

Liste des bisher nutzbaren Zeugs:

SYSTEM STUFF

InitGUI

MCP

WINDOW STUFF

AddWindow

ShowWindow / HideWindow / DeleteWindow

EnableUserGFX

CONTROL STUFF

AddSimpleGadget

AddToggleGadget

AddArrow / AddSmallArrow

GadgetOn / GadgetSleep / GadgetOff

GetSelect

SetSelect

SYSTEM STUFF

InitGUI		
sub InitGUI		
Beschreibung:	Voreinstellungen für sGUI, Erzeugen des sogenannten RootWindows	
Parameter:	/	
Hinweise:	<p>Muß direkt nach dem Screen-Befehl aufgerufen werden. Ohne den Aufruf von InitGUI funktioniert sGUI nicht!</p> <p>Einige der Voreinstellungen sind vom User seit dieser Version nachträglich veränderbar.</p> <p>Im allgemeinen betrifft dies die Farbgebung. Für alle Farbdefinitionen reichen die RGB Werte aus. Der separat ausgewiesene Alphawert (Colors.AlphaValue) wird den Farben dann intern zugewiesen!</p>	
	MinimumClearence	Ein in sGUI benutzter MindestAbstand, bestimmt beispielsweise die Dicke des Fensterrahmens usw. Im Zweifel auf 1 setzen, passt :)
	SelectBrightnessValue	Definiert ob und um wieviel ein Control beim Anklicken (Selektieren) dunkler oder heller wird. Erlaubt sind Werte von -100 (dunkler) bis +100 (heller).
	Colors.AlphaValue	Transparenzwert der GUI. Gültig sind Werte von 0=vollständig transparent bis 255=volle Deckung welches auch die Voreinstellung ist. Noch nicht realisiert!!!
	Colors.Cleaning	Interne „Reinigungsfarbe“, voreingestellt ist hier &HFF00FF bzw. RGB(255,0,255).
	Colors.Pen	Farbe des „Stiftes“, also der voreingestellten Schreibfarbe des Screens. DRAW STRING ohne Angabe einer Farbe benutzt Colors.Pen. Intern wird ein „COLOR Colors.Pen“ aufgerufen.
	Colors.Background	Allgemeine Hintergrundfarbe des Screens bzw. auch des Rootwindows. Achtung, „reguläre“ Fenster haben eine eigene Farbdefinition für den Fensterhintergrund. Intern wird ein „COLOR ,Colors.BackGround“ aufgerufen.
	Colors.WinFrame	Rahmenfarbe eines unselektierten Fensters
	Colors.WinFrameSelected	Rahmenfarbe eines selektierten Fensters
	Colors.WinText	Titeltextfarbe des unselektierten Fensters
	Colors.WinTextSelected	Titeltextfarbe des selektierten Fensters
	Colors.WinBackGround	Hintergrundfarbe des Fensterinhaltes. Im Falle EnableUserGFX() wird statt der Hintergrundfarbe das entsprechende Image im Fenster angezeigt.
	Colors.GadBody	Farbe des unselektierten Gadgets
	Colors.GadFrame	Farbe des Rahmens eines unselektierten Gadgets

	Colors.GadText	Farbe des Textes in einem unselektierten Gadget
	Colors.GadTextSelected	Farbe des Textes in einem selektierten Gadget
	Colors.GadGlow	Farbe des Gadgets wenn sich die Maus darüber befindet
	Colors.GadFrameGlow	Farbe des Rahmens eines Gadgets über dem sich die Maus befindet
	Colors.GadSleepGrid	Farbe des Rasters, welches über ein Gadget gelegt wird, wenn es sich im SLEEPModus befindet
Beispiel:		
Rückgabewert:	/	

MasterControlProgram		
sub MasterControlProgram (win as sGUIWindow ptr=RootWindow, FirstCall as integer=1)		
Beschreibung:	<p>Das MCP aufgerufen in einem Loop ist letztlich das Herzstück von sGUI. Es sammelt Systemereignisse, erzeugt daraus GUI Ereignisse und zeichnet die GUI in den Screen.</p> <p>Folgende System-Variablen stehen nach dem Aufruf „aktualisiert“ zur Verfügung:</p>	
	KEY	Ist der String einer gedrückten Taste des Keyboardes
	ASCCODE	Der ASCII-Code von KEY
	EXTENDED	Bei einigen Tasten des Keyboardes liefert die INKEY-Funktion einen String der 2 Zeichen enthält. Wenn dies der Fall ist hat EXTENDED einen Wert>0.
	MOUSEMOVED	Ist 1 wenn die Maus bewegt wurde
	MOUSEX	X-Position der Maus im Screen
	MOUSEY	Y-Position der Maus im Screen
	LMB	Status der linken Maustaste
	MMB	Status der mittleren Maustaste
	RMB	Status der rechten Maustaste
	LMB_DOUBLE_CLICK	Ist 1 wenn ein Doppelklick mit der linken Maustaste erfolgt
	MMB_DOUBLE_CLICK	Ist 1 wenn ein Doppelklick mit der mittleren Maustaste erfolgt
	RMB_DOUBLE_CLICK	Ist 1 wenn ein Doppelklick mit der rechten Maustaste erfolgt
	WHEEL	Ist <>0 wenn das Mousrad bewegt wurde. Genauer -1 wenn das Rad vom User weg bzw +1 wenn das Rad zum User

		hin bewegt wird.
	SCREENCLOSEBUTTON	Ist 1 wenn der ScreenCloseButton angeklickt wurde
	WINDOWMESSAGE	Enthält den Zeiger zu einem Fenster in welchem ein Control eine GADGETMESSAGE erzeugte. WINDOWMESSAGE und GADGETMESSAGE treten also immer gemeinsam auf. Eine WINDOWMESSAGE kann an ein ParentWindow „durch gereicht“ werden. Siehe WFLAG_SENDWMSGTOPARENT
	GADGETMESSAGE	Enthält den Zeiger eines Controls in welchem eine „Aktion“ stattfand. WINDOWMESSAGE und GADGETMESSAGE treten immer gemeinsam auf.
	MENUMESSAGE	Noch nicht realisiert
Parameter:	/	
Hinweise:	Die Parameter sind nur für den internen Gebrauch vorgesehen, zur Steuerung bei rekursiven Aufrufen.	
Beispiel:	/	
Rückgabewert:	/	

WINDOW STUFF

ShowWindow HideWindow DeleteWindow		
sub ShowWindow (win as sGUIWindow ptr) sub HideWindow (win as sGUIWindow ptr) sub DeleteWindow (win as sGUIWindow ptr)		
Beschreibung:	Versetzt jeweils ein Fenster in den SHOW- oder HIDE-Modus, d.h. es wird dargestellt oder ist nicht sichtbar. Fenster werden nach deren Erzeugung noch nicht dargestellt und müssen mit ShowWindow sichtbar gemacht werden. DeleteWindow löscht ein Fenster mitsamt dessen Inhalt!!!	
Parameter:	win	Zeiger auf ein sGUIWindow
Hinweise:	technisch passiert eigentlich nur folgendes, die Fenster-Eigenschaft win → Activation wird auf 1(sichtbar) bzw 0(nicht sichtbar) gesetzt. Statt ShowWindow/HideWindow wäre auch dies möglich. Bei DeleteWindow ist Vorsicht geboten. Zeiger, die ursprünglich auf das gelöschte Fenster und seine Childs verwiesen haben, zeigen nun auf freigegebenen Speicherbereich. Auch wenn das Fenster nicht mehr existiert, so kann ein Schreibzugriff zum Absturz führen.	
Beispiel:	/	
Rückgabewert:	/	

AddWindow		
function AddWindow (parent as any ptr, PosX as integer, PosY as integer, WinWidth as integer, WinHeight as integer, Text as string="", WinFlags as integer) as sGUIWindow ptr		
Beschreibung:	Erzeugt ein Fenster	
Parameter:	parent	Jedes neue Fenster (mit einer Ausnahme), ist ein Childobjekt, das kann sowohl ein Fenster als auch ein Gadget sein. Der Zeiger auf das ElternObjekt muß also hier übergeben werden. Wenn dieser Parameter 0 ist, dann wird das Fenster direkt auf dem Screen dargestellt.
	PosX PosY	Position dieses Fenster relativ zum ElternObjekt.
	WinWidth WinHeight	Breite und Höhe des Fensters, genauer die Contentgröße. Abhängig von vorhanden Rahmen kann das gesamte Fenster größer sein

	Text	Titeltext des Fensters, wird nur dargestellt wenn das Fenster eine Kopfzeile besitzt	
	WinFlags	Hier der komplizierteste Teil zur Konfiguration des zu erzeugenden Fensters, die Flags. Diese werden addiert oder logisch mit OR verknüpft. Einige Flags schließen sich aus, ich hoffe widersprüchliches Fensterverhalten ausgeschlossen zu haben. In sGUI.bas findet man die Flags als Konstanten.	
		WFLAG_WINDOWTYPE0	Das Fenster hat keinen Rahmen.
		WFLAG_WINDOWTYPE1	Das Fenster hat einen einfachen schmalen Rahmen
		WFLAG_WINDOWTYPE2	Das Fenster hat einen einfachen Rahmen mit Kopfzeile. Falls Text definiert ist, wird dieser angezeigt
		WFLAG_WINDOWTYPE3	Das Fenster hat einen einfachen schmalen Rahmen, gedacht für Menüs, bisher nicht realisiert!!!
		WFLAG_WINDOWTYPE4	reserviert
		WFLAG_WINDOWTYPE5	reserviert
		WFLAG_WINDOWTYPE6	reserviert
		WFLAG_WINDOWTYPE7	reserviert
		WFLAG_WINDOWTYPE8	reserviert
		WFLAG_DRAGABLE	Das Fenster ist verschiebbar, sinnvoll nur mit Fenstertyp 2
		WFLAG_CLOSEABLE	Das Fenster hat einen CloseButton, sinnvoll nur mit Fenstertyp 2
		WFLAG_MINIMIZABLE	unbenutzt, bisher nicht realisiert
		WFLAG_STACKABLE	Fenster wird im Fensterstapel nach vorn geholt, sobald es angeklickt wurde. Wie weit es nach vorn geht, hängt von den Flags anderer Fenster ab
		WFLAG_BACKDROP	Fenster wird immer hinten im Stapel gehalten.
		WFLAG_ALWAYSONTOP	Fenster wird immer vorn im Stapel gehalten.
		WFLAG_WARN	Fenster ist vorn. Der Rest der GUI ist blockiert solange das Fenster geöffnet ist.
		WFLAG_ALWAYSSELECTED	Fenster wird immer selektiert dargestellt, unabhängig von entsprechenden Mausklicks.
		WFLAG_ALWAYSUNSELECTED	Fenster wird immer unselektiert dargestellt, unabhängig von entsprechenden Mausklicks.

		WFLAG_DELETEONCLOSE	<p>Ohne dieses Flag wird ein Fenster beim Klicken auf den CloseButton nur in den HIDE-Modus versetzt. Das heißt es existiert noch und kann jederzeit wieder geöffnet werden.</p> <p>Mit diesem Flag wird das Fenster tatsächlich aus der GUI entfernt und gelöscht. Vorhandene Zeiger weisen nun allerdings auf freigegebenen Speicher, der nicht mehr dem Programm „gehört“. Es ist also Vorsicht geboten.</p>
		WFLAG_WFLAG_SENDWMSGTOPARENT	<p>Dieses Flag dient dazu, eine in einem Childwindow „erzeugte“ WINDOWMESSAGE als die des ParentWindows auszugeben. So kann man einen Fensterzweig in einer WINDOWMESSAGE zusammenfassen.</p> <p>Ein Beispiel: wenn man einen Dialog erstellt der mehrere Childs enthält. Egal wo das Ereignis im Fensterzweig auftritt, man erhält nur die WINDOWMESSAGE des DialogFensters.</p>
Hinweise:	/		
Beispiel:	/		
Rückgabewert:	Zeiger auf ein sGUIWindow		

EnableUserGFX		
function EnableUserGFX (win as _sGUIWindow ptr=0) as FB.Image ptr		
Beschreibung:	Der Rückgabewert ist ein Zeiger auf ein Image das im Fenster anstatt der Fensterhintergrundfarbe angezeigt wird. Dieses Image kann mit allen FB-eigenen Befehlen „bearbeitet“ werden.	
Parameter:	win	Zeiger auf ein sGUIWindow
Hinweise:	Technisch wird einfach nur ein weiteres Image erzeugt. Controls können in diesem Fenster natürlich weiterhin erzeugt und benutzt werden.	
Beispiel:		
Rückgabewert:	Ein FB.Image ptr	

CONTROL STUFF

GadgetOn GadgetSleep GadgetOff		
sub GadgetOn (gad as Gadget ptr) sub GadgetSleep (gad as Gadget ptr) sub GadgetOff (gad as Gadget ptr)		
Beschreibung:	Diese drei Befehle bestimmen die Benutz- und Sichtbarkeit eines Controls. Wenn ein Control ON ist, dann ist es sichtbar und benutzbar Wenn ein Control OFF ist, dann ist es unsichtbar und, logischerweise, nicht zu benutzen. Im SLEEP Modus sieht man ein Control zwar, es ist aber ohne Funktionalität.	
Parameter:	gad	Ein Zeiger auf ein Gadget
Hinweise:	die Gadget-Eigenschaft gad → Activation wird entsprechend mit einem Wert versehen 0=OFF, 1=ON, 2=SLEEP	
Beispiel:	/	
Rückgabewert:	/	

GetSelect		
function GetSelect (gad as Gadget ptr) as integer		
Beschreibung:	Liefert den Sektionszustand eines Controls	
Parameter:	Gad	Zeiger auf ein Gadget
Hinweise:	Diesen Zustand abzufragen ist nur sinnvoll bei ToggleGadgets, RadioButtons, CheckMarks. Es wird die Gadget-Eigenschaft gad → Selection zurückgeben.	
Beispiel:	/	
Rückgabewert:	0 oder 1	

SetSelect	
sub SetSelect (gad as Gadget ptr, Selection as integer)	
Beschreibung:	Liefert den Sektionszustand eines Controls

Parameter:	Gad	Zeiger auf ein Gadget
	Selection	Selektionszustand den das Control haben soll 0=unselktiert, 1=selektiert
Hinweise:	Diesen Zustand zu setzen ist nur sinnvoll bei ToggleGadgets, RadioButtons, CheckMarks. Es wird die Gadget-Eigenschaft gad → Selection gesetzt.	
Beispiel:	/	
Rückgabewert:	/	

AddArrow AddSmallArrow		
function AddArrow (win as sGUIWindow ptr, PosX as integer, PosY as integer, DirArrow as integer) as Gadget ptr function AddSmallArrow (win as sGUIWindow ptr, PosX as integer, PosY as integer, DirArrow as integer) as Gadget ptr		
Beschreibung:	Erzeugt ein Gadget mit einem Pfeilsymbol, in zwei Größen	
Parameter:	win	Ein Zeiger auf ein ParentWindow, in welchem das Gadget dargestellt werden soll. Wenn 0 übergeben wird, erscheint das Control auf dem Screen (das RootWindow).
	PosX PosY	Position dieses Fenster relativ zum ParentWindow.
	DirArrow	Richtung des Pfeilsymbols 0=links, 1=rechts, 2=hoch, 3=runter
Hinweise:	Beim Halten der Maustaste triggert das Gadget	
Beispiel:	/	
Rückgabewert:	Zeiger auf ein Gadget	

AddSimpleGadget		
function AddSimpleGadget (win as sGUIWindow ptr, PosX as integer, PosY as integer, GadWidth as integer=0, GadHeight as integer=0, Text as string) as Gadget ptr		
Beschreibung:	Erzeugt einen einfachen Button, beispielsweise für „Ok“, „Weiter“ oder „Abbruch“ usw.	
Parameter:	win	Ein Zeiger auf ein ParentWindow, in welchem das Gadget dargestellt werden soll. Wenn 0 übergeben wird, erscheint das Control auf dem Screen (das RootWindow).
	PosX PosY	Position dieses Fenster relativ zum ParentWindow.

	GadWidth GadHeight	Breite und Höhe des Gadgets. Wenn beide Parameter 0 sind, wird die Dimensionierung an den Gadgettext angepasst.
	Text	Gadgettext
Hinweise:	/	
Beispiel:	/	
Rückgabewert:	Zeiger auf ein Gadget	

AddToggleGadget		
function AddToggleGadget (win as sGUIWindow ptr, PosX as integer, PosY as integer, GadWidth as integer=0, GadHeight as integer=0, Selection as integer, Text as string) as Gadget ptr		
Beschreibung:	Erzeugt einen ToggleGadget, welches zwischen zwei Schaltzustände hat	
Parameter:	win	Ein Zeiger auf ein ParentWindow, in welchem das Gadget dargestellt werden soll. Wenn 0 übergeben wird, erscheint das Control auf dem Screen (das RootWindow).
	PosX PosY	Position dieses Fenster relativ zum ParentWindow.
	GadWidth GadHeight	Breite und Höhe des Gadgets. Wenn beide Parameter 0 sind, wird die Dimensionierung an den Gadgettext angepasst.
	Selection	Schaltzustand bei Erzeugung. Die Werte 0/1 entsprechen nicht selektiert/selektiert.
	Text	Gadgettext
Hinweise:	Den Selektionsstatus des Gadgets kann man mit SetSelect() und GetSelect setzen bzw auslesen. Alternativ dazu lässt sich dies auch direkt über die Gadget-Eigenschaft gad → Selection „regeln“.	
Beispiel:	/	
Rückgabewert:	Zeiger auf ein Gadget	

