

abc – Programmierungsübersicht

abc - Programming overview



- [Programmiersprache / programming language](#)
- [abc Struktur / abc structure](#)
- [Open Window](#)
- [Textfelder / text fields](#)
- [Lineedit](#)
- [Bilder / Pictures](#)
- [Buttons / pixbuttons](#)
- [Numeric Slider](#)
- [Combo Box](#)
- [File Open](#)
- [Info Dialog](#)
- [Movie](#)
- [RS232 abfragen / polling](#)
- [SQLite Datenbank / SQLite Database](#)
- [printout](#)
- [Icdhandling](#)
- [Debug](#)



YaBasic ist ein quelloffener BASIC-Interpreter für Linux/Unix und für Windows. Er zeichnet sich durch seine Kompaktheit und eine flexible Syntax aus: Für die Farbwahl sind zum Beispiel sowohl der Befehl `color` als auch `colour` möglich, und auch für `if ... then ... else ... endif`-Konstrukte gibt es eine Reihe verschiedener Syntax-Varianten usw.

YaBasic is an open source BASIC interpreter for Linux/Unix and Windows. It is characterized by its compactness and flexible syntax: for example, both the commands `color` and `colour` can be used to select colors, and there are also a number of different syntax variants for `if ... then ... else ... endif` constructs, etc.

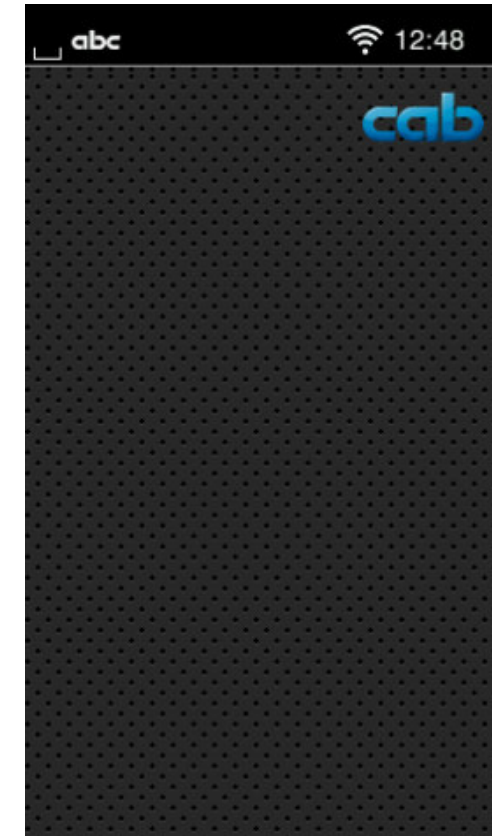
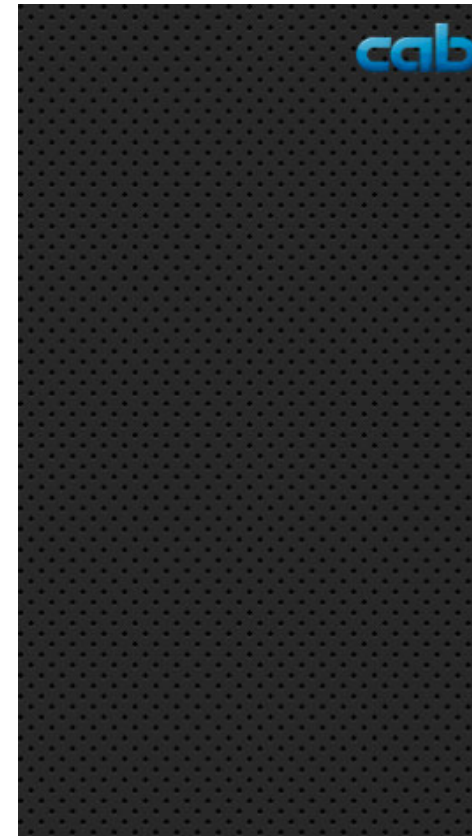


abc aktiv?
abc active?



Ob ein ABC-Programm aktiv ist, lässt sich entweder daran erkennen, dass keine Statuszeile angezeigt wird, oder daran, dass das abc-Symbol sichtbar ist.

You can tell whether an ABC program is active either by the fact that no status bar is displayed or by the fact that the abc icon is visible.



abc Struktur

abc structure



```
<ABC>
...
open window 480,272
poke("lcd"),1
...
routine()

    do
        ..
        openfile()
    Loop

sub routine()
    ...
end sub

sub openfile()
    ...
end sub

</ABC>
```



open window im Hoch- und Querformat open window in portrait and landscape format

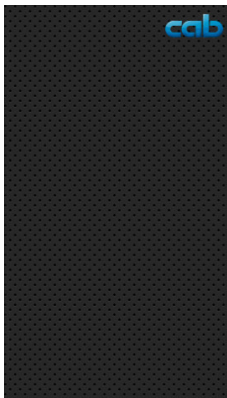


Hochformat

<ABC>

```
open window 272,480  
window read from "/pics/background.png"  
poke "gui.label.add","cab,195,-10,70,70"  
poke "gui.cab.image","/pics/logo.png"  
poke("lcd"),1
```

</ABC>

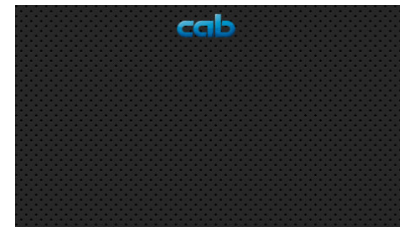


Querformat

<ABC>

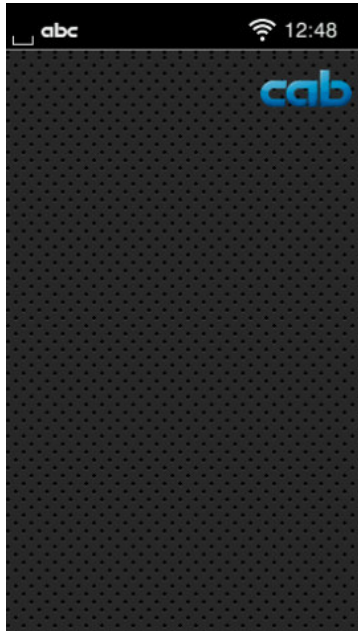
```
open window 480,272  
window read from "/pics/background.png"  
poke "gui.label.add","cab,195,-10,70,70"  
poke "gui.cab.image","/pics/logo.png"  
poke("lcd"),1
```

</ABC>



open window mit LCD Verschiebung X & Y

open window with LCD displacement X & Y



```
<ABC>  
  poke "lcdmodal",0  
  open window 272,440  
  poke "lcdx",0  
  poke "lcdy",40  
  window read from "/pics/background.png"  
  poke "gui.label.add","cab,195,-10,70,70"  
  poke "gui.cab.image","/pics/logo.png"  
  poke("lcd"),1  
</ABC>
```

on interrupt continue



Mit der Einstellung „*on interrupt continue*“ kann das ABC-Programm nicht über RS232, I/O oder TCP/IP beendet werden. Der Drucker ignoriert „Total Cancel“-Signale und bietet so eine zusätzliche Sicherheitsebene, die unerwünschtes beenden verhindert und den Druckvorgang stabil hält.

With the “on interrupt continue” setting, the ABC program is not terminated via RS232, I/O, or TCP/IP. The printer ignores “Total Cancel” signals, providing an additional level of security that prevents unwanted termination and keeps the printing process stable.

<ABC>

```
on interrupt continue
```

```
...
```

```
open window 480,272
```

```
poke("lcd"),1
```

```
do
```

```
    openfile()
```

```
Loop
```

```
sub openfile()
```

```
...
```

```
end sub
```

</ABC>



Definition von Textfeldern

Definition of text fields

<ABC>

```
open window 272,480  
window read from "/pics/background.png"
```

```
POKE("color#2"), dec("ffffff")  
POKE("fcolor"), 2  
FONT "Swiss 721, 20"
```

```
TEXT 50,10,"Kunden Programm"
```

```
POKE "gui.label.add", "textfeld,10,50,400,50,Mein Text....:"
```

```
POKE "gui.textfeld.visible", 1  
POKE "gui.textfeld.text", "Neuer Text "+value$  
POKE "gui.textfeld.visible", 0
```

```
poke("lcd"),1
```

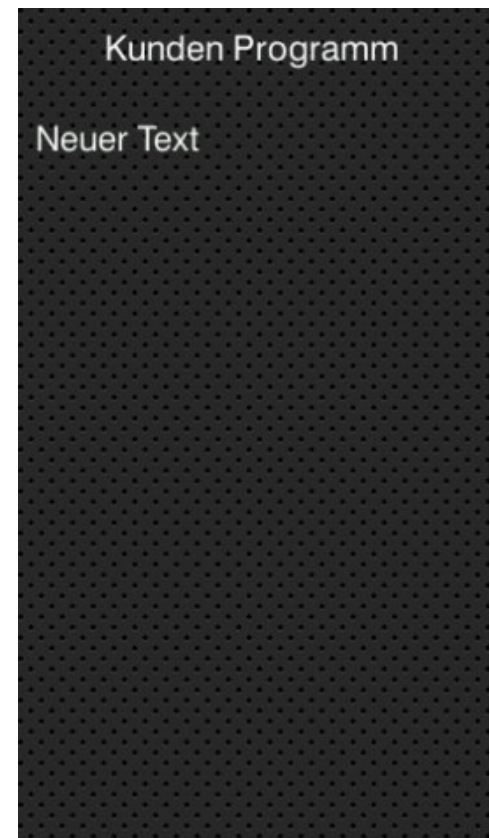
</ABC>

Textformatierung
text formatting

Fixer Inhalt
Fixed content

Flexibler Inhalt
Flexible content

Neuer Inhalt,
sicht- & unsichtbar
New content,
visible & invisible



Eingabe mit lineedit

Input with lineedit

1. Parameter: Object Name
2. Parameter: Position X on abc Window
3. Parameter: Position Y on abc Window
4. Parameter: Width of LineEdit
5. Parameter: Height of LineEdit
6. Parameter: Content (Optional)

```
poke "gui. lineedit.add", "myedit,10,10,200,40,cab Products"
```

```
poke("gui.myedit.focus"),1
```

```
poke("color#1"), dec("ffffff") // White
```

```
poke("color#2"), dec("000000") // Black
```

DO

```
pause 0.01
```

```
event$ = peek$("gui.event")
```

```
if (len(event$)) poke("stdout"),event$
```

```
if (instr(event$, "ReturnPressed")) or (instr(event$, "OnScreenKeyboardVisible:0")) break
```

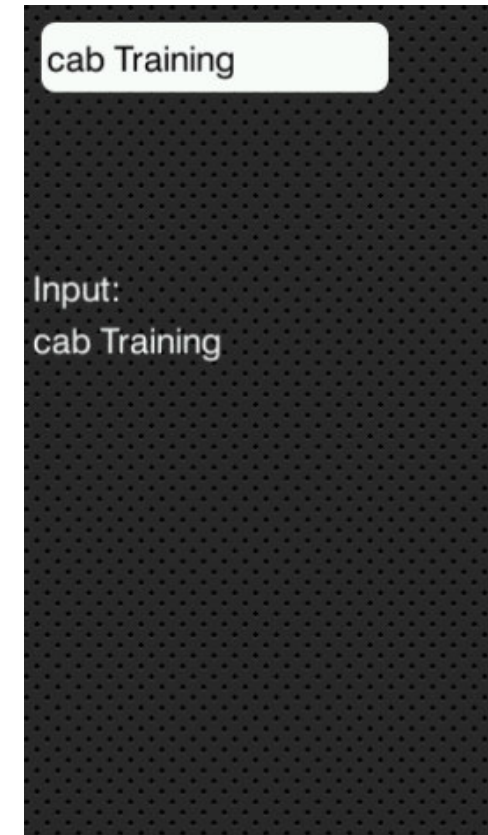
LOOP

```
poke("gui.myedit.visible"),0
```

```
FONT "Swiss 721,20":poke("fcolor"),1
```

```
Text 5,150,"Input:"
```

```
Text 5,180,peek$("gui.myedit.text")
```



LineEdit



Bild Picture



<ABC>

```
POKE "gui.label.add","logo,20,100,232,232,,
```

```
POKE "gui.logo.visible",1
```

```
POKE "gui.logo.image","logo_cutomer.png"
```

```
POKE "gui.logo.visible",0
```

</ABC>



Button & pixbutton

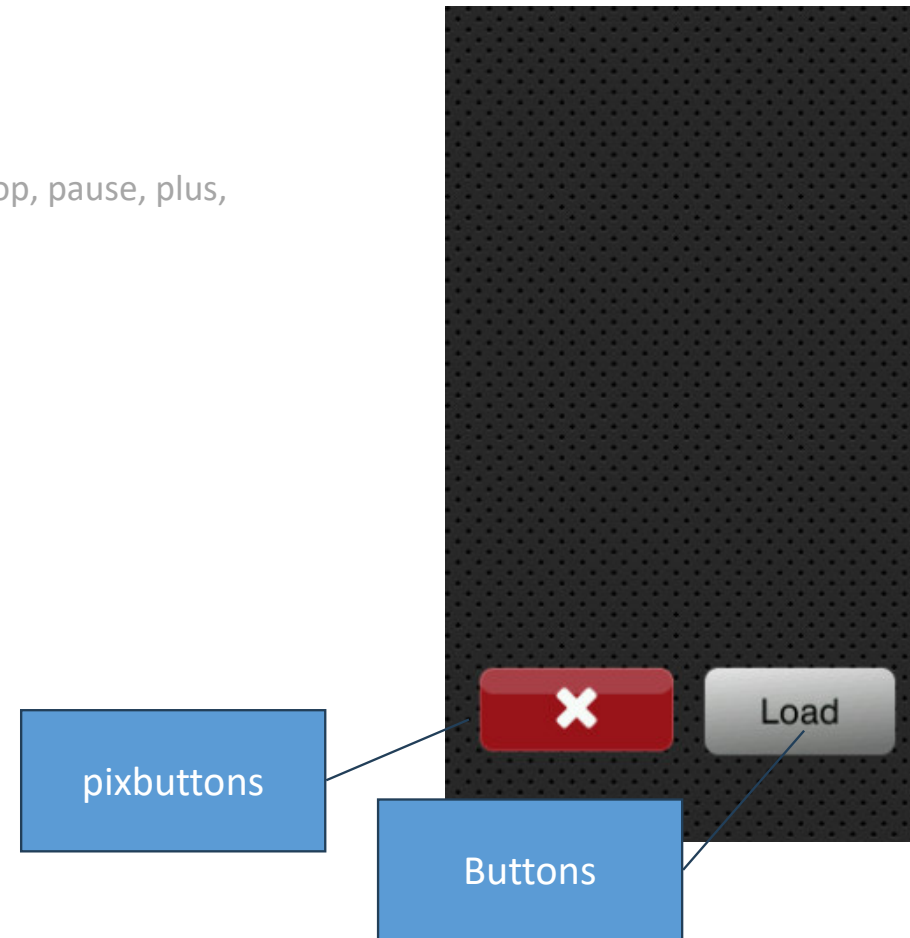
1. Parameter: **Object Name**
2. Parameter: **Position X on abc Window**
3. Parameter: **Position Y on abc Window**
4. Parameter: **Size (small , medium or large)**
5. Parameter: Type of Pixbutton (back, next, ok, cancel, feed, start, stop, pause, plus, minus, setup, trigger and help)
6. Parameter: **Content (Optional) for Button**

```
POKE "gui.button.add"," reload,150,380,110,50,Load"  
POKE "gui.pixbutton.add"," cancel_s,75,380,large,cancel"
```

DO

```
    pause 0.01  
    event$ = peek$("gui.event")  
    if (instr(var$, "cancel_s")) then  
        exit  
    endif  
    if (instr(var$, "reload")) then  
        openfile()  
    endif
```

LOOP



Beispiel Button

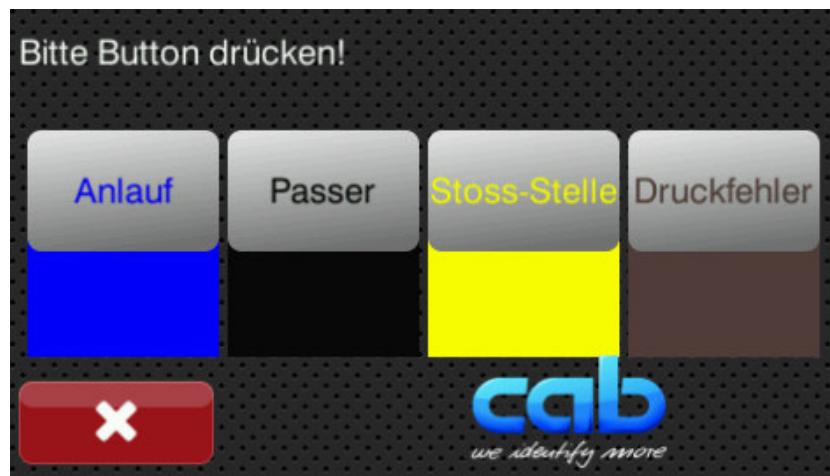
Example button

```
poke("color#3"), dec("0000ff")  
poke("fcolor"), 3  
poke("bcolor"), 3  
poke("gui.button.add"),"label1,10,70,110,70,Anlauf"  
fill rectangle 10,90 to 119,200
```

```
poke("color#4"), dec("0a0a0a")  
poke("fcolor"), 4  
poke("bcolor"), 4  
poke("gui.button.add"),"label2,125,70,110,70,Passer"  
fill rectangle 125,90 to 234,200
```

```
poke("color#5"), dec("ffff00")  
poke("fcolor"), 5  
poke("bcolor"), 5  
poke("gui.button.add"),"label3,240,70,110,70,Stoss-Stelle"  
fill rectangle 240,90 to 349,200
```

```
poke("color#6"), dec("563e39")  
poke("fcolor"), 6  
poke("bcolor"), 6  
poke("gui.button.add"),"label4,355,70,110,70,Druckfehler"  
fill rectangle 355,90 to 464,200
```



Beispiele pixbutton

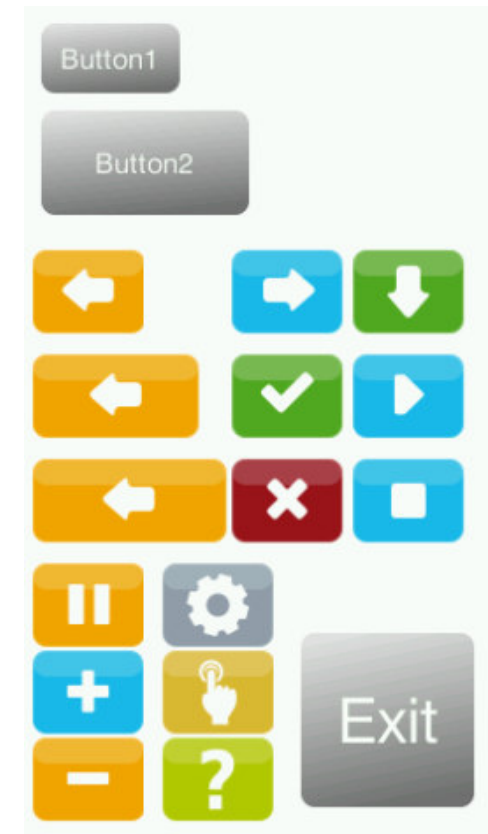
POKE "gui.pixbutton.add", "variable,75,380,groesse,type "

Pixbutton Größe:

tiny, small, medium or large

Pixbutton Type:

cancel, feed, help, minus, next, ok, pause, plus, setup, start, stop, trigger



Buttons

Numeric Slider

1. Parameter: **Caption Dialog**
2. Parameter: **Selected default Value after Open**
3. Parameter: **min. Value**
4. Parameter: **max. Value**
5. Parameter: **Step Size**
6. Parameter: **Digits behind Comma**

```
poke "gui.numeric.show", "How many?,30,0,100,1,0"
```

```
DO
```

```
    pause 0.01  
    event$ = peek$("gui.event")  
    if (instr(event$, "DialogClosed")) break
```

```
LOOP
```

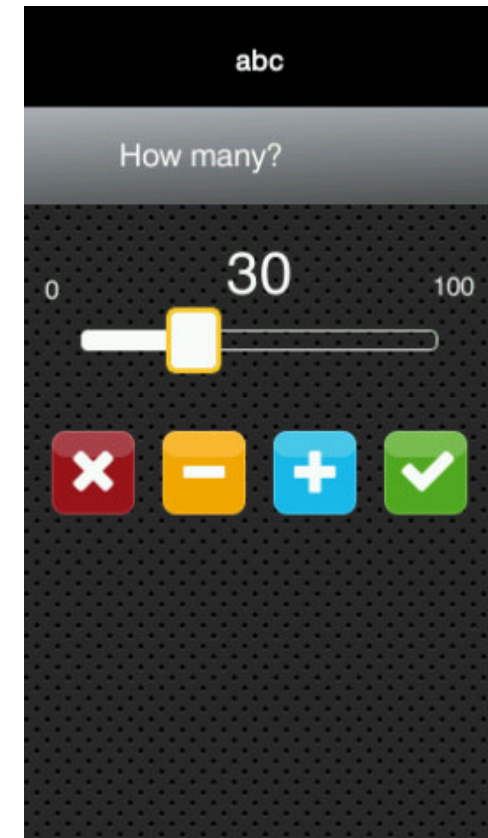
```
FONT "Swiss 721,20"
```

```
Text 5,10,"Selected Value:"
```

```
if (instr(event$, ":DialogClosed:ok:")) then
```

```
    TEXT 5,40,mid$(event$,len(":DialogClosed:ok:")+1)
```

```
endif
```



numericSlider



Combo Box

1. Parameter: **Object Name**
2. Parameter: **Position X on abc Window**
3. Parameter: **Position Y on abc Window**
4. Parameter: **Width of ComboBox**
5. Parameter: **Height of ComboBox**
6. Parameter: **Content for the Combobox, separated with Comma**

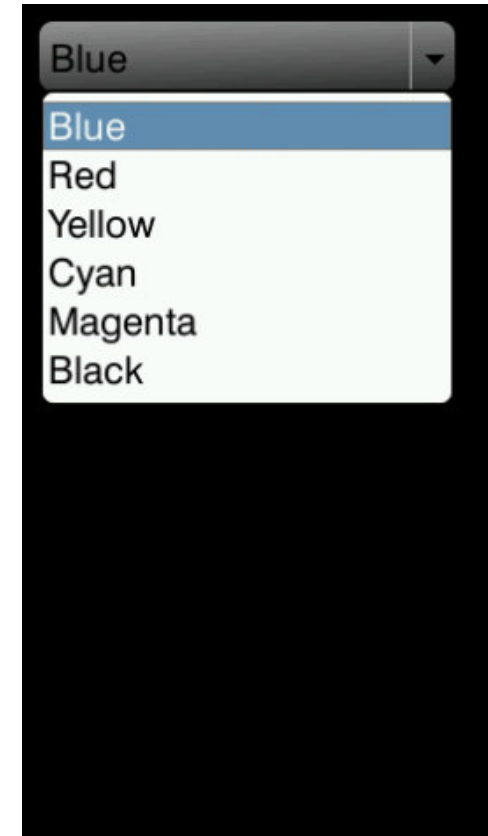
```
poke("gui.combobox.add"),"mycombo,10,10,240,40,Blue,Red,Yellow,Cyan,Magenta,Black"
```

DO

```
    pause 0.01  
    event$ = peek$("gui.event")  
    if (len(event$)) poke("stdout"),event$  
    if (instr(event$, "TextChanged")) break
```

LOOP

```
poke("gui.mycombo.visible"),0  
FONT "Swiss 721,20":poke("fcolor"),1  
Text 5,7,"Selected Content:"  
Text 5,30,mid$(event$,len("mycombo:TextChanged:")+1)
```



File Open Dialog

1. Parameter: **Caption Dialog**
2. Parameter: **Path**
3. Parameter: **Fileextension (Optional, if empty it uses *.*)**

```
poke("gui.fileopen.show"),"Select File;,labels,*.lbl"
```

```
DO
```

```
    pause 0.01  
    event$ = peek$("gui.event")  
    if (instr(event$, "DialogClosed")) break
```

```
LOOP
```

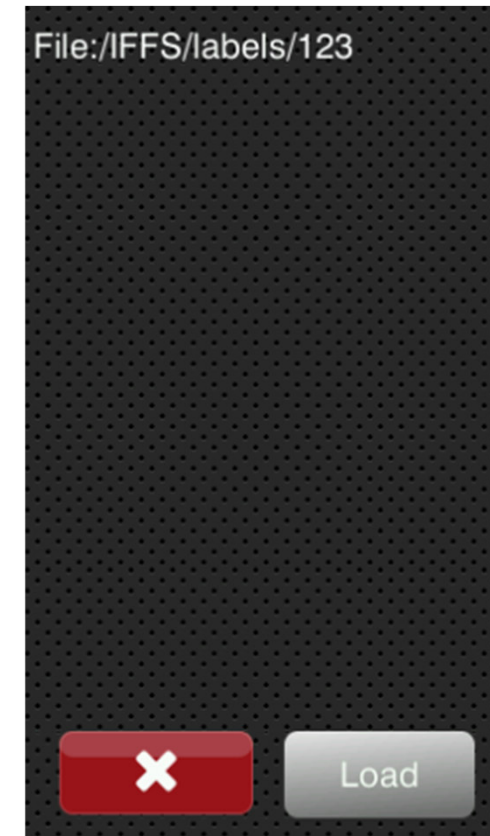
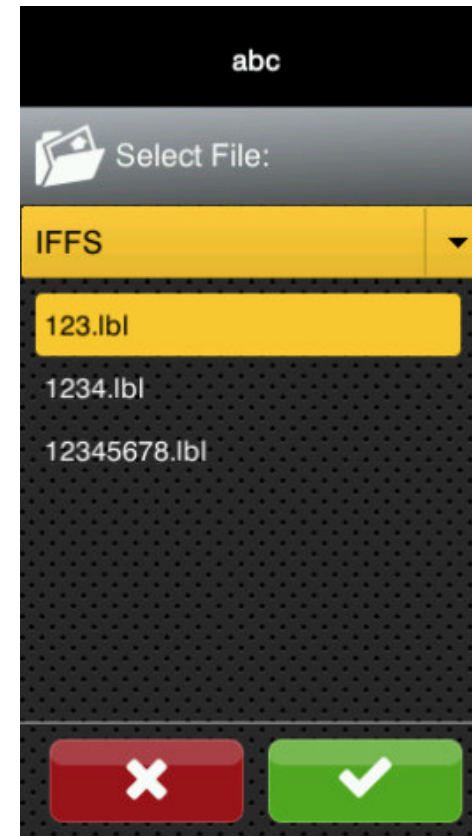
```
if (instr(event$, ":DialogClosed:ok:")) then
```

```
    // ***** Anzeige Dateiauswahl  
    lblname$=cutoff$(event$,1,4)  
    lblname$=cutoff$(lblname$,0,17)  
    TEXT 5,10,"File:"+lblname$
```

```
else
```

```
    // ***** Anzeige Dialog Abbruch  
    TEXT 5,10,"None"
```

```
endif
```



Info Dialog

1. Parameter: **Caption Dialog**
2. Parameter: **Text**

text\$="Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.
 At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
"

text\$=text\$+text\$

poke("gui. info.show"),"label source,"+text\$

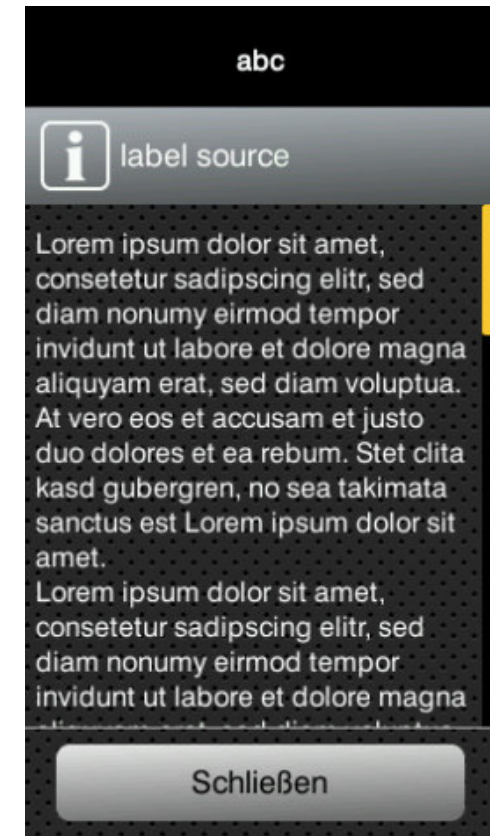
DO

```
    pause 0.01
    event$ = peek$("gui.event")
    if (instr(event$, "DialogClosed")) break
```

LOOP

FONT "Swiss 721,20"

```
if (instr(event$, ":DialogClosed:ok:")) then
    TEXT 5,40,"Dialog closed"
endif
```



Movie

1. Parameter: Object Name
2. Parameter: Position X on abc Window
3. Parameter: Position Y on abc Window
4. Parameter: Width of Movie
5. Parameter: Height of Movie
6. Parameter: FileName animated GIF File

```
poke("gui.movie.add"),"mymovie,10,10,250,160,closePhead.gif"  
poke("gui.mymovie.playing"),1
```

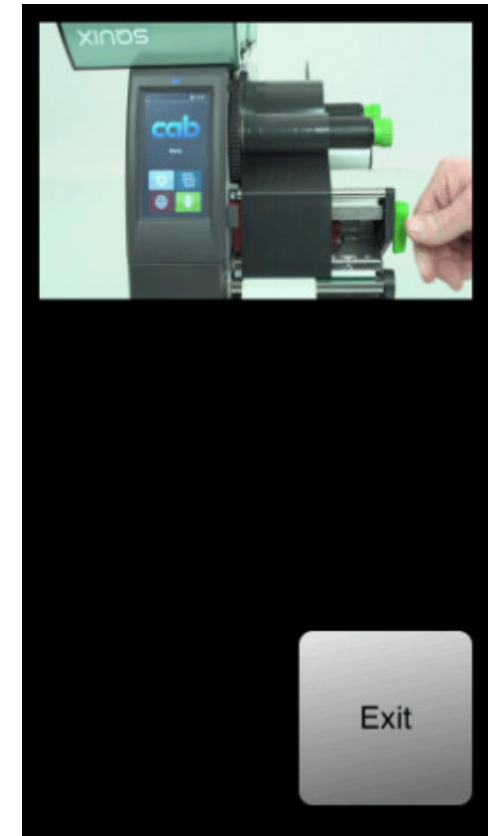
```
poke("gui.button.add"),"exit,160,360,100,100,Exit "
```



DO

```
    pause 0.01  
    event$ = peek$("gui.event")  
    if (len(event$)) poke("stdout"),event$  
    if (instr(event$, "Clicked")) break
```

LOOP

```
if (instr(event$, "exit")) end
```



 
closePhead.gif movie

rs232 zyklisches Abfragen

rs232 cyclic polling

```
<abc>... scaledata$=readrs232$() ...
sub readrs232$()
local tmp$,k
DO
k=peek(#1)
if k<>-1 and k>31 tmp$=tmp$+chr$(k)
if k=10 then
print #9 "Daten zuschneiden " +tmp$
if mid$(tmp$,2,2)="N:" then
// Daten vollständig
tmp$=cutoff$(tmp$,0,3)
tmp$=trim$(tmp$)
tmp$=cutoff$(tmp$,1,1)
print #9 "Daten verarbeiten / Trim data " +tmp$
return tmp$
else return
endif
endif
if k=-1 and len(tmp$)=0 return
LOOP
end sub
...</ABC>
```

Undefinierter Inhalt

Filter aller Sonderzeichen

Letztes Zeichen im Datenstrom

Voraussetzung / Condition

```
' ***** configuration RS232 Schnittstelle *****
open 1,"/dev/rs232:9600,XON/XOFF","r"
open 2,"/dev/rs232:9600,XON/XOFF","w"
```

SQLite Datenbank

SQLite database



```
print #9 „Database clean, wird bereinigt“  
print #1, "DELETE FROM Log WHERE STAMP<strftime('%Y-%m-%d',('now'),'-' +log_duration$+" days');“
```

```
print #9 “Datebase, aktuelle Messwerte werden eingefügt / Database, current measured values are inserted”  
print #1, "INSERT INTO Log VALUES ('"+article$+"', '"+current_weight$+"', DATE('now'))”  
line input #2 c_weight$
```

```
print #9 "Datenbank, Tagessumme vom Gewicht, des aktuellen Artikel wird ermittelt / Database, daily total weight of the current  
item is calculated"  
print #1, "SELECT SUM(WEIGHT) FROM Log WHERE STAMP=DATE('now') AND ITEM='"+article$+"”  
line input #2 sum_weight$
```

```
print #9 "Datenbank Summe Box / Database Sum Box "  
print #1, "SELECT COUNT(WEIGHT) FROM Log WHERE STAMP=DATE('now') AND ITEM='"+article$+"”  
line input #2 sum_box$
```

Voraussetzung / Condition

logging.db befindet sich im MISC
logging.db is located in MISC

```
' ***** SQLite open *****  
open 1,"sqlite:logging.db","w"  
open 2,"sqlite:logging.db","r "
```

```
' ***** SQLITE close *****  
close #1  
close #2
```



printout

<ABC>...

```
sub printout()
print "M I LBL;" +lblname$
print "R Replacefield;" +scaledata$

DO
    if right$(peek$("status"),1)="Y" break
LOOP
print "A 1"

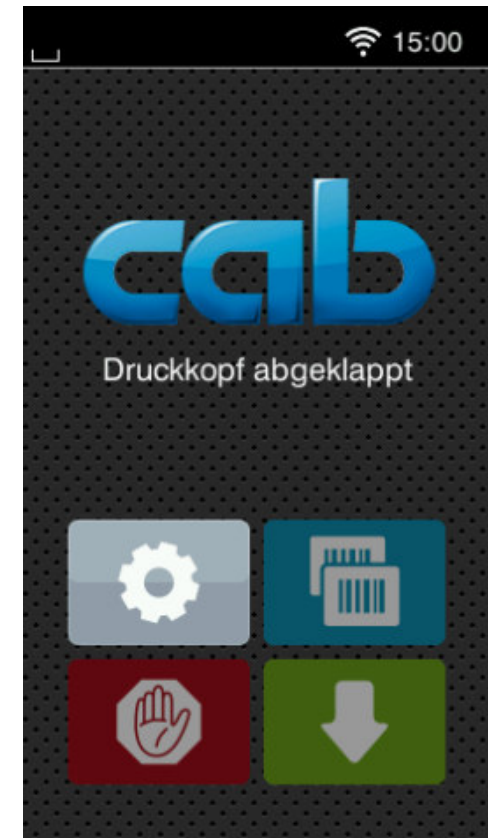
ctno=0

DO
    pause 0.01
    if right$(peek$("status"),1)="N" then ctno=ctno+1
        else ctno=0
    endif
    if ctno=5 break
LOOP
end sub
```

...</ABC>



```
<ABC>...
  sub Icdhandling()
    poke("wakeup"),1
    if mid$(peek$("status"),2,1)<>"-" then poke("Icd"),0
      else poke("Icd"),1
    endif
    return
  end sub
...</ABC>
```



Debug Schnittstelle mit Hilfe von Terra Term

Debug interface using Terra Term



```
<ABC>...  
' ***** Konfiguration Debug Schnittstelle *****  
if exists("/dev/rawip") open 9, "/dev/rawip", "w"  
  
print #9 "Start"  
  
print #9 lblname$  
  
print #9 "End"  
  
...</ABC>
```

The screenshot shows a terminal window titled "192.168.178.87:9100 - Terra Term VT". The window contains the following text:

```
Datei Bearbeiten Einstellungen Steuerung Fenster Hilfe  
Start  
/IFFS/labels/123  
End
```

Drucker mit Waagenanbindung Printer with scale connection



Projekte mit Waagen Hersteller / Projects with scales Manufacturer

- KERN
- RHEWA
- Mettler Toledo



Drucker mit Waagenanbindung Printer with scale connection

Die Daten Übertragung, kann über einen Start Button auf dem Drucker Display erfolgen oder über einen Button auf der Waage. Hier hat jeder Hersteller einen eigenen String, dieser über RS232 gesendet wird.

The data transfer can start via a button on the printer display or via a button on the scale. Each manufacturer has its own string, which is sent via RS232.



SUPPORT

cab Produkttechnik GmbH & Co KG
Wilhelm-Schickard-Str.14
76131 Karlsruhe

support.de@cab.de

+49 721 6626 333

