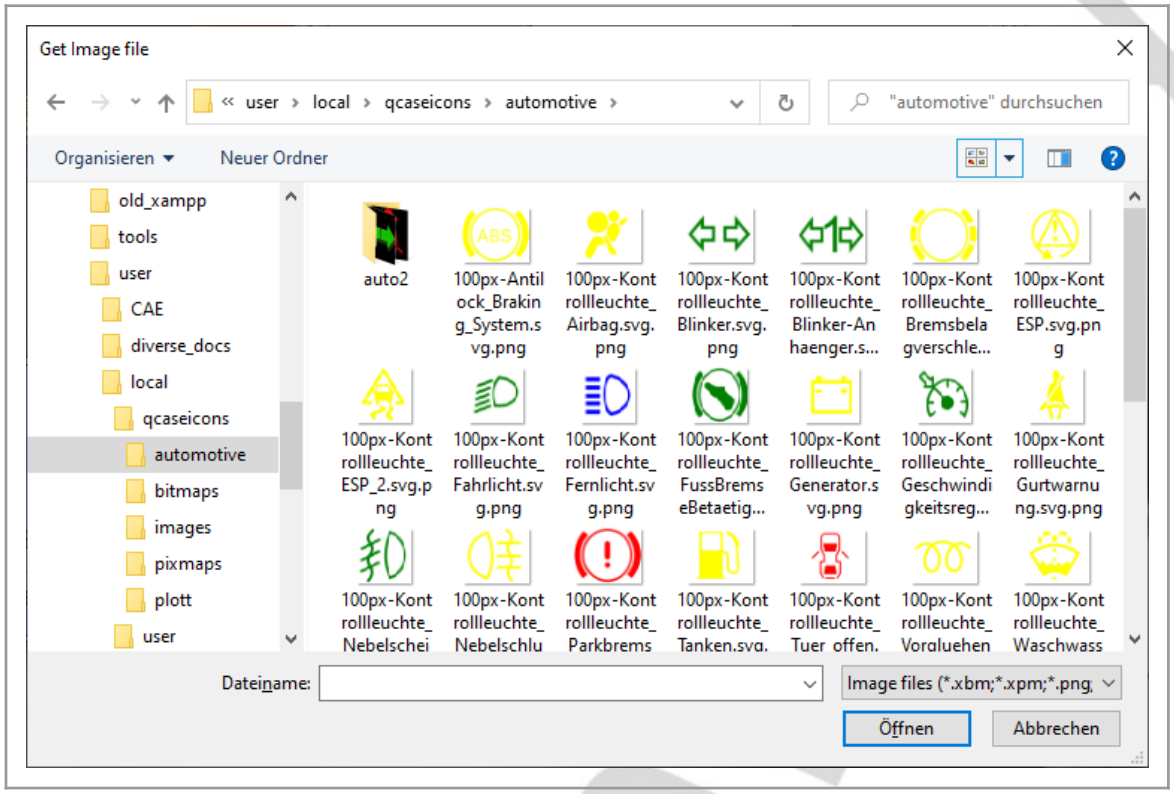
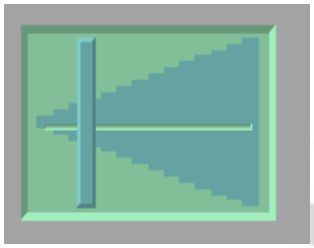


Abbildung 3.93: File Browser

Beispiele der Graphik Darstellung abhängig vom ausgewählten Bildformat.

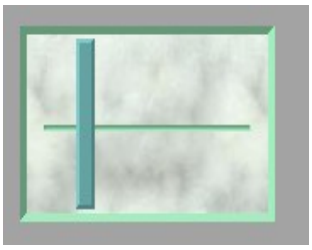


Background bitmap

Der Objekt Hintergrund wird in der 'field color' und die Bitmap in der 'text color' gezeichnet.

Background image

Das gesammte Objekt wird als Bild dargestellt.



Background muster

Das Objekt wird mit dem gekacheltem Image gefüllt.

Tabelle 6: Background Graphik

zeigt die Gruppenzugehörigkeit des Objekts an und kann durch das Betätigen des Button geändert werden (Abbildung 3.94). Die Gruppe hat keinen Einfluss auf die Eigenschaften des Objekts sondern dient lediglich der internen Kennzeichnung einer Zusammengehörigkeit verschiedener Elemente welche im 'Group Menü' gemeinsam bearbeitet werden können.

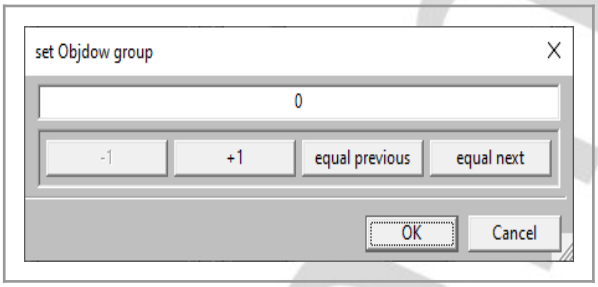
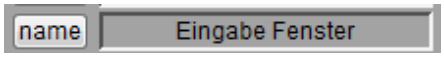
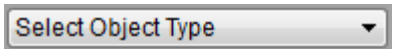


Abbildung 3.94: Group setting



Ebenso wie die Gruppe hat auch der Objekt Name keinen Einfluss auf die Eigenschaften. Er dient lediglich zur Dokumentation.



Erlaubt die Auswahl eines Objekt Typs. In der gezeigten Darstellung ist dem Objekt das erstellt wurde noch kein Typ zugewiesen worden. Nach dem Betätigen des Buttons erscheint ein Auswahlmenü der zur Zeit verfügbaren Objekte. (Abbildung 3.95)

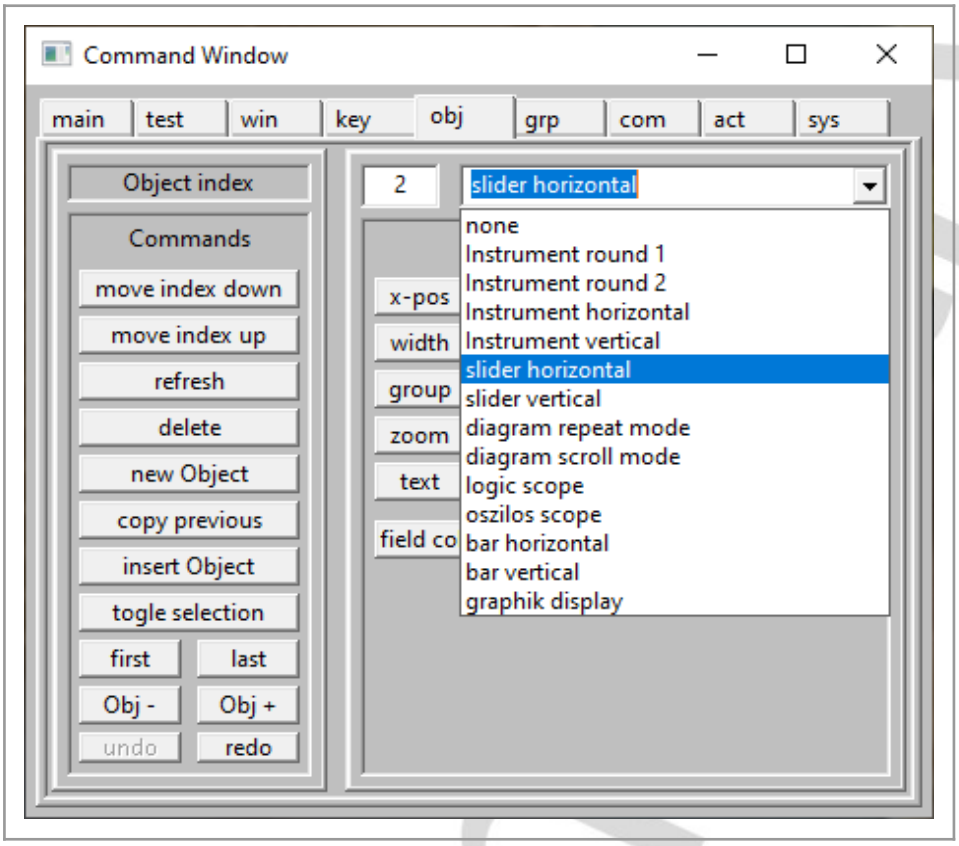


Abbildung 3.95: Objekt Auswahl Menü

Da es sich bei den Objekten um relativ komplexe Elemente handelt und diese immer mit einer Interaktion mit dem RPT-One Interface verbunden ist, sind zur Definition weitere Angaben erforderlich als bisher bei den Windows und den Keypicks besprochen. Diese zusätzlichen Attribute werden hier als Values bezeichnet. Abbildung 3.102 Skalen Parametereinstellung zeigt die Bereiche welche für diese Daten vorgesehen sind. Jeder Objekt Typ (im oberen Button) hat eine andere Anzahl und Bedeutung der Values.

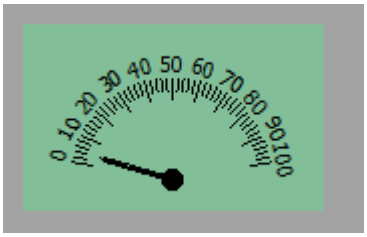
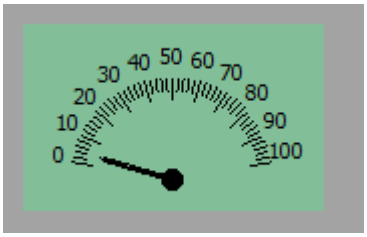
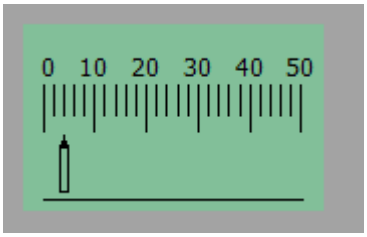
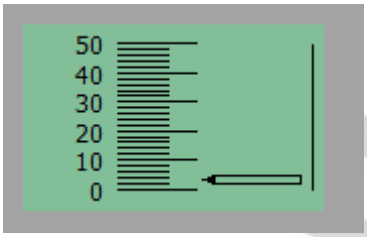
Das Betätigen des Button values öffnet den jeweiligen Parameter Dialog um die individuellen Werte für jedes Objekt zu setzen. Die Beschreibung der einzelnen Objekt Typen und deren Parameter folgt im anschließenden Kapitel.

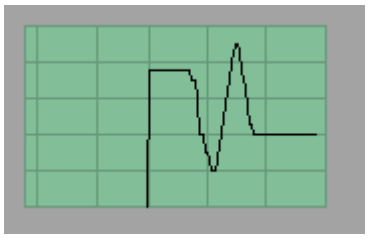
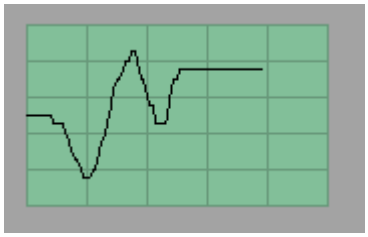
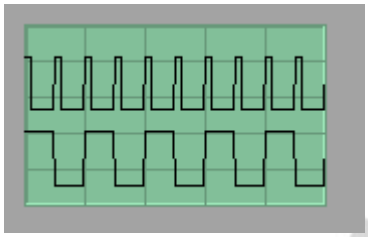
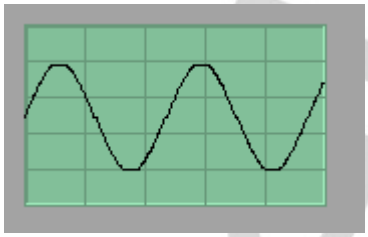
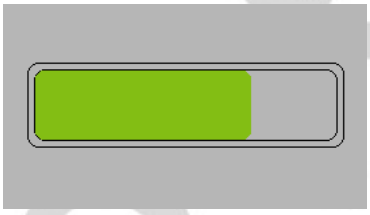
Für Diagramm Objekte ist das Setzen einiger Initialisierungswerte erforderliche, welche im folgendem Kapitel bei der Beschreibung der unterschiedlichen Objekt Typen ausführlich behandelt wird.

In Kapitel 3.6.3 Objekt Typen werden die verfügbaren Objekte tabellarisch aufgeführt und anschließend ausführlich behandelt.

3.6.3 Objekt Typen

In Tabelle 7 sind die Grundformen der Objekt Typen aufgelistet.

	Instrument round 1 entspricht einem rundem Zeigerinstrument bei dem die Beschriftung senkrecht zu den Skalenstrichen angebracht ist.
	Instrument round 2 entspricht einem rundem Zeigerinstrument bei dem die Beschriftung der Skala waagerecht angebracht ist.
	Instrument horizontal entspricht einem horizontalem Zeigerinstrument
	Instrument vertikal entspricht einem vertikalen Zeigerinstrument
	slider horizontal

	slider vertical
	diagram repeat mode Datendisplay bei dem jeder neue wert rechts eingefügt wird und die bereits dargestellten werte nach links verschoben werden.
	diagram scroll mode Datendisplay bei dem die daten von links startend ausgegeben werden. Nach erreichen des rechten Randes wird das Display gelöscht und ein neuer Zyklus von links gestartet.
	logic scope stellt ein Display für Digitale Signale in Form eines Logik Analysators dar.
	oszilos scope stellt ein Display für Analoge Signale in Form eines Oszilloskops dar.
	Bar horizontal

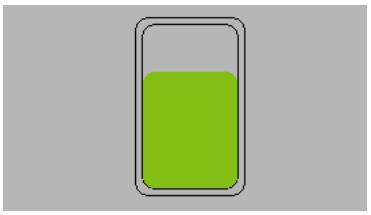
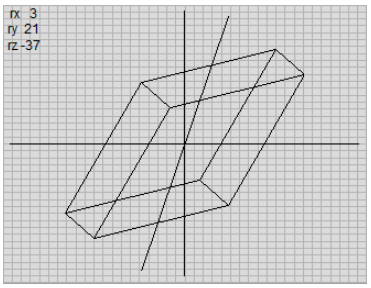
	Bar vertikal
	Graphik display stellt Vektorpaare als Graphik dar.

Tabelle 7: Grundformen der Objekt Typen

3.6.3.1 Instrument round 1 und round 2

Zeigerinstrument Typ 1 (Abbildung 3.96) entspricht einem rundem Instrument bei dem die Beschriftung der Skala senkrecht zu den Skalenstrichen angebracht ist.

Zeigerinstrument Typ 2 (Abbildung 3.97) entspricht einem rundem Instrument bei dem die Beschriftung der Skala waagerecht angebracht ist.

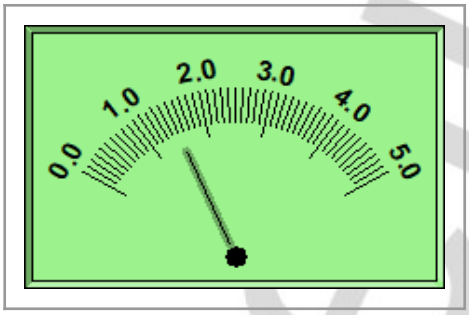


Abbildung 3.96: Zeigerinstrument Typ 1

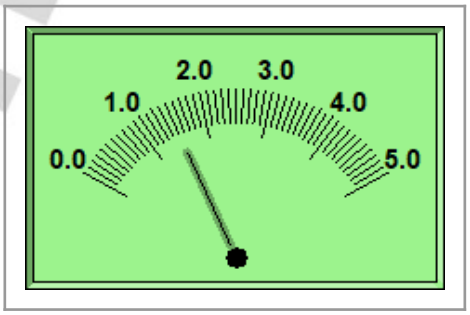


Abbildung 3.97: Zeigerinstrument Typ 2

Die Darstellung der Skala und weitere Parameter können nach dem Betätigen des Button values gesetzt werden. Zu der Darstellung in Abbildung 3.96 und Abbildung 3.97 gehört die in Abbildung 3.98 gezeigten Einstellungen der Werte.

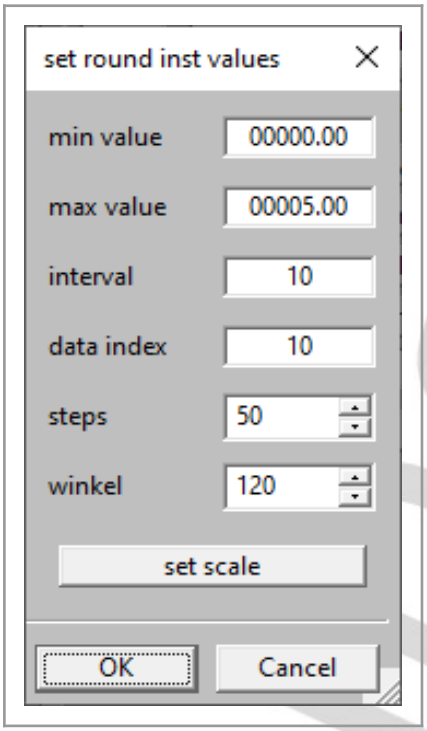


Abbildung 3.98: Beispiel Parametereinstellung

Die Bedeutung der einzelnen Werte ist wie folgt.

Stellt den Wert dar den der Zeiger in der linken Anfangsposition einnimmt.

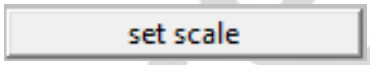
Stellt den Wert dar den der Zeiger in der rechten Endposition einnimmt.

Ist die Anzahl der Werte in die der Bereich Max – Min aufgeteilt wird.

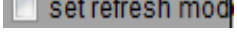
Ist der Index der variablen aus dem Interface welche den anzuzeigenden Wert enthält.

Ist die Anzahl der Einheiten der Skala von Min bis Max.

ist der Winkel in dem dem das Intervall abgebildet wird.

Der Button 

erlaubt das individuelle Einstellen der Erscheinung der Skala des Objekts.

Damit immer der aktuelle Wert der zugewiesenen Interface Variablen angezeigt wird, muss auch hier im Objekt Mode Dialog (Abbildung 3.89) die Option 

gesetzt sein.

3.6.3.2 Instrument horizontal und vertikal

Abbildung 3.99 entspricht einem Instrument mit horizontaler Ausrichtung.

Abbildung 3.100 entspricht einem Instrument mit vertikaler Ausrichtung.

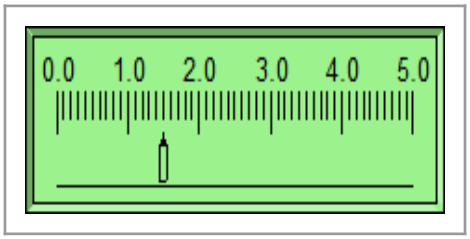


Abbildung 3.99: Horizontal Instrument

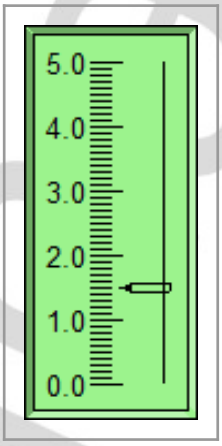


Abbildung 3.100: Vertikal Instrument

Die Darstellung der Skala und weitere Parameter können nach dem Betätigen des Button values gesetzt werden. Zu der Darstellung in Abbildung 3.99 und Abbildung 3.100 gehört die in Abbildung 3.101 gezeigten Einstellungen der Values.

Wird im Objekt Mode Dialog (Abbildung 3.89) die Option activate obj gesetzt, kann der Inhalt der dem Instrument zugewiesenen Variablen durch Verschieben mit dem Mauscursor bei gedrückter linker Maustaste geändert werden. Dieses wird durch den gefüllten Zeiger angezeigt.

Wie schon zuvor erwähnt, damit immer der aktuelle Wert der zugewiesenen Interface Variablen angezeigt wird, muss auch hier im Objekt Mode Dialog (Abbildung 3.89) die Option set refresh mod gesetzt sein.

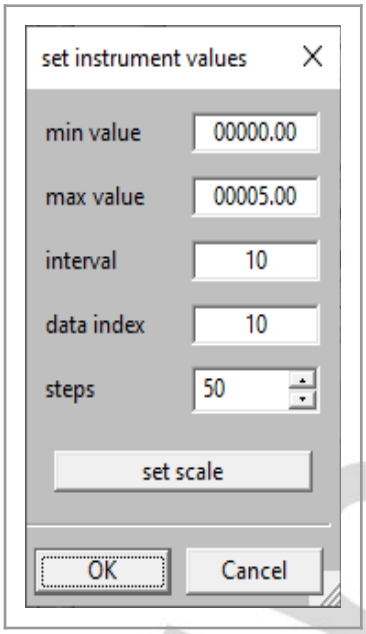


Abbildung 3.101: Beispiel Parametereinstellung

Die Bedeutung der einzelnen Werte ist wie folgt.

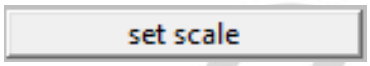
Stellt den Wert dar den der Zeiger in der linken Anfangsposition einnimmt.

Stellt den Wert dar den der Zeiger in der rechten Endposition einnimmt.

Ist die Anzahl der Werte in die der Bereich Max – Min aufgeteilt wird.

Ist der Index der variablen aus dem Interface welche den anzuzeigenden Wert enthält.

Ist die Anzahl der Einheiten der Skala von Min bis Max.

Der Button 

erlaubt das individuelle Einstellen der Erscheinung der Skala des Objekts.

3.6.3.2.1 Skalen Parametereinstellung

Mit Hilfe der Parametereinstellung kann man eine individuelle Gestaltung einer Skala vornehmen, das heißt alle Werte können den eigenen Vorstellungen entsprechend verändert werden.

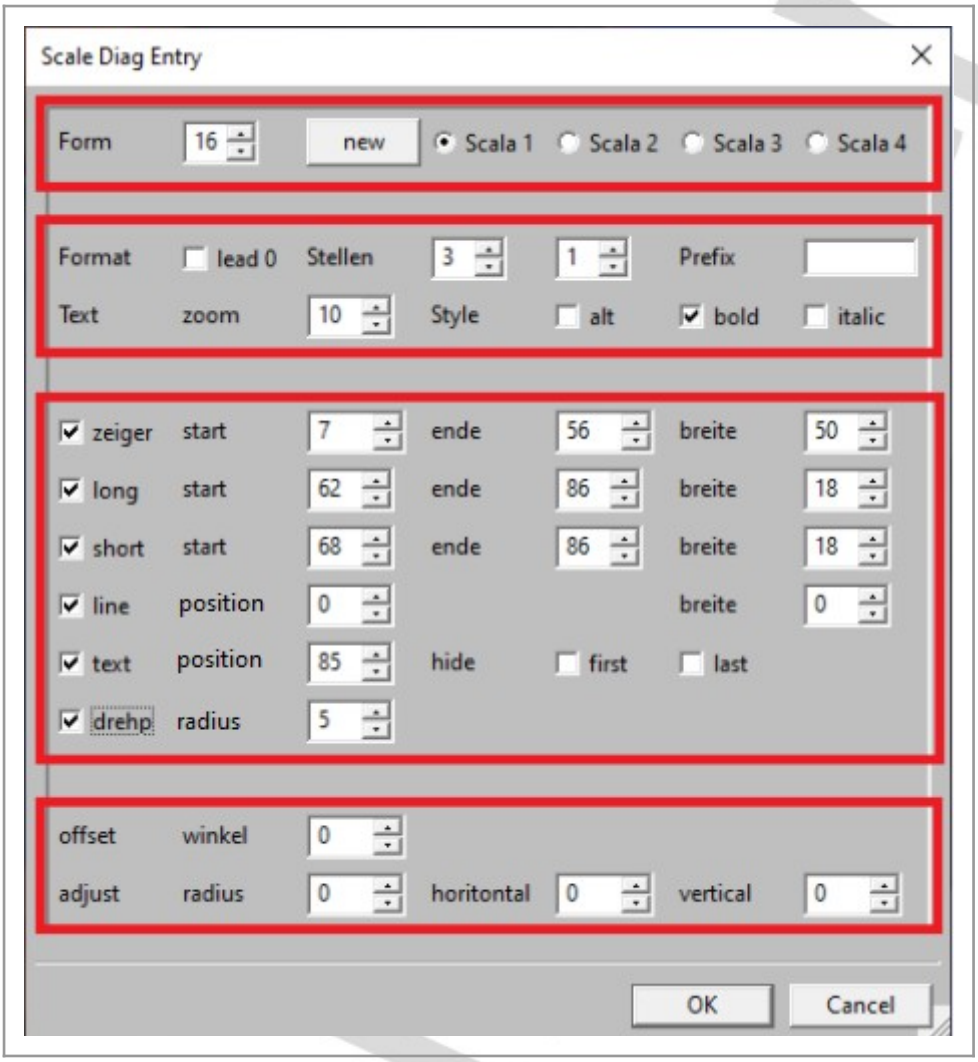


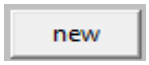
Abbildung 3.102: Skalen Parametereinstellung

Das ‚Scale Diag Entry' Fenster ist in 4 Bereiche aufgeteilt:

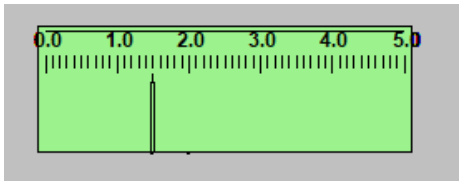
1. Der Form Bereich der die Form der Skala und die Skaleneinteilung bestimmt.
2. Der Format Bereich bestimmt das Aussehen der Skalen Beschriftung.
3. Der dritte Bereich dient zur Einstellung der graphischen Erscheinung der Skala.
4. Der vierte Bereich erlaubt die Feinjustierung des Erscheinungsbildes der Skala.

3.6.3.2.1.1 Form Bereich

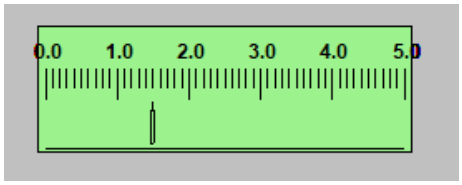
Abbildung 3.103 Skalen Form

Die Formen 0 bis 15 sind fest definiert, so das die Einstellung der graphischen Erscheinung der Skala als auch die Feinjustierung hier nicht möglich sind. Um eine neue Skalen Definition zu erstellen wählt man eine Form aus welche dem gewünschten Aussehen am nächstem kommt. Durch den Button 

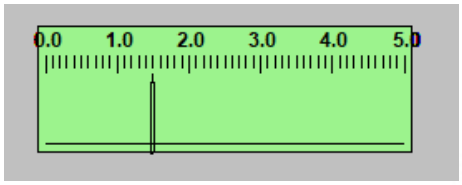
wird eine neue Form angelegt, welche Daten der Ausgangsform enthält. Diese können dann individuell angepasst werden. Achtung !! Beim Speicher der Applikation im Text Format gehen die Informationen von Skalen Definitionen welche keinem Objekt zugeordnet sind verloren. Um nicht benutzte Skalen weiterverwenden zu können muss das Projekt als Binär Datei abgespeichert werden.



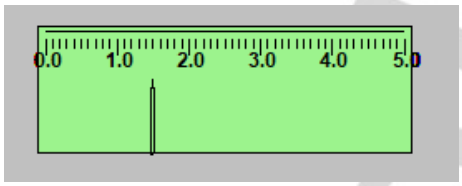
Form 0



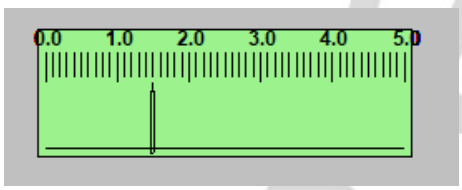
Form 1



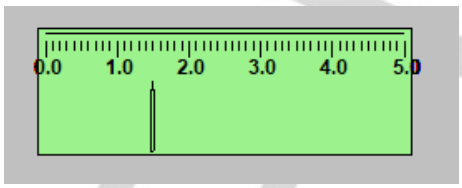
Form 2



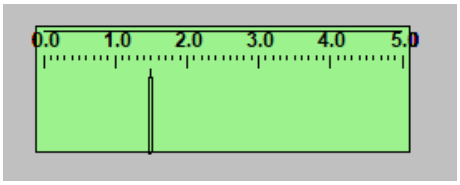
Form 3



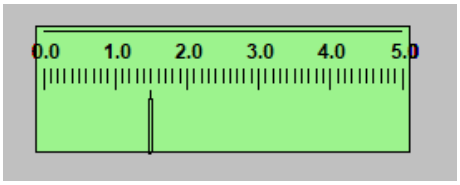
Form 4



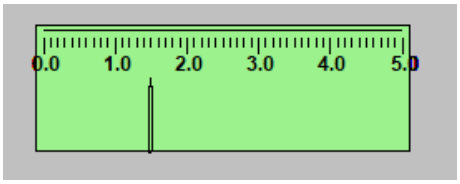
Form 5



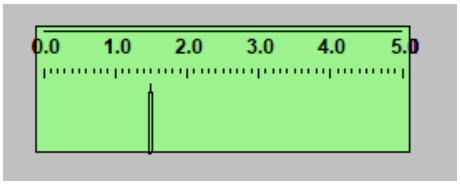
Form 6



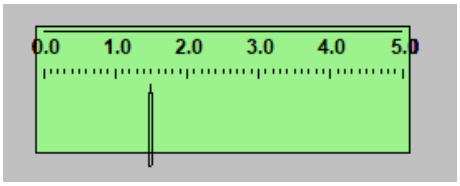
Form 7



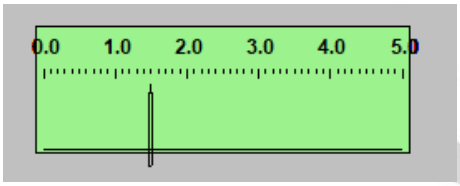
Form 8



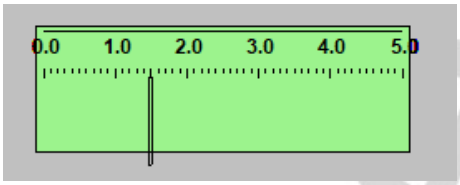
Form 9



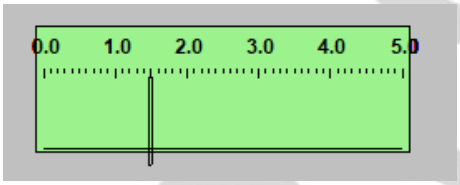
Form 10



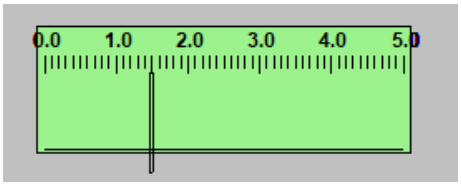
Form 11



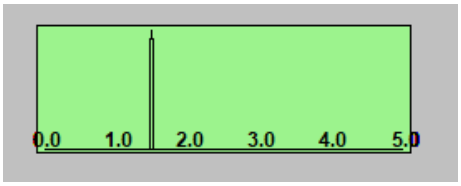
Form 12



Form 13



Form 14



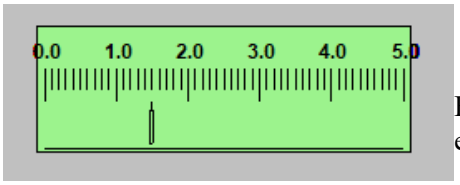
Form 15

Tabelle 8: Vordefinierte Skalen Formen

Mit der Skalierungs Einteilung können 4 unterschiedliche Aufteilungen der Scalierung gewählt werden.

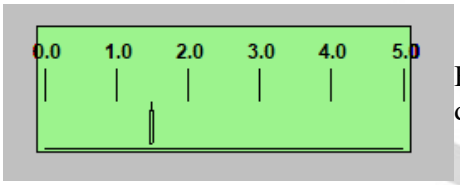
Abbildung 3.104 Skalierungs Einteilung

Die Unterschiede dieser Werte sind in Tabelle 9: Skalierungs Einteilung dargestellt.



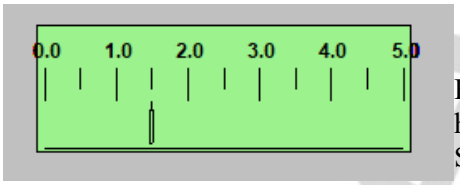
Skala 1

Es werden alle steps angezeigt, wobei alle 10 er Werte durch einen langen Skalenstrich hervorgehoben werden.



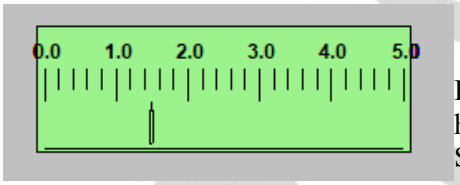
Skala 2

Es werden alle 10 er Werte durch einen langen Skalenstrich dargestellt.



Skala 3

Es werden alle 10 er Werte durch einen langen Skalenstrich hervorgehoben alle 5 er Werte werden durch einen kurzen Strich dargestellt..



Skala 4

Es werden alle 10 er Werte durch einen langen Skalenstrich hervorgehoben alle 2 er Werte werden durch einen kurzen Strich dargestellt..

Tabelle 9: Skalierungs Einteilung

3.6.3.2.1.2 Format Bereich

Hier kann man das Format der Skalen Beschriftung einstellen.

Abbildung 3.105: Skala Text Formatierung

Ist lead 0 gesetzt, werden führende Nullen in der Beschriftung der Skala angezeigt.

Die Darstellung der Beschriftung erfolgt immer im Foating Point Format.

Stellen 3 1 der erste Wert ist die mindest Anzahl der Ziffer einschließlich des Dezimalpunktes, wenn der zweitee Wert größer 0 ist.

Im Prefix kann ein Text eingefügt werden, welcher an die einzelnen Skalenziffern angefügt wird.

Die Text Größe sowie der Text Font kann hier alternativ zum Objekt Zoom Menü eingestellt werden.

Die Größe der Textdarstellung in den vorab beschriebenen Optionen kann mit dem Button zoom im Ojekt Zoom Dialog (Abbildung 3.106) geändert werden.

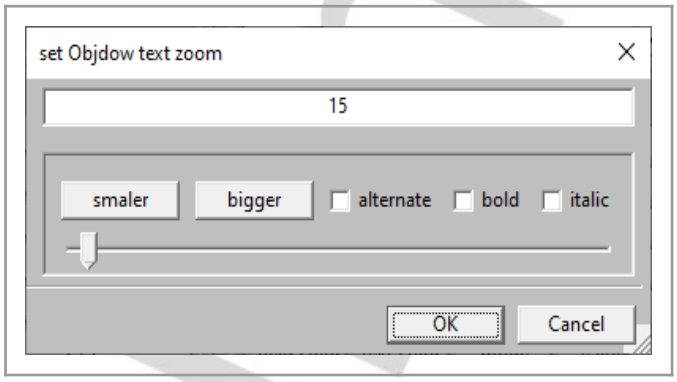


Abbildung 3.106: Ojekt Zoom Dialog

Die Größe der Textanzeige kann über die Buttons als auch über den Slider auf das gewünschte Maß gesetzt werden.

alternate schaltet auf App font 2 um. Siehe Seite 19 Kapitel 3.2.3.2 Konfiguration Menü. Abbildung 3.17: Font Selectionn

bold stellt den Text in Fettschrift dar.

italic stellt den Text in Schrägschrift dar.

3.6.3.2.1.3 Einstellung der graphischen Erscheinung der Skala.

In diesem Dialog Abbildung 3.108 können alle Werte welche das Aussehen der Skala bestimmen individuell angepasst werden. Abbildung 3.107: Scalen Definition zeigt den grundsätzlichen Aufbau einer solchen Skala.und deren Elemente am Beispiel eines horizontalen Instrumentes.

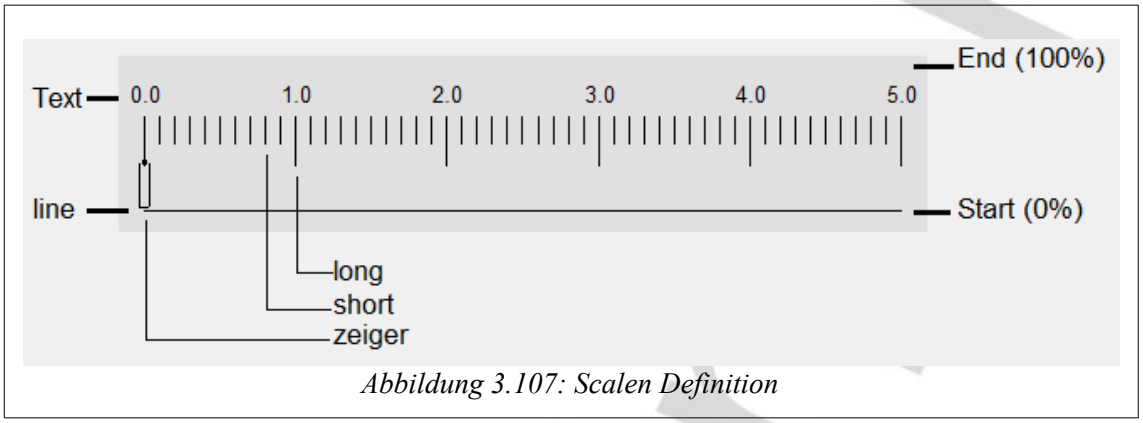


Abbildung 3.107: Scalen Definition

Die Werte beziehen sich bei einen horizontalen Instrument auf die Höhe des Objekts abzüglich eines internen Offsets als Prozentwert. Das heißt, das am Beispiel des ‘long‘ Wertes die Darstellung bei 34% beginnt und bei 72% endet, wobei die Strichbreite auf 1 Pixel gesetzt ist (intern wird der Wert 0 für die Breite als 1 behandelt). Durch das aktivieren der Checkbox vor dem entsprechendem Skalen Element bestimmt man ob dieses Element dargestellt wird. Das Beispiel ☑ long bedeutet, das der lange Skalenstrich gezeichnet wird. Die Darstellung des Drehpunktes (drehp) findet nur bei den Runden Skalen Anwendung. Für line und text gibt es nur eine position an Stelle von start und end.

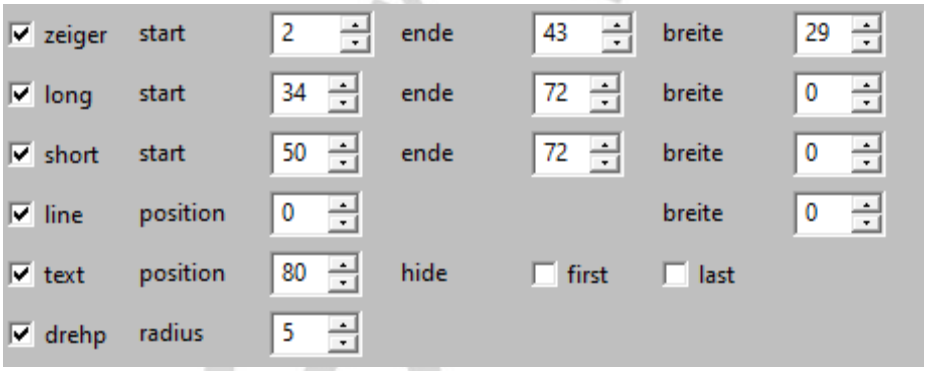
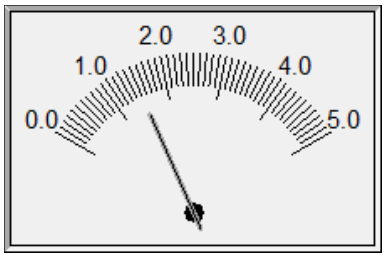
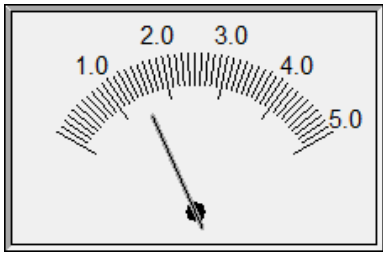


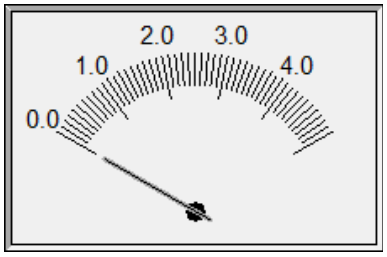
Abbildung 3.108: Graphische Skalen Einstellung

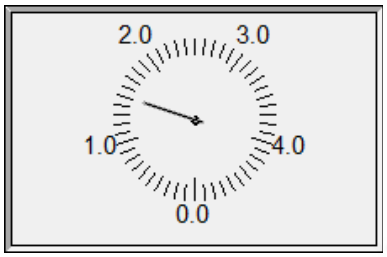
Dem Setting von hide ☐ first ☐ last bei den text Eigenschaften kommt eine besondere Bedeutung zu, welche im folgenden näher Erläutert wird.

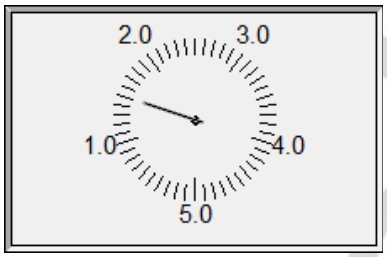


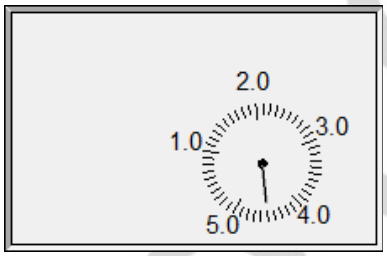
Die nebenstehende Darstellung zeigt ein normales Rundinstrument dar. Sowohl der erste als auch der letzte Skalenwert wird angezeigt.











hide ☑ first ☐ last

Wie der Begriff hide sagt, wird hier der erste Skalenwert versteckt, was zur nebenstehende Darstellung führt.

hide ☐ first ☑ last

Hier wird der letzte Skalenwert versteckt, was zur nebenstehende Darstellung führt.

hide ☐ first ☐ last

Der Sinn dieser Einstellungen wird klar, wenn maqn sich das nebenstehende Bild ansieht. Da hier der Winkel des Instruments auf 360 grad gesetzt wurde wird der erste Skalen Wert durch den Letzten überschrieben, was zu einer nicht leserlichen Darstellung führt.

hide ☐ first ☑ last

Durch das unterdrücken des letzten Werte erhält man die nebenstehende Ausgabe.

hide ☑ first ☐ last

Durch das unterdrücken des ersten Werte erhält man die nebenstehende Ausgabe.

hide ☑ first ☐ last

offset winkel 30

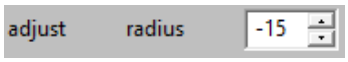
adjust radius -15 horitontal 40 vertical 20

Das nebenstehende Bild zeig die Auswirkungen der restlichen Einstellungen der Skala.

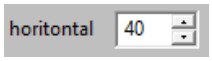
offset winkel 30

Addiert auf alle Werte eine

Winkeloffset. In diesem Beispiel 30 Graqd.

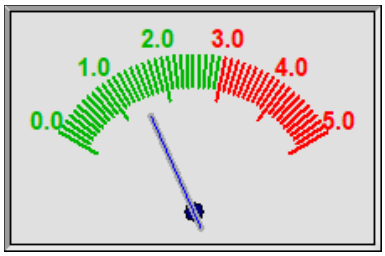


Skaliert den Radius der Skala in diesem Fall um 15%.

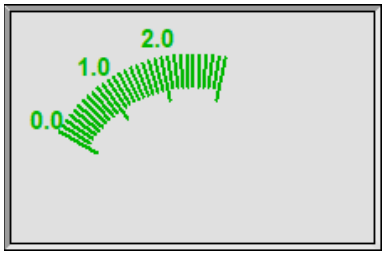


Verschiebt die Skala in horizontaler Richtung.

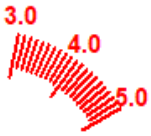
Verschiebt die Skala in vertikalerRichtung



Mit Hilfe dieser Einstellungen, und der Möglichkeit mehrere Skalen übereinander legen zu können, ist es möglich auch komplexere Darstellungen zu erreichen, wie zum Beisiel eine mehrfarbige Skala..

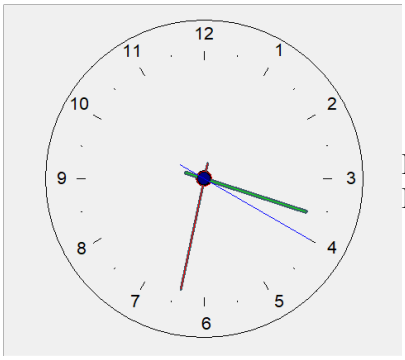


Die erste Skala stellt den Hintergrund und den erste Skalen Abschnitt dar.

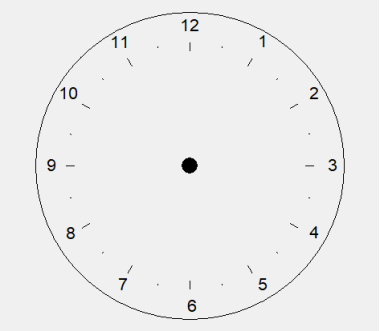


Die zweite Skala stellt den restlichen erste Skalen Abschnitt dar.

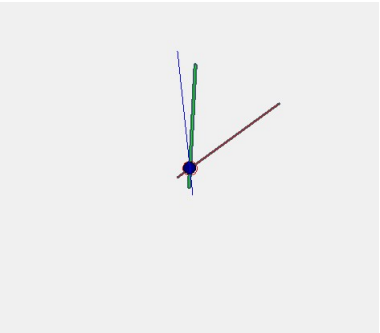

Der letzten Skala wird die Zeigerdarstellung zugeordnet.



Mit den oben genannten Einstellungen ist es möglich die Darstellung einer analogen Uhr zu erstellen.



Die erste Skala wird durch ein round Instrument erstellt das lediglich die Zeiteinteilung zeigt.



Drei w4eitere Skalen bilden den Stundenzeiger, den Minutenzeiger und den Sekundenzeiger.

3.6.3.3 Slider horizontal und vertikal

Slider horizontal (Abbildung 3.109) entspricht einem horizontalem Schieberegler ohne Beschriftung.

Slider vertikal (Abbildung 3.110) entspricht einem vertikalem Schieberegler ohne Beschriftung.

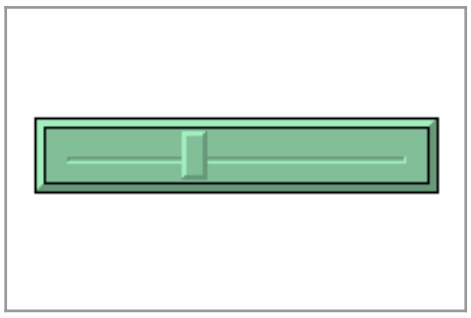


Abbildung 3.109: Slider horizontal

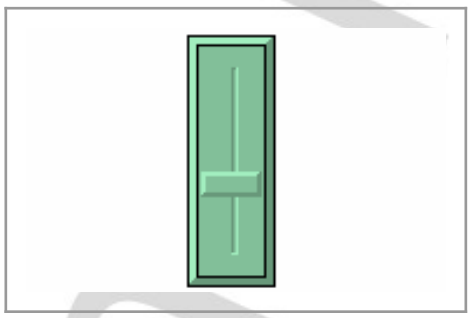


Abbildung 3.110: Slider vertikal

Die Parameter können nach dem Betätigen des Button values gesetzt werden. Zu der Darstellung in Abbildung 3.109 und Abbildung 3.110 gehört die in Abbildung 3.111 gezeigten Einstellungen der Values.

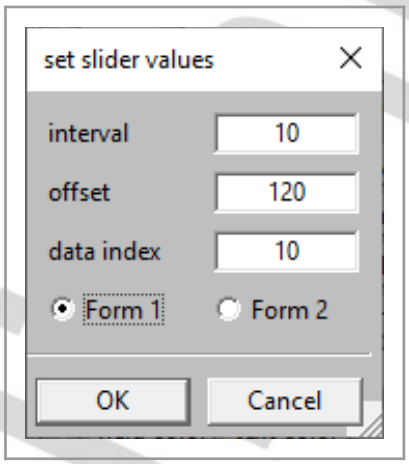
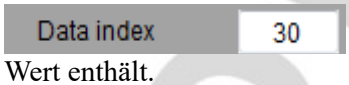


Abbildung 3.111: Beispiel Parametereinstellung

Die Bedeutung der einzelnen Werte ist wie folgt.

Ist die Anzahl der Werte die der Slider durchläuft.

Ist der Startwert des Sliders. Ist hier statt 0 ein Wert von 20 eingesetzt, ist der Wertebereich 20 bis 275 .



Ist der Index der variablen aus dem Interface welche den eingestellten Wert enthält.

Damit der Slider als Eingabe genutzt werden kann und die Schieberstellung dem aktuelle Wert der zugewiesenen Interface Variablen folgt, müssen im Objekt Mode Dialog (Abbildung 3.89) die

Optionen pick aktivieren und set refresh mod gesetzt sein.

Wird nur die Option set refresh mod gesetzt ist keine Eingabe möglich und der Schieber folgt lediglich den Werten in der Variablen.

Form 1 Form 2 Wird der Button für Form 2 gesetzt, erfolgt die unten gezeigte alternative Darstellung der Slider.

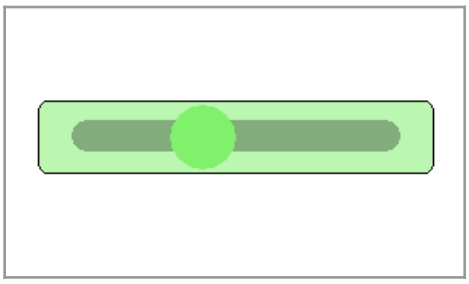


Abbildung 3.112: Slider horizontal

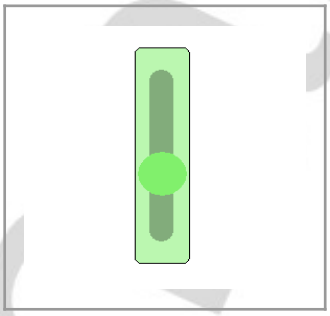


Abbildung 3.113: Slider vertikal

Eine weitere Option der Darstellung ist die Zuordnung einer Graphik als Zeiger des Sliders. Hierzu wird zusätzlich zur ausgewählten Graphik die Option zeiger image ausgewählt. Die Graphik wird jetzt nicht mehr dem Hintergrund des Objekts, sondern dem Zeiger des Sliders zugeordnet.

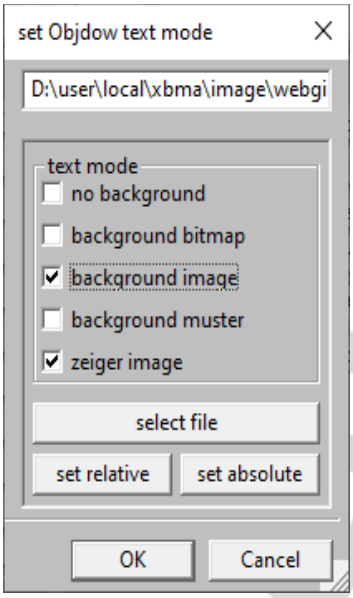


Abbildung 3.114: Objekt Textmode

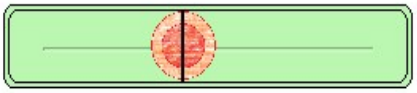


Abbildung 3.115: Slider mit Graphik Zeiger

Wobei hier im mode x-adjust und y-adjust gesetzt sein sollten, um im zoom Menue die Größe der Graphik anpassen zu können. Die Kombination von Background muster und zeiger image macht hier jedoch keinen Sinn. Abbildung 3.115: Slider mit Graphik Zeiger zeigt das Ergebnis.

3.6.3.4 Diagramm Repeat

Der Objekt Type Diagramm Repeat stellt eine rechteckige Displayfläche zur Anzeige von Kurvenverläufen dar. Die Darstellung der Werte erfolgt von Links nach Rechts. Ist der rechte Rand erreicht, startet der nächste Zyklus wieder am linken Rand. Für die Darstellung stehen 3 verschiedene Display Mode zur Verfügung.

Einstellung 

in Abbildung 3.122: ParameterEinstellung Set.

Bei Display Mode Repeat 1 Abbildung 3.116 wird das Display nach dem Beenden eines Zyklus gelöscht und die Darstellung der Werte wird von links beginnend fortgesetzt.

Bei Display Mode Repeat 2 Abbildung 3.117 wird das Display nach dem Beenden eines Zyklus nicht gelöscht sondern die Darstellung der Werte wird von links beginnend fortgesetzt, wobei die neuen Werte den vorherigen Zyklus überschreiben.

Der Display Mode Repeat 3 Abbildung 3.118 unterscheidet sich von Repeat 2 lediglich durch das Einblenden einer senkrechten Cursorlinie an der aktuellen Kurvenposition.

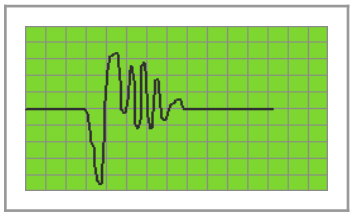


Abbildung 3.116: Repeat 1

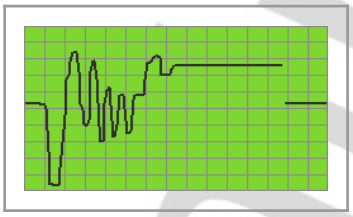


Abbildung 3.117: Repeat 2

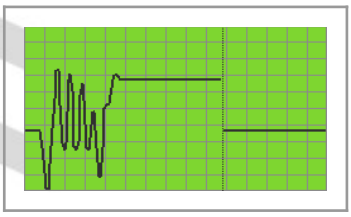


Abbildung 3.118: Repeat 3

Bei diesen Objekten sind außer dem refresh keine Mode Einstellungen möglich. Deshalb erscheint beim Aufruf des Mode Dialogs das folgende Bild. (Abbildung 3.119)

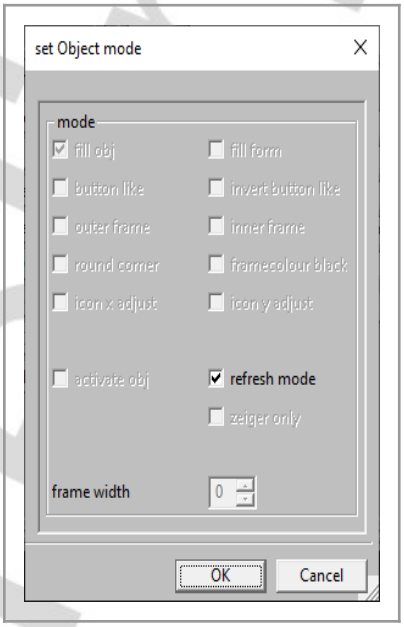


Abbildung 3.119: Diagramm Mode Dialog

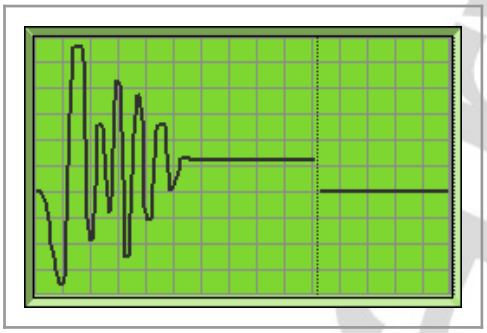


Abbildung 3.120: Diagram inv Button

Um Eine Darstellung wie in Abbildung 3.120 zu erhalten, muss dem Diagrammfeld ein Window mit dem gewünschten Aussehen unterlegt werden.

Die Eingabe der Parameter des Diagramms erfolgt nach der Betätigung des Button values im Parameter Dialog. Das Beispiel Abbildung 3.122 zeigt die Werte für die Abbildung 3.120 welche im folgenden erläutert werden.

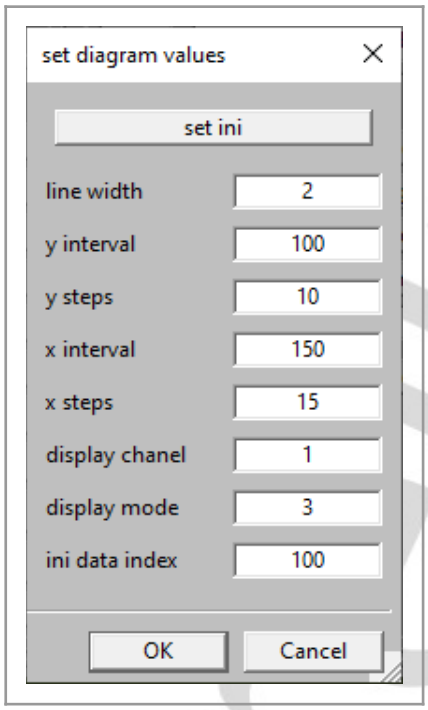


Abbildung 3.122: ParameterEinstellung Set

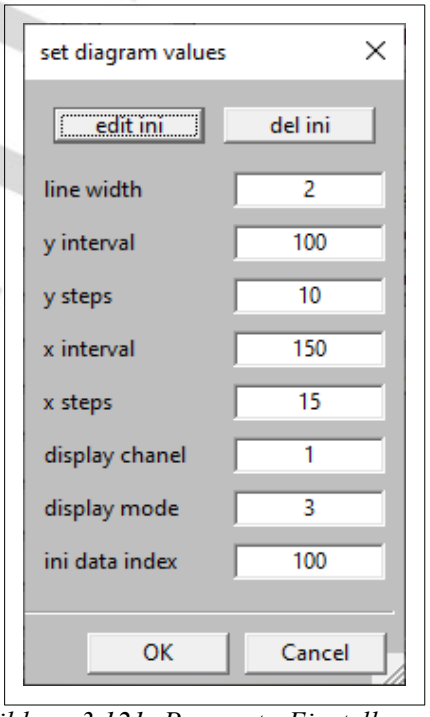
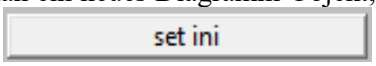


Abbildung 3.121: ParameterEinstellung Edit

Da für Diagramm Objekte auf Werte im Interface zugegriffen wird, die das Erscheinungsbild des Diagramms beeinflussen, sind hier weitere Eingaben zwingend erforderlich. Diese Werte stellen eine Initialisierung des Objekts dar, welche beim Applikationsstart in das Interface geladen werden. Sind diese Werte nicht gesetzt, kann es zu einem Programmabsturz führen. Mit der Veränderung der Werte

können die Parameter des Diagramms während der Laufzeit gesteuert werden. Das Verändern der Werte kann aus der Applikation heraus erfolgen als auch durch eine externe User Funktion. Das setzen der Startwerte erfolgt im Fehler: Verweis nicht gefunden. Für das Beispiel aus Abbildung 3.120 sind hier die Werte in Fehler: Verweis nicht gefunden dargestellt.

Erstellt man ein neues Diagramm Objekt, wird im oberen Teil des Parameter Dialogs ein ini Button in der Form 

dargestellt, um anzuzeigen das für dieses Objekt noch keine Initialisierungswerte erstellt wurden. ***Achtung !*** *Wird keine Initialisierung erstellt, werden die Werte aus dem Interface benutzt auf welche* 

zeigt. Wird jetzt ein Test gestartet, ist das Ergebnis nicht vorhersagbar, da diese Werte nicht definiert sind.

Bei einem bereits komplett erstelltem Objekt erscheint 

wodurch das Ändern oder Löschen der Werte ermöglicht wird.

Mit 

wird die Stärke der Kurvenlinie bestimmt.

Ist die Anzahl der Werte die in der Vertikalen des Diagramms dargestellt werden.

Ist die Anzahl der vertikalen Rasteraufteilung.

Ist die Anzahl der Werte die in der Horizontalen des Diagramms dargestellt werden.

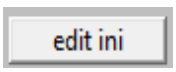
Ist die Anzahl der horizontalen Rasteraufteilung.

Ist die Anzahl der dargestellten Kurven. (maximal 4)

Bestimmt die Art der Kurvendarstellung wie bereits am Anfang des Kapitels beschrieben. (Abbildung 3.116: Repeat 1, Abbildung 3.117: Repeat 2 und Abbildung 3.118: Repeat 3)

Ist der Index der Basis Variablen aus dem Interface welche dem Objekt zugeordnet werden. Es werden 20 Variablen belegt deren Bedeutungen in Tabelle 10 Repeat Diagramm Interface Definitionen beschrieben werden.

Die Button 

b.z.w 

öffnen den Objekt Init Dialog aus Abbildung 3.123 mit dem die Initialisierungswerte aus Tabelle 10 gesetzt werden können. Vor den Eingabefeldern wird jeweils der Index des Interfaces angezeigt in den die Werte geschrieben werden. Mit dem Button  wird ein Color Dialog zur Auswahl der jeweiligen Kurvenfarbe geöffnet.

Die hier erstellten Daten werden bei jedem Programmstart in den Interface Bereich geladen. Danach können die Werte dynamisch durch internen als auch externem Zugriff verändert werden

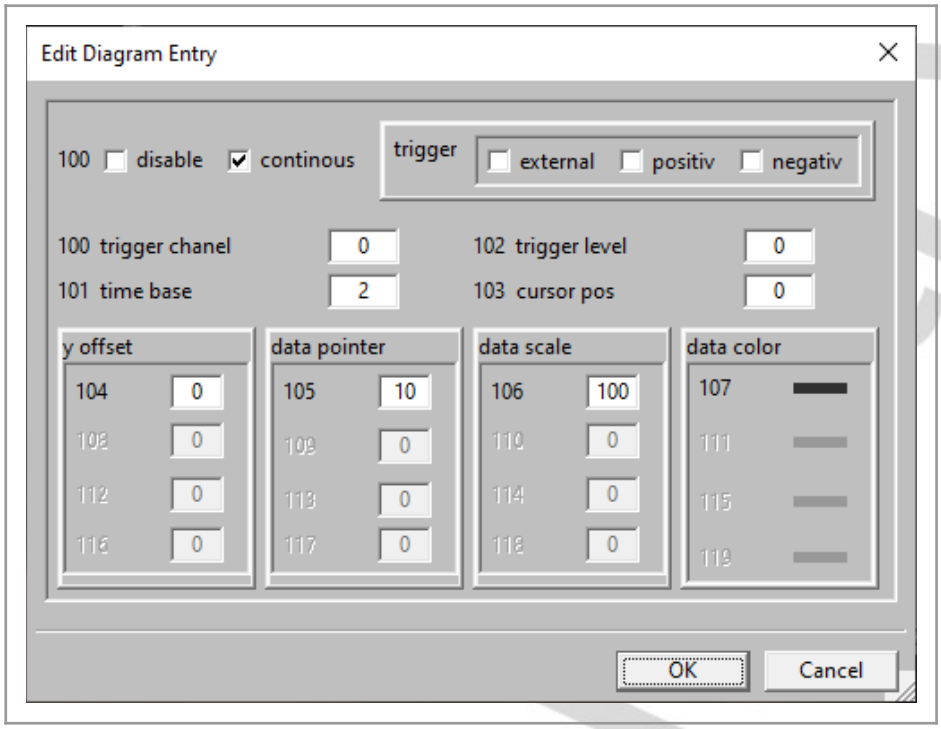


Abbildung 3.123: Objekt Init Dialog

Index Offset	Bedeutung	
0 Bits 0 bis 7	Kontroll Bits	Die Kontroll Bits diehnen der Steuerung des Diagramms im Run Mode. In Tabelle 11 wird die Bedeutung der Kontroll Bits erläutert.
0 Bits 8 bis 11	Trigger Kanal	Der Triggerkanal bestimmt auf welchen Eingangswert die Triggerbedingung angewendet werden soll.
1	Faktor Sample Zeit	Die Eingangswerte des Diagramms werden im Run Mode alle xx Millisekunden eingelesen. Diese Zeit wird mit dem Faktor multiplieziert, um zum Beispiel auch Werte anzuzeigen die sich nur sehr langsam ändern.

2	Trigger_Level	Sind in den Kontroll Bits 'Trigger Positiv' oder 'Trigger Negativ' gesetzt, und der 'Trigger_Level' durch den im 'Trigger Kanal' gewählten Eingangswert über b.z.w unterschritten, wird die Ausgabe gestartet b.z.w gestoppt falls es sich um ein Scroll Diagramm handelt. Der Wert des 'Trigger_Level' muss im Bereich des im 'Parameter Dialog' gesetztem 'Y Intervalls' liegen.
3	Cursor Position	Legt die Position einer vertikalen Cursorlinie im Oszidiagram und Logikdiagram fest.
4	Werte Offset `Kanal` 1	Verschiebt die Kurve des Eingangswertes um Offset in der Darstellung. Auch dieser Wert muss im Bereich des gesetztem 'Y Intervalls' liegen.
5	Daten Index `Kanal` 1	Enthält den Index des Eingangswertes der diesem Kanal zugeordnet ist. Die Werte der Daten auf die der Index verweist müssen im Bereich des gesetzten 'Y Intervalls' liegen.
6	Daten Faktor Kanal 1	Skaliert die Darstellung der Eingangs-werte in Prozent.
7	Kurven Farbe Kanal 1	Legt die Farbe der Kurve für diesen Kanal fest.
8	Werte Offset `Kanal` 2	Wie 'Werte Offset `Kanal` 1'
9	Daten Index `Kanal` 2	Wie 'Daten Index `Kanal` 1'
10	Daten Faktor Kanal 2	Wie 'Daten Faktor Kanal 1'
11	Kurven Farbe Kanal 2	Wie 'Kurven Farbe Kanal 1'
12	Werte Offset `Kanal` 3	Wie 'Werte Offset `Kanal` 1'
13	Daten Index `Kanal` 3	Wie 'Daten Index `Kanal` 1'
14	Daten Faktor Kanal 3	Wie 'Daten Faktor Kanal 1'
15	Kurven Farbe Kanal 3	Wie 'Kurven Farbe Kanal 1'
16	Werte Offset `Kanal` 4	Wie 'Werte Offset `Kanal` 1'
17	Daten Index `Kanal` 4	Wie 'Daten Index `Kanal` 1'

18	Daten Faktor Kanal 4	Wie 'Daten Faktor Kanal 1'
19	Kurven Farbe Kanal 4	Wie 'Kurven Farbe Kanal 1'

Tabelle 10: Repeat Diagramm Interface Definitionen

xx 0000	Disabled	Sperrt den Start einer Ausgabe der Kurve. Wird dieser Mode gesetzt während eine Ausgabe läuft, so wird diese zu Ende gebracht und ein weiterer Start gesperrt.
xx 0001	Conntinnous Run	Die Ausgabe wird unabhängig von den Triggerbedingungen gestartet und kontinuierlich wiederholt.
xx 0010	External Trigger	Es wird eine Ausgabe gestartet wenn das External Trigger Start Bit gesetzt wird. Während eine Ausgabe läuft wird das Trigger Bit ignoriert.
xx 0100	Trigger Positiv	Es wird eine Ausgabe gestartet wenn der Eingangswert des gewählten Trigger Kanals den Wert des Trigger_Levels überschreitet. Während eine Ausgabe läuft wird der Trigger ignoriert.
xx 1000	Trigger Negativ	Es wird eine Ausgabe gestartet wenn der Eingangswert des gewählten Trigger Kanals den Wert des Trigger_Levels unterschreitet. Während eine Ausgabe läuft wird der Trigger ignoriert.
01 0010	External Trigger Start	Es wird eine Ausgabe gestartet wenn das Bit im Eexternal Trigger Mode gesetzt wird und keine Ausgabe läuft. Nach dem Start einer Ausgabe wird das Bit intern gelöscht.
1x xxxx	Retrigger Start / One Shot	Das Setzen des Bits startet unabhängig vom gesetzten Mode eine Ausgabe. Wird das Bit gesetzt während eine Ausgabe läuft, wird diese unterbrochen und sofort eine neue Ausgabe gestartet. Nach dem Start einer Ausgabe wird das Bit intern gelöscht.

Tabelle 11: Bedeutung der Kontroll Bits

3.6.3.5 Diagramm scroll Mode

Der Objekt Type Diagramm Repeat stellt ebenfalls eine rechteckige Displayfläche zur Anzeige von Kurvenverläufen dar, jedoch erfolgt die Darstellung hier von Rechts nach Links und wird bei jedem Einlesen eines neuen Wertes um eine Position nach Links verschoben. Des weiteren ist die Anzahl der horizontalen Kurvenpunkte gleich der Objektbreite und ein 'Trigger' stoppt die Ausgabe anstatt Sie zu starten. Da bis auf die oben beschriebenen Unterschiede für ein Scroll Diagramm die gleichen Einstellungen gelten, sind an dieser Stelle nur die abweichenden Eigenschaften beschrieben.

Abbildung 3.124 Scroll Diagramm zeigt als Beispiel den Bildaufbau eines Diagramms.

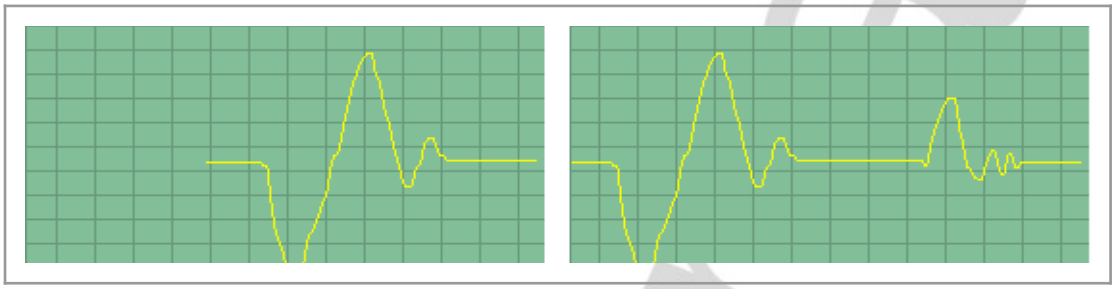
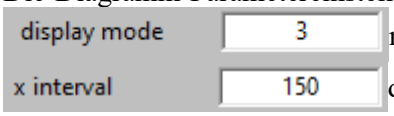


Abbildung 3.124: Scroll Diagramm

.Auch hier sind keine Mode Einstellungen möglich (siehe Abbildung 3.119 Diagramm Mode Dialog) und es muss ebenfalls ein Window mit dem gewünschten Aussehen unterlegt werden, wenn eine Darstellung wie in Abbildung 3.120: Diagram inv Button gewünscht ist.

Die Diagramm Parametereinstellung unterscheidet sich durch den Wegfall der Eingabe des (es gibt



nur eine Darstellungsart im Scroll Diagramm) und der Eingabe des , da die Anzahl der Werte die in der Horizontalen des Diagramms dargestellt werden der Objektbreite entspricht und hier nicht verändert werden kann.

Das Beispiel Abbildung 3.125 zeigt die Werte für die Abbildung 3.124.

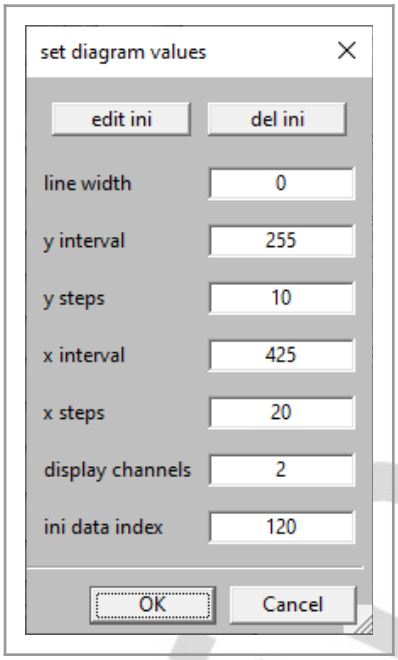


Abbildung 3.125: Diagramm Parametereinstellung

Das unterschiedliche Triggerverhalten ist in Tabelle 12: Bedeutung der Kontroll Bits aufgelistet.

xx 0000	Disabled	Sperrt den Start einer Ausgabe der Kurve. Wird dieser Mode gesetzt während eine Ausgabe läuft, so wird diese angehalten.
xx 0001	Conntinnous Run	Die Ausgabe wird unabhängig der Triggerbedingungen gestartet und kontinuierlich fortgesetzt bis 'Disable' gesetzt wird oder eine gesetzte Triggerbedingung erfüllt ist. Nach einem Trigger Stopp kann durch das Setzen des Bits die Ausgabe weiterlaufen, wobei an der Triggerposition der Kurve eine senkrechte Linie eingeblendet wird (Abbildung 3.126 Diagramm mit eingeblendeten Triggerlinien)
xx 0010	External Trigger	Es wird eine Ausgabe angehalten wenn das External Trigger Start Bit gesetzt wird. Wenn keine Ausgabe läuft wird das Trigger Bit ignoriert.
xx 0100	Trigger Positiv	Es wird eine Ausgabe angehalten wenn der Eingangswert des gewählten Trigger Kanals den Wert des Trigger_Levels überschreitet. Wenn keine Ausgabe läuft wird der Trigger ignoriert.
xx 1000	Trigger Negativ	Es wird eine Ausgabe angehalten wenn der

		Eingangswert des gewählten Trigger Kanals den Wert des Trigger_Levels unterschreitet. Wenn keine Ausgabe läuft wird der Trigger ignoriert.
01 0010	External Trigger Start	Es wird eine Ausgabe angehalten wenn das Bit im Eexternal Trigger Mode gesetzt wird und eine Ausgabe läuft. Nach dem Stop einer Ausgabe wird das Bit intern gelöscht.
1x xxxx	Retrigger Start / One Shot	Wird das Bit gesetzt während eine Ausgabe läuft, wird diese unterbrochen, die laufende Kurvendarstellung gelöscht und sofort eine neue Ausgabe gestartet. Nach dem Start einer Ausgabe wird das Bit intern gelöscht.

Tabelle 12: Bedeutung der Kontroll Bits

Der Fehler: Verweis nicht gefunden entspricht der Fehler: Verweis nicht gefunden da auch alle Eingaben den Werten aus Tabelle 10: Repeat Diagramm Interface Definitionen die gleiche Funktion haben.

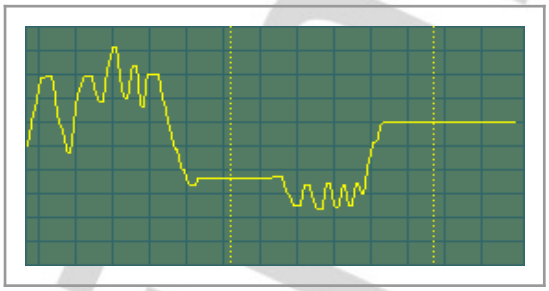


Abbildung 3.126: Diagramm mit eingeblendeten Triggerlinien

3.6.3.6 Oszillographen Diagramm

Der Oszillographen Diagramm ist eine weitere Displayfläche zur Anzeige von Kurvenverläufen das in der Funktion einem Oszillographen ähnelt. Da bei jedem Refresh der gesamte Kurvenzug dargestellt wird müssen die Daten als Feld im Interface abgelegt sein. Dazu sind im Interface 4 Datenfelder von jeweils 1000 Integerwerten vorhanden. Damit können somit 4 Kurvenzüge als Zeitdiagramm dargestellt werden. Im Displaymode '2' können die Datenfelder 0 und 1 auch als X-Y Darstellung ausgegeben werden. ***Die Benutzung dieses Diagramms setzt eine externe Funktion voraus welche das Datenfeld mit Werten füllt und auch das Triggersignal liefert.***

Abbildung 3.127 Oszillographen Diagramm zeigt als Beispiel den Bildaufbau zweier Diagramme.

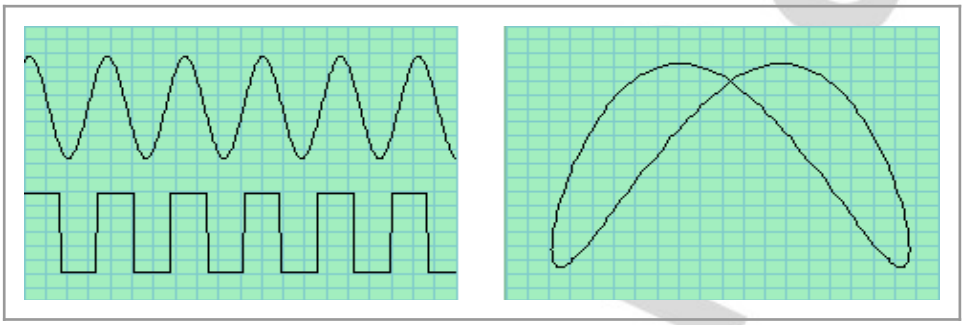


Abbildung 3.127: Oszillographen Diagramm

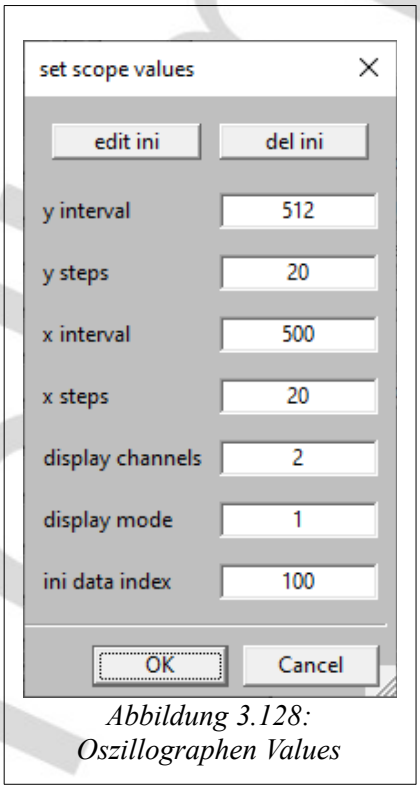


Abbildung 3.128: Oszillographen Values

Die Initialisierungswerte entsprechen denen aus der Tabelle 10 Repeat Diagramm Interface Definitionen wobei sich die Bedeutung der Werte wie im Folgenden unterscheidet.

Index Offset	**Bedeutung**	
0,2	Kontroll Bits Trigger Kanal Trigger_Level	Diese Werte werden während der Laufzeit in einer Variablen in den Datenfeldern gespiegelt.
5, 9, 13, 17	Daten Index `Kanal` x	Der Index ist hier die Nummer des Datenfeldes welches dargestellt werden soll.

Tabelle 13: Unterschiede zum Repeat Diagramm

01 0010	External Trigger Start	Dieses bit kann nur über die Kontrollbits im Datenfeld gesetzt werden.

Tabelle 14: Unterschiede der Kontroll Bits

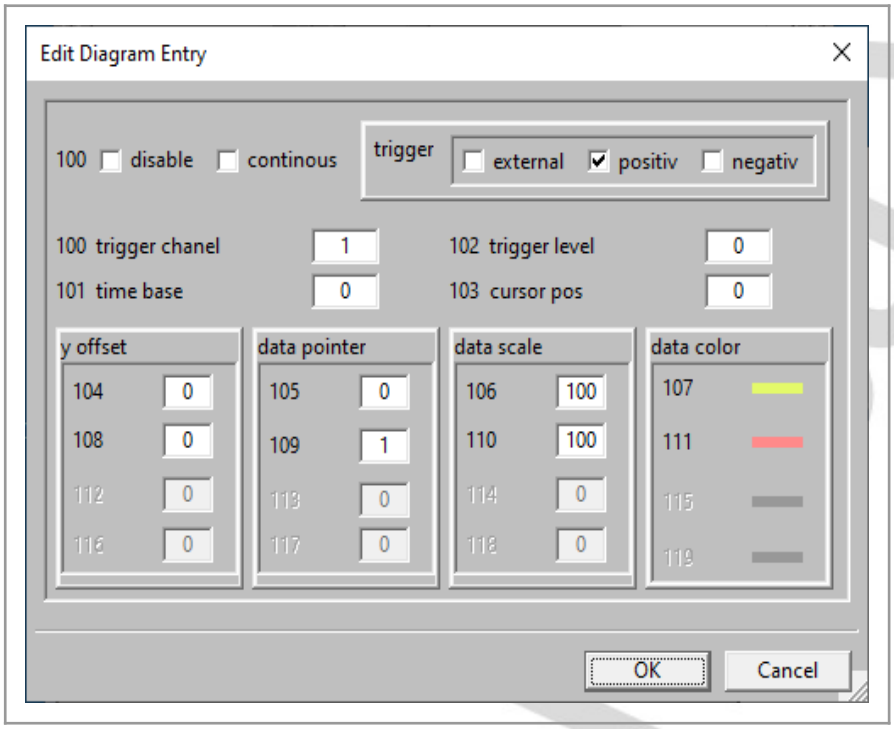


Abbildung 3.129: Objekt Init Dialog

Im Gegensatz zu den vorherigen Diagrammen kann hier eine vertikale Cursorlinie eingeblendet werden, deren Position durch den Wert in Basis + Offset 3 bestimmt wird. Ist der Wert dieser Variablen -1 wie in Abbildung 3.129 dargestellt, ist die Cursorlinie nicht sichbar.

3.6.3.7 Logicscope Diagramm

Diese Diagrammform dient zur Darstellung digitaler Daten. Es gibt hier nur eine Ausgabe als Zeitdiagramm. Die Kurven entsprechen dem Zustand der Bits in einem Wert des Eingabefeldes. Es können bis zu 12 Bit abgebildet werden, wobei die oberste Kurve immer dem LSB eine Wertes entspricht. ***Die Benutzung dieses Diagramms setzt eine externe Funktion voraus welche das Datenfeld mit Werten füllt und auch das Triggersignal liefert.***

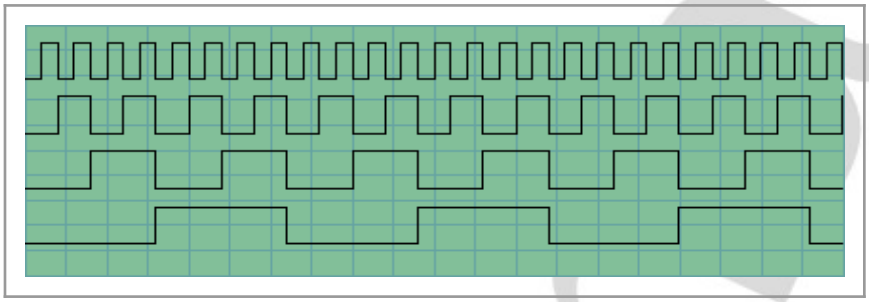


Abbildung 3.130: Logig Diagramm

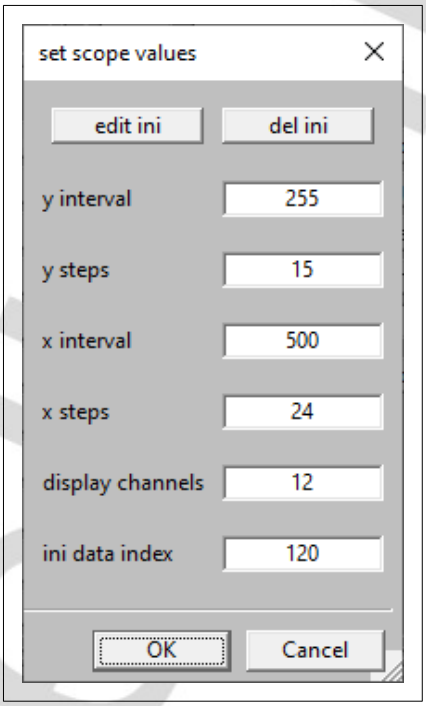


Abbildung 3.131: Logic Diagramm Values

Es werden 20 Variablen belegt deren Bedeutungen in Tabelle 15 Logic Diagramm Interface

Definitionen beschrieben sind.

Index Offset	Bedeutung	
0 Bits 0 bis 7	Kontroll Bits	Diese Werte werden während der Laufzeit in einer Variablen in den Datenfeldern gespiegelt.
0 Bits 8 bis 11	Trigger Kanal	Nummer des Bits auf das getriggert werden soll. Diese Werte werden während der Laufzeit in einer Variablen in den Datenfeldern gespiegelt.
1	Sample Zeit	Diese Werte werden während der Laufzeit in einer Variablen in den Datenfeldern gespiegelt.
2	Trigger_Level	Diese Werte werden während der Laufzeit in einer Variablen in den Datenfeldern gespiegelt.
3	Cursor Position	Legt die Position einer vertikalen Cursorlinie im Oszidiagram und Logikdiagram fest.
4	Kurven Y Lage	Legt die vertikale Position des Kurvenfeldes fest.
5	Daten Index	Der Index des Datenfeldes welches dargestellt werden soll.
6	Daten Faktor	Skaliert die Darstellung der Eingangs-werte.
7	Y Delta	Legt den vertikalen Abstand der Kurven fest.
8	Farbe Bit 0	Legt die Farbe der Kurve für diese Kurve fest.
9	Farbe Bit 1	
10	Farbe Bit 2	

11	Farbe Bit 3
12	Farbe Bit 4
13	Farbe Bit 5
14	Farbe Bit 6
15	Farbe Bit 7
16	Farbe Bit 8
17	Farbe Bit 9
18	Farbe Bit 10
19	Farbe Bit 11

Tabelle 15: Logic Diagramm Interface Definitionen

Die Funktion der Kontrollbits ist die gleiche wie unter 3.6.3.6 Oszillographen Diagramm bereits beschrieben.

Da die Interface Definitionen sich von den anderen Diagrammen unterscheiden, stellt sich der Logic Diagramm Init Dialog wie in Abbildung 3.132 dar.

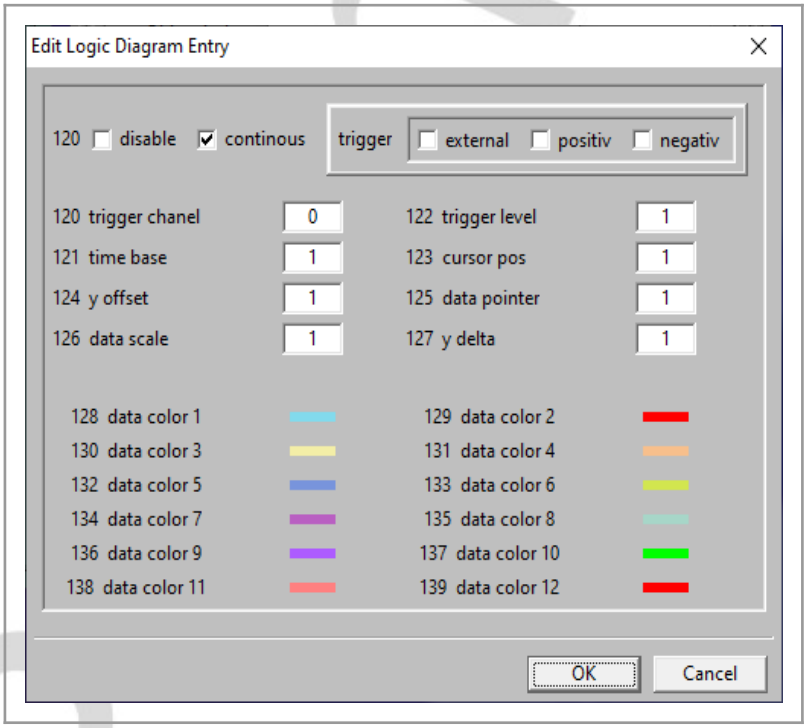


Abbildung 3.132: Logic Diagramm Init Dialog

3.6.3.8 Bar Horizontal und Vertikale

Bar horizontal Abbildung 3.133 entspricht einem horizontalem Balken ohne Beschriftung.

Bar vertical Abbildung 3.134(entspricht einem vertikalem Balken ohne Beschriftung.

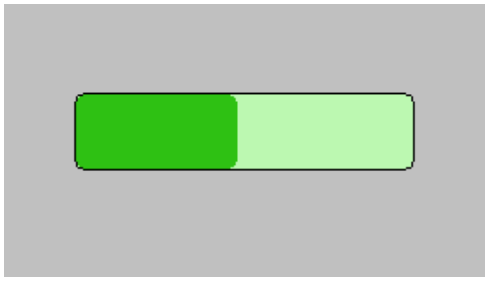


Abbildung 3.133: Bar horizontal

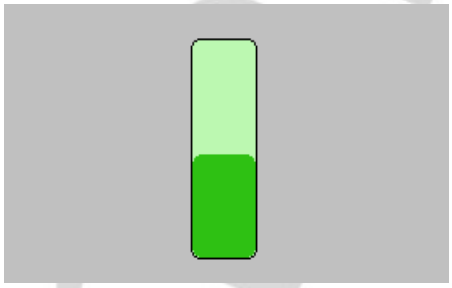


Abbildung 3.134: Bar vertical

Die Parameter können nach dem Betätigen des Button values gesetzt werden. Zu der Darstellung in Abbildung 3.133 und Abbildung 3.134 gehört die in Abbildung 3.135 gezeigten Einstellungen der Values.

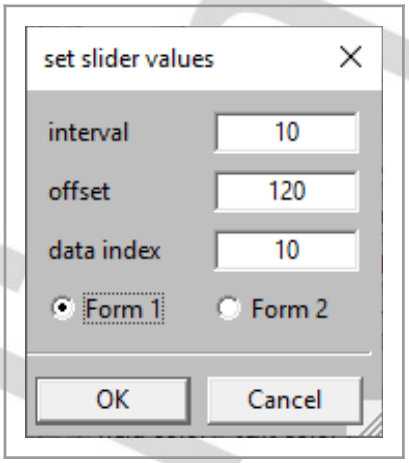
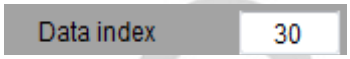


Abbildung 3.135: Beispiel Parametereinstellung

Die Bedeutung der einzelnen Werte ist wie folgt.

Ist die Anzahl der Werte die der Balken durchläuft.

Ist der Startwert des Balkens. Ist hier statt 0 ein Wert von 20 eingesetzt, ist der Wertebereich 20 bis 275 .



Ist der Index der variablen aus dem Interface welche den eingestellten Wert enthält.

Damit der Balken als Eingabe genutzt werden kann und die Balkengroeße dem aktuelle Wert der

zugewiesenen Interface Variablen folgt, müssen im Objekt Mode Dialog (Abbildung 3.89) die Optionen pick aktivieren und set refresh mod gesetzt sein.

Wird nur die Option set refresh mod gesetzt ist keine Eingabe möglich und der Schieber folgt lediglich den Werten in der Variablen.

Form 1 Form 2 Wird der Button für Form 2 gesetzt, wird die Richtung des Balkens umgekehrt dargesztellt.

Eine weitere Option der Darstellung ist die Zuordnung einer Graphik als Zeiger des Balkens. Hierzu wird zusätzlich zur ausgewählten Graphik die Option zeiger image ausgewählt. Die Graphik wird jetzt nicht mehr dem Hintergrund des Objekts, sondern dem Balken zugeordnet. Hierbei ergibt jedoch lediglich die Zuweisung eines Musters einen Sinn.

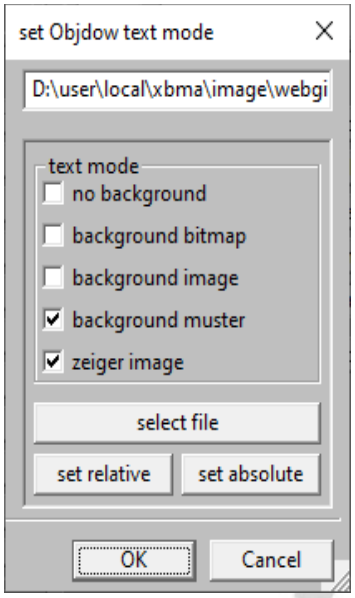


Abbildung 3.136: Objekt Textmode

Abbildung 3.137: Bar mit Graphik Muster

.

3.6.3.9 Graphik Display

Das Graphik Display Objekt stellt eine Zeichenfläche für Vektoren dar. Die Vektoren können durch ein externes Programm in einem Intrrface Feld abgelegt werden. Die Vektordaten belegen jeweils 4 Integer Werte (xstart , ystart , xend , yend) . Da die Feldgröße auf 1000 Werte beschränkt ist, können auf diese Art maximal 250 Vektoren gespeichert werden. Ist das Feld mit den gewünschten Daten gefüllt worden, kann man durch das Setzen des externen Trigger Signals die Anzeige der Vektoren auslösen.

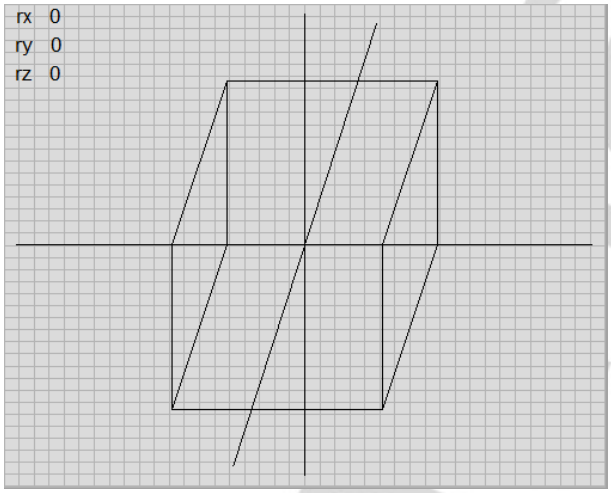


Abbildung 3.138: Graphik Display Example

Im Objekt Mode sind nur das setzen des ☑ fill obj und des ☑ refresh mode möglich (Abbildung 3.140: Display Objekt Modes). Ist ‚fill obj' gesetzt, wird die Objektfläche mit der ‚field color' gefüllt und das Raster in der ‚text color' gezeichnet, wie in Abbildung 3.138: Graphik Display Example dargestellt. Ist ‚fill obj' nicht gesetzt, kann über den Objekt Text auch ein Hintergrundbild als auch ein Hintergrundmuster ausgewählt werden.

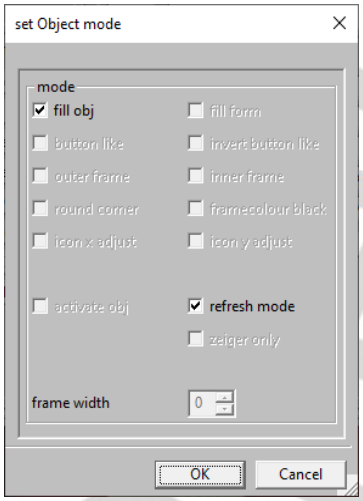


Abbildung 3.140: Display Objekt Modes

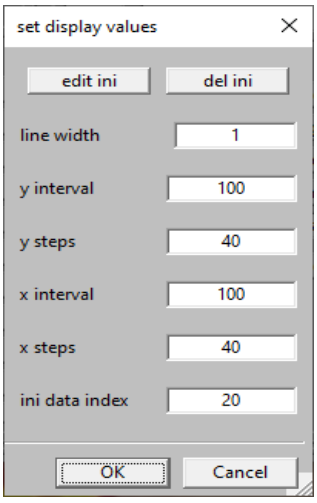


Abbildung 3.139: Display Objekt Values

In Abbildung 3.139: Display Objekt Values ist das dazugehörige Display Value Menue dargestellt.

3.7 Kommando Menü

Kommandos stellen Funktionen dar welche während der Laufzeit entsprechend ihrem 'event type' ausgeführt werden und unter anderem Variable im Interface bearbeiten können.

Die Erstellung von Kommandos und deren Eigenschaften werden hier im folgendem beschrieben. Da Kommandos keine sichtbaren Elemente sind, gibt es logischerweise keine Darstellung im Applikationsfenster.

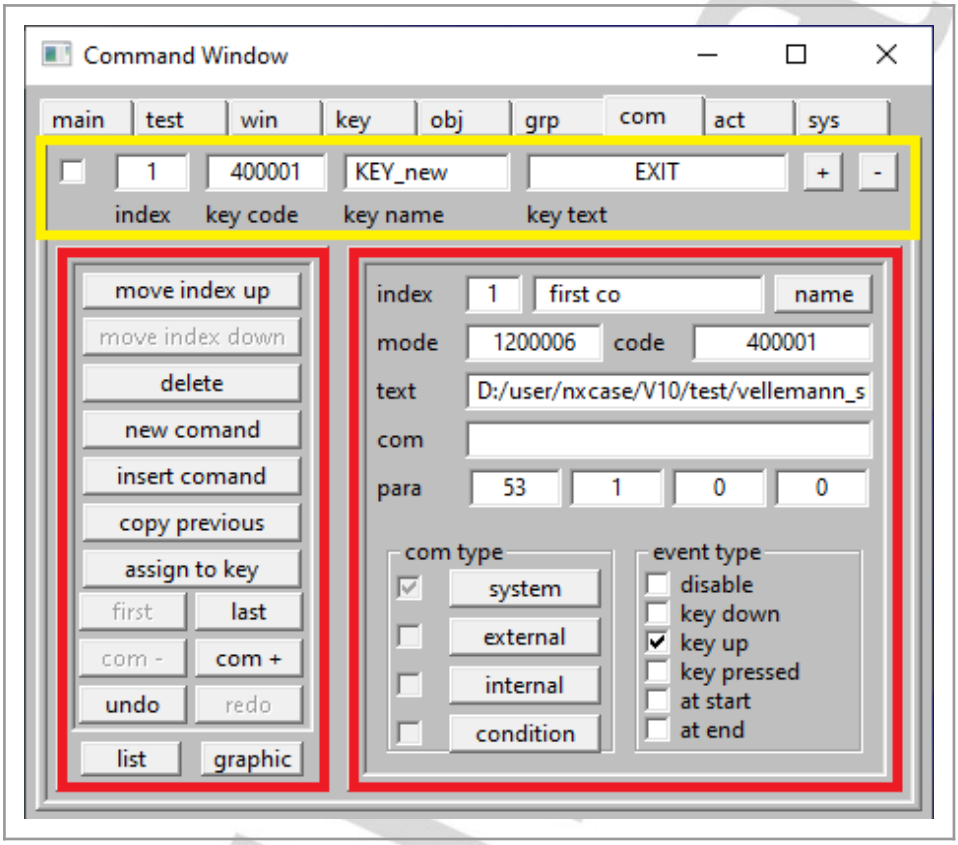


Abbildung 3.141: Kommando Menü

Wie in Abbildung 3.141 gezeigt, besteht das Kommando Menü aus drei Hauptbereichen. Dem 'Commands' Bereich (Rot) und dem 'Eigenschaften' Bereich (Rot) und einem 'Keypick' Bereich (Gelb).

Der 'Commands' Bereich dient der Verwaltung der Kommandos, während der 'Eigenschaften' Bereich wie der Name schon sagt zur Einstellung der Eigenschaften der Kommando vorgesehen ist.

Neu ist hier der 'Keypick' Bereich welcher die Daten des Keypicks anzeigt welches gerade selectiert , oder dem aktuellem Kommando zugeordnet ist.

3.7.1 Kommando Verwaltung

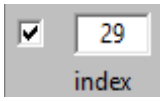
Wie schon vorab beschrieben sind alle Elemente in Feldern abgelegt. Auf die Kommandos kann über deren Feldindex zugegriffen werden. Der Index startet mit 1 und reicht bis zum letzten erstelltem Kommando.

Das Feld 

zeigt die Nummer des zur Bearbeitung ausgewählten Kommandos an.

Mit den Buttons com - und com + kann man sich durch das Feld aufwärts und abwärts bewegen.

Mit den Buttons first und last springt man direkt zum Anfang b.z.w zum Ende der Liste.



wird die Checkbox neben dem Index Zähler gesetzt, wird durch das Anklicken eines Keypicks auch das dazugehörige Kommando angezeigt (sofern dem gewählten Element ein Kommando zugewiesen wurde). Im anderen Fall wird das Anklicken eines Keypicks benutzt um einem Kommando mit dem Button assign to key einen Keypick zuzuordnen.

move index up Verschiebt das aktuelle Kommando um eine Position in der Liste nach oben.

move index down Verschiebt das aktuelle Kommando um eine Position in der Liste nach unten

Da die Kommandos in der Reihenfolge von 1 bis last während der Laufzeit ausgeführt werden, und somit ein Kommando mit höherem Index nach einem Kommando mit niedrigerem Index ausgeführt wird, kann man durch die Indexposition eine Berechnungsreihenfolge innerhalb eines 'event type' bestimmen.

new com Erstellt ein neues Kommando ohne Eigenschaften am Ende der Liste.

insert com Fügt ein neues Kommando ohne Eigenschaften hinter dem aktuellem Kommando ein.

copy previous Kopiert die Eigenschaften des vorherigen Kommandos in das aktuelle Kommando.

assign to key verknüpft das aktuelle Kommando mit dem im Keypick Bereich (Abbildung 3.143) gesetztem Keypick, welches auch im Applikationsfenster markiert ist. Die Auswahl eines Keypicks erfolgt mit den Tasten - + oder durch das Anklicken des gewünschten Keypicks im Applikationsfenster. (wenn die Checkbox 27 neben der Index Anzeige nicht gesetzt ist).

delete Löscht das gerade aktuelle Kommando aus der Liste.

undo Erlaubt das Rückgängig machen der letzten Änderungen im Kommando Menü.

Alternativ zu der oben aufgeführten Art der Bedienung über die Buttons, können einige Funktionen auch über Popup Menü aufgerufen werden. Befindet sich der Mauscursor im Applikationsfenster und die rechte Maustaste wird betätigt erscheint das Kommando Pop Up Menü (Abbildung 3.142).

Mit 'copy' wird das aktuelle Kommando mit allen seinen Eigenschaften in das Clipboard copiert.

Mit 'paste' wird der Inhalt des Clipboards hinter dem Index des aktuellen Kommandos eingefügt.

Die Funktionen 'delete' und 'undo' entsprechen den Button delete und undo.

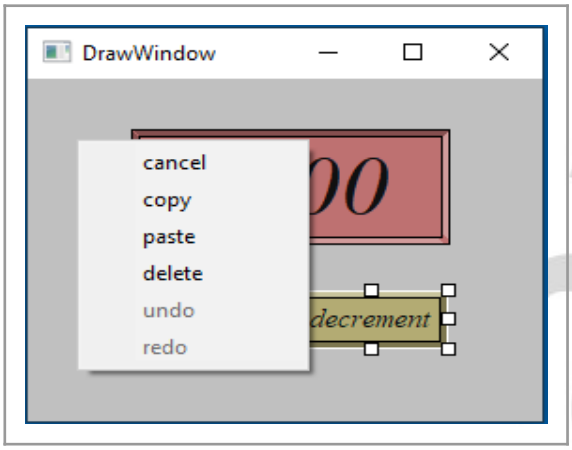


Abbildung 3.142: Kommando Pop Up Menü

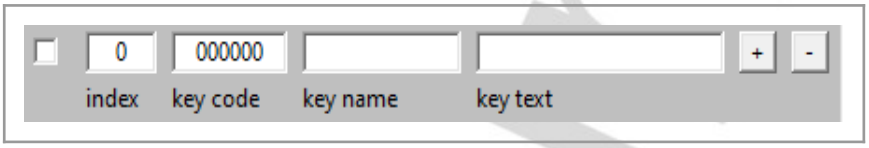


Abbildung 3.143: Keypick Bereich

Um die Kommandos besser bearbeiten zu können werden mit den Button list und graphic zusätzliche Fenster geöffnet.

list öffnet das Kommando List Fenster (Abbildung 3.144) welche alle Kommandos in der Reihenfolge ihres Indizes auflistet.

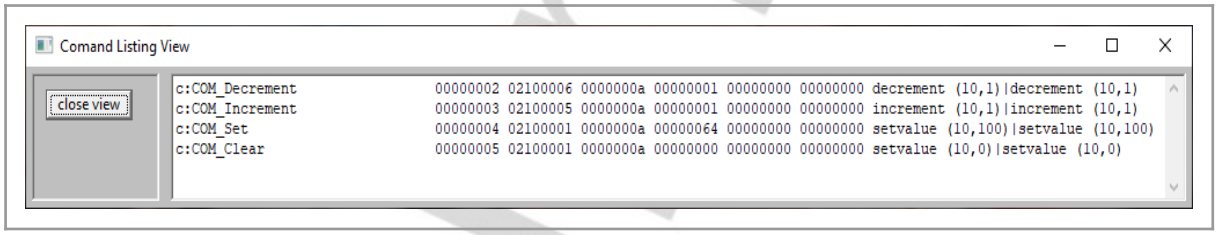


Abbildung 3.144: Kommando List Fenster

Alternativ zur Textanzeige kann man die Kommandos auch in einer Graphischen Ansicht anzeigen.

graphic Öffnet das Fenster mit der Graphische Darstellung des aktuellen Kommandos und dessen Eigenschaften (Abbildung 3.145) welche in Abbildung 3.146 erläutert wird.

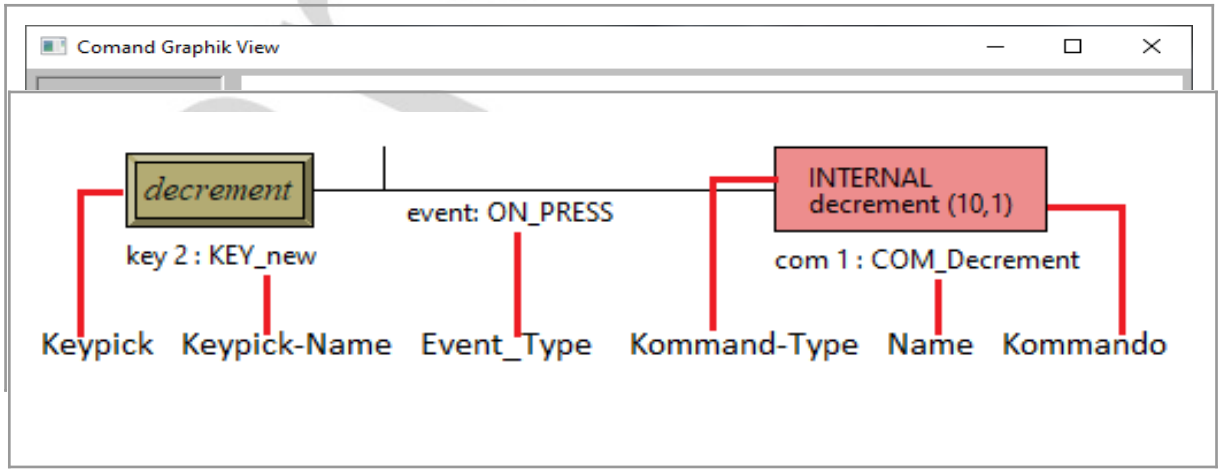


Abbildung 3.146: Elemente der Kommandodarstellung

Im linken Teil wird das 'Keypick' mit dem Namen dargestellt welches mit dem Kommando verknüpft ist. Das rechte Element zeigt das Kommando, den Typ des Kommandos sowie dessen Name. Die Verbindung zwischen den Elementen ist der Event Type über den beide miteinander verbunden sind.

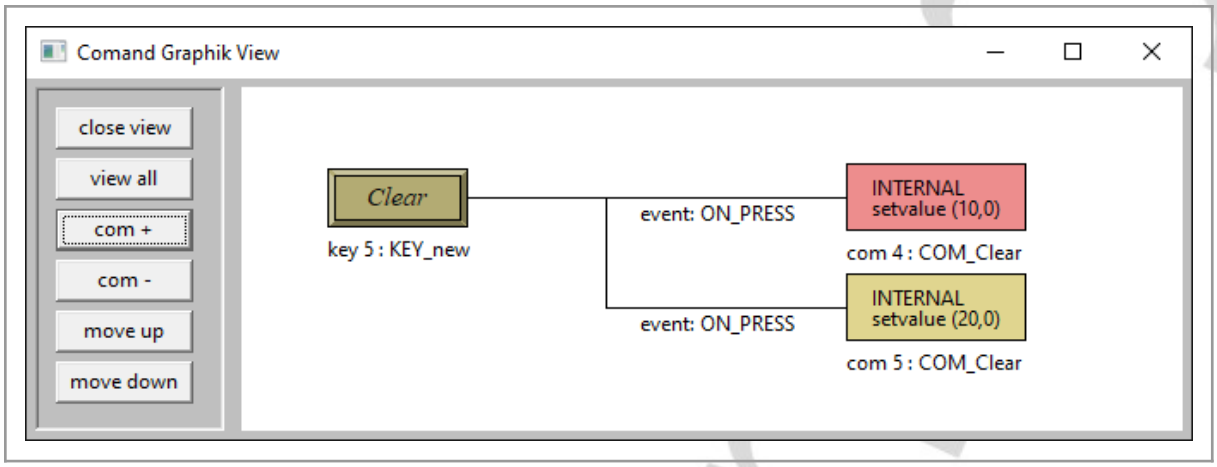


Abbildung 3.147: Mehrfache Kommando Zuordnung

Abbildung 3.147 Zeigt die Mehrfache Kommando Zuordnung zu einem Keypick, wobei das aktuelle Kommando rot dargestellt wird. Sind die Kommandos nicht mit einem Keypick verbunden sondern Teil der Applikation ergibt sich das Bild aus Fehler: Verweis nicht gefunden.

Auch hier haben die Button 

die gleichen Funktionen wie die entsprechenden Button im Kommando Bereich.

Schaltet die Anzeige in die in Abbildung 3.148 gezeigte Darstellung um. Hier werden alle Kommandos in der Reihenfolge ihres Indizes aufgeführt. Das aktuelle Kommando ist hier wieder rot markiert. Mit der linken Maustaste kann ein Kommando zur Bearbeitung ausgewählt werden. Mit der rechten Maustaste öffnet sich ein Pop Up Menü mit den gleichen Funktionen wie das Pop Up Menü im Applikationsfenster welches schon weiter oben beschrieben wurde.

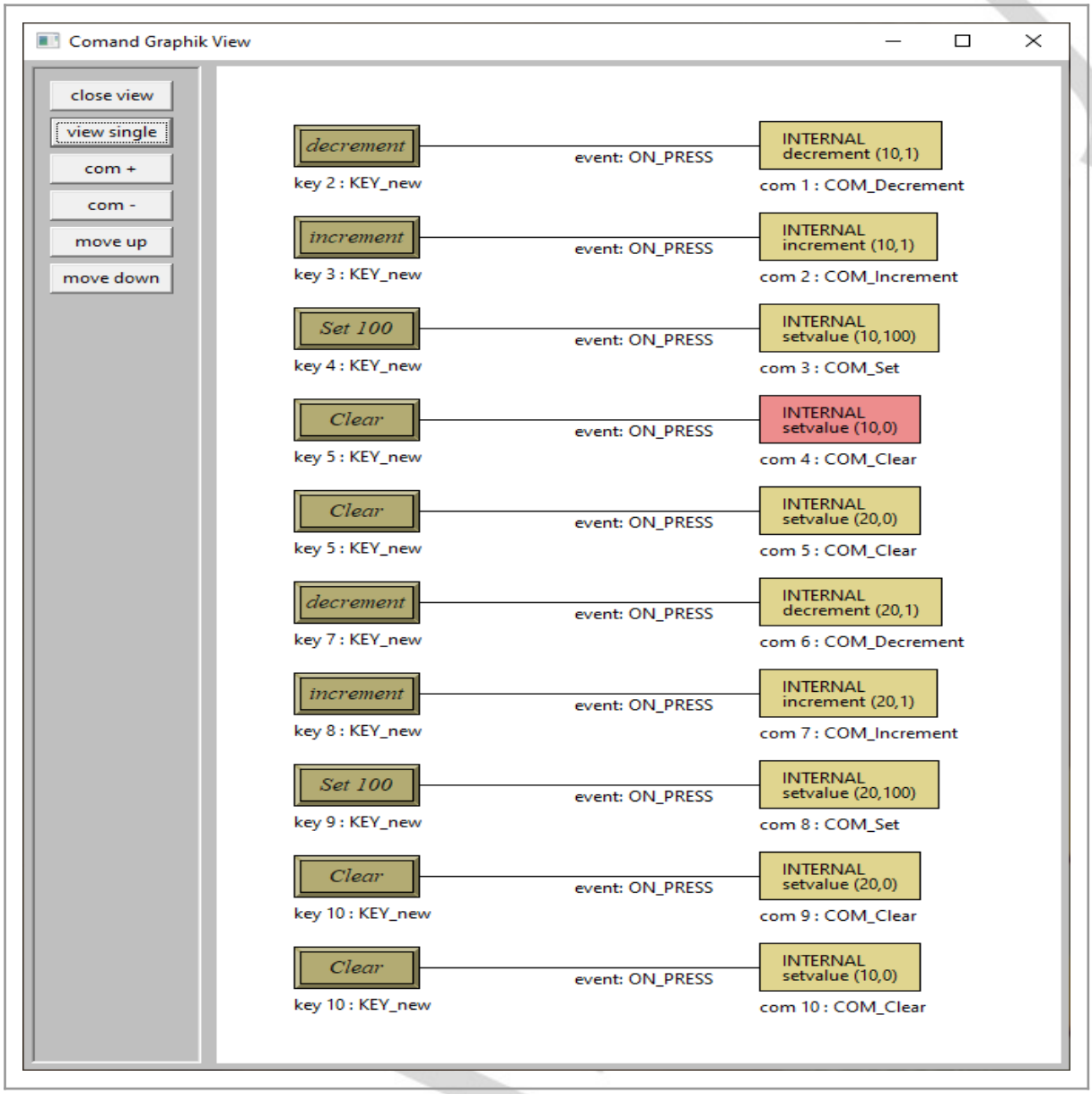


Abbildung 3.148: 'Show all' Ansicht mit Pop Up Menü

Setzt die Anzeige wieder in die Einzeldarstellung zurück.

3.7.2 Kommando Eigenschaften

Die Eigenschaften eines Kommandos bestehen aus der Kommandoart (com type) und der Auslösebedingung (event type).

Mit dem Button name kann man das Kommando mit einem Namen versehen, der zwar keine Bedeutung für dier Funktion hat, aber bei der Verwaltung hilfreich sein kann. Die funktionalen Eigenschaften werden mit dem in Abbildung 3.149 dargestelltem Kommando Eigenschaften Setup Bereich gesetzt.

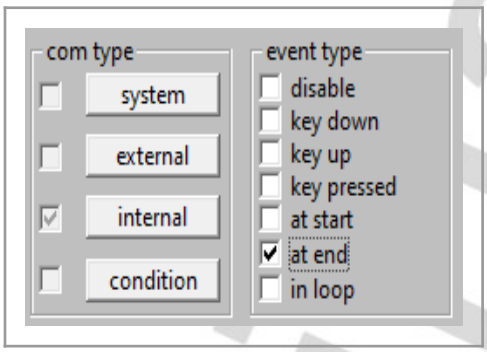


Abbildung 3.149: Kommando Eigenschaften Setup

Die Art eines Kommandos wird mit den Button im Feld 'com type' ausgewählt.

3.7.2.1 Kommando Typen.

Wie Abbildung 3.149 zeigt, gibt es vier verschiedene Arten von Kommandos. ***Systemkommandos*** beeinflussen das Verhalten der Applikation. ***Externe Kommandos*** rufen Programme oder Funktionen auf die nicht zu RPT-One gehören und Daten aus dem Interface lesen, schreiben und bearbeiten können. Hierbei handelt es sich meistens um User Funktionen. Einige Basisfunktionen zur Bearbeitung von Daten im Interface sind in RPT-One bereits integriert und als ***Internen Kommandos*** verfügbar. ***Condition Kommandos*** prüfen die gewählte Bedingung und überspringen die folgenden Kommandos wenn diese Bedingung erfüllt ist.

3.7.2.1.1 Systemkommandos.

Mit dem Button System öffnet man den System command call submenue (Abbildung 3.151) in welchem die verfügbaren System Kommandos aufgeführt sind. Mit der linken Maustaste kann das gewünschte Kommando ausgewählt werden. Verfügt das Kommando über einen Parameter (in der Klammer hinter dem Kommando angezeigt). Mit set parameter kann man den System command chain to programm (Abbildung 3.150) öffnen um den Parameter direkt einzugeben oder über den Button filename einen Filedialog zur Auswahl der Datei aufrufen.

3.7.2.1.1.1 Chain to programm (file.ndat)

ersetzt die aktuelle Applikation durch die im Parameter 'file.ndat' gewählte Applikation, wobei andere gestarteten Applikationen oder User Programme davon nicht beeinflusst werden.

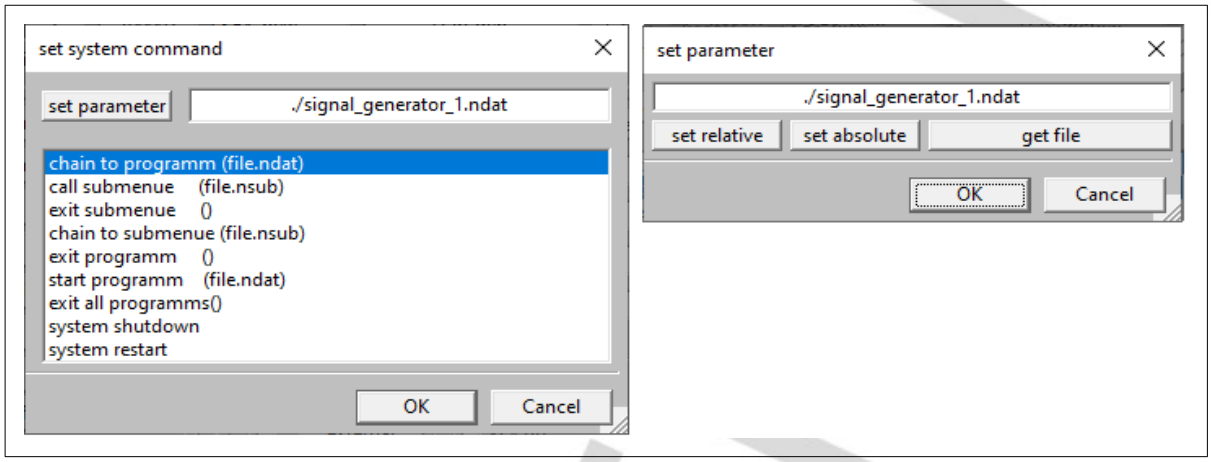


Abbildung 3.150: System command chain to programm

3.7.2.1.1.2 Call Submenue (file.nsub)

Blendet eine Unterapplikation in die laufende Applikation ein. Dabei wird die Hauptapplikation bis zur Schließung dieses Fensters durch den Aufruf 'exit submenue' angehalten und erste danach weiter ausgeführt. Eine genaue Beschreibung der Benutzung von Submenüs folgt an anderer Stelle dieser Dokumentation. In einem Submenü darf kein 'Call Submenue' aufgerufen werden.

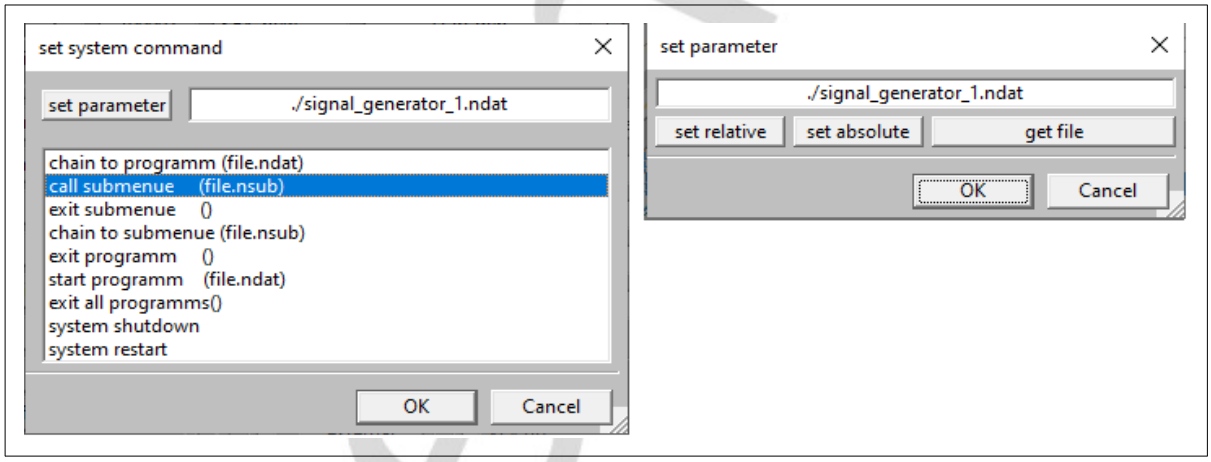


Abbildung 3.151: System command call submenue

3.7.2.1.1.3 Exit submenue ()

Beendet wie schon erwähnt die Ausführung eines Untermenüs.

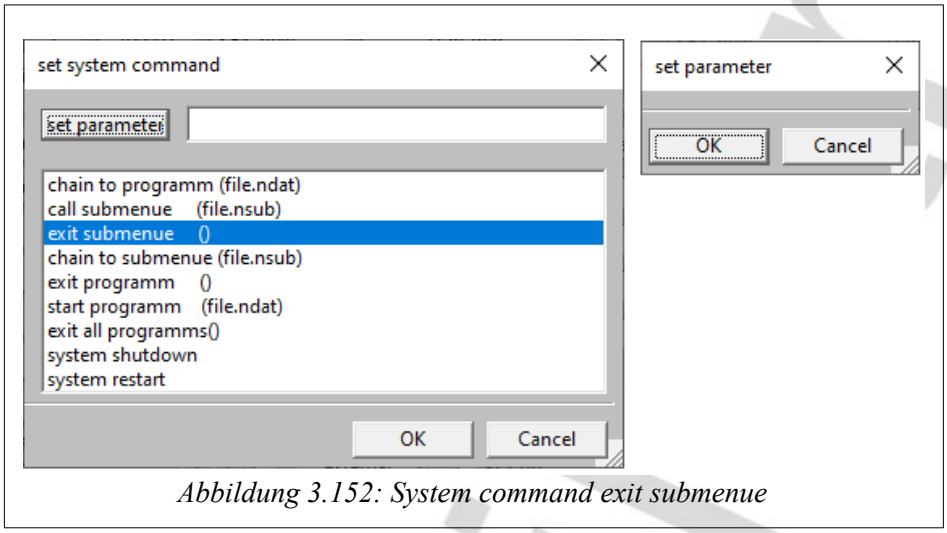


Abbildung 3.152: System command exit submenue

3.7.2.1.1.4 Chain to submenue (file.nsub)

Da ein Submenü keinen Aufruf 'Call Submenue' enthalten darf bietet das Kommando 'Chain to submenue' die Möglichkeit von einem Submenü zu einem anderem Submenü zu wechseln.

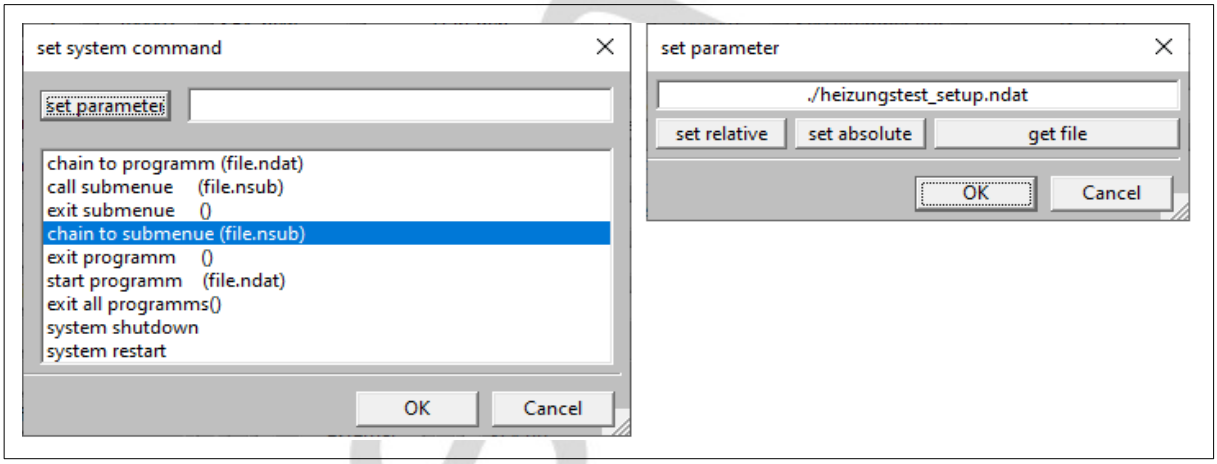


Abbildung 3.153: System command chain to submenue

3.7.2.1.1.5 Exit programm ()

Beendet die Applikation ohne andere gestarteten Applikationen oder User Programme zu beeinflussen.

Da es nur maximal 32 parallele Prozesse geben kann, sind nur Prozess Nummern von 1 bis 31 zulässig. Wird hier eine Nummer größer als 32 eingegeben, so werden alle RPT Prozesse größer 0 gestoppt.

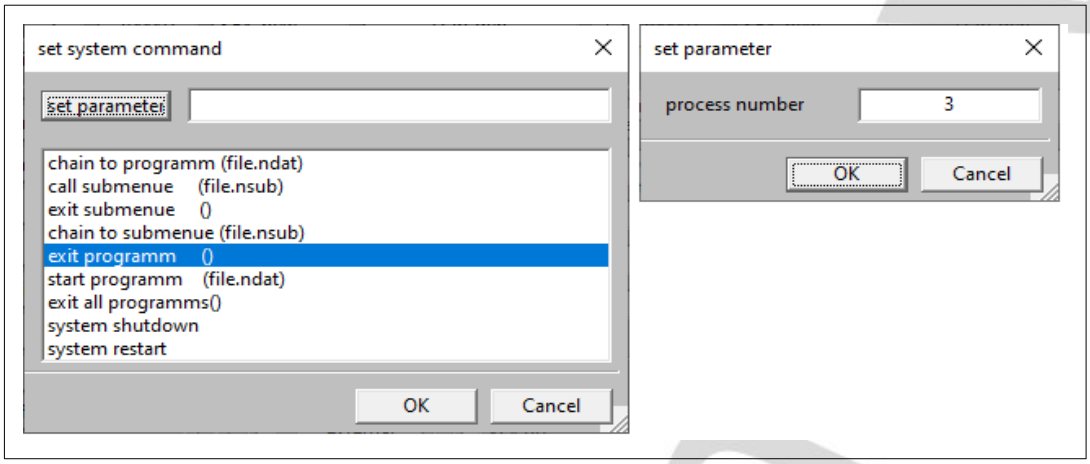


Abbildung 3.154: System command exit programm

3.7.2.1.1.6 Start programm (file.ndat)

Startet eine neue Applikation parallel zur laufenden Applikation.

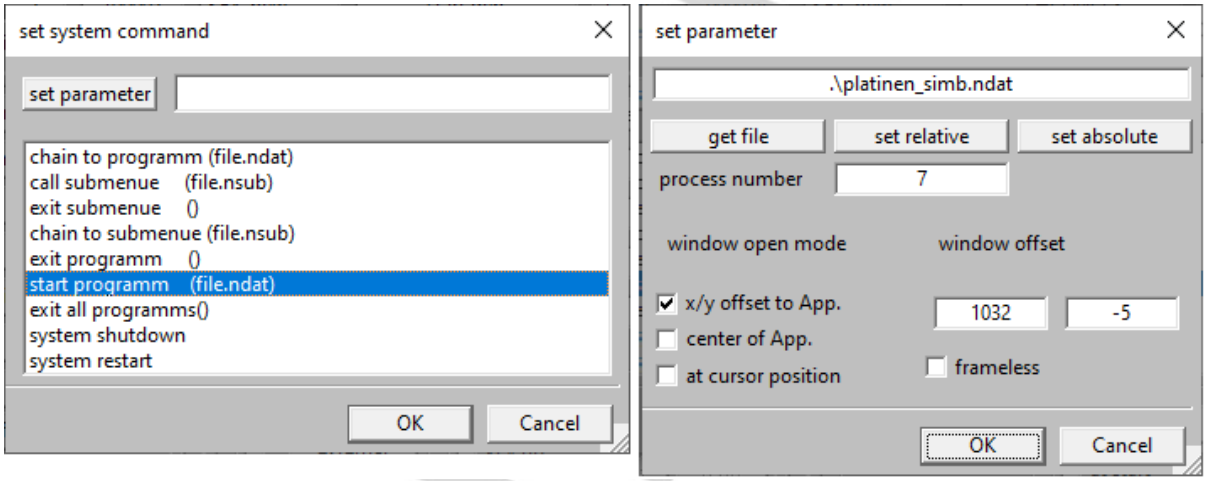
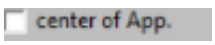


Abbildung 3.155: System command start programm

Die neue Applikation wird auf dem Desktop relativ zur Position der aktuellen Applikation im Abstand horizontal / vertkal geöffnet. In diesem Beispiel 

ist die horizontale Position gleich alter Position + 1032 Pixel. Die vertikale Position ist demnach -5 Pixel zur alten Position.



Die neue Applikation wird auf dem Desktop zentriert in der alten Position geöffnet.

Die Startposition der neue Applikation auf dem Desktop entspricht den aktuellen Cursor Daten.

Wird diese Option gesetzt, wird die neue Applikation ohne die normale Frame Dekoration geöffnet. Das bedeutet, daß das Fenster nicht mit den Mausfunktionen bearbeitet werden kann.

3.7.2.1.1.7 Exit all programms()

Beendet alle laufenden Applikationen und Userprogramme (sofern diese entsprechend aufgebaut wurden).

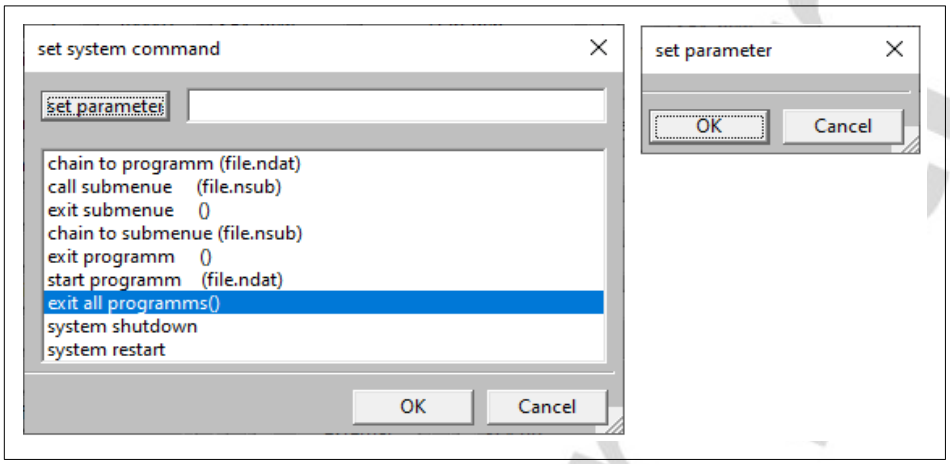


Abbildung 3.156: System command exit all programms

3.7.2.1.1.8 System Shutdown

Führt einen Shutdown des Betriebssystems durch, das heißt der Rechner wird ausgeschaltet.

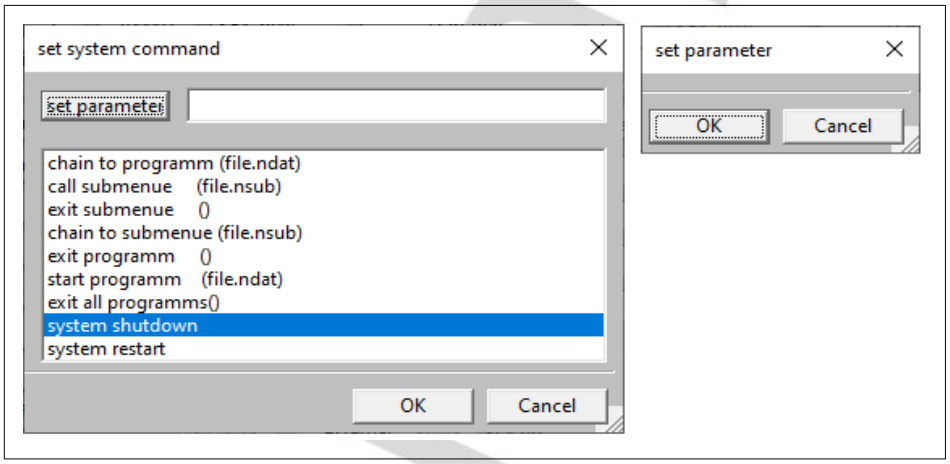


Abbildung 3.157: System command system shutdown

3.7.2.1.1.9 System Restart

Führt einen Neustart des Rechner aus.

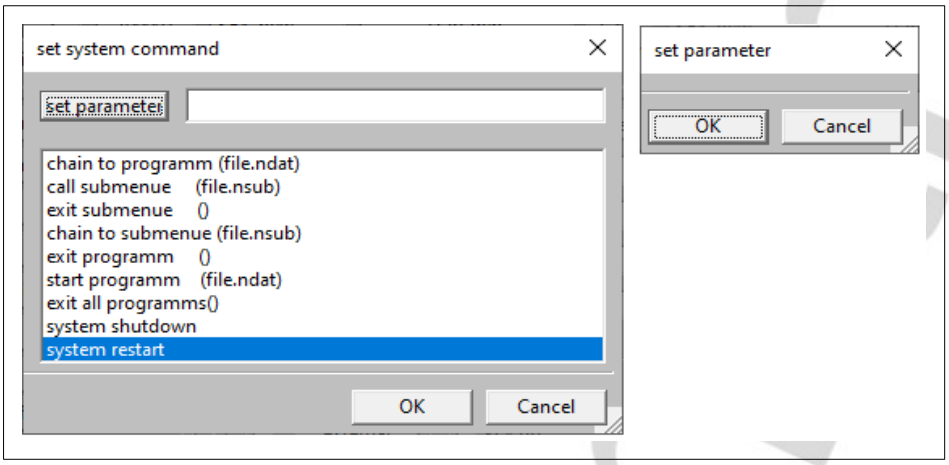


Abbildung 3.158: System command system restart

3.7.2.1.2 Externe Kommandos

Mit dem Button External öffnet man den External Kommando Dialog (Abbildung 3.159).

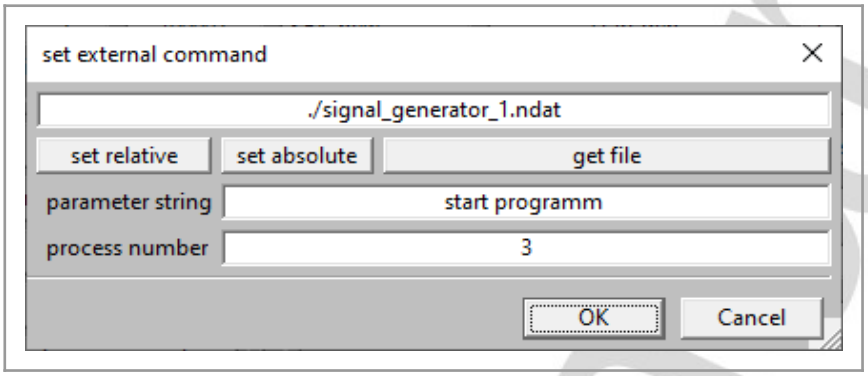


Abbildung 3.159: External Kommando Dialog

Mit dem Button get file wird ein Filedialog geöffnet der die Auswahl des externen Programms ermöglicht. Wie bei einem Programmaufruf in einem 'DOS Fenster' unter MS Windows oder einem Terminal Fenster unter Liniux können hier Parameter an das Programm übergeben werden. Die vom Programm benötigten Variablen müssen dazu in den Parameter String eingegeben werden.Auf die Anforderungen an externe Programme wird in einem späteren Kapitel noch näher eingegangen.

3.7.2.1.3 Interne Kommandos

Mit dem Button Internal öffnet man den Innternal Kommando Dialog (Abbildung 3.160). Hier sind alle intern verfügbaren Kommandos aufgelistet. Mit der linken Maustaste wird das gewünschte Kommando ausgewählt und danach mit set parameter der Internal Parameter Dialog (Abbildung 3.161) geöffnet um die Übergabeparameter einzugeben.

Abbildung 3.160: Innternal Kommando Dialog

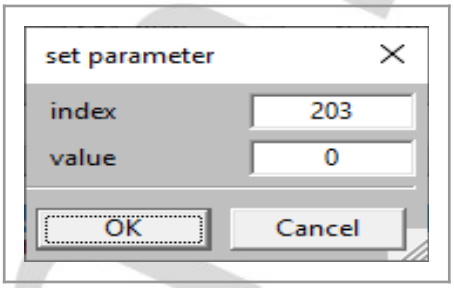


Abbildung 3.161: Internal Parameter Dialog

3.7.2.1.3.1 Eingabe(int index, int stellen, int value)

Ist eine Funktion zum Aufbau einer Tastatureingabe. Der Index verweist auf eine Variable im Interface. Die Stellen sind die Anzahl der Ziffern der Eingabe und Value ist der Wert welcher auf der Einerstelle angefügt werden soll. Der Aufruf der Funktion verschiebt den Inhalt der Variablen um eine Dezimalstele nach Links und addiert dann den Value. Beim Verschieben werden die Ziffern die Rechts die Anzahl der Stellen überschreiten gelöscht.

Beispiel:

Nehmen wir an die Variable 10 enthält den Startwert 0 dann erhalten wir folgende Ergebnisse.

eingabe 10 3 1 setzt den Wert auf 001

eingabe 10 3 4 setzt den Wert auf 014

eingabe 10 3 2 setzt den Wert auf 142

eingabe 10 3 0 setzt den Wert auf 420

3.7.2.1.3.2 Setwert(int index, int value, int number)

Schreibt den Wert Value in die Variable Index. Ist der Wert von number größer 1 wird der Wert Value in alle variablen von index bis index +number copiert .

3.7.2.1.3.3 Setbit(int index, int bit)

Setzt das Bit bit der Variablen Index auf 1.

3.7.2.1.3.4 Resbit(int index, int bit)

Setzt das Bit bit der Variablen Index auf 0.

3.7.2.1.3.5 Invbit(int index, int bit)

Invertiert das Bit bit der Variablen Index.

3.7.2.1.3.6 Increment(int index, int value)

Erhöht den Wert der Variablen Index um den Wert Value.

3.7.2.1.3.7 Decrement(int index, int value)

Erniedrigt den Wert der Variablen Index um den Wert Value.

3.7.2.1.3.8 Copy(int source_index, int dest_index, int number)

Kopiert den Wert der Variablen Source nach Destination. Ist der Wert von number größer 1 wird der Inhalt von Source so oft nach Destination copiert wie number angibt. Nach jedem kopieren wird der Zeiger auf Destination um 1 erhöht, so das am Ende alle variablen von Destination bis Destination + number auf den Wert von Source gesetzt sind.

3.7.2.1.3.9 Increment_to_limit(int index, int value, int limit)

Erhöht den Wert der Variablen Index um den Wert Value wenn der Wert von Index kleiner ist als Limit.

3.7.2.1.3.10 Decrement_to_limit(int index, int value, int limit)

Erniedrigt den Wert der Variablen Index um den Wert Value wenn der Wert von Index größer ist als Limit.

3.7.2.1.3.11 hexinput(int index, int stellen, int value)

Entspricht der Funktion Eingabe jedoch für Hexadezimal Werte.

3.7.2.1.3.12 tofloat(int source_index, int dest_index, float faktor)

Der Integer Wert aus source index wird zu einenm floating point wert umgewandelt und mit faktor multipliziert als floating point wert in der Variablen dest_index abgelegt.

3.7.2.1.3.13 scale(int source_index, int dest_index, int percent)

Der Wert aus source index wird mit dem Prozentwert percent multipliziert in der Variablen dest_index abgelegt.

3.7.2.1.3.14 shiftleft(int source_index, int dest_index , int positions , int mask)

Der Wert aus source index wird um position Bits nach oben geshiftet in dest index abgelegt. In mask auf ‚1' gesetzte Bits werden von dieser Aktion ausgeschlossen. (hat die Maske den Wert 0, werden alle Bits aus Souce Index um positions bit nach oben geshiftet in dest index abgelegt.

3.7.2.1.3.15 shiftright(int source_index, int dest_index , int positions , int mask)

Der Wert aus source index wird um position Bits nach unten geshiftet in dest index abgelegt. In mask auf ‚1' gesetzte Bits werden von dieser Aktion ausgeschlossen. (hat die Maske den Wert 0, werden alle Bits aus Souce Index um positions bit nach unten geshiftet in dest index abgelegt.

3.7.2.1.3.16 setmaskedwert(int index, int value , int mask)

Setzt Wert der index Variablen auf den Wert value, wobei die in mask gesetzten Bits ausgenommen sind.

3.7.2.1.3.17 copymasked(int source_index, int dest_index , int mask)

Kopiert den Wert der source_index Variablen in die dest_index Variable, wobei die in mask gesetzten Bits ausgenommen sind.

3.7.2.1.3.18 gettime(int index, int code)

In die Variable index wird der durch code definierte Wert der Systemzeit abgelegt.

code	Bedeutung
1	Es werden die Milli Sekunden nach index kopiert.
2	Es werden die Sekunden nach index kopiert.
3	Es werden die Minuten nach index kopiert.
4	Es werden die Stunden im 24 Stunden Format nach index kopiert.
5	Es werden die Stunden im 12 Stunden Format nach index kopiert.
6	Spezielle Funktion um die Position des Stunden Zeigers auf einem Uhren Ziffernblatts darzustellen.

3.7.2.1.3.19 damping(int source_index, int dest_index , int temp_index , int faktor)

Die Differenz aus dem Inhalt von source_index – Inhalt von temp_index geteilt durch den Inhalt von faktor wird bei jedem Schleifendurchlauf in dest_index gespeichert, wodurch sich eine Dämfpung nach einer E-funktion ergibt, deren Zeitkonstante abhängig ist von dem Inhalt von faktor und der Applikations Loop Time.

3.7.2.1.3.20 addint(int dest_index, int index_1 , int index_2)

Der Inhalt von dest_index entspricht der Summe aus dem Inhalt von index_1 und index_2.

3.7.2.1.3.21 subint(int dest_index, int index_1 , int index_2)

Der Inhalt von dest_index entspricht der Differnz aus dem Inhalt von index_1 - index_2.

3.7.2.1.3.22 multint(int destination, int source , int factor)

Der Inhalt von destination entspricht dem Inhalt von source multipliziert mit factor.

3.7.2.1.3.23 divint(int destination, int source , int divisor)

Der Inhalt von destination entspricht dem Inhalt von source dividiert durch divisor.

3.7.2.1.3.24 textinput(int stringindex, int stellen, int character)

Entspricht der Funktion Eingabe jedoch für Texte, wobei stringindex für einen Zeiger auf ein charakter feld im Interface verweist.

3.7.2.1.3.25 strtofloat(int str_index, int dest_index , int faktor)

Der im charakter feld str_index befindliche Text wird in einen floating point value umgewandelt und mit dem durch faktor indiziertem floatingpoint wert multiplizier unter dem dest_index im Interface abgelegt.

3.7.2.1.3.26 logarythem(int source_index, int dest_index , int faktor)

Der Inhalt von destination entspricht dem Integer Wert des logaryhtmus von source multipliziert mit faktor.

3.7.2.1.3.27 copyfloat(int dest_index, int source_index)

Kopiert den floating point Wert der Variablen source_index nach dest_index.

3.7.2.1.3.28 addfloat(int dest_index, int index_1 , int index_2)

Der Inhalt von dest_index entspricht der Summe aus dem Inhalt von index_1 und index_2.

3.7.2.1.3.29 subfloat(int dest_index, int index_1 , int index_2)

Der Inhalt von dest_index entspricht der Differnz aus dem Inhalt von index_1 - index_2.

3.7.2.1.3.30 multfloat(int destination, int source , int factor)

Der Inhalt von destination entspricht dem Inhalt von source multipliziert mit factor.

3.7.2.1.3.31 divfloat(int destination, int source , int divisor)

Der Inhalt von destination entspricht dem Inhalt von source dividiert durch divisor.

3.7.2.1.3.32 floattostr(int stringindex, int sourceindex , int num1 , int num2)

Der floating point value sourceindex wird in den durch stringindex definierten Text umgewandelt, wobei num1 die Anzahl der darzustellenden Stellen angibt und num2 die Anzahl der Nachkommastellen.

3.7.2.1.3.33 floattoint(int source_index, int dest_index, float faktor)

Der float Wert aus source index wird mit faktor multipliziert als integer wert in der Variablen dest_index abgelegt.

3.7.2.1.4 Condition Kommandos

Condition Kommandos dienen dazu, nachfolgende Kommandos zu Überspringen, wenn die gewählte Bedingung erfüllt ist.

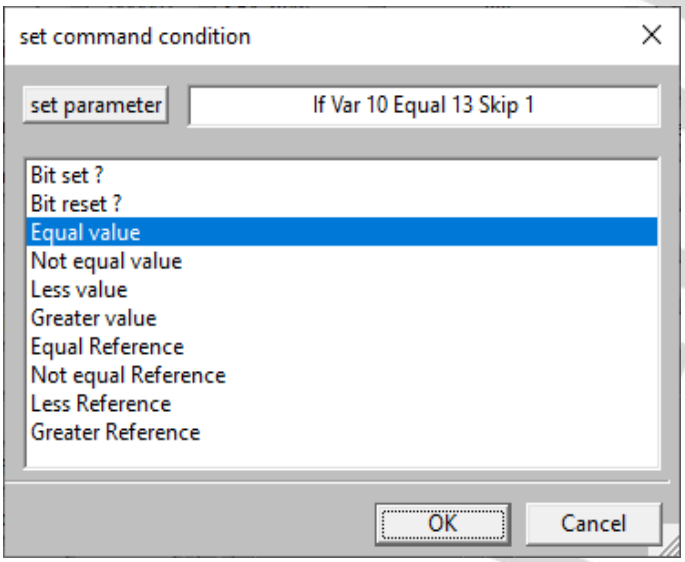


Abbildung 3.162: Kondition Kommandos

Ist wie in Beispiel Abbildung 3.162: Kondition Kommandos die Variable im Index 10 gleich 13, so wird 1 folgendes Kommando übersprungen.

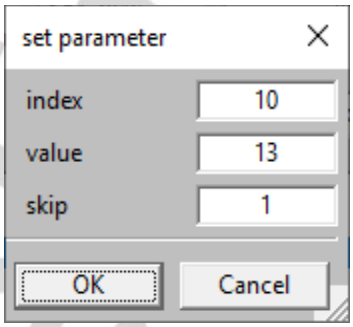


Abbildung 3.163: Condition Parameter set

3.7.2.1.4.1 Bit set ?

Ist das Bit der Variablen index gesetzt, werden skip Kommandos übersprungen.

3.7.2.1.4.2 Bit reset ?

Ist das Bit der Variablen index nicht gesetzt, werden skip Kommandos übersprungen.

3.7.2.1.4.3 Equal value

Ist der Wert der Variablen index gleich Value, werden skip Kommandos übersprungen.

3.7.2.1.4.4 Not equal value

Ist der Wert der Variablen index ungleich Value, werden skip Kommandos übersprungen.

3.7.2.1.4.5 Less value

Ist der Wert der Variablen index kleiner als Value, werden skip Kommandos übersprungen.

3.7.2.1.4.6 Greater value

Ist der Wert der Variablen index größer als Value, werden skip Kommandos übersprungen.

3.7.2.1.4.7 Equal Reference

Ist der Wert der Variablen index gleich der Variablen reference, werden skip Kommandos übersprungen.

3.7.2.1.4.8 Not equal Reference

Ist der Wert der Variablen index ungleich der Variablen reference, werden skip Kommandos übersprungen.

3.7.2.1.4.9 Less Reference

Ist der Wert der Variablen index kleiner als der Variablen reference, werden skip Kommandos übersprungen.

3.7.2.1.4.10 Greater Reference

Ist der Wert der Variablen index größer als der Variablen reference, werden skip Kommandos übersprungen.

3.7.2.2 Event Typen.

Der Event Typ bestimmt die Auslösebedingung eines Kommandos.

☑ disabled Das Kommando ist gesperrt und wird nicht aufgerufen.

☐ key down Das Kommando wird aufgerufen wenn das Keypick bei einem Mausclick in die Alternativansicht schaltet.

☐ key up Das Kommando wird aufgerufen wenn das Keypick bei einem Mausclick in die Normalansicht schaltet.

☐ key presse Das Kommando wird in der Applikationsschleife aufgerufen solange die linke Maustaste gedrückt ist. Die Geschwindigkeit der Aufrufe erhöht sich mit der Dauer des Tastendrucks.

☐ at start Das Kommando wird beim Start der Applikation einmalig aufgerufen. Diese Funktion wird normaler Weise zur Initialisierung von Variablen im Interface benutzt.

☐ at end Das Kommando wird vor dem Verlassen der Applikation einmalig aufgerufen.

☐ in loop Das Kommanda wird zyklisch während der Laufzeit einer Applikation aufgerufen.

3.8 Aktions Menü

Aktionen stellen die Reaktion eines Keypicks auf den Inhalts einer Variablen aus dem Interface während der Laufzeit dar. Dabei ändert sich die Darstellung des Keypicks in Abhängigkeit vom gewählten Mode, wobei die Parameter der Darstellung durch eine Variable im Interface bestimmt wird. Das erlaubt einem externem Programm die Form der Darstellung während der Laufzeit zu ändern. Eine Beschreibung der Modi und deren Parameter erfolgt in Kapitel 3.8.2 Aktions Eigenschaften.

Die Erstellung von Aktionen und deren Eigenschaften werden hier im folgendem beschrieben. Da Aktionen keine sichtbaren Elemente sind, gibt es logischerweise keine Darstellung im Applikationsfenster.

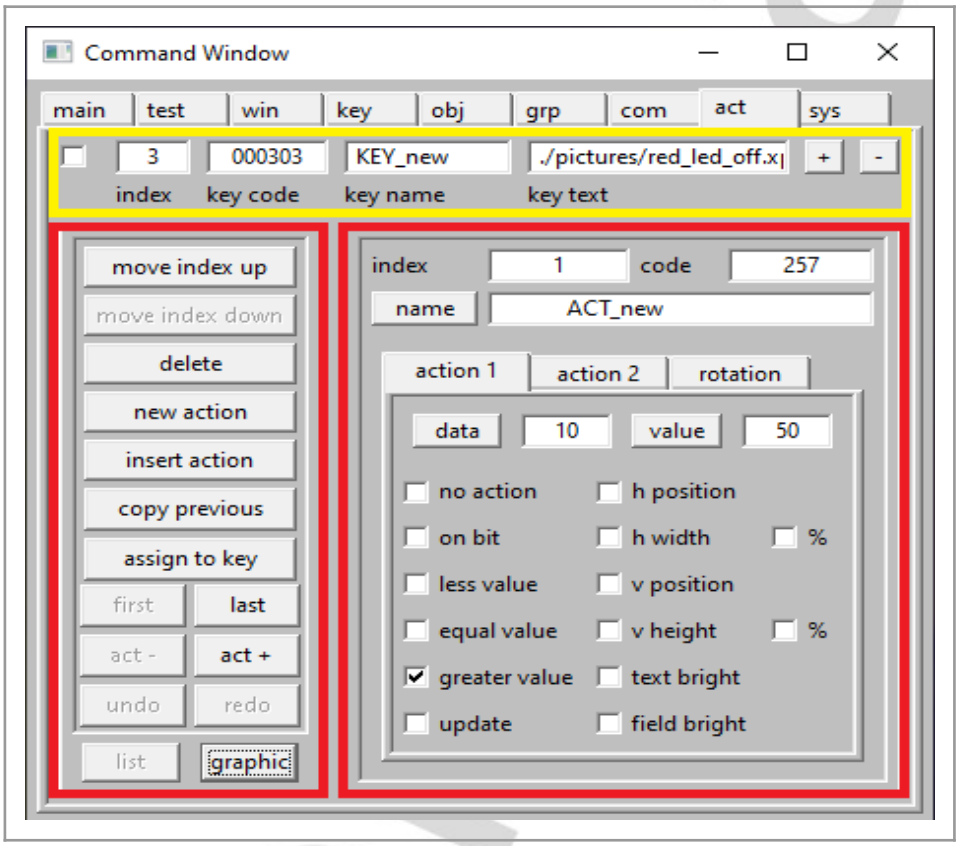


Abbildung 3.164: Aktions Menü

Wie in Abbildung 3.164 gezeigt, besteht das Aktionen Menü aus drei Hauptbereichen. Dem 'Commands' Bereich (Rot) und dem 'Eigenschaften' Bereich (Rot) und einem 'Keypick' Bereich (Gelb).

Der 'Commands' Bereich dient der Verwaltung der Aktionen, während der 'Eigenschaften' Bereich wie der Name schon sagt zur Einstellung der Eigenschaften der Aktionen vorgesehen ist.

Neu ist hier der 'Keypick' Bereich welcher die Daten des Keypicks anzeigt welches gerade selectiert , oder der aktuellen Aktion zugeordnet ist.

3.8.1 Aktions Verwaltung

Wie schon vorab beschrieben sind alle Elemente in Feldern abgelegt. Auf die Aktionen kann über

deren Feldindex zugegriffen werden. Der Index startet mit 1 und reicht bis zur letzten erstellten Aktion.

Das Feld index 1 zeigt die Nummer der zur Bearbeitung ausgewählten Aktion an.

Mit den Buttons act - und act + kann man sich durch das Feld aufwärts und abwärts bewegen.

Mit den Buttons first und last springt man direkt zum Anfang b.z.w zum Ende der Liste.

move index down Verschiebt die aktuelle Aktion um eine Position in der Liste nach unten.

move index up Verschiebt die aktuelle Aktion um eine Position in der Liste nach oben.

Da die Aktionen in der Reihenfolge von 1 bis last während der Laufzeit ausgeführt werden, und somit eine Aktion mit höherem Index nach einer Aktion mit niedrigerem Index ausgeführt wird, kann man durch die Indexposition eine Darstellungsreihenfolge bestimmen.

delete Löscht die gerade aktuelle Aktion aus der Liste.

undo redo Erlaubt das Rückgängig machen der letzten Änderungen im Aktionen Menü.

new action Erstellt eine neue Aktion ohne Eigenschaften am Ende der Liste.

copy previous Kopiert die Eigenschaften der vorherigen Aktion in die aktuelle Aktion.

insert action Fügt eine neue Aktion ohne Eigenschaften hinter der aktuellem Aktion ein.

Alternativ zu der oben aufgeführten Art der Bedienung über die Buttons, können einige Funktionen auch über Popup Menü aufgerufen werden. Befindet sich der Mauscursor im Applikationsfenster und die rechte Maustaste wird betätigt erscheint das Kommando Pop Up Menü (Abbildung 3.165).

Mit 'copy' wird die aktuelle Aktion mit allen seinen Eigenschaften in das Clipboard kopiert.

Mit 'paste' wird der Inhalt des Clipboards hinter dem Index der aktuellen Aktion eingefügt.

Die Funktionen 'delete' und 'undo' entsprechen den Button delete und undo.

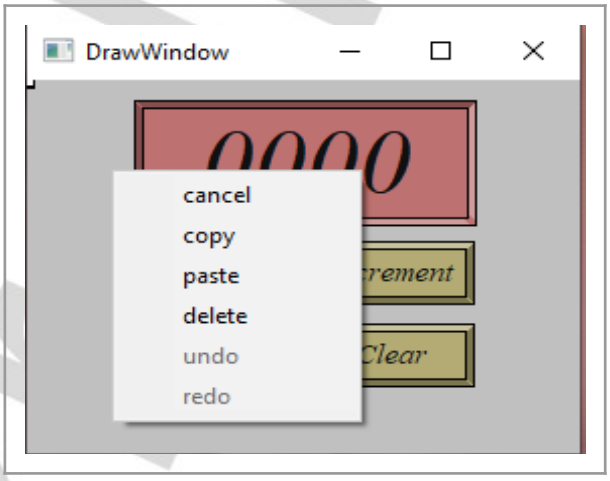


Abbildung 3.165: Kommando Pop Up Menü

assign to key verknüpft die aktuelle Aktion mit dem im Fehler: Verweis nicht gefunden (Fehler: Verweis nicht gefunden) gesetztem Keypick, welches auch im Applikationsfenster markiert ist. Die Auswahl eines Keypicks erfolgt mit den Tasten - + oder durch das Anclicken des gewünschten Keypicks im Applikationsfenster.

graphic Öffnet das Fenster mit der Graphische Darstellung des aktuellen Aktionen und dessen Eigenschaften (Abbildung 3.166) welche in Abbildung 3.167 erläutert wird.

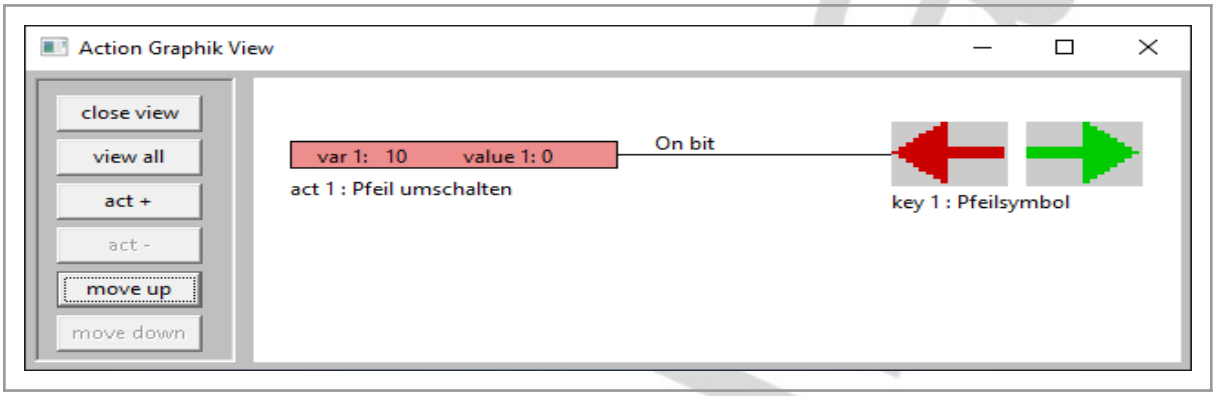


Abbildung 3.166: Graphische Darstellung des aktuellen Aktionen

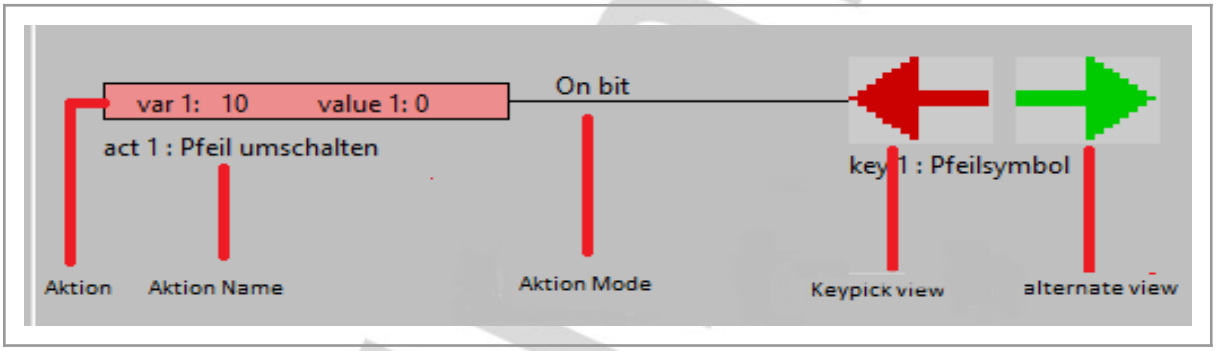


Abbildung 3.167: Elemente der Aktionsdarstellung

Im rechten Teil wird das 'Keypick' mit dem Namen und der alternativen Ansicht dargestellt welches mit dem Keypick verknüpft ist. Das linke Element zeigt die Aktion und die Parameter der Bedingung für die Auslösung der Aktion an. Die Verbindung zwischen den Elementen entspricht der Bedingung zur Auslösung der Aktion.

Die Button act + act - move up move down haben auch hier die gleichen Funktionen wie die entsprechenden Button im Kommando Bereich.

Show all Schaltet die Anzeige in die in Abbildung 3.170 gezeigte Darstellung um. Hier werden alle Aktionen in der Reihenfolge ihres Indizes aufgeführt. Das aktuelle Aktion ist hier wieder rot markiert. Mit der linken Maustaste kann eine Aktion zur Bearbeitung ausgewählt werden. Mit der rechten Maustaste öffnet sich ein Pop Up Menü mit den gleichen Funktionen wie das Pop Up Menü im Applikationsfenster welches schon weiter oben beschrieben wurde.

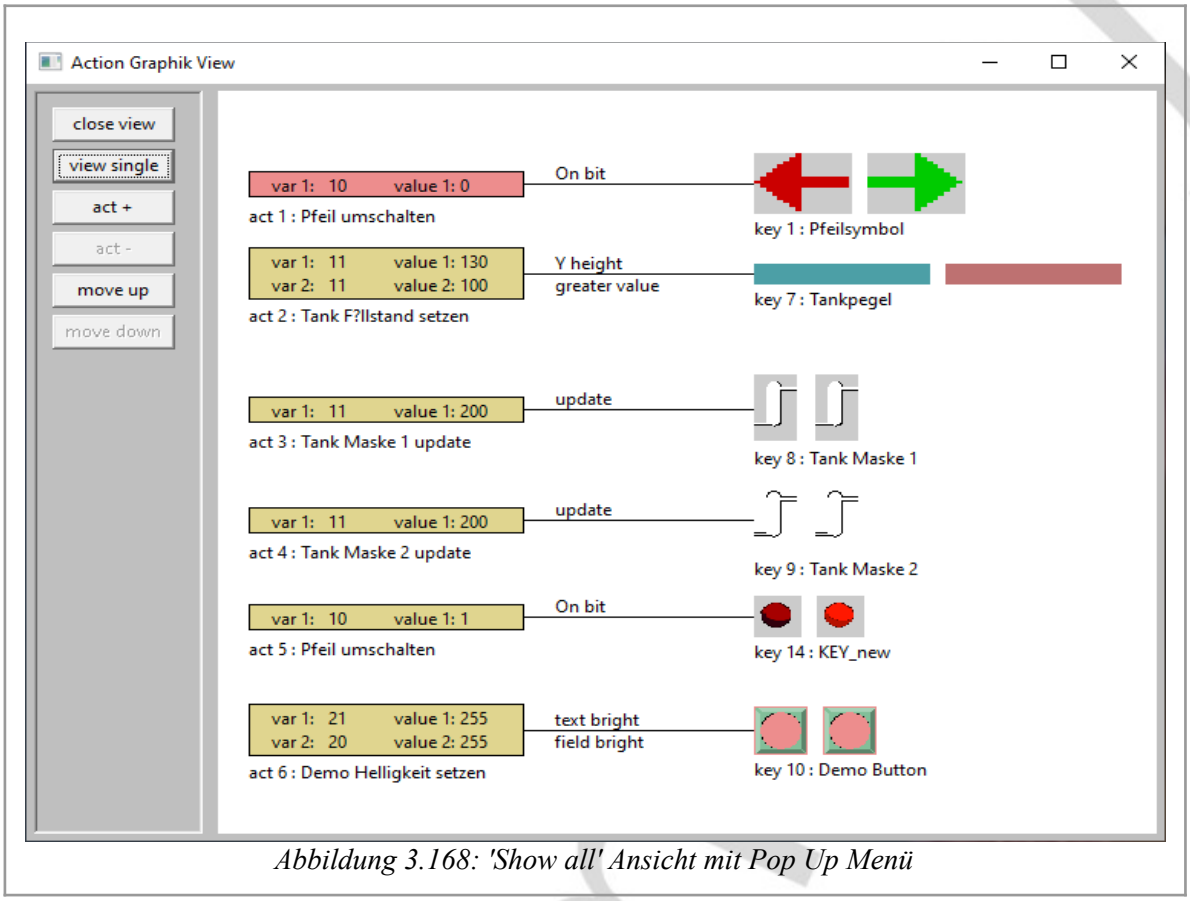


Abbildung 3.168: 'Show all' Ansicht mit Pop Up Menü

Show selected Setzt die Anzeige wieder in die Einzeldarstellung zurück.

3.8.2 Aktions Eigenschaften

Die Eigenschaften sowie die Auslösebedingung einer Aktion werden durch den action mode 1 und action mode 2, sowie den dazugehörigen Variablen data und value bestimmt.

data Bestimmt die Variable im Interface von welcher die Darstellung des zugeordneten Keypicks abhängig ist.

value Bestimmt den Parameter der Abhängigkeit.

action mode Bestimmt die Art der Abhängigkeit von der Variablen.

Mit dem Button name kann man die Aktion mit einem Namen versehen, der zwar keine Bedeutung für deren Funktion hat, aber bei der Verwaltung hilfreich sein kann. Die Aktivitäten Bedingungen werden mit dem in Abbildung 3.169 dargestelltem Menü gesetzt.

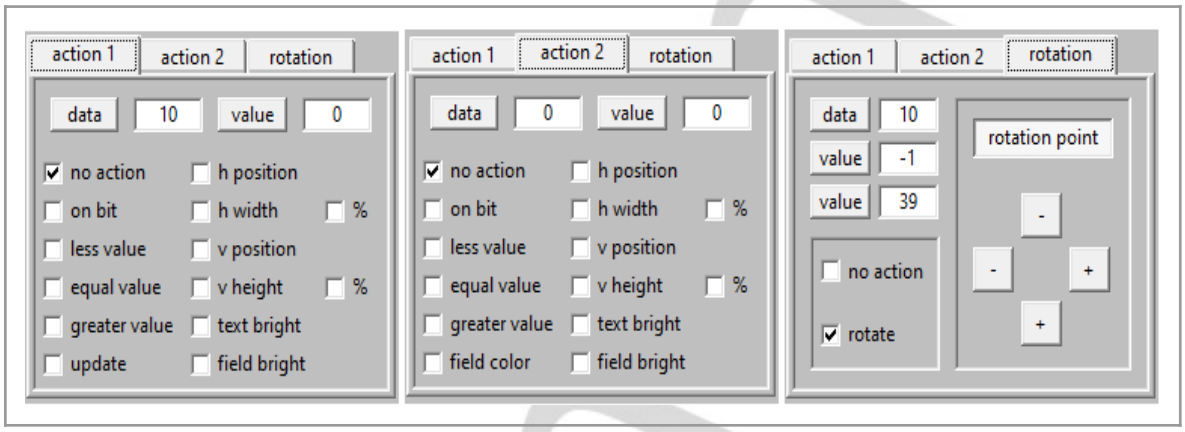


Abbildung 3.169: Aktivitäten Bedingungen

Einer Aktivität können jeweils 3 Eigenschaften zugeordnet werden. Mit den Button action 1, action 2 und rotation kann zwischen den Eingaben action mode 1, action mode 2 und action rotation gewechselt werden. Es kann in action mode 1 und action mode 2 nur jeweils eine Bedingung gesetzt werden. action rotation stellt nur eine Rotation des Keypicks zur Verfügung. ***Der Refresh Mode des Keypicks welches mit der Aktivität verbunden ist sollte nicht gesetzt sein, um undefinierte Anzeigen zu vermeiden***. Wenn noch keine der 3 Aktivitäten gesetzt wurde erscheint ein leeres Aktivitätsmenue wie in Abbildung 3.170 dargestellt. Mit dem Mausclick auf eines der Label action 1 action 2 rotation kann die gewünschte Eingabe geöffnet werden. Wenn in eine oder mehrere Aktivitäten gesetzt sind, wird das entsprechende Label als Button dargestellt (siehe Abbildung 3.169).

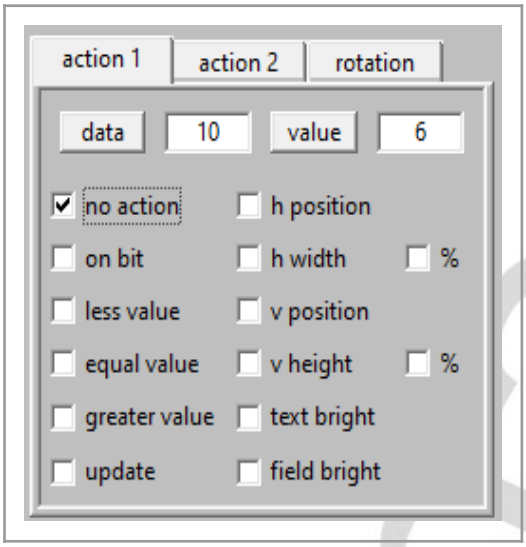


Abbildung 3.170: leeres Aktivitätsmenue

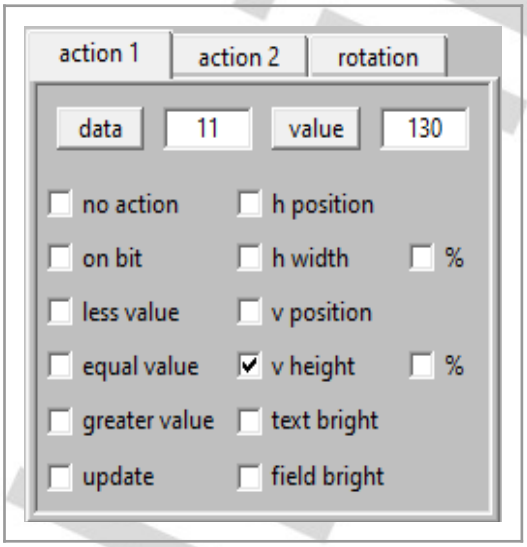


Abbildung 3.171: Acction mode 1

Die Bedeutung der einzelnen Aktivitäten werden hier am Beispiel von action mode 1 beschrieben da der Unterschied nur im Fehlen des update Feldes in action mode 2 besteht.

No Action Sperrt die Aktivität. Es wird immer die Normalansicht des Keypicks dargestellt.

Die Action Modes On bit, less value, equal value und greater value bewirken die alternative Darstellung eines Keypicks wenn die gewählte Bedingung erfüllt ist. Die Tabelle 16 zeigt Beispiele der Darstellung der Keypicks bei verschiedenen Keypick Modi.

On bit Die Bedingung ist erfüllt wenn der in value definierte Wert des Bits in der Variablen data eins ist.

less value Die Bedingung ist erfüllt wenn Wert in der Variablen data kleiner als der in value definierte Wert ist.

equal value Die Bedingung ist erfüllt wenn Wert in der Variablen data gleich dem in value

definierte Wert ist.

greater value Die Bedingung ist erfüllt wenn Wert in der Variablen data größer als der in value definierte Wert ist.

update Führt ein bedingungsloses Update des Keypicks aus, das im Gegensatz zum refresh mode des Keypicks eine Bestimmung des refresh Zeitpunkts erlaubt.

Die Auswirkung der oben beschriebenen Aktivitäten auf ein Keypick sind in der Tabelle 16 als Beispiel aufgeführt.

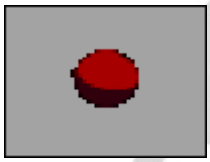
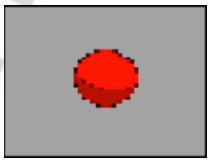
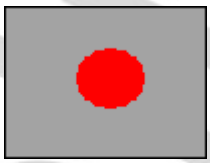
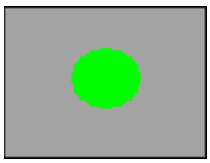
Mode	Normalansicht	Alternative Ansicht
	 Keypick Mode Button	 Keypick alternate Mode Button Invertiert
On bit oder less value oder equal value oder greater value	 Keypick entspricht der Pixmap Datei red_led_off.xpm	 Keypick alternate entspricht der Pixmap Datei red_led_on.xpm
	 Keypick entspricht der Bitmap Datei point3.xbm mit der Textfarbe Rot	 Keypick alternate entspricht der Bitmap Datei point3.xbm mit der Textfarbe Grün

Tabelle 16: Keypick Darstellung

Die Action Modes X position, X width, Y position, Y height, text bright und field bright wirken sich im Gegensatz zu den vorab beschriebenen Aktivitäten sowohl auf die Normaldarstellung als auch auf die Alternativdarstellung des Keypicks aus. In der Tabelle 17 sind einige Beispiele für diese Modi dargestellt. Um diese Darstellungen zu erhalten muss dem Keypick Bereich ein Window mit aktiviertem Refresch Mode unterlegt sein. In den gezeigten Beispielen ist das Window durch einen schwarzen Rand gekennzeichnet.

X position Die horizontale Position des Keypicks im Applikationsfenster wird um den Wert der Variablen data verschoben. Der Wert value dient dabei als Maximalwert auf den die Verschiebung begrenzt wird.

X width Die Breite des Keypicks im Applikationsfenster wird um den Wert der Variablen data vergrößert. Der Wert value dient dabei als Maximalwert auf den die Vergrößerung begrenzt wird.

Y position Die vertikale Position des Keypicks im Applikationsfenster wird um den Wert der Variablen data verschoben. Der Wert value dient dabei als Maximalwert auf den die Verschiebung begrenzt wird.

Y height Die Höhe des Keypicks im Applikationsfenster wird um den Wert der Variablen data vergrößert. Der Wert value dient dabei als Maximalwert auf den die Vergrößerung begrenzt wird.

text bright Die Helligkeit des Textes oder der Bitmap des Keypicks entspricht dem Wert der Variablen data. Der Wert value dient dabei als Maximalwert für die Helligkeit. Die größte Helligkeit entspricht dem Wert 255.

field bright Die Helligkeit des Keypicks entspricht dem Wert der Variablen data. Der Wert value dient dabei als Maximalwert für die Helligkeit. Die größte Helligkeit entspricht dem Wert 255.

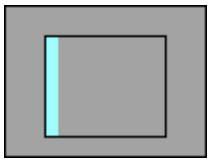
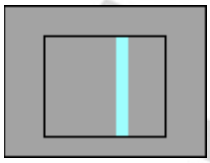
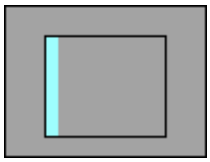
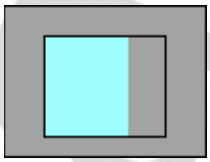
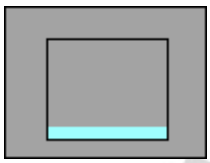
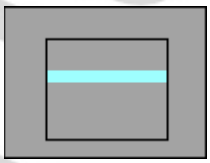
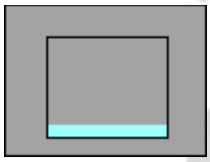
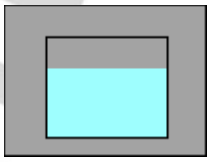
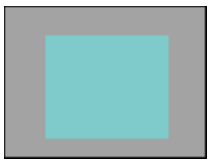
Mode	Normalansicht	Aktive Ansicht
X position		
X width		
Y position		
Y height		
text bright		
field bright		

Tabelle 17: Keypick Darstellung

Mit dem action rotation Mode wird das Keypick um den Winkel des Wertes der Variablen data um einen durch die Werte rpx und rpy definierten Rotationspunkt gedreht wenn rotate gesetzt ist. Die Einstellung erfolgt im Rotaion Setup (Abbildung 3.172). Der Winkel der Drehung erfolgt von 0 bis 360 grad.

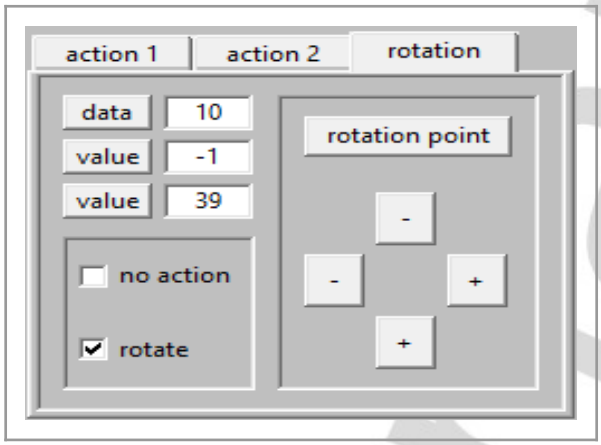


Abbildung 3.172: Rotaion Setup

Defaultmäßig liegt der Rotationspunkt in der Mitte des Keypicks, wie in Abbildung 3.173 durch den roten Punkt gekennzeichnet. Diese Darstellung entspricht den Werten rpx = 0 und rpy = 0.

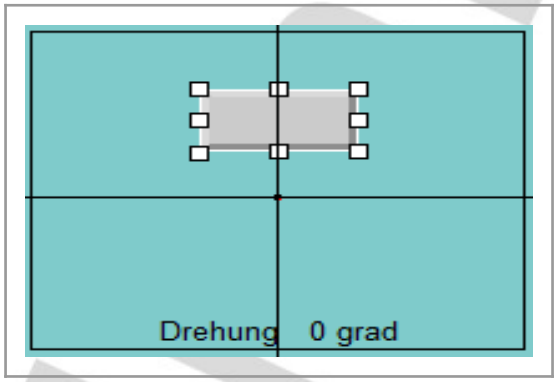


Abbildung 3.173: Rotation Point

Mit rpx und rpy kann der Mittelpunkt der Rotation an einen beliebigen Punkt des Applikationsfensters gesetzt werden. Die eingegebenen Werte werden immer relativ zum defaultmäßigem Mittelpunkt des Keypicks interpretiert. Zur Erleichterung der Einstellung des Rotationspunktes kann mit dem Button rotationpoint im Rotation Point change Menü (Abbildung 3.174) ein Fadenkreuz im Applikationsfenster eingeblendet werden, welches den aktuelle Rotationspunkt anzeigt (Abbildung 3.175: Rotations Point Marker). Dabei wechselt der Button rotationpoint in die Ansicht rotationpoint. Ein Mausclick im Applikationsfenster setzt den Rotationpoint an die aktuelle Cursorposition. Mit den Button + und - kann dann die punktgenaue Einstellung vorgenommen werden. Mit einem Click auf rotationpoint wird die Anzeige des Rotationspunkt Cursors deaktiviert.

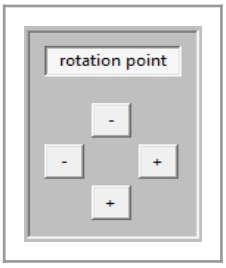


Abbildung 3.174: Rotation Point change Menü

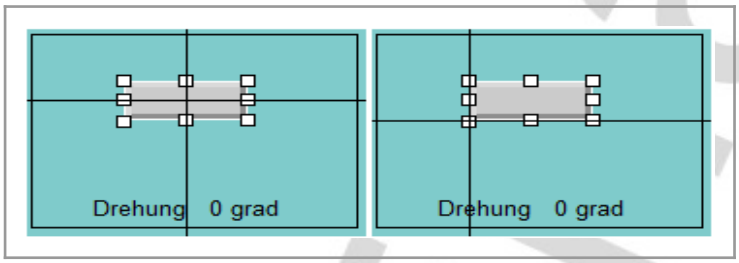


Abbildung 3.175: Rotations Point Marker

In der Tabelle 18 sind einige Beispiele für die Auswirkung der Rotation mit verschiedenen Mittelpunkten dargestellt.

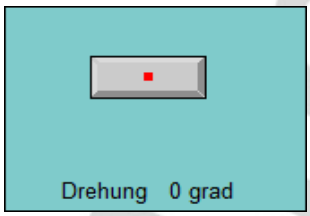
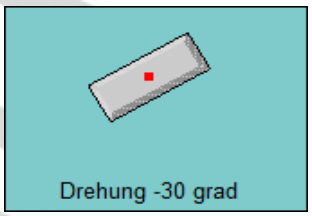
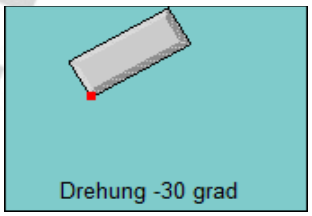
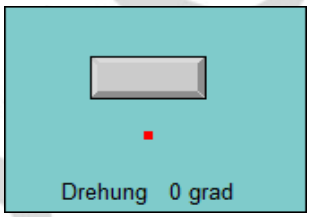
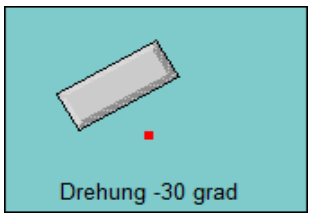
Normalansicht	Aktionsdarstellung
	
	
	

Tabelle 18: Rotation Beispiele

3.9 Gruppen Menü

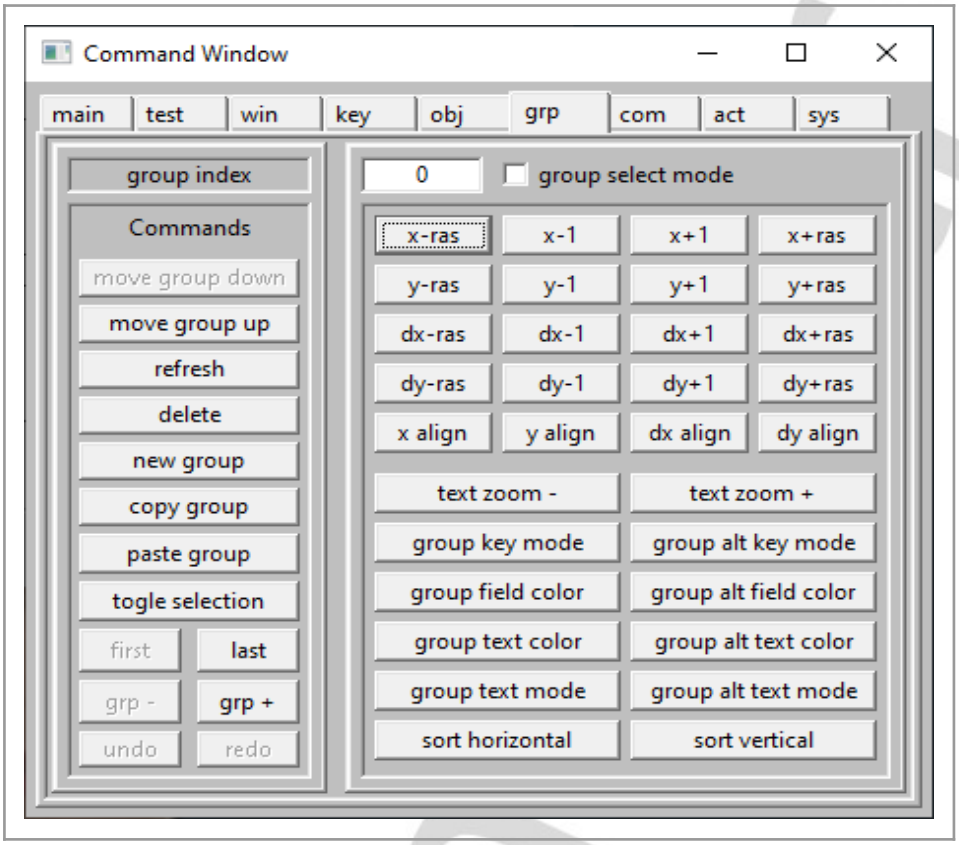
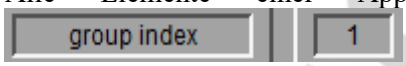


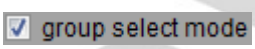
Abbildung 3.176: Gruppen Menü

Gruppen stellen kein neues Element zur Verfügung, sondern erlauben lediglich das Verändern von Eigenschaften von Windows, Keypicks und Objekten welche der gleichen Gruppe angehören. Zu einer Gruppe gehören alle Elemente die den gleichen Gruppenindex besitzen. Die Gruppenzugehörigkeit kann beim Erstellen der Elemente gesetzt werden oder im Gruppenmenü geändert werden. Wurde beim Erstellen der Elemente kein Gruppenindex gesetzt, dann gehören alle Elemente zur GruppeAuch hier ist das Menü in die zwei Bereiche Kommandos und Eigenschaften eingeteilt, wobei der Kommandobereich der Verwaltung der Gruppen und der Eigenschaftsbereich der Änderung der Eigenschaften der Elemente einer Gruppe dient.

3.9.1 Gruppen Verwaltung

Alle Elemente einer Applikation sind einer Gruppe zugeordnet welche im 

angezeigt wird. Gleichzeitig werden alle zu dieser Gruppe gehörenden Elemente im Applikationsfenster wie in Abbildung 3.177 gezeigt markiert. Wird der angezeigte Index durch Tastatureingabe geändert, und mit der 'Return' Taste bestätigt, werden alle selektierten Elemente der eingegebenen Gruppe zugeordnet.

Wenn 

gesetzt ist kann durch das Anklicken eines Elementes die Gruppe ausgewählt werden zu der dieses Element gehört. Ist

nicht gesetzt, wird das gewählte Element der gerade aktuellen Gruppe hinzugefügt. Wird die linke Maustaste gedrückt gehalten, kann die ganze Gruppe mit der Maus verschoben werden.

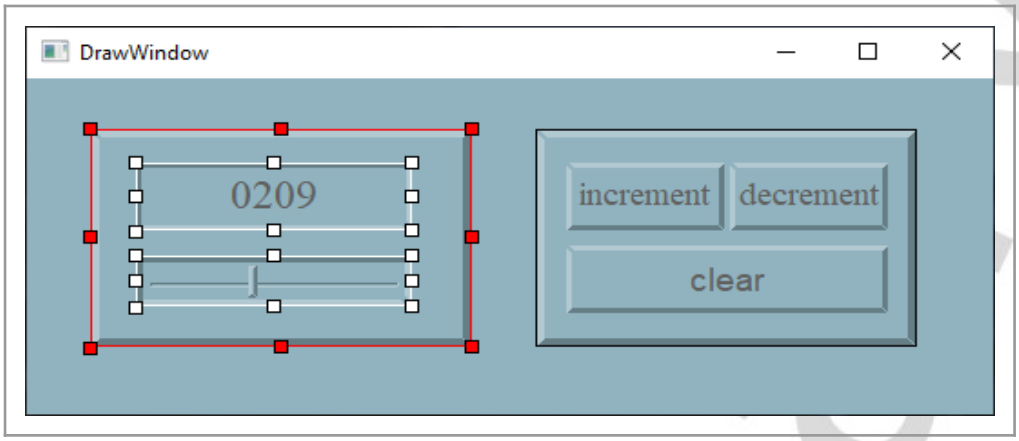


Abbildung 3.177: Gruppen Markierung

move group dowr und move group up verschieben den Index der Gruppe nach unten, b.z.w nach oben.

refresh Erzwingt ein Neuzeichnen der Applikation.

delete Löscht alle Element der Gruppe. ***! Achtung zur Zeit ist ein UNDO noch nicht möglich.***

new group Erstellt eine neue Gruppe am Ende der Liste, welcher dann auf die oben beschriebene Art Elemente zugewiesen werden können.

copy group kopiert alle Elemente einer Gruppe ins Clipboard.

insert group fügt am Ende der Liste eine neue Gruppe ein die alle Elemente aus dem Clipboard enthält.

togle selection schaltet die Markierung der Gruppe im Applikationsfenster aus / ein.

first Spring auf den Anfang der Gruppenliste.

last Spring auf das Ende der Liste.

grp - grp + Erlauben die Auswahl der gewünschten Gruppe.

3.9.2 Gruppen Eigenschaften

x-ras x-1 x+1 x+ras Verschieben alle Elemente einer Gruppe in der Horizontalen.

y-ras y-1 y+1 y+ras Verschieben alle Elemente einer Gruppe in der Vertikalen.

dx-ras dx-1 dx+1 dx+ras Ändern die Breite aller Elemente der Gruppe.

dy-ras dy-1 dy+1 dy+ras Ändern die Höhe aller Elemente der Gruppe.

Die folgenden Tasten des Gruppenmenüs haben nur eine Auswirkung auf die Keypicks einer Gruppe.

x-align y-align Es werden alle Elemente des gleichen Typs einer Gruppe jeweils auf die horizontale b.z.w vertikale Anfangsposition des ersten Elementes (das Element mit dem niedrigstem Index) dieses Typs gesetzt.

dx-align dy-align Es werden alle Elemente des gleichen Typs einer Gruppe jeweils auf die Breite b.z.w Höhe des ersten Elementes (das Element mit dem niedrigstem Index) dieses Typs gesetzt.

Die folgenden Tasten des Gruppenmenüs haben nur eine Auswirkung auf die Keypicks einer Gruppe.

text zoom - text zoom + Ändert die Größe des Textes der Keypicks der Gruppe jeweils um 1.

group key mode group alt key mode es wird der Keypick Mode Dialog aus Abbildung 3.62 auf Seite 50 aufgerufen. Dabei wird allen Keypicks dieser Gruppe der gleiche Mode b.z.w der gleiche alternative Mode zugewiesen.

group field color group alt field color Es wird das Color Menü aus Abbildung 3.63 auf Seite 56 aufgerufen. Dabei wird allen Keypicks dieser Gruppe die gleiche Feldfarbe b.z.w die gleiche alternative Feldfarbe zugewiesen.

group text color group alt text color Es wird das Color Menü aus Abbildung 3.63 auf Seite 56 aufgerufen. Dabei wird allen Keypicks dieser Gruppe die gleiche Textfarbe b.z.w die gleiche alternative Textfarbe zugewiesen.

group text mode group alt text mode Es wird der Keypick Text Dialog aus Abbildung 3.64 auf Seite 56 aufgerufen. Dabei wird allen Keypicks dieser Gruppe der gleiche Textmode b.z.w die gleiche alternative Textmode zugewiesen.

sort horizontal sort vertical Erlauben das Anordnen der Keypicks einer Gruppe. Ausgehend von der in Abbildung 3.178 dargestellten Situation ergibt sich nach Aufruf sort horizontal und der im Sort Parameter Dialog (horizontal) aus Abbildung 3.179 die in Abbildung 3.180 gezeigte Anodnung.

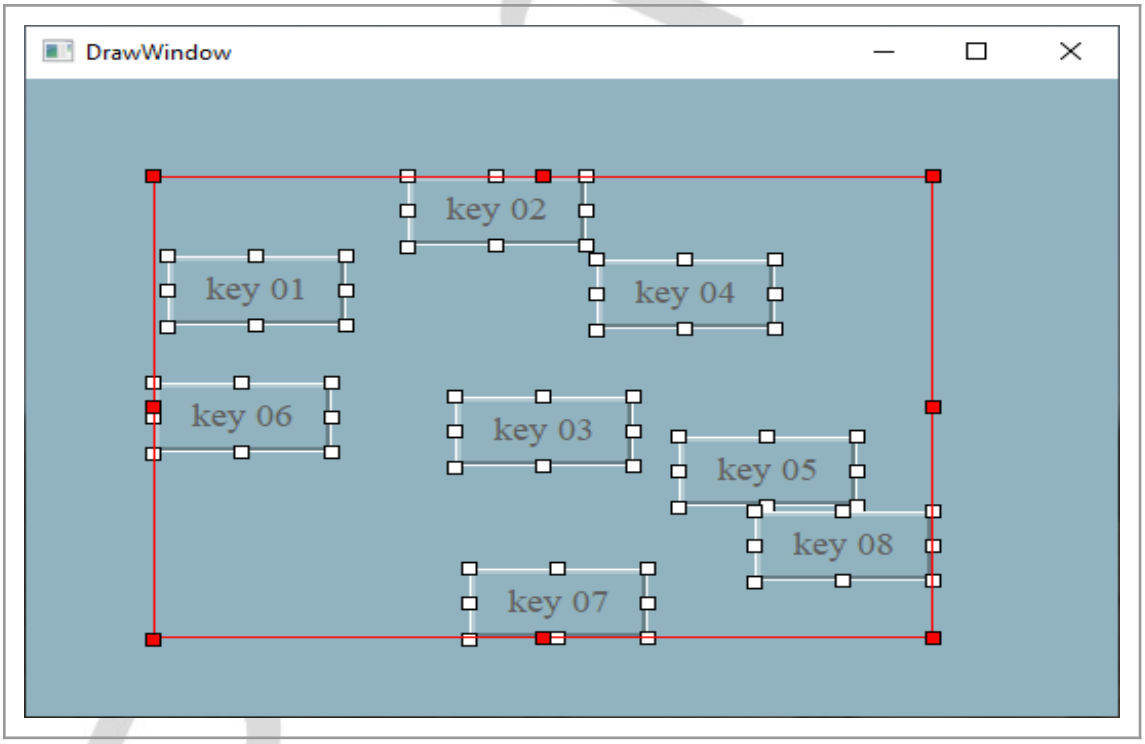


Abbildung 3.178: Unsortierte Keypick Gruppe

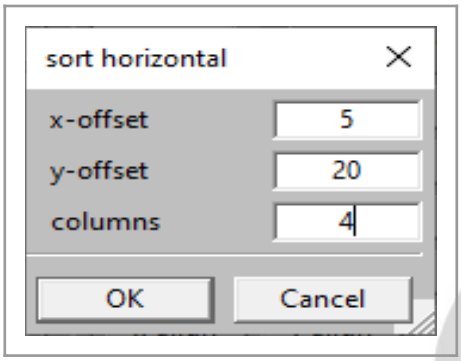


Abbildung 3.179: Sort Parameter Dialog (horizontal)

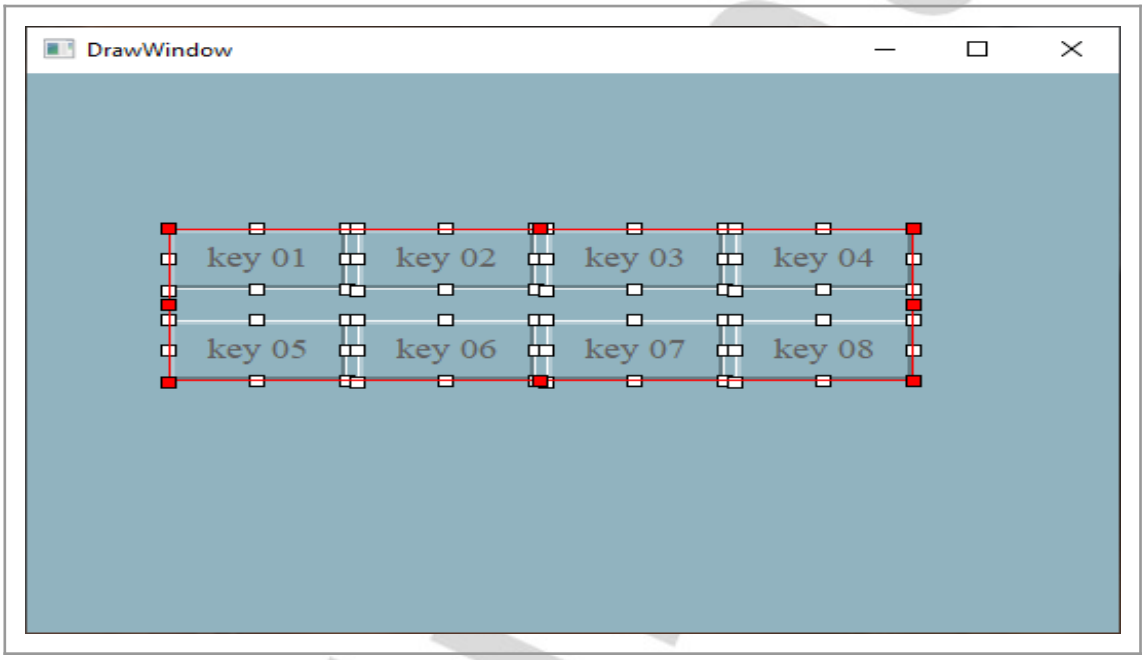


Abbildung 3.180: Horizontale Sortierung

sort vertical Mit den Parametern aus Abbildung 3.181 führt zu einer Anordnung der Keypicks wie in Abbildung 3.182 gezeigt.

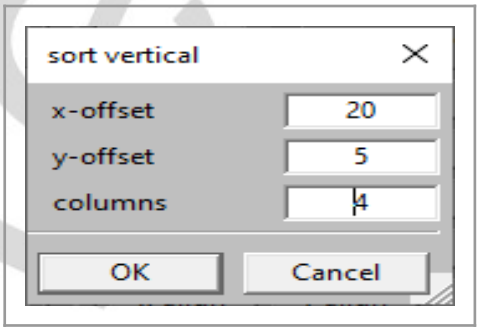


Abbildung 3.181: Sort Parameter Dialog (vertikal)

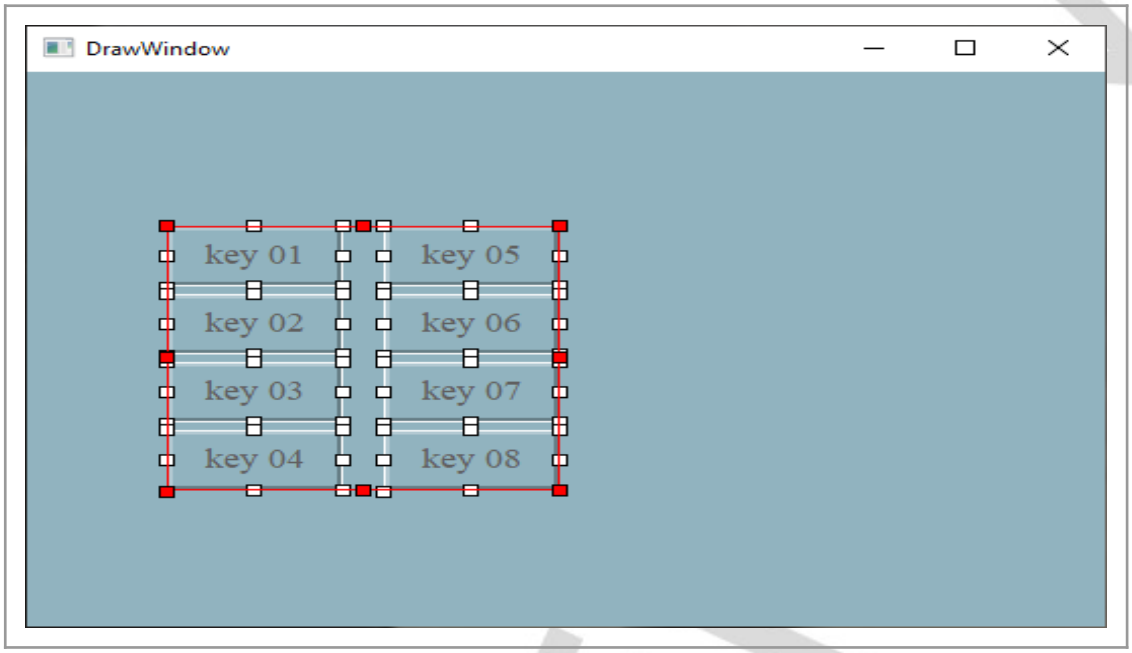


Abbildung 3.182: Vertikale Sortierung

3.10 System Menü

Das System Menü aus Abbildung 3.183 stellt Funktionen zur Verwaltung der erstellten Applikation zur Verfügung. Application Infos liefert Informationen über die zur Applikation gehörenden Dateien und verwendeten Resourcen, während Application Exports das Exportieren der Applikationsdaten in verschiedenen Formaten ermöglicht.

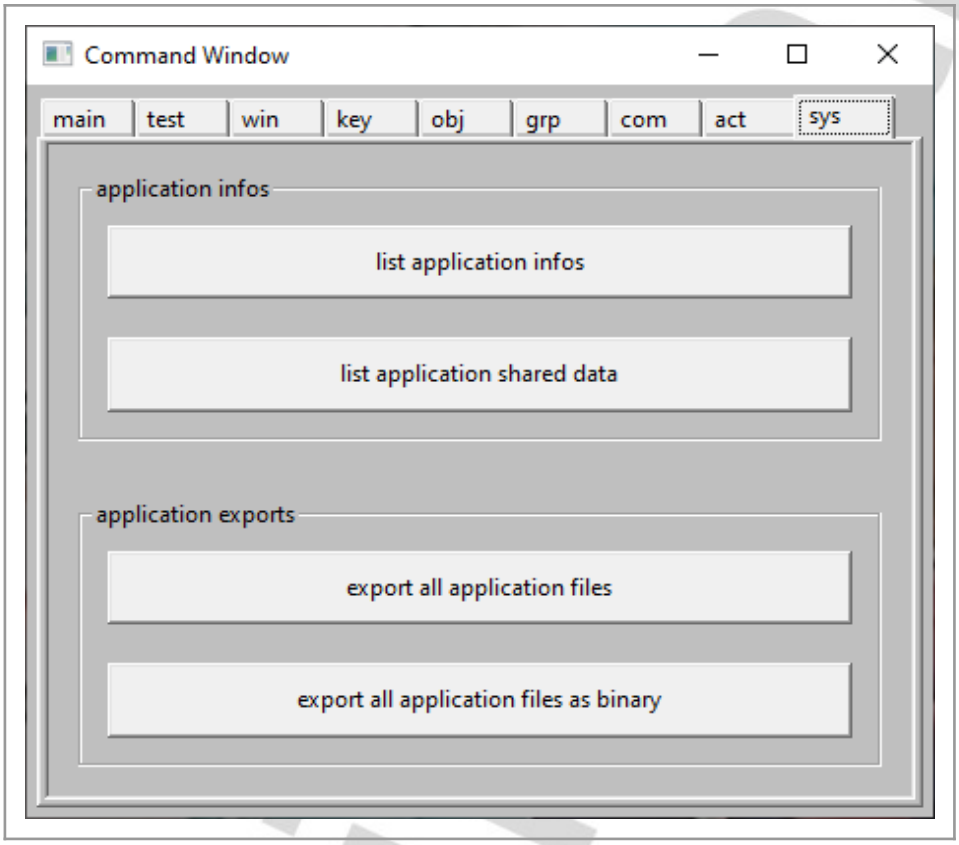


Abbildung 3.183: System Menü

Zum besseren Verständnis dieser Funktionen dient die in Abbildung 3.184, Abbildung 3.185 und Abbildung 3.186 gezeigte Demoapplikation bei der mit dem Schieber 'Füllstand' die Höhe des Pegel in 'Tank 1' geändert werden kann. Mit der Tasten 'Nach Rechts' und 'Nach Links' wird der dazwischen liegende Richtungspfeil umgeschaltet. Mit den Schiebern 'Helligkeitsregler' wird die Helligkeit der Feldfarbe sowie die Helligkeit der Textfarbe des links der Regler dargestellten Keypicks verändert. Die rote Led blinkt durch eine externe Funktion gesteuert, sobald die Applikation gestartet wird. Ein Mausclick auf den Tank öffnet ein neues Fenster mit einer großen Darstellung des Tanks (Abbildung 3.186) welche mit dem Button 'Schließen' beendet werden kann.

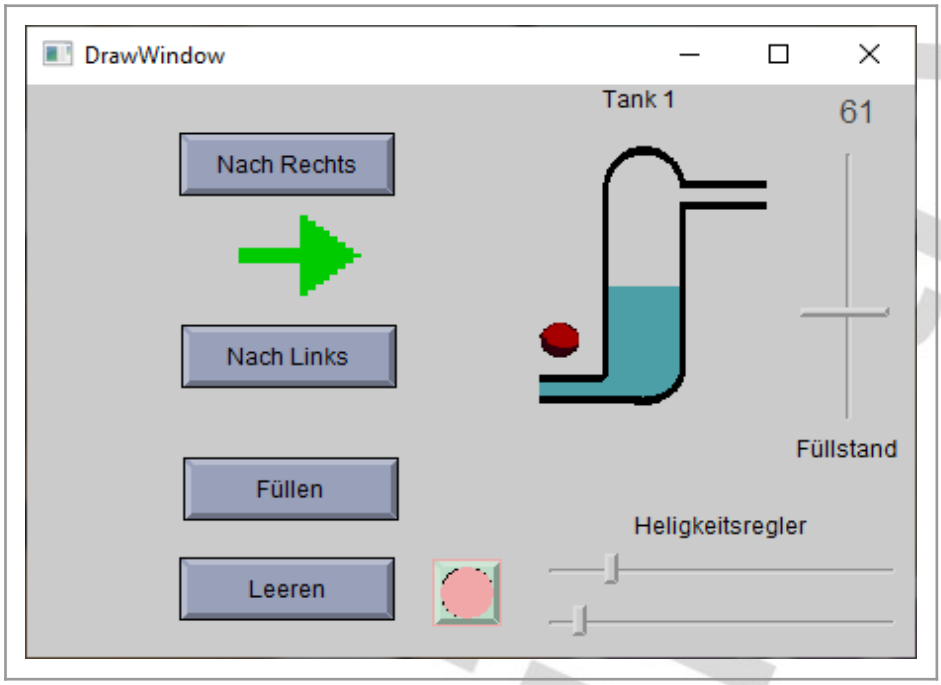


Abbildung 3.184: Beispiel Demo_01 Ansicht 1

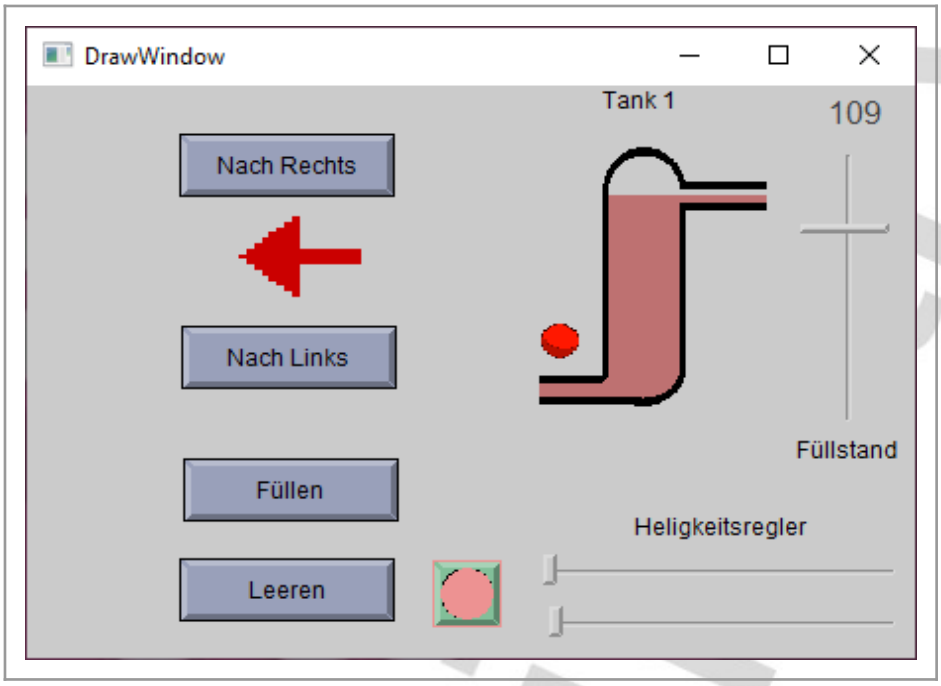


Abbildung 3.185: Beispiel Demo_01 Ansicht 2

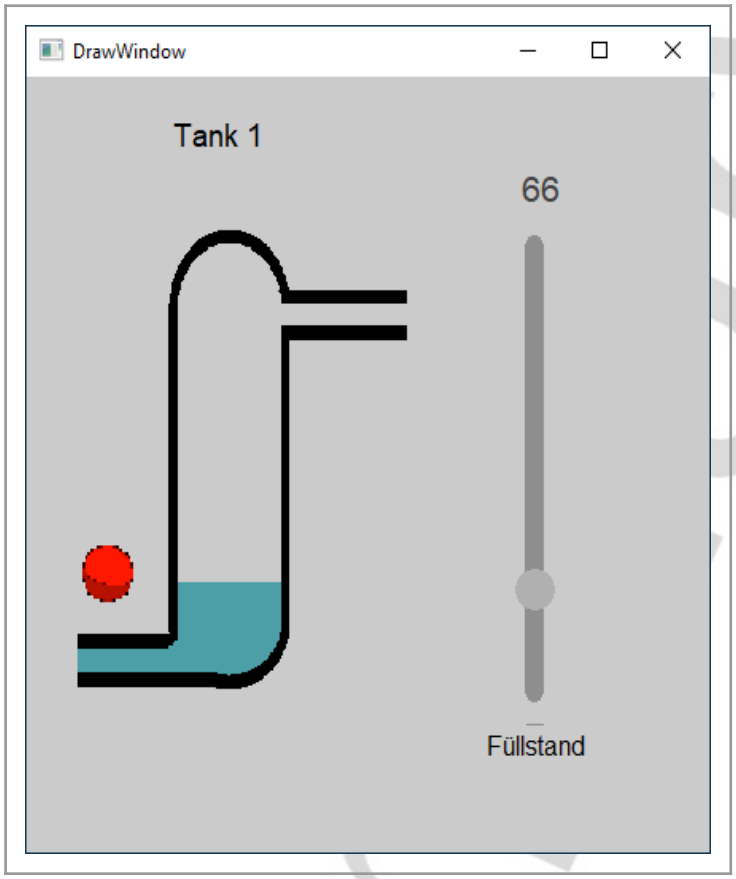


Abbildung 3.186: Beispiel Demo_01 Ansicht 3

List Application Infos Öffnet das Fenster mit der Applikation Info Ausgabe (Abbildung 3.187) in welcher alle zur Applikation gehörenden Dateien aufgelistet sind. Durch das Setzen der entsprechenden ‚checkboxen' kann man bestimmen welche Informationen angezeigt werden sollen.

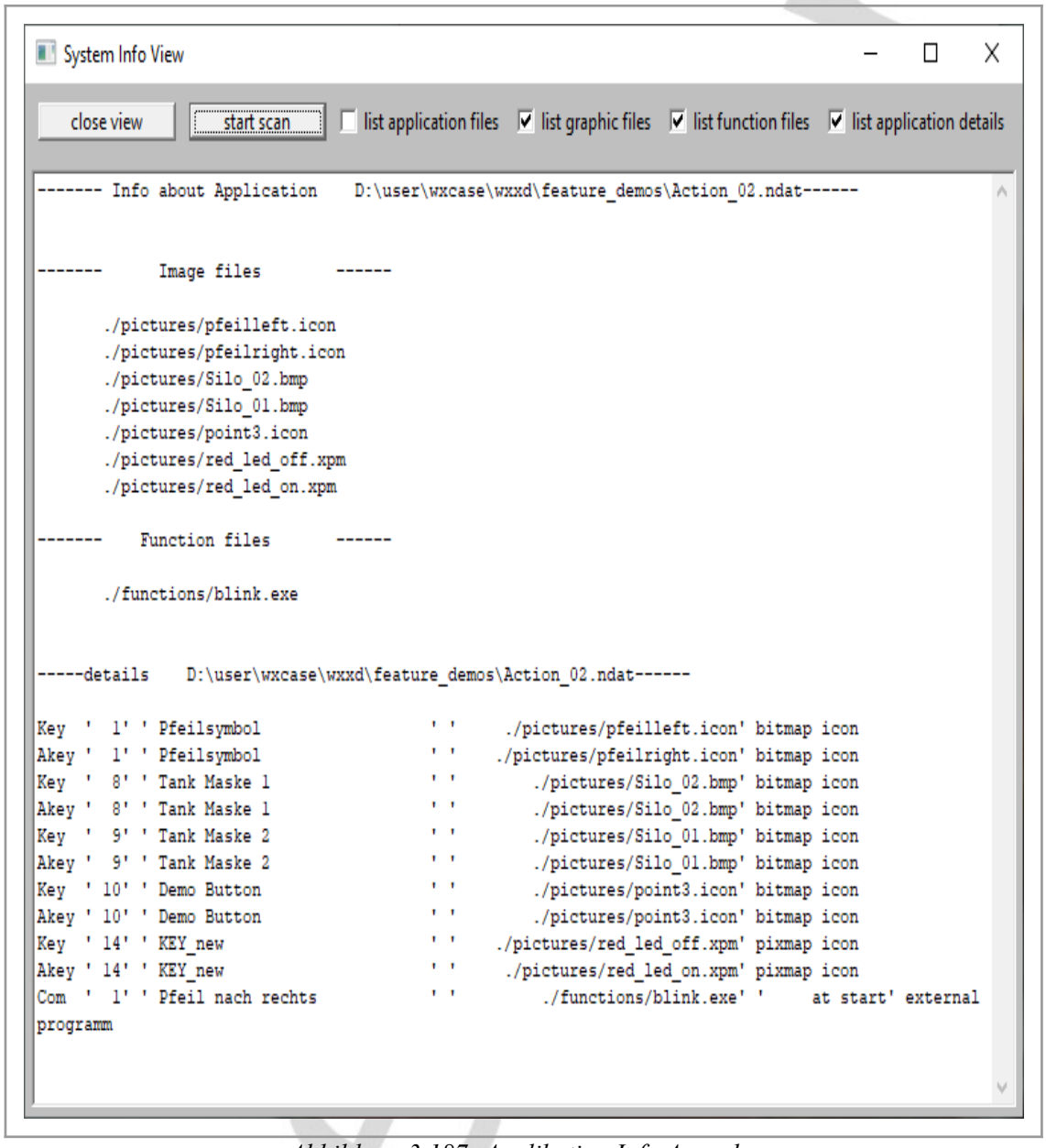


Abbildung 3.187: Applikation Info Ausgabe

List Application shared data Öffnet das Fenster mit der Applikation Shared Data Ausgabe (Abbildung 3.188) in welcher alle zur Applikation gehörenden Dateien und die von diesen belegten Variablen im Interface aufgelistet sind.

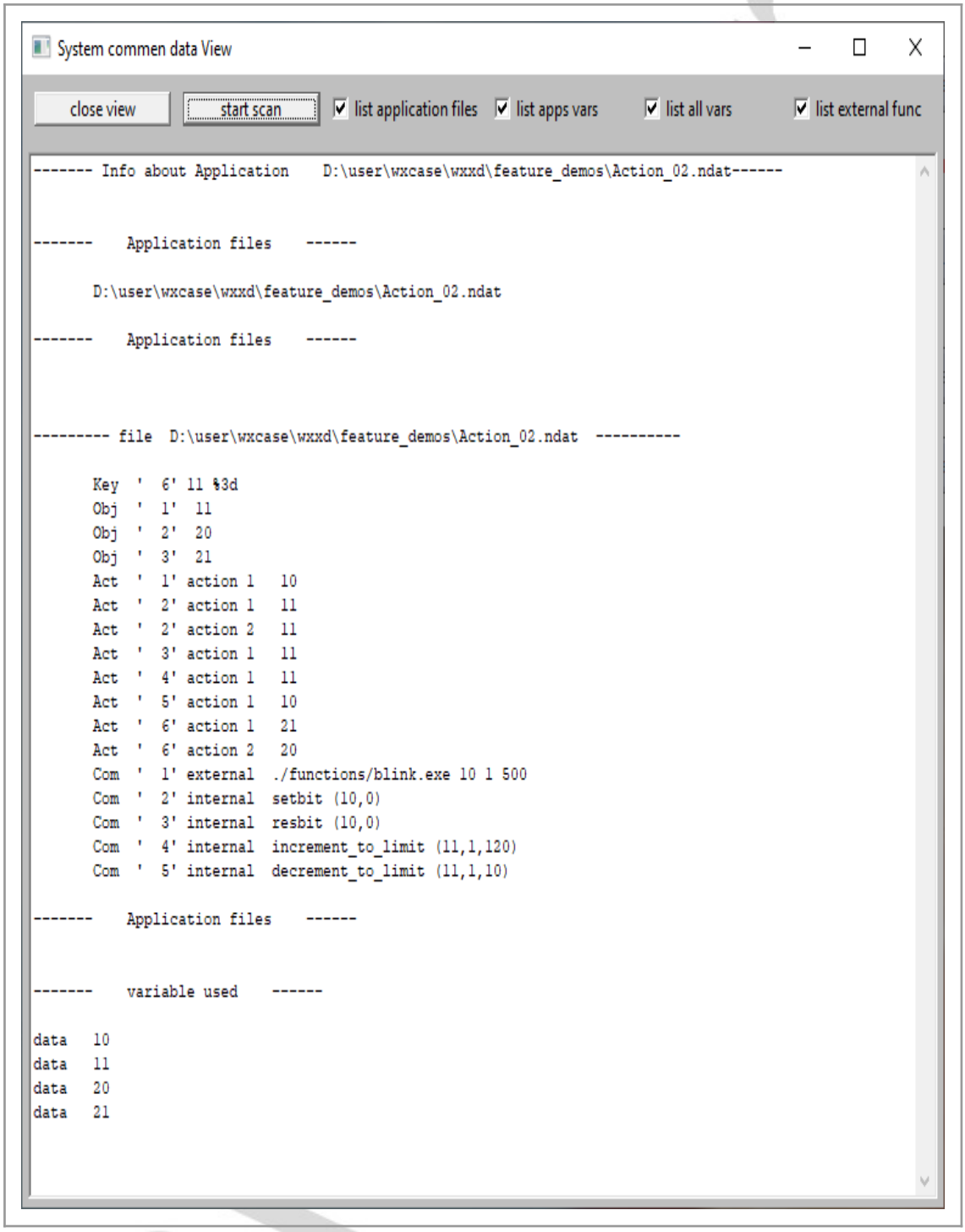


Abbildung 3.188: Applikation Shared Data Ausgabe

!

Export all Application Files Kopiert alle zur Applikation gehörenden Dateien in ein neues Zielverzeichnis. Da wie in Abbildung 3.187 (Applikation Info Ausgabe) gezeigt alle Dateien irgendwo im Dateibaum verteilt gespeichert sind, und einen absolutem Dateipfad besitzen, ist es nicht ohne weiteres möglich die Applikation auf ein anderes System zu kopieren. Deshalb können mit diesem Export alle zur Applikation gehörenden Dateien in ein neues Verzeichnis kopiert und die absoluten Dateinamen in der Applikation durch relative Pfade ersetzt werden. Es wird ein Dateibrowser zur Bestimmung des Zielverzeichnisses und ein Protokollfenster geöffnet. Im gezeigten Beispiel erfolgte der Export in das erstellte Verzeichnis 'Neuer Ordner' in welchem die Unterverzeichnisse 'pictures' und 'functions' erstellt wurden. List Application Infos liefert jetzt nach dem Öffnen der Datei 'D:/NxDemos/Neuer Ordner/Demo_01.ndat' das in Abbildung 3.189 gezeigte Ergebnis.

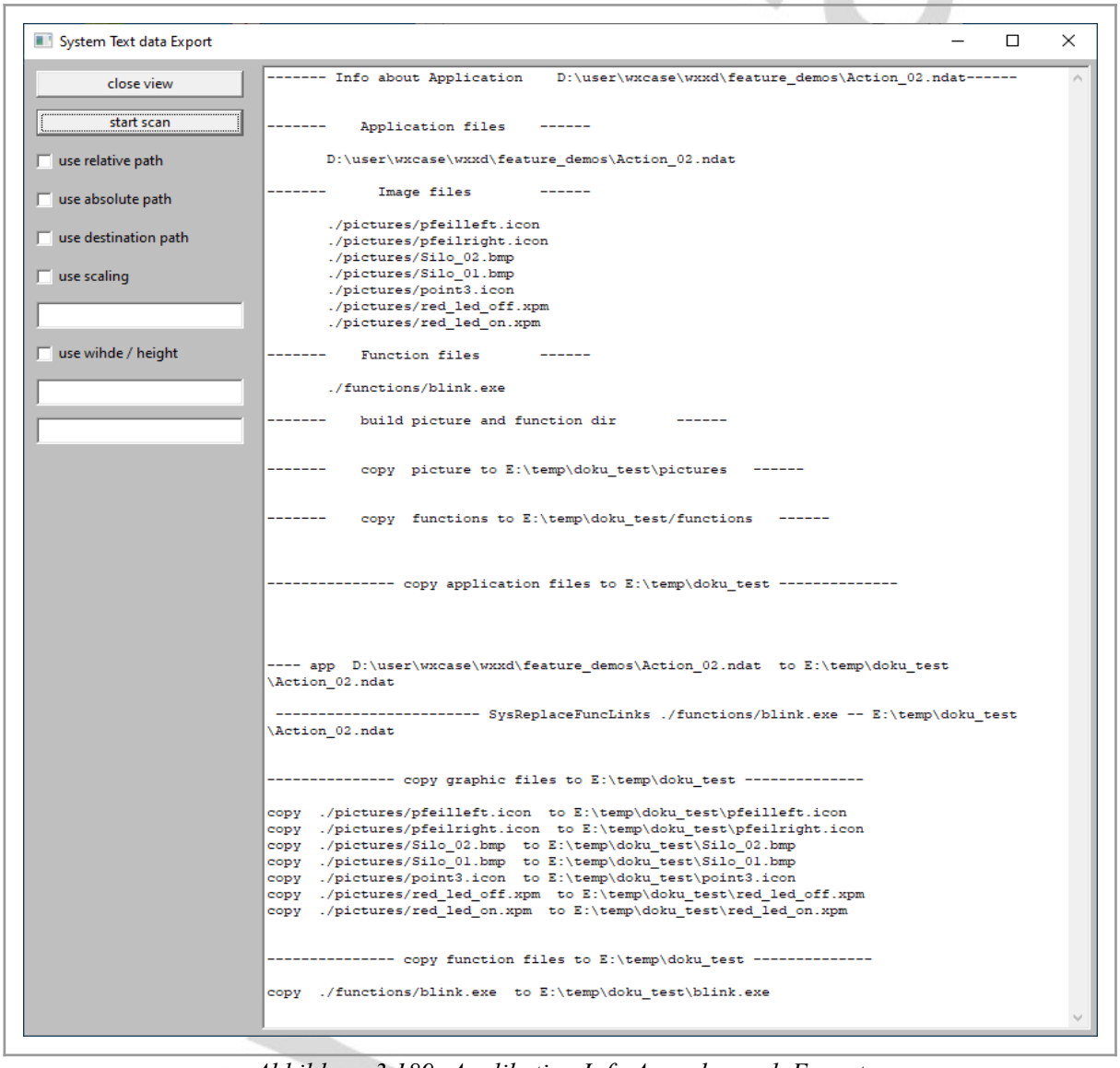


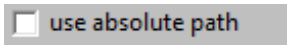
Abbildung 3.189: Applikation Info Ausgabe nach Export

Export Application as binary Files entspricht der vorab beschriebenen Funktion mit dem Unterschied das die Applikationsdaten in einem binärem Format mit der Dateiendung '.qdat' gespeichert werden.

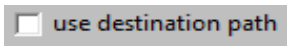
Diese Dateien können mit dem 'Runtime' Interpreter direkt ausgeführt werden.

Das Kopieren der Daten kann durch das Setzen der folgenden Optionen beeinflusst werden.

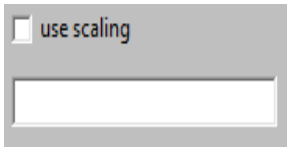
Ist diese Option ausgewählt worden, werden im Zielverzeichnis die Ordner 'pictures' und 'functions' erstellt, alle zur Applikation gehörigen Graphiken und Funktionen dort abgelegt und in den Aufrufen die Dateipfade als relative auf diese Ordner geändert. Damit ist es später möglich den Zielordner beliebig zu kopieren und an einer anderen Stelle im Dateisystem zu erstellen ohne die Funktion der Applikation zu stören.



Diese Option kopiert alle Dateien der Applikation in ein neues Zielverzeichnis und ersetzt dabei alle relativen Dateipfade durch deren absoluter Version.

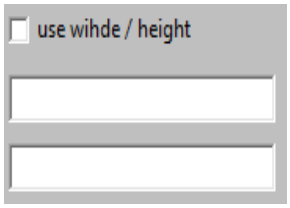


In diesem Fall werden alle Dateien ins Zielverzeichnis kopiert und auch alle Verweise auf diese angepasst.



Zusätzlich zu den oben angeführten Optionen kann hier im Textfeld ein Prozentwert eingegeben werden. Ist die Option ‚use scaling' gesetzt, werden alle Elemente der Applikation mit diesem Prozentwert scaliert in das Zielverzeichnis kopiert.

Die scalierte Version muss ewentuel noch mal händisch korrigiert werden, da es bei bestimmten Elementen oder Fontzuweisungen zu unschönen Umwandlungen kommen konnte.



Mit dieser Optionen werden alle Elemente der Applikation auf die in den beiden Textfeldern eingegebenen Werte für die Breite (oberes Feld) und Höhe (unteres Feld) in Pixeln skaliert in das Zielverzeichnis kopiert.

Die scalierte Version muss auch hier ewentuel noch mal händisch korrigiert werden, da es bei bestimmten Elementen oder Fontzuweisungen zu unschönen Umwandlungen kommen konnte.

4 RPT-One Interface

Den zentrale Bestandteil in RPT-One stellt das Interface dar. Dieses besteht aus einem Datenbereich auf den alle Keypicks, Objekte, Aktivitäten und Kommandos zugreifen können. Um auch von anderen Programmen und externen Funktionen auf diese Daten zugreifen zu können ist dieser Datenbereich als sogenanntes 'Shared Memory Segment' ausgeführt. Das bedeutet das alle Programme welche mit diesem Memory Segment verlinkt wurden, lesenden und schreibenden Zugriff auf die hier abgelegten Daten haben. Die Zugriffskontrolle auf diese Daten wird durch das Betriebssystem geregelt.

Die folgende C-Struktur zeigt die Aufteilung des 'Shared Memory Segments' dar.

```
typedef struct termda__
{
   int incount;
   int outcount;
   int status;
   char data[128];
} TERMDEF;


   typedef struct commda__
   {
      int system[10];
      TERMDEF term[10] ;
      char text[20][80];
      struct dapu_
      {
         int control ;
         int sample_time ;
         int trigger_level ;
         int da[DATPUFLEN];
      } dapu[DATPUFNUM];
      int data[1000];
   } COMMEMDEF;
```

int system[10]	ist der internen Verwaltung von Applikationen vorbehalten und darf nicht von User Programmen verändert werden.
TERMDEF term[10]	ist für den Datenaustausch von externen Programmen mit internen Terminalfenstern vorgesehen.
char text[10][80]	ist für Texte vorgesehen welche einem Keypick zur Anzeige zugewiesen werden können.
struct dapu[4]	dient zur Dateneingabe für Objekte des Typs Oszillographen Diagramm (Seite 108) und (Seite 110).
int control dapu[n]	entspricht den Kontrollbits des Datenpuffers n wie in Tabelle 12 auf Seite 107 aufgelistet.
int da[1000] dapu[n]	Datenpuffer für die Kurvenwerte der oben angegebenen Objekte.
int data[1000]	Standard Variablenbereich zum Datenaustausch von Keypicks, Objekten untereinander als auch mit externen Programmen und anderen Applikationen.

Das Zusammenspiel von RPT-One Elementen mit dem Interface soll hier an Hand mehrerer Beispiele dargestellt werden.

4.1 Datenaustausch innerhalb einer Applikation.

Das Zusammenspiel der Elemente innerhalb einer Applikation wird hier am Beispiel 1 aus Abbildung 4.1 beschrieben. Die Applikation besteht aus folgenden Elementen:

1. Einem Keypick mit dem Textmode Formatstring '%5 %03d'.
2. Einem Keypick mit dem Textmode Bitmap file 'point3.icon', der Textfarbe 'Gruen' und der alternativen Textfarbe 'Rot'.
3. Einem activen Keypick als clear Button.
4. Einem horizontalem Slider mit den 'values' 'Data index 5' und 'Interval 500'.
5. Dem internem Kommando 'setwert(5,0)' 'key down' welches dem clear Button zugeordnet ist
6. Einer Action 'greater value' 'data 5' und 'value 200' welche dem Bitmap Keypick zugeordnet ist.

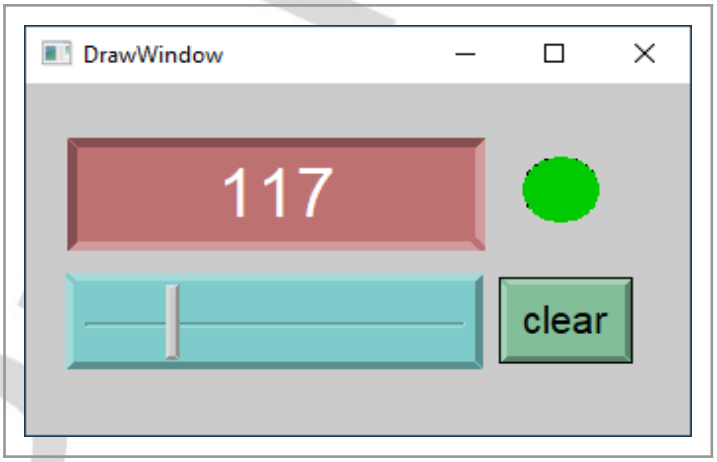


Abbildung 4.1: Beispiel 1

Wird diese Applikation gestartet, ergibt sich die in Abbildung 4.2: Beispiel 1 a dargestellte Situation. Der Wert des 'Sliders horizontal' wird in der Variablen 'Interface Integer Feld' mit dem Index 5 abgelegt und in dem Keypick 'Display' angezeigt. Da der Wert der Variablen kleiner ist als der in der action 'greater value' vorgegebene Wert 'value 1 (200)' wird das Keypick 'Warning Led' in der Normalansicht dargestellt.

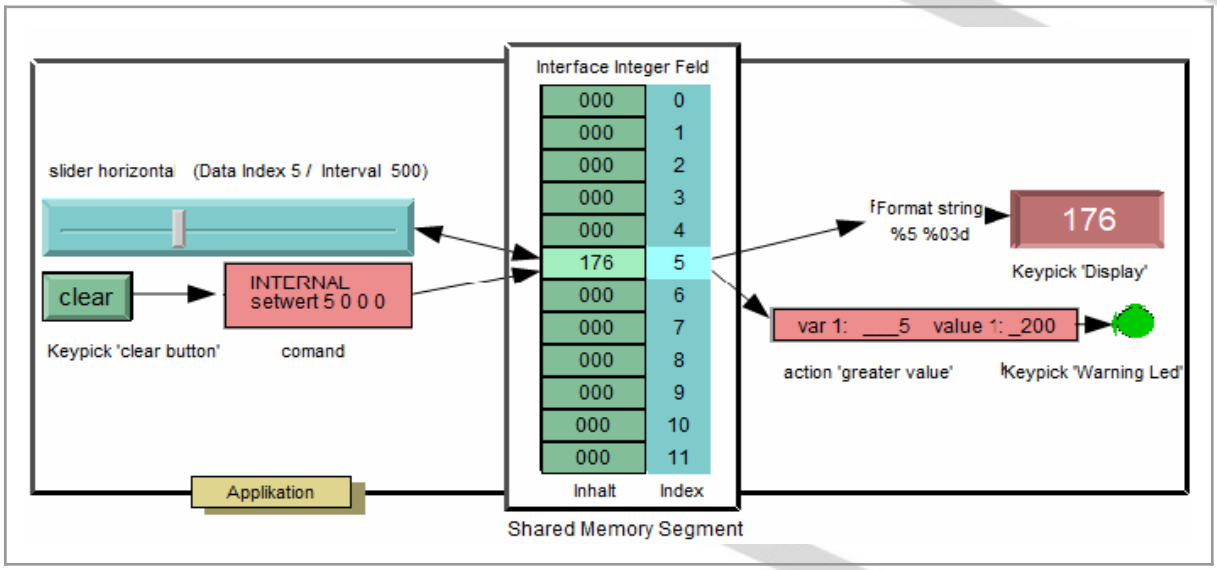


Abbildung 4.2: Beispiel 1 a

Wenn der Button 'clear' betätigt wird, wird das interne Kommando 'setwert(5,0)' ausgeführt, wodurch der Inhalt der Variablen mit dem Index 5 auf 0 gesetzt wird. Damit wird auch der Slider auf die Position 0 gesetzt (Abbildung 4.3: Beispiel 1 b).

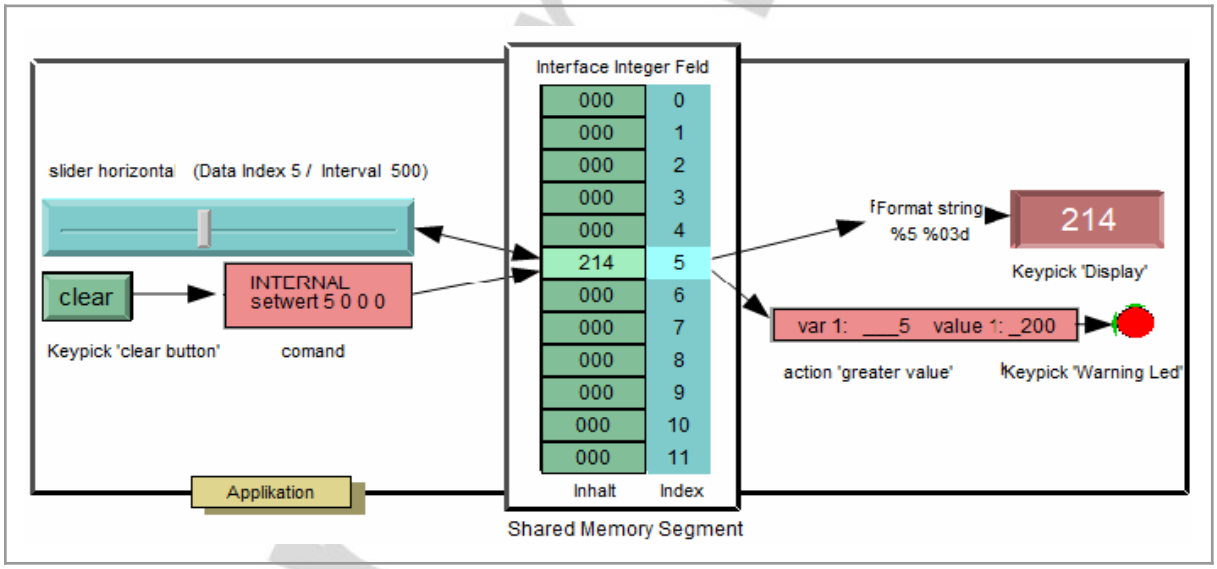


Abbildung 4.3: Beispiel 1 c

Wird der Wert der Variablen größer als der in der 'action greater value' vorgegebene Wert von 'value 1', wird statt der Normalansicht die Alternativansicht von Keypick 'Warning Led' dargestellt (Abbildung 4.3: Beispiel 1 c).

4.2 Datenaustausch zwischen Applikationen.

Das folgende Beispiel zeigt den Zugriff zweier Applikationen auf das Interface. Dazu wurde eine neue Applikation mit den folgenden Elementen erstellt:

1. Einem Objekt vom Typ 'Instrument round 1' mit dem Data Index '5' und
2. Einem Objekt vom Typ 'diagram scroll mode' ebenfalls mit dem Data Index '5'.

Werden beide Applikationen gestartet ergibt sich die in Abbildung 4.4 gezeigte Darstellung auf dem Desktop.

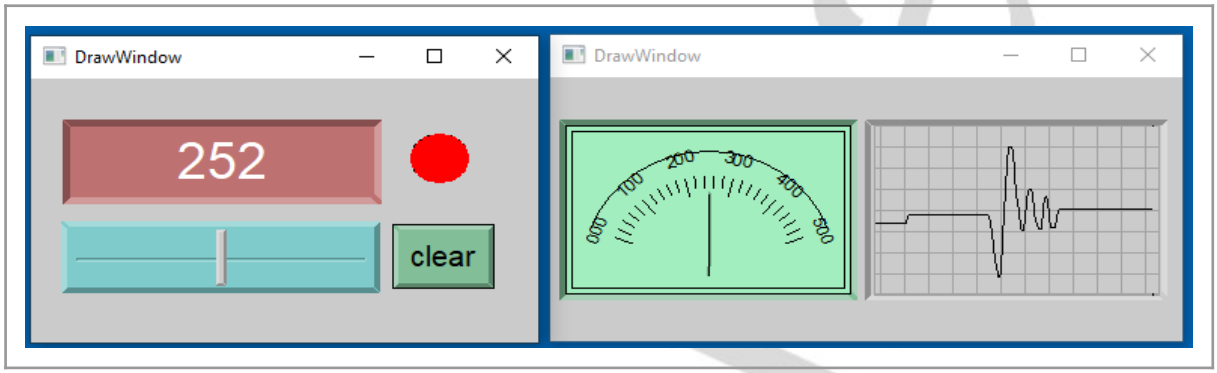


Abbildung 4.4: Beispiel 2

In Abbildung 4.5 ist das Zusammenspiel der beiden Applikationen mit dem 'Shared Memory Segment' dargestellt. Applikation 1 läuft ab wie vorab beschrieben. Applikation 2 benutzt die gleichen Daten um das Instrument zu steuern und im Srollobjekt wird der Verlauf der Schieberbewegung aus Applikation 1 aufgezeichnet.

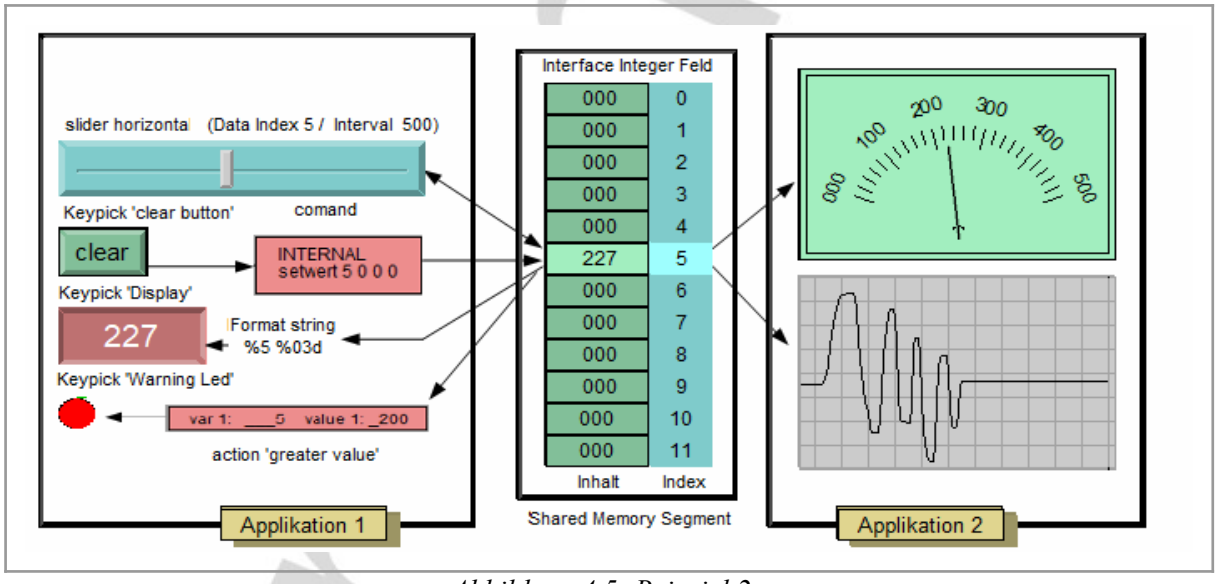


Abbildung 4.5: Beispiel 2a

4.3 Datenaustausch mit externen Funktionen.

Im nächsten Beispiel wurde die action 'greater value' in Appliktion 1 durch die action 'On Bit' mit dem Index 10 und dem Bit 0 ersetzt. Wird nun die Applikation 1 gestartet und in einem Doswindow der Befehl 'blink 10 0 500' (siehe Abbildung 4.7) gestartet, ergibt sich der in Abbildung 4.6 dargestellte Zustand.

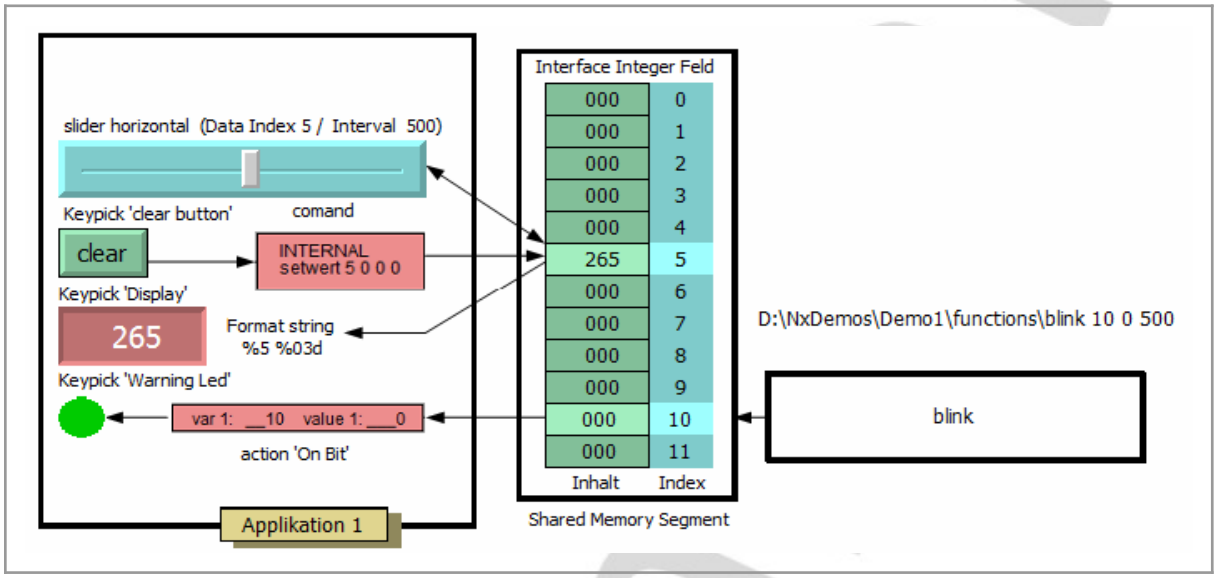


Abbildung 4.6: Beispiel 3a

Die Funktion 'blink.exe' ist ein Programm dem beim Aufruf die Parameter 'Index bit time' übergeben wurden. Das Programm läuft in einer Schleife 'time, welches im 'Index' das Bit 'bit' invertiert. Da das Programm eine Verlinkung auf das Shared Memory Segment beinhaltet, blinkt die 'Warning Led' aus Applikation 1 im Rhythmus 'time'.

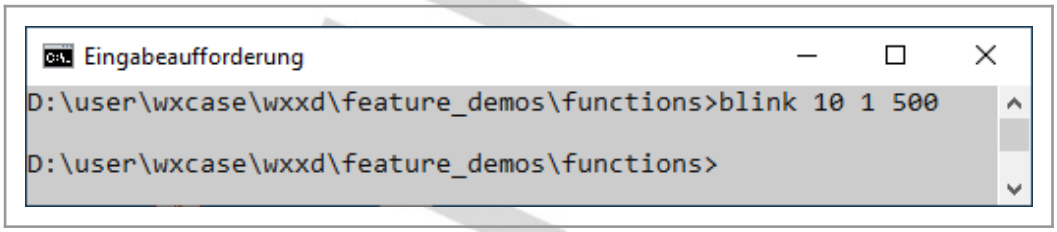


Abbildung 4.7: Beispiel 3b

Statt das Programm 'blink' über eine Dosbox aufzurufen kann es auch aus der Applikation heraus gestartet werden indem man den Aufruf als 'External Command' mit dem 'event type at start' der Applikation hinzufügt wie in Abbildung 4.8 gezeigt

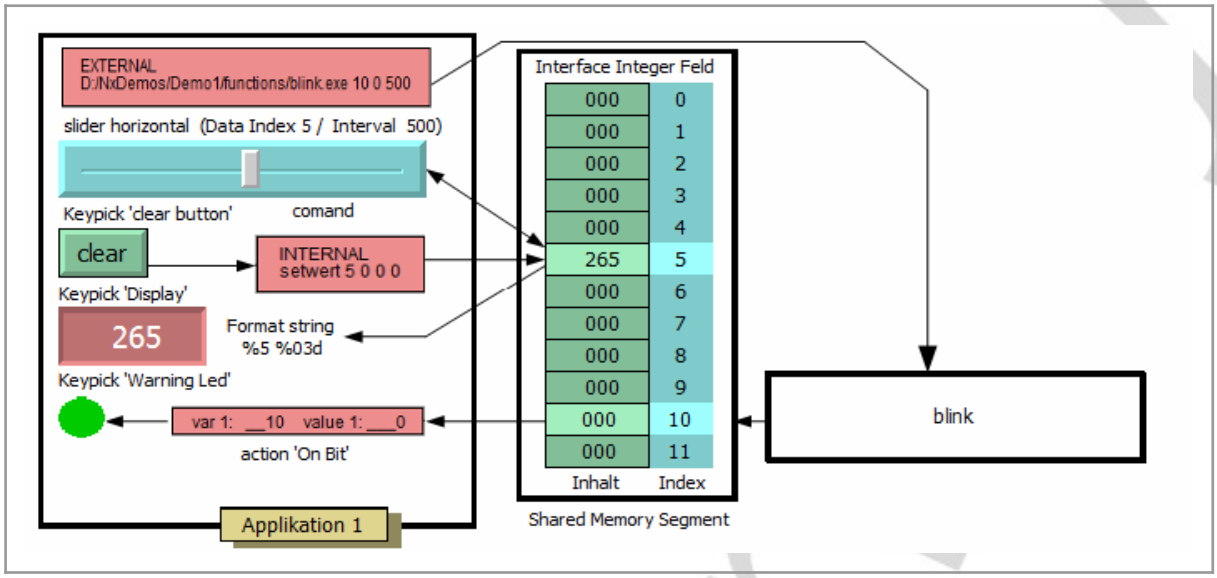


Abbildung 4.8: Beeispiel 3c

4.4 Datenaustausch über Netzwerkfunktionen.

Das nächste Beispiel Abbildung 4.9 zeigt die Verteilung der Applikation 1 und der Applikation 2 auf zwei Rechnern.

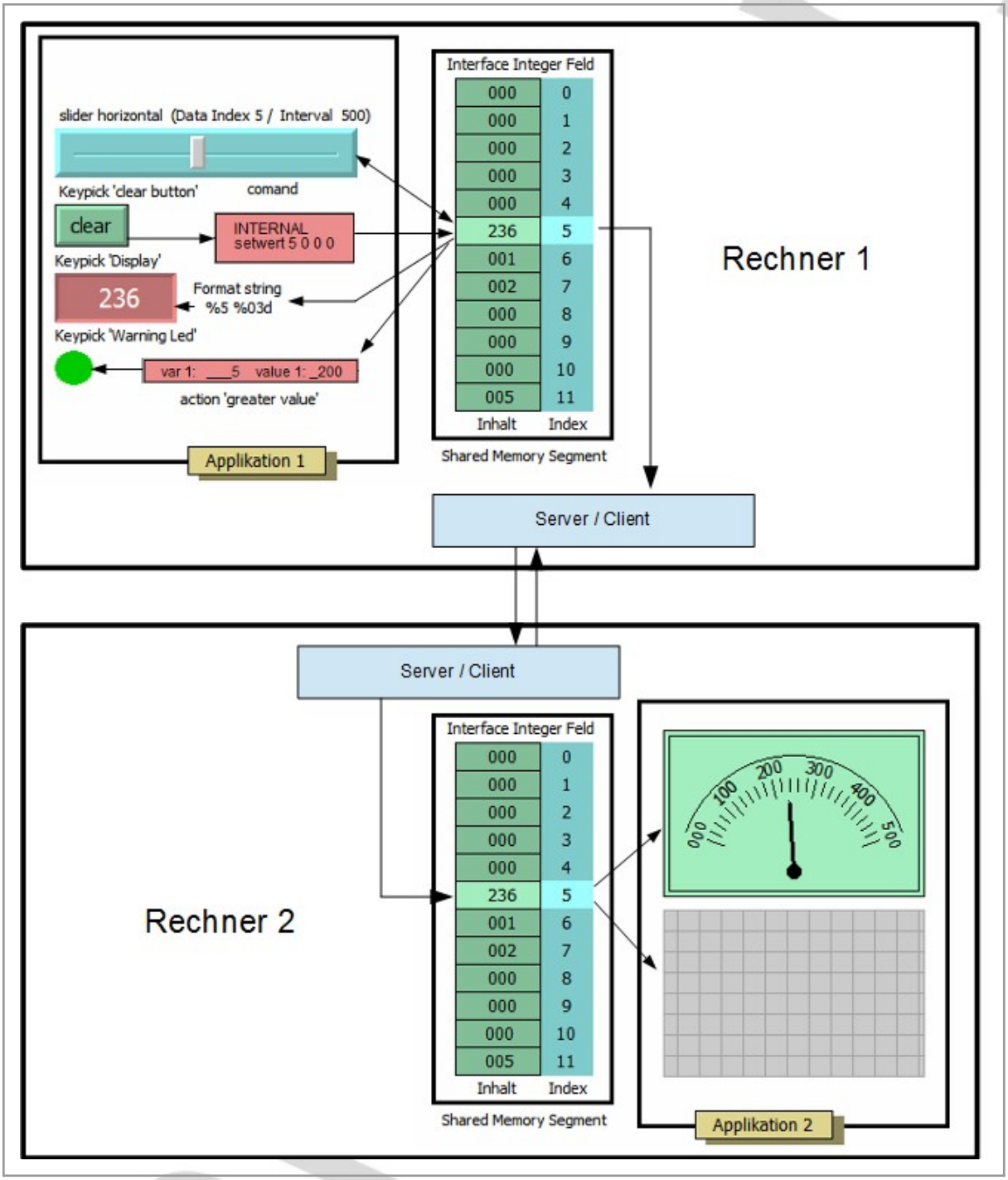


Abbildung 4.9: Beispiel 4

Wird Applikation 1 auf Rechner 1 gestartet, wird auf diesem ein Shared memory Segment erzeugt. Applikation 2 erzeugt beim Start ebenfalls ein Shared Memory Segment auf Rechner 2. Eine Server-

Client auf beiden Systemen kann nun zyklisch die Daten zwischen dem beiden Memory Segmenten austauschen. Die Schieberbewegungen auf Rechner 1 werden dann in Applikation 2 auf Rechner 2 ausgegeben. Da RPT-One Anwendungen auf Microsoft Windows Systemen als auch unter Linux Systemen lauffähig sind, können Applikation 1 und Applikation 2 auch auf Rechner mit verschiedenen Betriebssystemen kommunizieren.

4.5 Datenaustausch mit externer Hardware.

Das nächste Beispiel beschreibt die Ansteuerung einer externen Hardware an Hand eines USB Moduls K8055N der Firma Velleman. Die Hardware verfügt über folgende Ein-Ausgabe Funktionen:

Zwei 8 Bit Analog – Digitalwandler.

Zwei 8 Bit Digital – Analogwandler.

Sechs Digitalen Eingängen.

Acht Digitalen Ausgängen.

Zwei Impulszählern welche mit den Digitaleingängen 0 und 1 verbunden sind.

Die Ansteuerung der Hardware erfolgt über eine USB Schnittstellen und der mitgelieferten Treibersoftware. Abbildung 4.10 zeigt den Aufbau der K8055N - Demo Applikation mit der dazugehörigen Hardware Ausgabe in Abbildung 4.11.

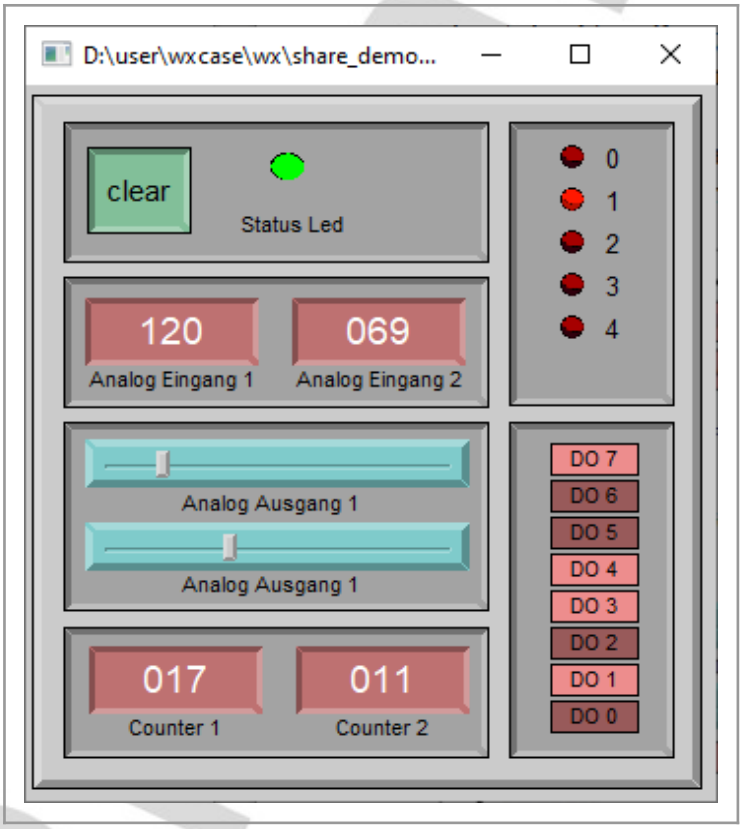


Abbildung 4.10: K8055N - Demo Applikation

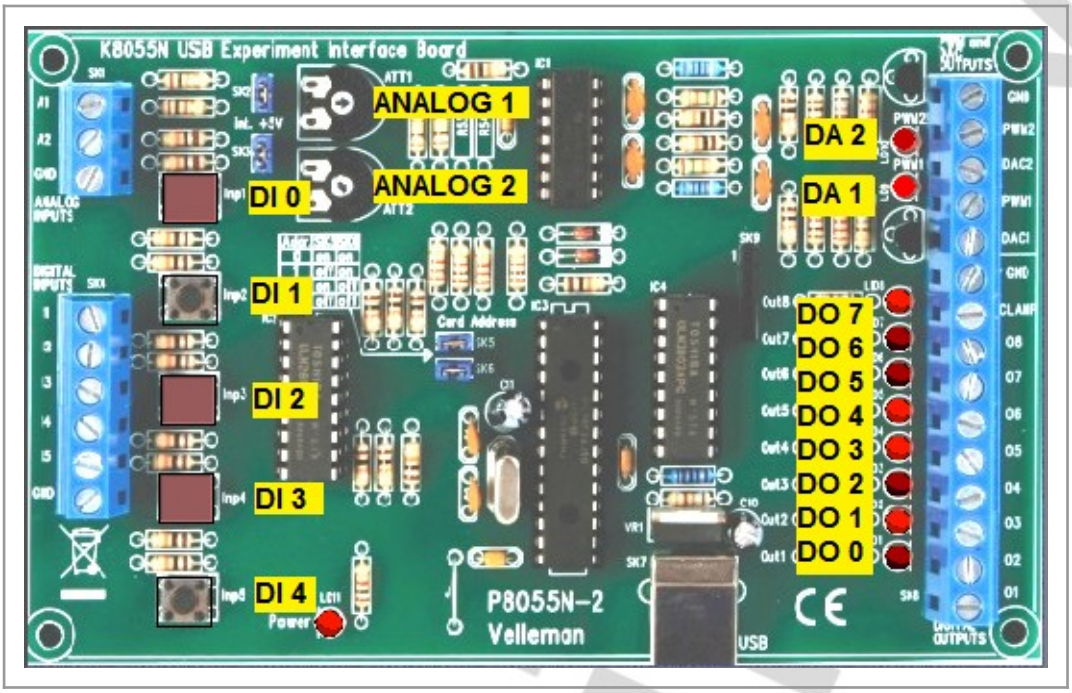


Abbildung 4.11: Hardware Ausgabe

Das Zusammenspiel der Applikation mit dem Interface und dem USB Treiber ist als Teildarstellung in der Abbildung 4.12 zu sehen. Als Schnittstelle zwischen dem Shared Memory und dem USB Treiber dient das User Programm 've_run.exe' mit den Übergabeparametern 3 für den Basisindex im Interface und dem Parameter 50 als Aufrufrate der Hardwareabfrage in Milisekunden. Das Programm liest zyklisch die Werte der Analog und Digitaleingänge ein und legt diese im Shared Memory ab. Die analogen und digitalen Ausgabewerte werden im gleichen Zyklus vom Shared Memory gelesen und an den USB Treiber übergeben.

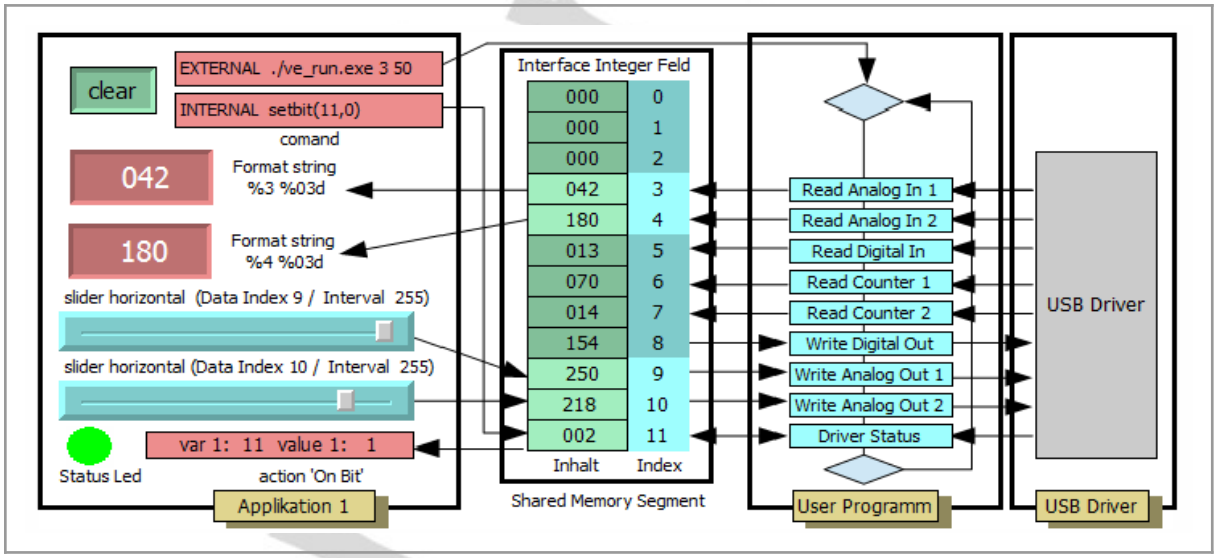


Abbildung 4.12: Aufbau Demo Applikation

Auch hier kann zwischen das Shared Memory und dem User Programm eine Netzwerkfunktion wie in Abbildung 4.9 geschaltet werden, so das die Hardwareabfrage auf einem anderem System stattfinden kann.

4.6 Library Funktionen zum Zugriff auf das RPT Interface.

4.6.1 Beispiel Blinken

Am Beispiel xxx soll die wirking …

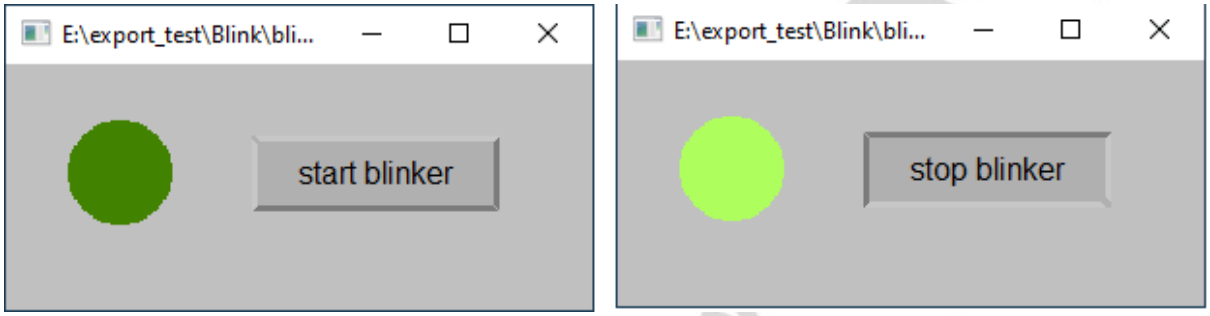


Abbildung 4.13: text 1

xxxx

```
h: rpt_one 17 01 0303 0127 E:\export_test\Blink\blinktest_0.ndat
s:af! 1;8.25;-11;0;0;0;400;0;0;0;0;3;2;1;34;Arial
s:bf! 1;8.25;-11;0;0;0;400;0;0;0;0;3;2;1;18;ZapfEllipt BT
s:tf! 1;11.25;-15;0;0;0;400;0;0;0;0;3;2;1;49;Courier New
s:rt! 20
w:Background Window             000000 000000 000303 000127 00000000 00c0c0c0 00ffffff 00028004 0000 0000 0008
k:Start Stop Key                000127 000037 000128 000039 00000001 00b0b0b0 00b0b0b0 00000000 00000000 2221060b 2220160b 0000 0000 0011 start blinker|stop blinker
k:Led Symbol                    000032 000029 000060 000060 00000002 00b0b0b0 00b0b0b0 00008241 005effae 20200100 20200900 0000 0000 0021 .\pictures\point3.icon|.\pictures\point3.icon
a:Set Led Status                000010 000001 000000 000000 000000 000000 000000 00000002 00000001 00000000 00000000
c:Start External Blink Task     00000001 04100004 00000005 00000001 00000000 00000000 .\functions\blink.exe|10 1 500
c:Stop External Blink Task      00000001 01200004 00000005 00000000 00000000 00000000 |exit process nr. 5
c:Set default Led Status        00000001 02200002 0000000a 00000001 00000000 00000000 setbit (10,1)|setbit (10,1)
```

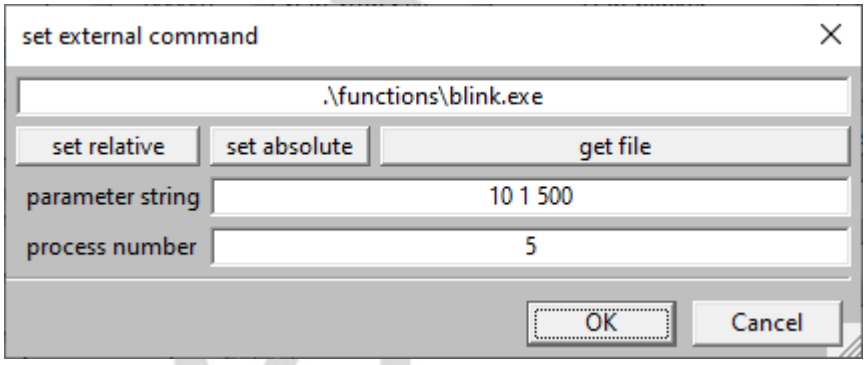


Abbildung 4.14: text 2

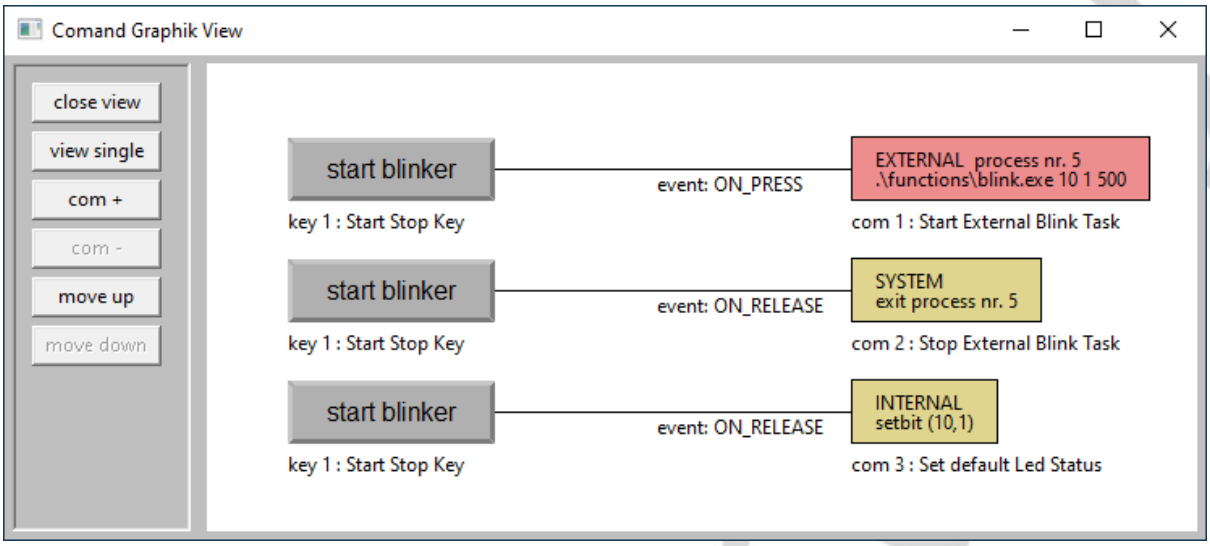


Abbildung 4.15: text 3

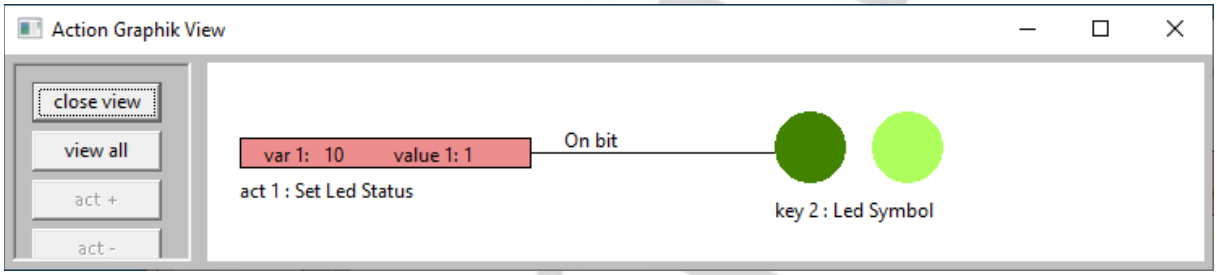


Abbildung 4.16: text 4

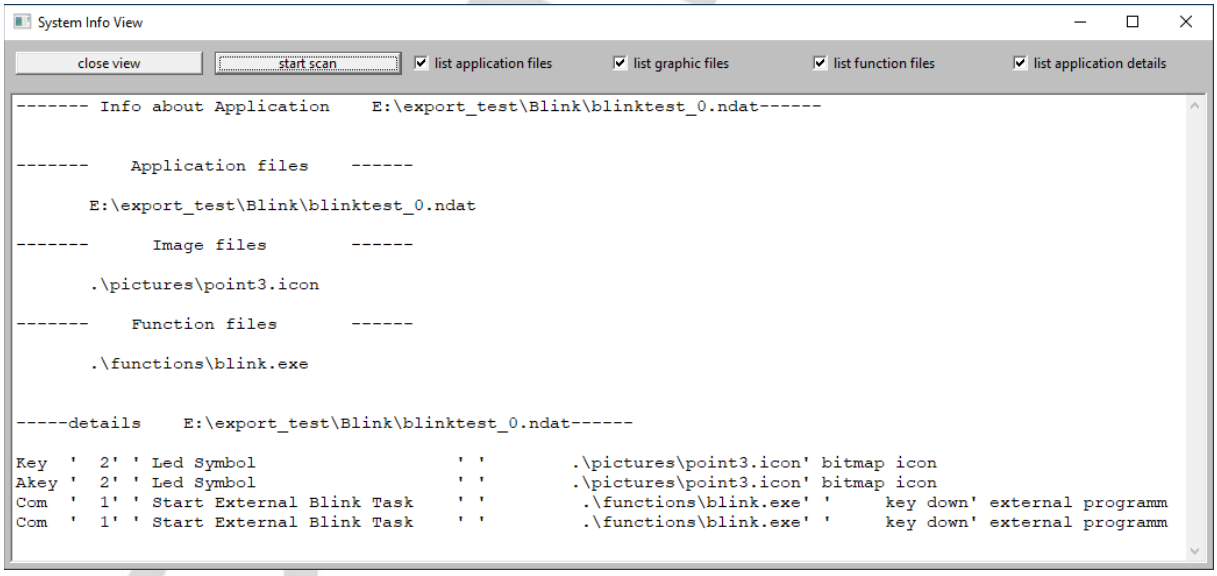


Abbildung 4.17: text 5

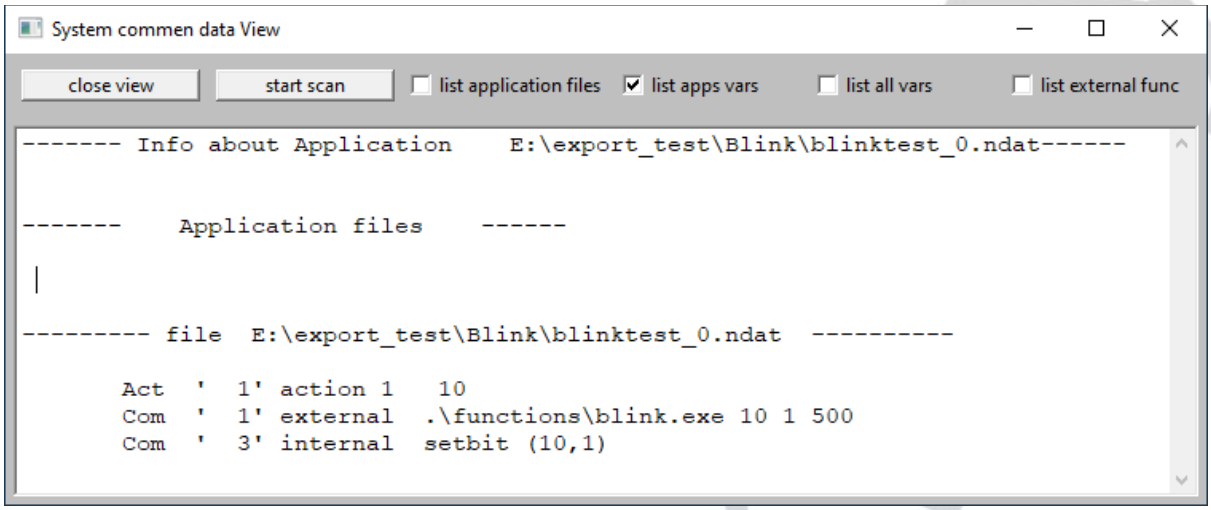


Abbildung 4.18: text 6

Um auf das Rpt Interface zuzugreifen wird das folgende C – Headerfile benötigt.

```
/*****************************************************************************
*                                                                            *
* File name   :  fxinterface.h                                               *
*                                                                            *
* Copyright Wilfried Schude, all rights reserved                             *
*                                                                            *
*                                    ###    ##   ###                        *
*                                     ###   ###   ###    #####              *
* Wilfried Schude                     ###    ###  ###  ###                   *
* Elisabeth Muehlenweg Str. 3          ###   ###  ###   ###                  *
* 78476 Allensbach                      ###  #### ####    ###               *
*                                        ###### ####       ####            *
* wilfried.schude@wischu.de               #####  #####       ###           *
*                                         #####   ####       ###           *
*                                                           #####            *
*                                                                            *
* PROPRIETARY:                                                               *
*    This code and all features are proprietary to  Wilfried Schude          *
*    and may not be reproduced, modified or distributed in any manner        *
*    without the written permission of Wilfried Schude.                      *
*                                                                            *
* PROJECT ID:   wx_functions                                                 *
*                                                                            *
* FILE DESCRIPTION:                                                          *
*    ---------------------------                                             *
*   History:                                                                 *
*   ---------------------------------------------------------                *
*                                                                            *
*****************************************************************************
#ifndef FX_COMMON_INCLUDE_FXINTERFACE_H_
#define FX_COMMON_INCLUDE_FXINTERFACE_H_

#define DATPUFNUM    4
#define DATPUFLEN 1000

typedef struct termda__
{
   int incount;
```

```
   int outcount;
   int status;
   char data[1024]; // termmode
} TERMDEF;

   typedef struct commda__
   {
      int system[10];
      TERMDEF term[10] ;
      char text[20][80];
      struct dapu_
      {
         int control ;
         int sample_time ;
         int trigger_level ;
         int da[DATPUFLEN];
      } dapu[DATPUFNUM];
      int data[1000];
   } COMMEMDEF;


   enum term_status_defs
   {
      TERM_ENAB = 0x00000001,
      TERM_USED = 0x00000002
   };

int ProNum ;

extern bool rpt_interface_exists(const char*) ;
extern void rpt_interface_connect(void) ;
void user_call(void) ;
extern bool prog_controll(int process_number) ;
void end_call(void) ;
void user_init(void) ;
extern void start_loop(int argc, char *argv[],int process_num, int time) ;

extern void lock_data(void) ;
extern void unlock_data(void);
extern int get_var(int index) ;
extern void set_var(int index, int value);
extern void set_text(int index, char *  text) ;
extern void get_text(int index, char *  text) ;
extern float get_float(int index) ;
extern void set_float(int index, float value);

#endif /* FX_COMMON_INCLUDE_FXINTERFACE_H_ */
```

Beschreibung der Interface Zugriffe am Beispiel der externen Funktion blink.cpp, welche als parallele Task zur aufrufenden Applikation gestartet wird.

```
/*****************************************************************************
*                                                                            *
* File name   :  blink.h                                                     *
*                                                                            *
* Copyright Wilfried Schude, all rights reserved                             *
*                                                                            *
*                                    ###    ##   ###                        *
*                                     ###   ###   ###    #####              *
* Wilfried Schude                     ###    ###  ###  ###                   *
* Elisabeth Muehlenweg Str. 3          ###   ###  ###   ###                  *
* 78476 Allensbach                      ###  #### ####    ###               *
*                                        ###### ####        ####           *
```

```
* wilfried.schude@wischu.de               #####  #####        ###             *
*                                          #####   ####        ###             *
*                                                            #####             *
*                                                                              *
* PROPRIETARY:                                                                 *
*    This code and all features are proprietary to  Wilfried Schude            *
*    and may not be reproduced, modified or distributed in any manner          *
*    without the written permission of Wilfried Schude.                        *
*                                                                              *
* PROJECT ID:   wx_functions                                                   *
*                                                                              *
* FILE DESCRIPTION:                                                            *
*    --------------------------                                                *
*   History:                                                                   *
*   ------------------------------------------------------------               *
*                                                                              *
********************************************************************************
#ifndef BLINK_H_
#define BLINK_H_

#include "../fx_common/include/fxinterface.h"

int n;
int m;
int v;

int cvar ;

#endif /* BLINK_H_ */



/*******************************************************************************
*                                                                              *
* File name   :  blink.cpp                                                     *
*                                                                              *
* Copyright Wilfried Schude, all rights reserved                               *
*                                                                              *
*                                  ###    ##   ###                           *
*                                   ###    ###    ###    #####                *
* Wilfried Schude                   ###    ###  ###  ###                      *
* Elisabeth Muehlenweg Str. 3        ###   ###  ###   ###                     *
* 78476 Allensbach                    ###  #### ####    ###                  *
*                                    ###### ####        ####                 *
* wilfried.schude@wischu.de           #####  #####        ###                 *
*                                     #####   ####        ###                 *
*                                                        #####                 *
*                                                                              *
* PROPRIETARY:                                                                 *
*    This code and all features are proprietary to  Wilfried Schude            *
*    and may not be reproduced, modified or distributed in any manner          *
*    without the written permission of Wilfried Schude.                        *
*                                                                              *
* PROJECT ID:   rptblink                                                       *
*                                                                              *
* FILE DESCRIPTION:                                                            *
*    --------------------------                                                *
*   History:                                                                   *
*   ------------------------------------------------------------               *
*                                                                              *
********************************************************************************
#include <stdlib.h>
#include "blink.h"

int main(int argc, char **argv)
{
```

```
   if (argc >= 4)
   {
      if (rpt_interface_exists("wtymem"))
      {
         ProNum = 0;

         n = atoi(argv[1]); // index
         v = atoi(argv[2]); // bitnumber
         m = atoi(argv[3]); // time

         if (argc >= 4)
         {
            ProNum = atoi(argv[4]); // process number
         }
         if (prog_controll(ProNum))
         {
            start_loop(argc, argv, ProNum, m);
         }
      }
   }
   return 0;
}


void user_init(void)
{
}

void user_call(void)
{
   /* ************************ include user action here ***************** */
   cvar = get_var(n) ;

   if (cvar & (0x01 << v))
   {
      cvar &= ~(0x01 << v);
   }
   else
   {
      cvar |= (0x01 << v);
   }

   set_var(n,cvar) ;
   /* ************************   end of user action     ***************** */
}

void end_call(void)
{
}
```

4.6.2 Library Functions Call

4.6.2.1 `bool rpt_interface_exists(const char*) ;`

4.6.2.2 `extern void rpt_interface_connect(void) ;`

4.6.2.3 `void user_call(void) ;`

4.6.2.4 `extern bool prog_controll(int process_number) ;`

4.6.2.5 `void end_call(void) ;`

4.6.2.6 `void user_init(void) ;`

4.6.2.7 `extern void start_loop(int argc, char *argv[],int process_num, int time)`

4.6.2.8 `void lock_data(void) ;`

4.6.2.9 `extern void unlock_data(void);`

4.6.2.10 `extern int get_var(int index) ;`

4.6.2.11 `extern void set_var(int index, int value);`

4.6.2.12 `extern void set_text(int index, char * text) ;`

4.6.2.13 `extern void get_text(int index, char * text) ;`

4.6.2.14 `extern float get_float(int index) ;`

4.6.2.15 `extern void set_float(int index, float value);`

5 Anhang

5.1 Formate der Instrument Objekte

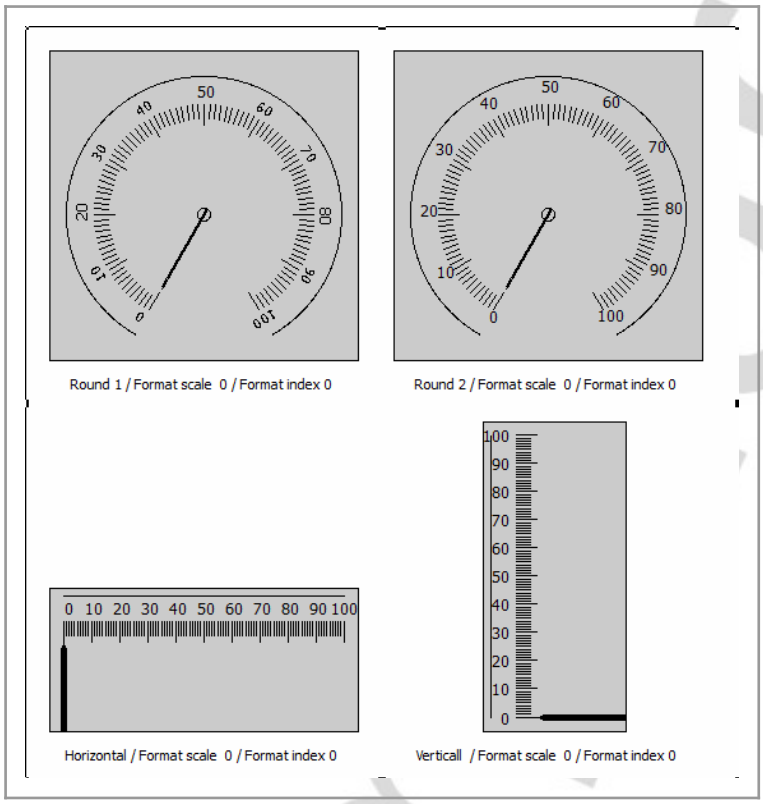


Abbildung 5.1: Form 0

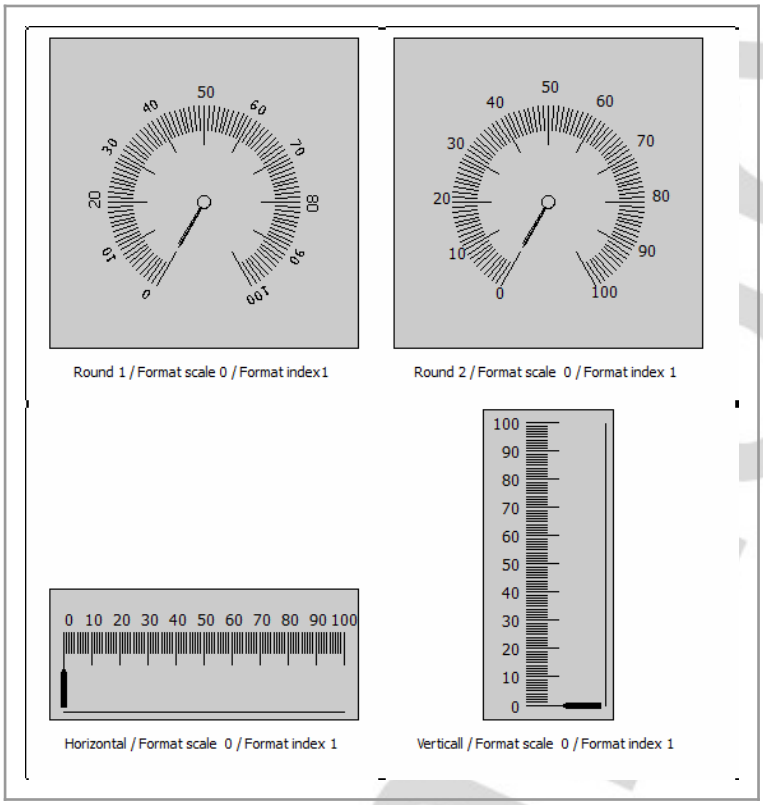


Abbildung 5.2: Form 1

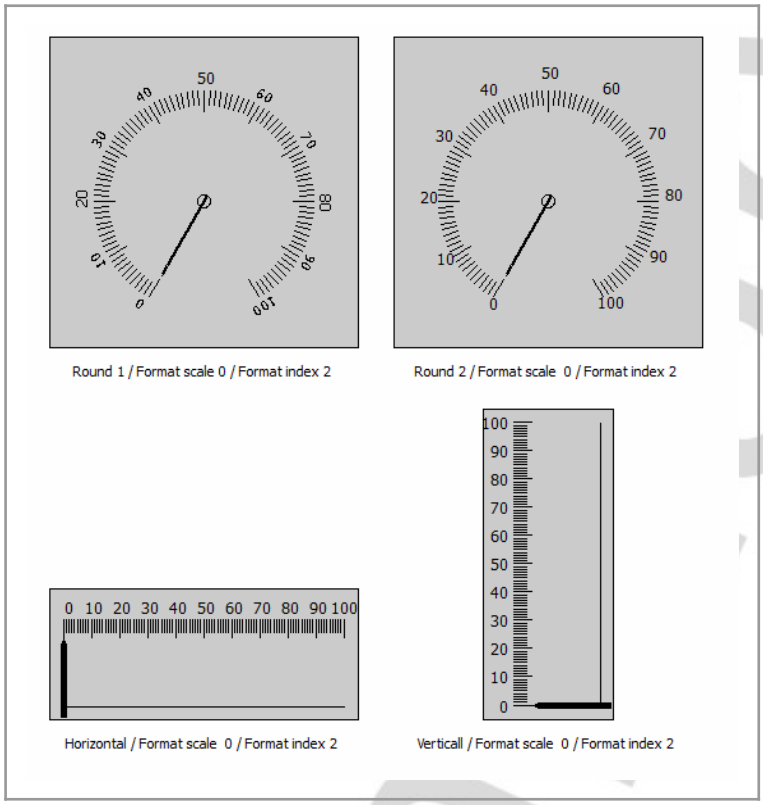


Abbildung 5.3: Form 2

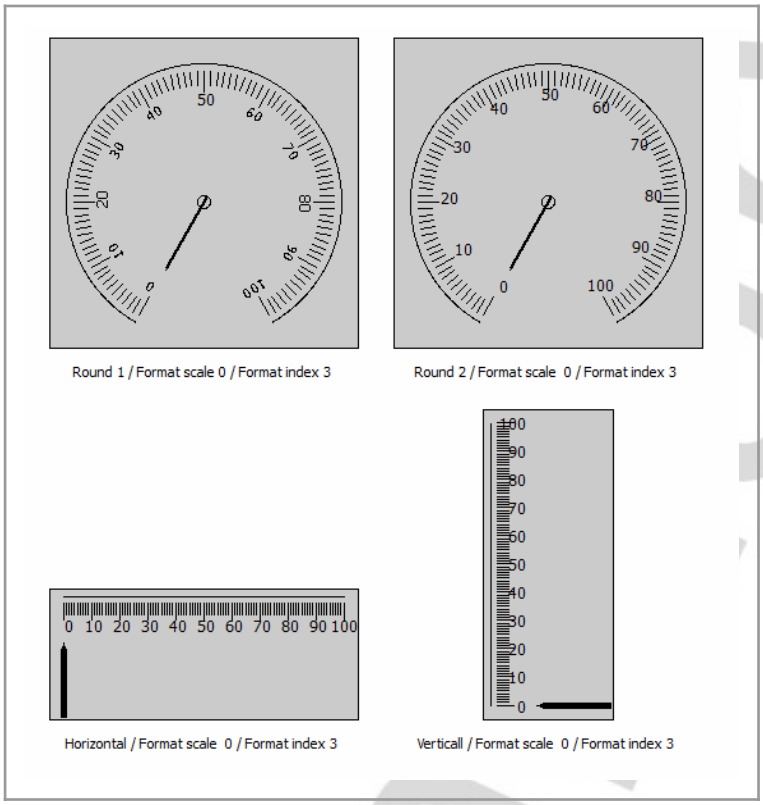


Abbildung 5.4: Form 3

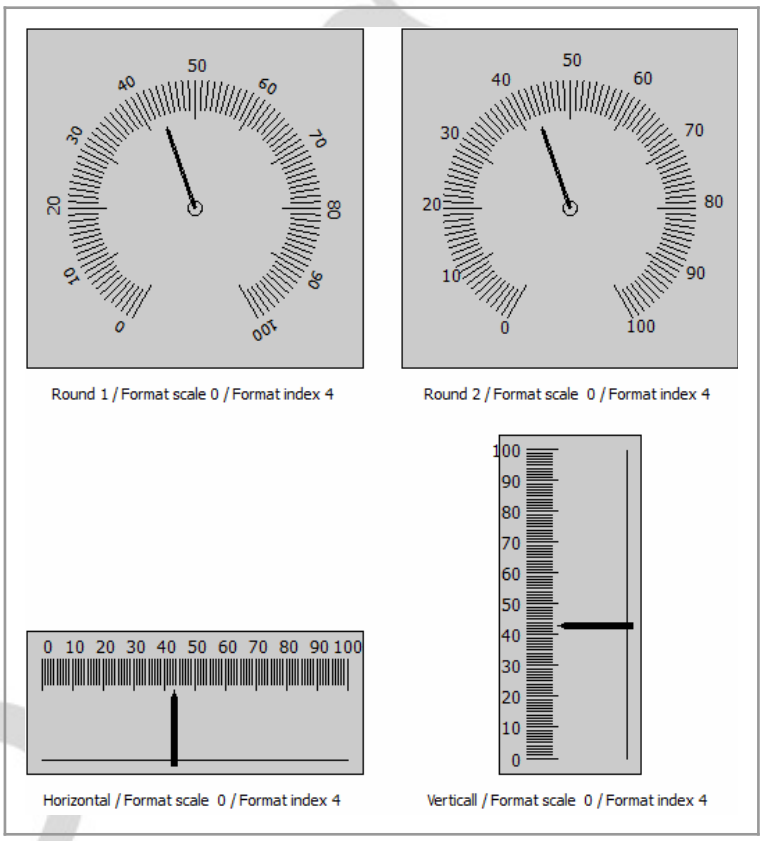


Abbildung 5.5: Form 4

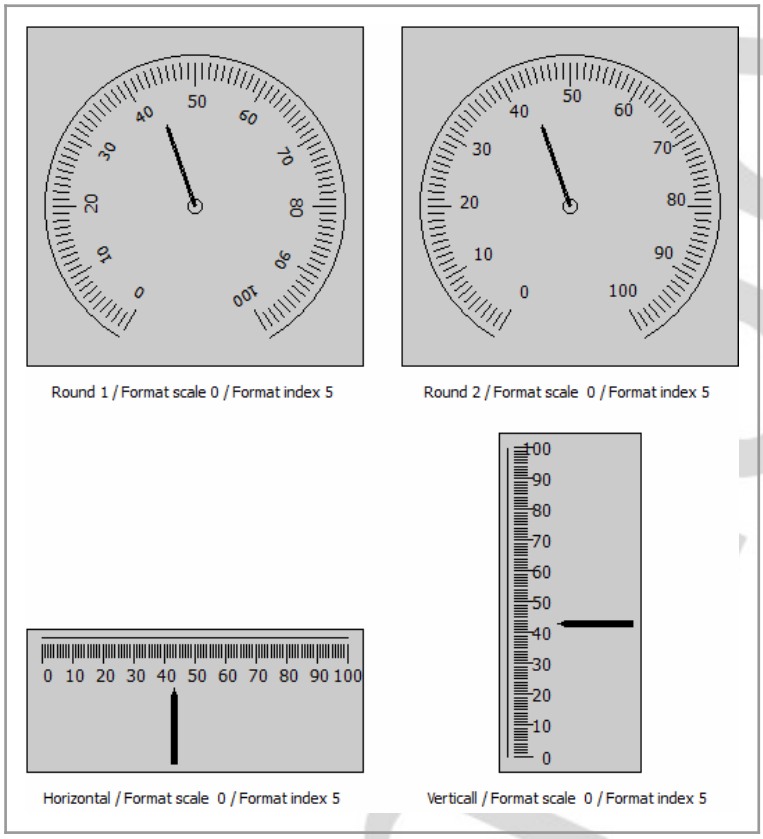


Abbildung 5.6: Form 5

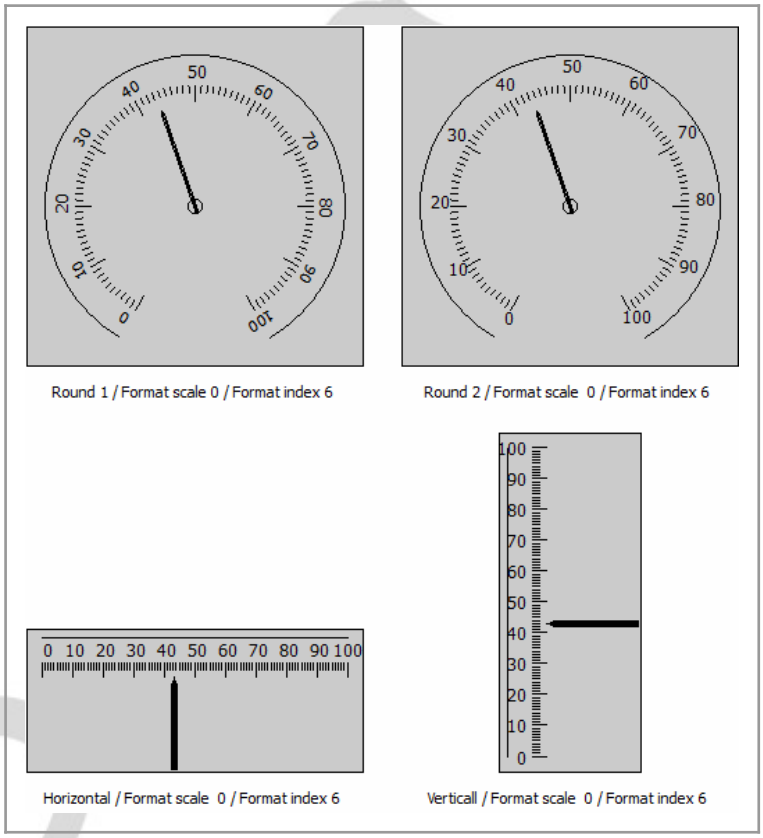


Abbildung 5.7: Form 6

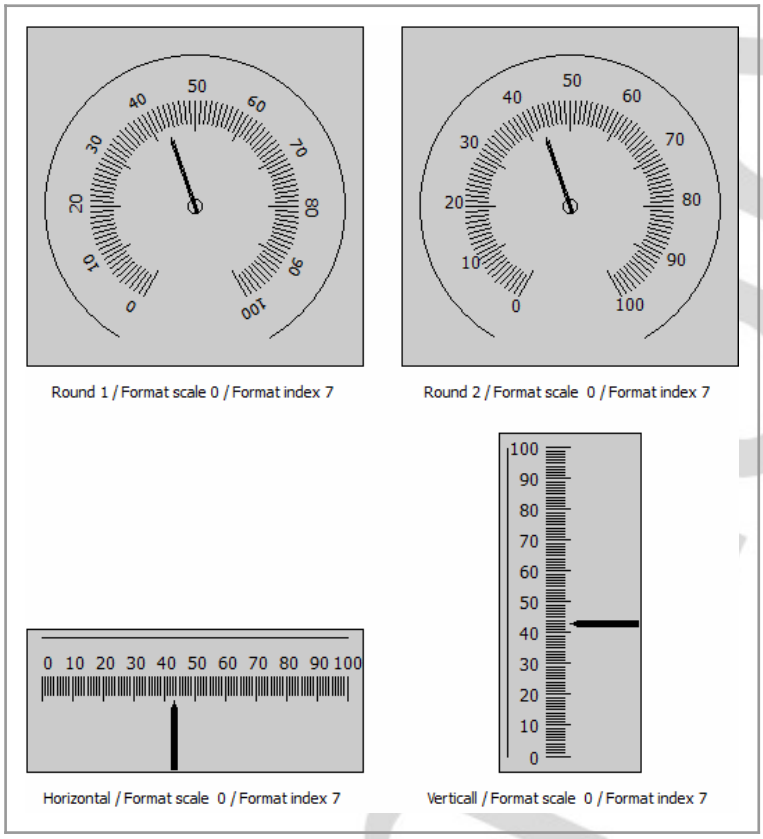


Abbildung 5.8: Form 7

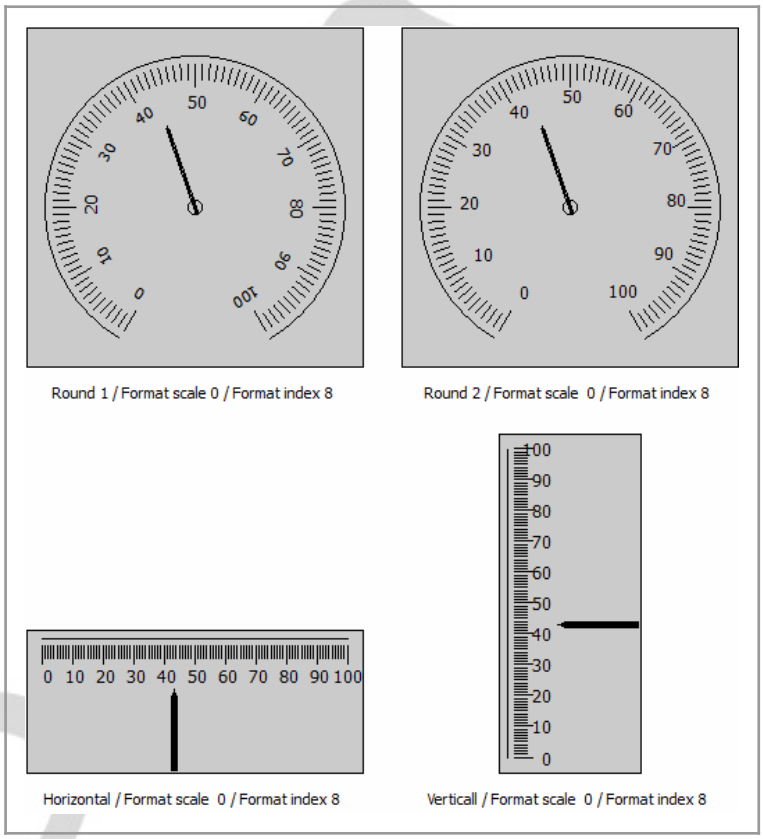


Abbildung 5.9: Form 8

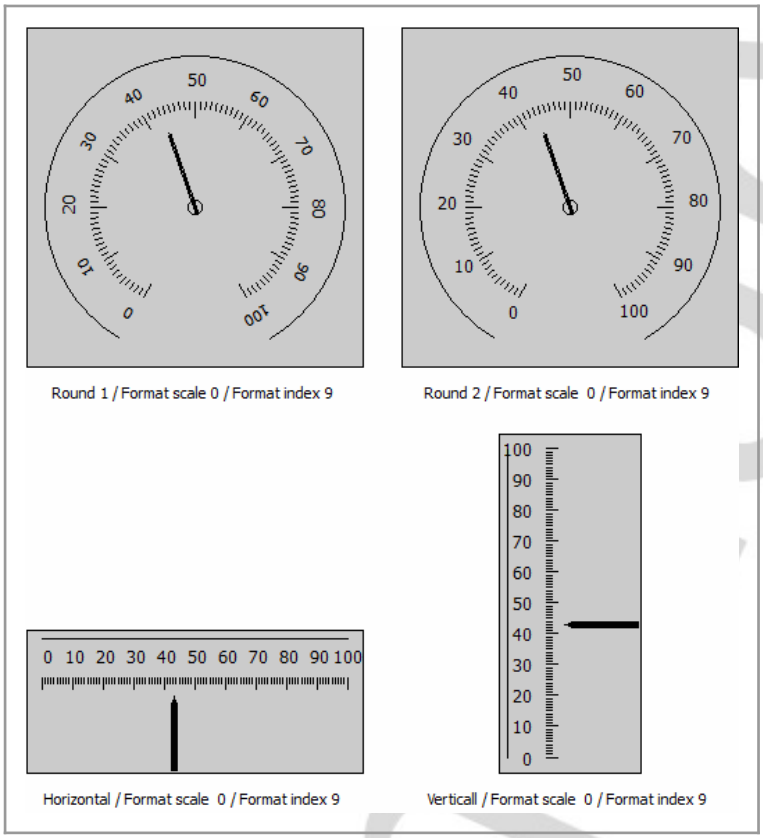


Abbildung 5.10: Form 9

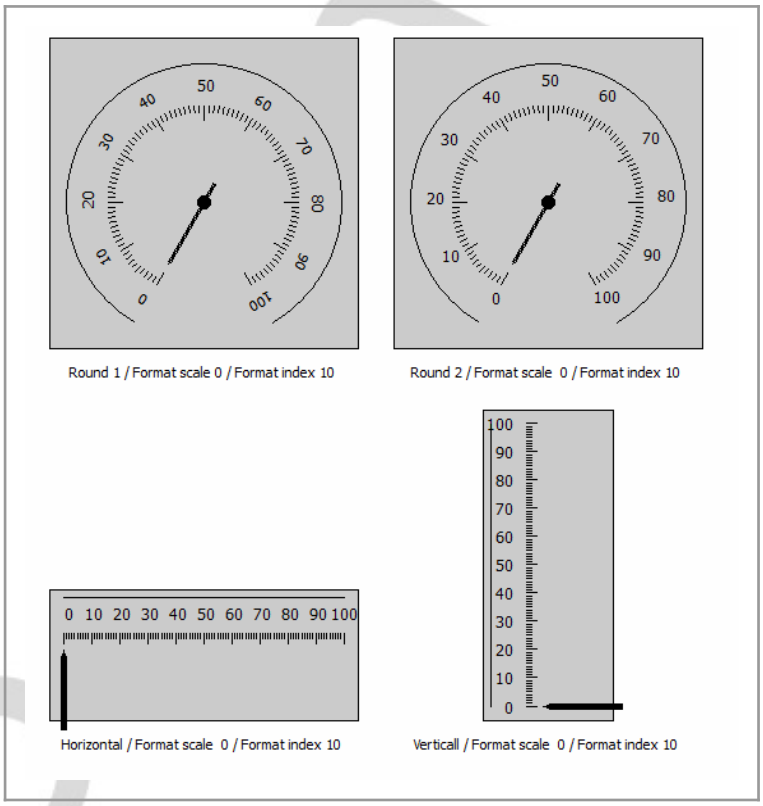


Abbildung 5.11: Form 10

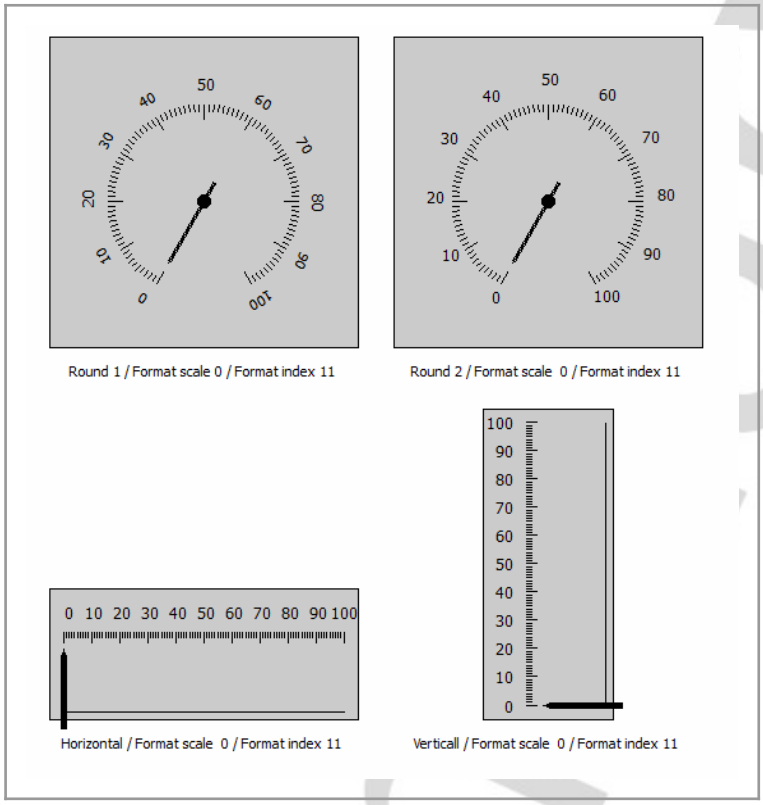


Abbildung 5.12: Form 11

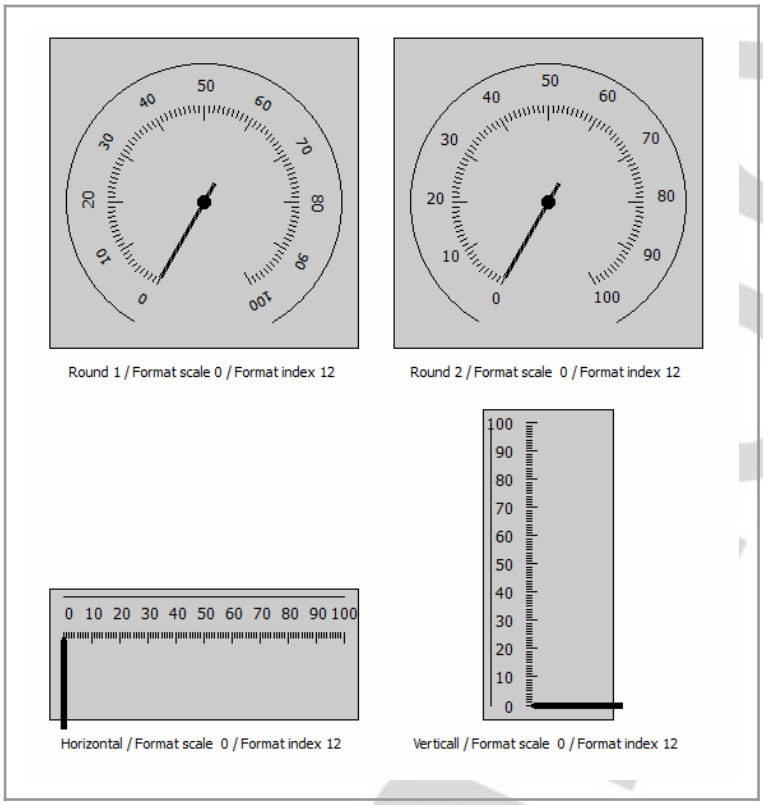


Abbildung 5.13: Form 12

5.2 Scalierung der Instrument Objekte

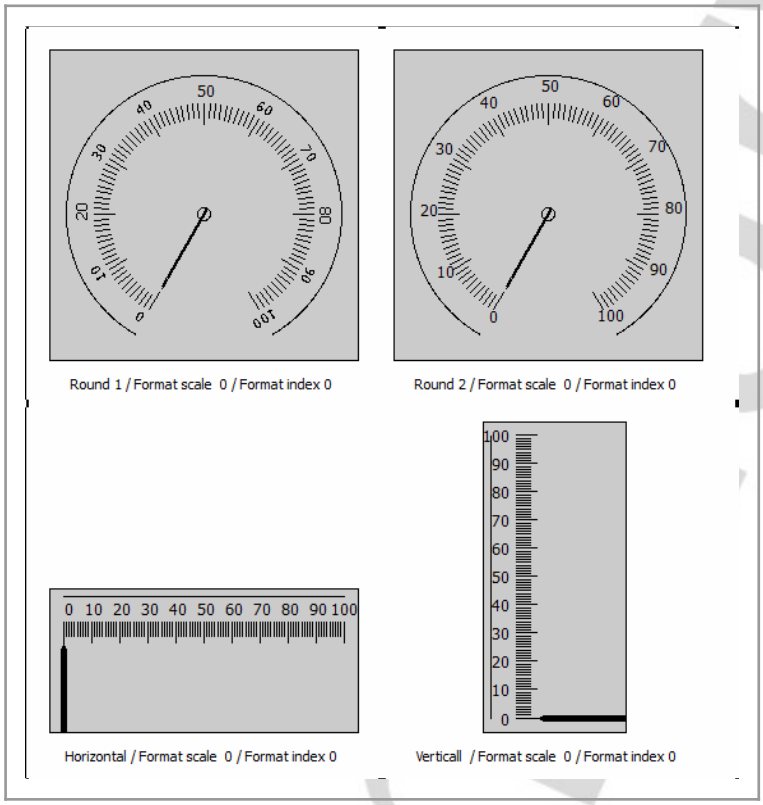


Abbildung 5.14: Scala 0

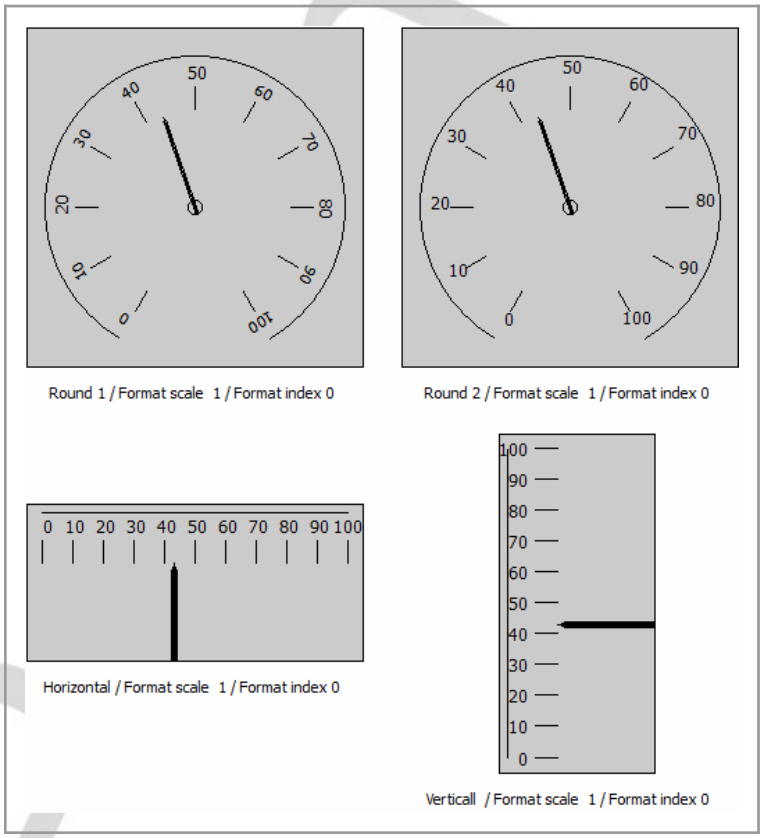


Abbildung 5.15: Scala 1

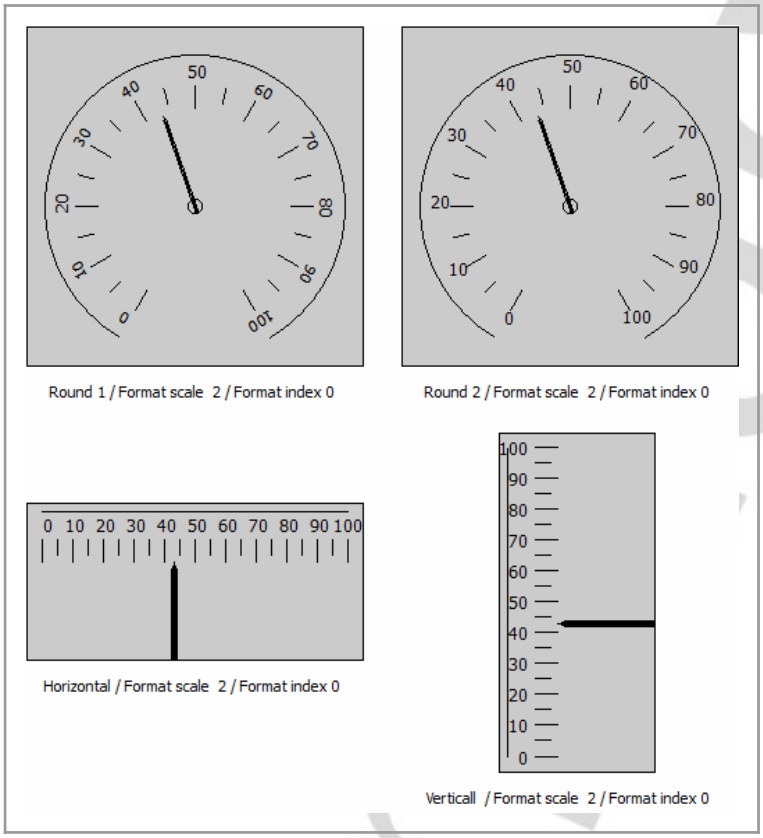


Abbildung 5.16: Scala 2

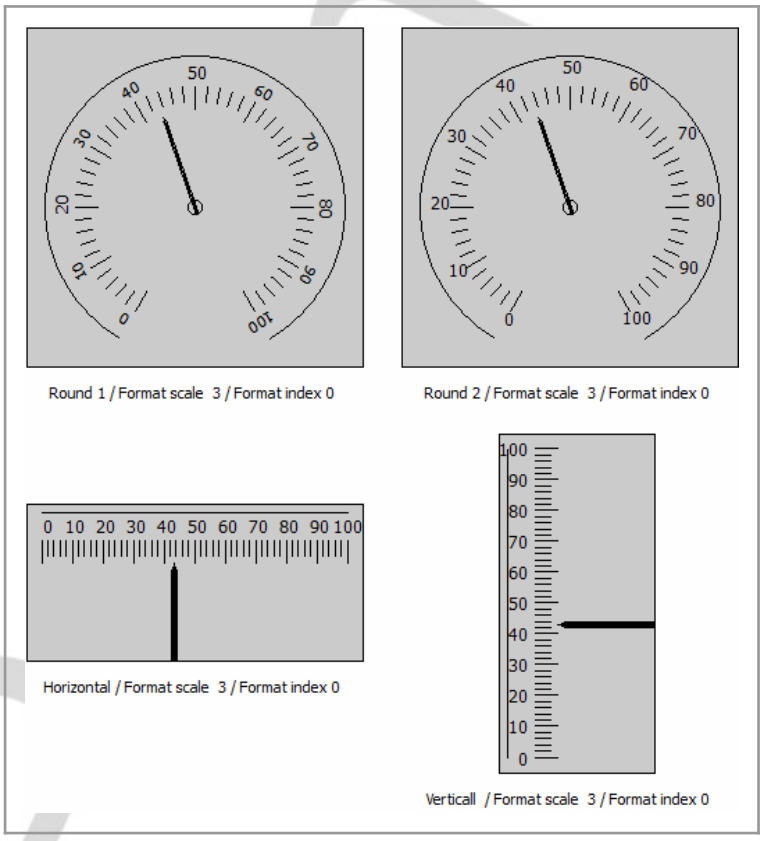


Abbildung 5.17: Scala 3