

Das LWB-Grafikpaket "gfx"

```
/* Autor: Stefan Schmidt
 * Datum: 03.03.2015 ; letzte Änderung: 26.04.2021
 * 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 :- )
 *        AM GO-QUELLTEXT MÜSSEN KEINE ÄNDERUNGEN VORGENOMMEN WERDEN!
 *
 * ACHTUNG: Die Darstellung von Grafikobjekte im Grafikfenster ist nur dann
 *          sichergestellt, wenn sie vollständig im sichtbaren Bereich liegen. */
```

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, ohne Transparenz. 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.

Erg.: Die Anzahl der Grafikfensterzeilen (Pixelzeilen) des gfx-Fensters ist geliefert.

Grafikzeilen () uint16

Vor.: Das Grafikfenster ist offen.

Erg.: Die Anzahl der Grafikfensterspalten (Pixelspalten) des gfx-Fensters ist geliefert.

Grafikspalten () uint16

Vor.: Das Grafikfenster ist offen.

Eff.: Das gfx-Fenster hat sichtbar den neuen Fenstertitel s. In der Regel verwendet man hier den Programmnamen.

Fenstertitel (s string)

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. Die Transparenz der Stiftfarbe kann mit der Funktion Transparenz eingestellt werden.

Stiftfarbe (r,g,b uint8)

Vor.: Das Grafikfenster ist offen.

Eff.: Die Transparenz der Stiftfarbe bzw. die von "Zeichenoperationen" ist neu gesetzt. 0 bedeutet keine Transparenz (Standard), 255 komplett durchsichtig. Wenn also etwas nach dem Aufruf gezeichnet wird, so scheint vorher Gezeichnetes ggf. durch.

Transparenz (t 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 (x1,y1) bis (x2,y2) ist eine Strecke mit der Strichbreite 1 Pixel in der aktuellen Stiftfarbe gezeichnet.

Linie (x1,y1,x2,y2 uint16)

Vor.: Das Grafikfenster ist offen.

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

Kreis (x,y,r uint16)

Vor.: Das Grafikfenster ist offen.

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

Vollkreis (x,y,r uint16)

Vor.: Das Grafikfenster ist offen.

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

Ellipse (x,y,rx,ry uint16)

Vor.: Das Grafikfenster ist offen.

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

Vollellipse (x,y,rx,ry 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. w1 ist dabei der Startwinkel in Grad, w2 der Endwinkel in Grad. Ein Winkelmaß von 0 Grad bedeutet in Richtung Osten geht es los, dann entgegengesetzt zum Uhrzeigersinn.

Kreissektor (x,y,r,w1,w2 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. w1 ist dabei der Startwinkel in Grad, w2 der Endwinkel in Grad. Ein Winkelmaß von 0 Grad bedeutet in Richtung Osten geht es los, dann entgegengesetzt zum Uhrzeigersinn.

Vollkreissegment (x,y,r,w1,w2 uint16)

Vor.: Das Grafikfenster ist offen.

Eff.: In der aktuellen Stiftfarbe ist ein 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.

Rechteck (x1,y1,b,h 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 und ist ein ASCII-Code-String.

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

Schreibe (x, y uint16, s string)

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

Eff.: Wenn es die ttf-Datei gibt, so ist der angegebene Font nun der aktuelle Font, der bei Aufruf von SchreibeFont () verwendet wird.

Erg.: -true- ist geliefert, gdw. die ttf-Datei an der Stelle lag und der Font als aktuellerFont gesetzt werden konnte.

SetzeFont (s string, groesse int) bool

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 uint16, 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 gemäß der aktuell eingestellten Transparenz eingefügt. Die Position ist die linke obere Ecke des Bildes. Die Bildkanten verlaufen parallel zu den Achsen.

LadeBild (x, y uint16, s string)

Vor.: Das Grafikfenster ist offen. s beinhaltet maximal 255 Bytes und stellt den Dateinamen eines Bildes im bmp-Format dar. r,g und b geben eine Pixelfarbe an (ColorKey).

Eff.: Ab der Position (x,y) ist das angegebene rechteckige Bild gemäß der eingestellten Transparenz eingefügt. Die Position ist die linke obere Ecke des Bildes. Die Bildkanten verlaufen parallel zu den Achsen. Alle Pixel des Bildes mit den Farbwerten r,g und b werden jedoch vollkommen transparent dargestellt! Ursprüngliche Pixel im Grafikfenster werden hier nicht überzeichnet!

LadeBildMitColorKey (x,y uint16, s string, r,g,b uint8)

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 gesamte 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. Die gesetzte Transparenz hat keinen Einfluss auf die Funktion.

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, wobei die gesetzte Transparenz entsprechenden Einfluss hatte.

Clipboard_einfuegen (x, y uint16)

Vor.: Das Grafikfenster ist offen, Clipboard_kopieren wurde vorher mindestens einmal aufgerufen und seitdem wurde das Fenster nicht geschlossen. r,g und b geben eine Pixelfarbe an.

Eff.: Der Inhalt des Zwischenspeichers (Clipboard) ist an die angegebene Position (x,y) ins Grafikfenster unter Beachtung der gesetzten Transparenz kopiert. Alle Pixel des Clipboards mit dem durch r,g und b festgelegten Farbwertes sind jedoch vollkommen transparent und ändern so das ursprüngliche Pixel im Grafikfenster an dieser Stelle nicht.

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

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 gfx-Anweisungen 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 gfx-Anweisungen werden wieder direkt umgesetzt.

UpdateAn ()

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.: Die Tastaturbelegung ist deutsch.

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 Steuertasten-kombination), 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 zurückgegeben: 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) zwischengespeichert. 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.: Das Grafikfenster ist offen. 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)

Vor.: Das Grafikfenster ist offen.

Erg.: Das aktuelle Noten-Tempo ist geliefert, d.h. die Anzahl der vollen Noten pro Minute.

GibNotenTempo () uint8

Vor.: $30 \leq \text{tempo} \leq 240$; Das Grafikfenster ist offen.

Eff.: Das Noten-Tempo ist auf den Wert t gesetzt, es gibt also t volle Noten pro Minute.

SetzeNotenTempo (t uint8)

Vor.: Das Grafikfenster ist offen.

Erg.: Anschlagzeit, Abschwellzeit, Haltepegel und Ausklingzeit sind in ms geliefert.

GibHuellkurve () (float64,float64,float64,float64)

Vor.: Das Grafikfenster ist offen.

a ist die Anschlagzeit in ms mit $0 \leq a \leq 1$, d ist die Abschwellzeit in ms mit $0 \leq d \leq 5$,

s ist der Haltepegel in Prozent vom Maximum mit $0 \leq s \leq 1.0$,

r ist die Ausklingzeit in ms mit $0 \leq r \leq 5$.

Eff.: Für die Hüllkurve zukünftig zu spielender Töne bzw. Noten sind die Parameter gesetzt.

SetzeHuellkurve (a,d,s,r float64)

Vor.: Das Grafikfenster ist offen.

Erg.: Geliefert sind:

Abtastrate der WAV-Daten in Hz, z.B. 44100, Auflösung der Klänge (1: 8 Bit; 2: 16 Bit),

die Anzahl der Kanäle (1: mono, 2: stereo), die Signalform (0: Sinus, 1: Rechteck, 2: Dreieck,

3: Sägezahn) und die Pulsweite HIGH bei Rechteckform als Prozentsatz zw. 0 und 1.

GibKlangparameter () (uint32,uint8,uint8,uint8,float64)

Vor.: Das Grafikfenster ist offen. rate ist die Abtastrate, z.B. 11025, 22050 oder 44100.

auflösung ist 1 für 8 Bit oder 2 für 16 Bit. kanaele ist 1 für mono oder 2 für stereo.

signal gibt die Signalform an: 0: Sinus, 1: Rechteck, 2: Dreieck, 3: Sägezahn

p ist die Pulsweite für Rechtecksignale und gibt den Prozentsatz ($0 \leq p \leq 1$) für den HIGH-Teil an.

Eff.: Die klangparameter sind auf die angegebenen Werte gesetzt.

SetzeKlangparameter(rate uint32, aufloesung,kanaele,signal uint8, p float64)

Vor.: Das Grafikfenster ist offen. Das erste Zeichen von tonname ist eine Ziffer von 0 bis 9 und

gibt die Oktave an. Erlaubte weitere Zeichen für den Notennamen sind "C", "D", "E", "F", "G",

"A", "H", "C#", "D#", "F#", "G#", "A#". $0 < \text{laenge} \leq 1$; laenge 1: volle Note; 1.0/2: halbe

Note, ..., 1.0/16: sechzehntel Note $0.0 \leq \text{wartedauer}$; Die Wartedauer gibt die Dauer in

Notenlänge an, nach der nach dem Anspielen der Note im Programmablauf fortgefahren wird. 0:

keine Wartedauer; 1.0/2: Dauer einer halben Note, ...

Es werden gerade höchstens 9 Noten oder WAV-Dateien abgespielt.

Eff.: Der Ton wird gerade gespielt bzw. ist gespielt. Je nach Wartedauer wurde die Fortsetzung des

Programms verzögert. Der voreingestellte Standard ist aus 'GibHuellkurve ()' und

'GibKlangParameter()' ersichtlich. Die Einstellungen mit 'SetzeHuellkurve' und

'SetzeKlangparameter' haben Einfluss auf den "Ton".

SpielerNote (tonname string, laenge float64, wartedauer float64)