

Inhaltsverzeichnis

Fenster (breite, hoehe uint16).....	2
FensterOffen () bool.....	2
FensterAus ().....	2
Cls ().....	2
Stiftfarbe (r,g,b uint8).....	2
Punkt (x,y uint16).....	2
GibPunktfarbe (x,y uint16) (r,g,b uint8).....	3
Linie (x1,y1,x2,y2 uint16).....	3
Kreis (x,y,r uint16).....	3
Vollkreis (x,y,r uint16).....	3
Ellipse (x,y,rx,ry uint16).....	3
Vollellipse (x,y,rx,ry uint16).....	3
Kreissektor (x,y,r,w1,w2 uint16).....	3
Vollkreissektor (x,y,r,w1,w2 uint16).....	3
Rechteck (x1,y1,b,h uint16).....	4
Vollrechteck (x1,y1,b,h uint16).....	4
Dreieck (x1,y1,x2,y2,x3,y3 uint16).....	4
Volldreieck (x1,y1,x2,y2,x3,y3 uint16).....	4
Schreibe (x, y int16, s string).....	4
SetzeFont (s string, groesse int).....	4
GibFont () string.....	4
SchreibeFont (x,y int16, s string).....	4
LadeBild (x, y int16, s string).....	4
LadeBildInsClipboard (s string).....	5
Archivieren ().....	5
Restaurieren (x,y,b,h uint16).....	5
Clipboard_kopieren (x,y,b,h uint16).....	5
Clipboard_einfuegen (x, y uint16).....	5
Sperren ().....	5
Entsperren ().....	5
UpdateAus ().....	5
UpdateAn ().....	6
Grafikzeilen () uint16.....	6
Grafikspalten () uint16.....	6
TastaturLesen1 () (taste uint16, gedrueckt uint8, tiefe uint16).....	6
Tastaturzeichen (taste, tiefe) rune.....	6
TastaturpufferAn ().....	6
TastaturpufferAus ().....	6
TastaturpufferLesen1 () (taste uint16, gedrueckt uint8, tiefe uint16).....	7
MausLesen1 () (taste uint8, status int8, mausX, mausY uint16).....	7
MauspufferAn ().....	7
MauspufferAus ().....	7
MauspufferLesen1 () (taste uint8,status int8, mausX, mausY uint16).....	7
SpieleSound (s string).....	7

package gfx

```
/* Autor: Stefan Schmidt
 * Datum: 03.03.2015 ; letzte Änderung: 07.10.2017
 * Zweck: Das Paket unterstützt (derzeit) Befehle zur Erstellung
 * einfacher Grafiken in einem eigenen Fenster
 * inklusive Schrift mit beliebigen Fonts und
 * Soundausgabe sowie Eingabe per Tastatur und Maus!
 *
 * SOWOHL UNTER X (LINUX) ALS AUCH UNTER MS WINDOWS - getestet :- )
 *
 * als auch unter MAC OS X (??) - mangels HW nicht getestet :-(
 *
 * AM GO-QUELLTEXT MÜSSEN KEINE ÄNDERUNGEN Vorgenommen werden!
 *
 * notwendige Voraussetzungen:
 * - TODO
 */
```

Vor.: Das Grafikfenster ist nicht offen. breite <=1920; hoehe <=1200
Eff.: Ein Fenster mit einer 'Zeichenfläche' von breite x hoehe Pixeln wurde geöffnet. Die Zeichenfarbe ist Schwarz. Der Ursprung (0,0) ist oben links im Fenster. Die x-Koordinate wächst horizontal nach rechts, die y-Koordinate vertikal nach unten.

Fenster (breite, hoehe uint16)

Vor.: -

Erg.: True ist geliefert, gdw. das Grafikfenster offen ist.

FensterOffen () bool

Vor.: Das Grafikfenster ist offen.

Eff.: Das Grafikfenster ist geschlossen.

FensterAus ()

Vor.: Das Grafikfenster ist offen.

Eff.: Alle Pixel des Grafikfenster haben nun die aktuelle Stiftfarbe, d.h., der Inhalt des Fensters ist gelöscht.

Cls ()

Vor.: Das Grafikfenster ist offen.

Eff.: Die Zeichenfarbe ist gemäß dem RGB-Farbmodell neu gesetzt.

Beispiel: Stiftfarbe (0xFF, 0, 0) ist Rot.

Stiftfarbe (r,g,b uint8)

Vor.: Das Grafikfenster ist offen.

Eff.: An der Position (x,y) ist ein Punkt in der aktuellen Stiftfarbe gesetzt.

Punkt (x,y uint16)

Vor.: Das Grafikfenster ist offen.

Erg.: Der Rot-, Grün- und Blauanteil des Punktes mit den Koordinaten (x,y) im Grafikfenster ist geliefert.

GibPunktfarbe (x,y uint16) (r,g,b uint8)

Vor.: Das Grafikfenster ist offen.

Eff.: Von der Position (x_1, y_1) bis (x_2, y_2) eine Strecke mit der Strichbreite 1 Pixel in der aktuellen Stiftfarbe gezeichnet.

Linie $(x_1, y_1, x_2, y_2 \text{ uint16})$

Vor.: Das Grafikfenster ist offen.

Eff.: Um den Mittelpunkt $M(x, y)$ ein Kreis mit dem Radius r mit der Strichbreite 1 Pixel in der aktuellen Stiftfarbe gezeichnet.

Kreis $(x, y, r \text{ uint16})$

Vor.: Das Grafikfenster ist offen.

Eff.: Um den Mittelpunkt $M(x, y)$ ein ausgefüllter Kreis mit dem Radius r in der aktuellen Stiftfarbe gezeichnet.

Vollkreis $(x, y, r \text{ uint16})$

Vor.: Das Grafikfenster ist offen.

Eff.: Um den Mittelpunkt $M(x, y)$ ist mit der horizontalen Halbachse r_x und der vertikalen Halbachse r_y mit der Strichbreite 1 Pixel in der aktuellen Stiftfarbe eine Ellipse gezeichnet.

Ellipse $(x, y, r_x, r_y \text{ uint16})$

Vor.: Das Grafikfenster ist offen.

Eff.: Um den Mittelpunkt $M(x, y)$ ist mit der horizontalen Halbachse r_x und der vertikalen Halbachse r_y in der aktuellen Stiftfarbe eine ausgefüllte Ellipse gezeichnet.

Vollellipse $(x, y, r_x, r_y \text{ uint16})$

Vor.: Das Grafikfenster ist offen.

Eff.: Um den Mittelpunkt $M(x, y)$ ist mit dem Radius r in der aktuellen Stiftfarbe ein Kreissektor(Tortenstück:-)) gezeichnet. w_1 ist dabei der Startwinkel in Grad, w_2 der Endwinkel in Grad. Ein Winkelmaß von 0 Grad bedeutet in Richtung Osten gehts los, dann entgegengesetzt zum Uhrzeigersinn.

Kreissektor $(x, y, r, w_1, w_2 \text{ uint16})$

Vor.: Das Grafikfenster ist offen.

Eff.: Um den Mittelpunkt $M(x, y)$ ist mit dem Radius r in der aktuellen Stiftfarbe ein gefüllter Kreissegment gezeichnet. w_1 ist dabei der Startwinkel in Grad, w_2 der Endwinkel in Grad. Ein Winkelmaß von 0 Grad bedeutet in Richtung Osten gehts los, dann entgegengesetzt zum Uhrzeigersinn.

Vollkreissektor $(x, y, r, w_1, w_2 \text{ uint16})$

Vor.: Das Grafikfenster ist offen.

Eff.: In der aktuellen Stiftfarbe ist ein Rechteck gezeichnet. Die Position (x_1, y_1) gibt die linke obere Ecke des Rechtecks an, b die Breite in x -Richtung, h die Höhe in y -Richtung. Die Seiten des Rechtecks verlaufen parallel zu den Achsen.

Rechteck $(x_1, y_1, b, h \text{ uint16})$

Vor.: Das Grafikfenster ist offen.

Eff.: In der aktuellen Stiftfarbe ist ein gefülltes Rechteck gezeichnet. Die Position (x1,y1) gibt die linke obere Ecke des Rechtecks an, b die Breite in x-Richtung, h die Höhe in y-Richtung. Die Seiten des Rechtecks verlaufen parallel zu den Achsen.

Vollrechteck (x1,y1,b,h uint16)

Vor.: Das Grafikfenster ist offen.

Eff.: In der aktuellen Stiftfarbe ist ein Dreieck mit den Eckpunktkoordinaten (x1,y1), (x2,y2) und (x3,y3) gezeichnet.

Dreieck (x1,y1,x2,y2,x3,y3 uint16)

Vor.: Das Grafikfenster ist offen.

Eff.: In der aktuellen Stiftfarbe ein gefülltes Dreieck mit den Eckpunktkoordinaten (x1,y1), (x2,y2) und (x3,y3) gezeichnet.

Volldreieck (x1,y1,x2,y2,x3,y3 uint16)

Vor.: Das Grafikfenster ist offen. s beinhaltet maximal 255 Bytes.

Eff.: In der aktuellen Stiftfarbe ist der Text s hingeschrieben ohne den Hintergrund zu verändern. Die Position (x,y) ist die linke obere Ecke des Bereichs des ersten Buchstaben von S.

Schreibe (x, y int16, s string)

Vor.: s gibt die ttf-Datei des Fonts mit Pfad an. groesse gibt die gewünschte Punkthöhe der Buchstaben an.

Eff.: Der angegebene Font ist nun der aktuelle Font, der bei Aufruf von SchreibeFont () verwendet wird.

SetzeFont (s string, groesse int)

Vor.: keine

Erg.: Der mit SetzeFont () hinterlegte Pfad inklusive Dateiname des aktuell gewünschten Fonts ist geliefert.

GibFont () string

Vor.: Das Grafikfenster ist offen. s beinhaltet maximal 255 Bytes.

Eff.: In der aktuellen Stiftfarbe ist der Text s mit dem zuletzt mit SetzeFont() gesetzten Font hingeschrieben ohne den Hintergrund zu verändern. Die Position (x,y) ist die linke obere Ecke des Bereichs des ersten Buchstaben von S.

SchreibeFont (x,y int16, s string)

Vor.: Das Grafikfenster ist offen. s beinhaltet maximal 255 Bytes und stellt den Dateinamen eines Bildes im bmp-Format dar.

Eff.: Ab der Position (x,y) ist das angegebene rechteckige Bild eingefügt. Die Position ist die linke obere Ecke des Bildes. Die Bildkanten verlaufen parallel zu den Achsen.

LadeBild (x, y int16, s string)

Vor.: Das Grafikfenster ist offen. s beinhaltet maximal 255 Bytes und stellt den Dateinamen eines Bildes im bmp-Format dar.

Eff.: Das angegebene Bild ist in einen Zwischenspeicher (das Clipboard) geladen. Vorher im Clipboard enthaltene Daten wurden damit überschrieben.

LadeBildInsClipboard (s string)

Vor.: Das Grafikfenster ist offen.

Eff.: Der gesamter Inhalt des Fensters ist in einen (versteckten) Zwischenspeicher kopiert. Daten, die vorher in diesem Zwischenspeicher waren, wurden überschrieben.

Archivieren ()

Vor.: Das Grafikfenster ist offen. Archivieren wurde vorher mindestens einmal aufgerufen und seit dem das Fenster nicht geschlossen.

Eff.: Der angegebene rechteckige Bereich des versteckten Zwischenspeichers (s. Archivieren) ist an seine ursprüngliche Stelle ins Grafikfenster zurückkopiert.

Restaurieren (x,y,b,h uint16)

Vor.: Das Grafikfenster ist offen.

Eff.: Der angegebene rechteckige Grafikfensterbereich ist in einem Zwischenspeicher (das Clipboard) kopiert. Daten, die vorher in diesem Zwischenspeicher waren, wurden überschrieben.

Clipboard_kopieren (x,y,b,h uint16)

Vor.: Das Grafikfenster ist offen, Clipboard_kopieren wurde vorher mindestens einmal aufgerufen und seitdem wurde das Fenster nicht geschlossen.

Eff.: Der Inhalt des Zwischenspeichers (Clipboard) ist an die angegebene Position (x,y) ins Grafikfenster kopiert. Dort vorher vorhandene Daten wurden überschrieben.

Clipboard_einfuegen (x, y uint16)

Vor.: Das Grafikfenster ist offen. Sperren wurde noch nicht aufgerufen bzw. der Aufruf wurde mit einem Aufruf von Entsperrern 'neutralisiert'.

Eff.: Das Grafikfenster ist nun nur noch vom aufrufenden Prozess 'beschreibbar', wenn alle anderen Prozesse vor einem Schreibzugriff auf das Grafikfenster ebenfalls Sperren aufrufen. Gegebenenfalls war der aufrufende Prozess solange blockiert, bis er den Zugriff erhielt. Andere Prozesse, die nun Sperren ausführen, sind blockiert.

Sperren ()

Vor.: Das Grafikfenster ist offen. Sperren wurde aufgerufen und seit dem das Grafikfenster nicht geschlossen.

Eff.: Das Grafikfenster ist für andere Prozesse wieder zum 'Beschreiben' freigegeben.

Entsperren ()

Vor.: Das Grafikfenster ist offen.

Eff.: Die abgesetzten Grafikbefehle werden nicht sofort im Fenster, sondern lediglich im 'Double-Buffer-Bereich' verdeckt durchgeführt.

UpdateAus ()

Vor.: Das Grafikfenster ist offen und wurde nach einem 'UpdateAus()' nicht geschlossen.

Eff.: Alle nach 'UpdateAus ()' durchgeführten Änderungen durch abgesetzte Grafikbefehle sind nun sichtbar geworden. Folgende Befehle werden wieder direkt umgesetzt.

UpdateAn ()

Vor.: Das Grafikfenster ist offen.

Erg.: Die Anzahl der Grafikfensterzeilen (Pixelzeilen) ist geliefert.

Grafikzeilen () uint16

Vor.: Das Grafikfenster ist offen.

Erg.: Die Anzahl der Grafikfensterspalten (Pixelspalten) ist geliefert.

Grafikspalten () uint16

Vor.: Das Grafikfenster ist offen.

Erg.: Der aufrufende Prozess war solange blockiert, bis eine Taste auf der Tastatur gedrückt oder losgelassen wurde. Geliefert ist mit 'taste' die Tastennummer. 'gedrückt' ist 1 (0), falls die Taste gedrückt (losgelassen) wurde. 'tiefe' liefert die Kombination der gedrückten Steuerungstasten.

TastaturLesen1 () (taste uint16, gedrueckt uint8, tiefe uint16)

Vor.: Das Grafikfenster ist offen.

Erg.: Wenn -tiefe- nur SHIFT oder STANDARD (also kein SHIFT) in Kombination mit NUMLOCK und/oder ALT GR ist und eine Tastaturzeichen-Taste mit -taste- übergeben wurde (also keine Steuertastenkombination), so ist das entsprechende Tastaturzeichen als Rune geliefert. Andernfalls ist rune(0) geliefert. -tiefe- und -taste- erhält man i.d.R. durch Tastaturlesen1().

Tastaturzeichen (taste, tiefe) rune

Vor.: Das Grafikfenster ist offen.

Eff.: Ab jetzt werden bis zu 255 Tastaturereignisse in einem versteckten Tastaturpuffer zwischengespeichert. Darüber hinausgehende eingehende Tastaturevents gehen verloren.

TastaturpufferAn ()

Vor.: Das Grafikfenster ist offen.

Eff.: Der Tastaturpuffer ist aus. Enthaltene Events sind verloren.

TastaturpufferAus ()

Vor.: Das Grafikfenster ist offen.

Erg.: Das vorderste Element (gespeicherte Event) des Tastaturpuffers ist ausgelesen, aus dem Puffer entfernt und zurueckgegeben: Geliefert ist mit 'taste' die Tastennummer. 'gedrückt' ist 1 (0), falls die Taste gedrückt (losgelassen) wurde. 'tiefe' liefert die Kombination der gedrückten Steuerungstasten. War der Puffer leer, so war der aufrufende Prozess solange blockiert, bis etwas gelesen werden konnte.

TastaturpufferLesen1 () (taste uint16, gedrueckt uint8, tiefe uint16)

Vor.: Das Grafikfenster ist offen.

Erg.: Der aufrufende Prozess war solange blockiert, bis Daten von der Maus gelesen werden konnten. Mit 'taste' erhält man die Nummer der betreffenden Maustaste. Mit 'status' (1/0/-1), ob sie gedrückt bzw. unverändert ist oder losgelassen wurde. 'mausX' und 'mausY' sind die Koordinaten der Mauszeigerspitze.

MausLesen1 () (taste uint8, status int8, mausX, mausY uint16)

Vor.: Das Grafikfenster ist offen.

Eff.: Ab jetzt werden bis zu 255 Mausereignisse (Events) zwischen- gespeichert. Darüber hinaus eingehende Maus-Events gehen verloren.

MauspufferAn ()

Vor.: Das Grafikfenster ist offen.

Eff.: Der Mauspuffer ist deaktiviert. Enthaltene Ereignisse sind verloren.

MauspufferAus ()

Vor.: Das Grafikfenster ist offen.

Erg.: Das vorderste Mausereignis ist aus dem Puffer gelesen, dort entfernt und zurückgegeben: Mit 'taste' erhält man die Nummer der betreffenden Maustaste. Mit 'status' (1/0/-1), ob sie gedrückt bzw. unverändert ist oder losgelassen wurde. 'mausX' und 'mausY' sind die Koordinaten der Mauszeigerspitze.

War der Puffer leer, so war der aufrufende Prozess solange blockiert, bis er etwas lesen konnte.

MauspufferLesen1 () (taste uint8,status int8, mausX, mausY uint16)

Vor.: s ist der Dateiname der wav-Datei inklusive Pfad. Zum Zeitpunkt des Aufrufs werden gerade höchstens 9 .wav-Dateien abgespielt.

Eff.: Die angegebene wav-Datei wird ab jetzt auch abgespielt. Das Programm läuft ohne Verzögerung weiter.

SpieleSound (s string)