uVariotest

Functionional Description

© Gerhard Mutz

Version. 1 958 from 23.12 .2015

This English version was translated by google translate and has a lot of funny translation errors yet!

Overview

UVariotest is a development environment for creating programs in the context of psychological research. The aim in developing it was to be able to do as many of the necessary in experimental psychology tasks with just one program.
It can, for example create, complex psychological experiments, each with a screen for experimenter and subject and simultaneous acquisition and processing of bio-signals. The bio-signals may be used already during the acquisition pre-processed and for controlling the experiment itself. In addition to the storage of raw biodata in "VPD" format, for example results of the experiment can be saved as CSV or Excel xls files at the same time.
In addition, can be programmed with uVariotest, any evaluations of experiments and biosignals.
In addition to the built-in programming language TCL (test construction language) the language Realtime SPIL (Signal Processing and inferencing language) is supported. This is possible with the biosignal devices of the Varioport family a custom online preprocessing during measurement even with field measurements.
The syntax colorizing the editor also supports scripting language QDF (questionaire Definition Files) for writing scripts for iDialogPad.

(EMA data sampling App for IOS devices (G.Mutz 2008-2015)
Even a syntax check and a simulation (both the iPod and the iPad version) from QDF files is supported, including the "sorted export" output as text and Excel file. In the first line of the script file the following options can be specified after SAN:
ipod => an iPod instead of an iPad is simulated
auto => the timer controlled portion of the scripts is started instead of the user portion
center => the window is centered in the middle of the screen (with black background)
debug => the response code of each item is displayed immediately in a text window.

To support the Varioport biodata devices are extensive built interfaces for example also an editor for definition files.
There is a version for Mac OSX 10.5 or higher and a version for Windows from XP.

uVariotest is basically a text editor with many of the usual features.
Only the description of the specific menu options:
Window Menu => Run:
Compiles the script in the top window and starts immediately
Window Menu => Colorize:
Colors the text in the top window reclassified to conform to the syntax description
Window Menu => Startup:
Creates after query a standalone program from the top window.
Window Menu => Compile:
Compiles the script in the top window, but does not start the execution

Menu Varioport => Serial Port
Here you can select one of the available serial ports

Menu Varioport => Serial Baud Rate

Here you can select the desired baud rate of the serial interface

Menu Varioport => Check Device

Tests if a device of Varioport Series is connected and reports some parameters of the device

Menu Varioport => Edit defintion file

Here you can load and edit a Varioport definition file

Menu Varioport => Serial Terminal

Opens a serial terminal with which one can access serial devices

Syntax Description TCL

TCL is a simple programming language with allusions to Basic and C. The language is compiled directly into a bytecode, which is then executed immediately, so that programs can be tested directly.

It is only ever allowed one command per line.
Arguments are separated by spaces. Newline character is line feed (implicitly)
Comments start with // for one line:
// This is a comment
should be commented out several lines at the beginning / * and * at the end /
At the end of the program must always be END
Most commands in TCL must be written in uppercase. However, the variables and some flow commands may be written in lower case.
In order to standardize the language, there is the proposal to defined variables in lower case and to define macros in case sensitive.
Programs can be canceled by holding down the Shift Ctrl Alt keys. If you hold down the Shift and Function (Fn) keys simultaneously you can print out the screen.

Paths:

To include data, images, files, etc. required under one file paths.
In order for these paths are not too cumbersome and long, you can work out with complete paths with so-called. Relative paths. This requires first the global reference path can be defined as:
DEFINE PATH *"hd / uVariotest / Progs / TestProg" (OSX)*
DEFINE PATH *"c: / uVariotest / Progs / TestProg" (WIN32)*
Sets the path to the folder TestProg in which then the other files you want to be accommodated.
Basic here is the following path:
DEFINE PATH *"home"*
Sets the path to the folder where is the script itself.

Within the program, you can now access this folder, in which one accesses the relative path starting with a slash:
PlaySound *"/Sounds/sound.wav" 100 1*
Plays the sound from the Sounds folder

<u>Space, and most special characters are not allowed in paths.</u>

Labels:

Label (brands) are jump targets within the program and are indicated by a colon before the name:
:DoCalc
......
return

and eg ***gosub :****DoCalc*

Numeric variables:

Number variables (buffers) are always in the format Double. Speed

range floating point 16 points.
There are 1000 pre-defined buffers labeled B1 to B1000:
b1 = 5
b2 = 4.3
b3 = b1 + b2
Any number of additional buffers are defined as:
ACCU1 = 3
ACCU2 = 7.2
b3 = ACCU1 + ACCU2
The first assignment to a constant defines the buffer. After that it can be used as desired.

If a name can be assigned to one of the 1000 pre-defined buffer, can be defined that as follows:
Counter: 5 = 0
Is the Buffer B5 the name "Counter"

Arrays:

Arrays (vectors) are also always in the format Double. There are 100 predefined arrays A1 to A 100
Arrays must be defined prior to use with respect to their length:
***SET ALEN** a1 1000*
Sets the length of the array A1 to 1000 elements.
If an array is copied to another the target array must not be previously defined.
Is the size of the target array determines also needs the target array in many array functions not to be defined.

In addition, any number of arrays can be defined by the name of:
DFA: "array"
Creates an array with the name "array"
Should be assigned a pre-defined array a name happens with:
DFA: "array" 3
Assigns predefined array a3 the name "array" to.

Array cells can be addressed using their index (1 ..N):
a1 [2] = 5
Sets the second element of the array 1 to the value. 5

Arrays can be except by their index in the "shift mode" is set:
a1 = 5
a1 = 3
Here, the first element is described as 5 and the second element 3 etc. If the index is arrived at the last element again started at first.

Smaller Array s let n assign constants also as follows:
a1 = 1 7 3 6 8
Occupies the 5 elements of the array with these numbers

The "shift index" can read and write on the array cell 0th

The length of an array can be read out with the array cell -1:
b1 = a1 [-1]
Reads the length of the arrays a1 in B1

But arrays can also be copied as a whole, for example:
a3 = a1
copies the contents of the array A1 in array. 3
It should be noted that with different sized arrays, a conversion takes place. (either values multiplication or averaging)

Arrays can also be added together, is subtracted, multiplied or divided:
a1 = a3 a5 +
Adds the arrays A3 and A5 and stores the contents in A1

It can also adds each value of an array with a value subtracted be multiplied or divided:
a1 + = 5

Adds the value 5 to all the cells of the array 1

String variables:

String variables can accommodate any texts (Unicode, including, for example, Chinese characters).
There are 100 predefined string variables: s1 to s100:
s1 = "Hello world"
In addition, any number of string variables can be defined:
text = "This is a text"
generates the string variable text and assigns the text "This is a text" to.
Should a predefined string variables a name can be assigned to the done so:
text: 5 = ""
is the string variable s5 the name "text" and then stores an empty string.

Can string variables are "added":
s1 = s2 + "/" + s3
then appends the string s2 the character /, then the string s3 succession.

Characters in string variables can be written and read by their index:
s1 [5] = 32
Sets the 5th character in the string s1 to the character code 32 (spaces)

The index 0 returns the length of the string.

In direct String entries with quotes may occur no newline character. Instead, the sequence "\ r" must be used , For example:
s1 = "Hello \ r"
Likewise, no quotation marks may appear in the string. Here you must use "\ '".

String arrays:

There is a single string or combined string array with 512 entries Numbers
The cells are described with **DEFINE TEXT** NUM "text" val with a text and a number. (NUM 1-512)
This array is used for example when reading folder entries.
Read the cells with the **DTEXT** [NUM] command. Immediately after the
Reading of the text is also to stored number with **WEIGHT** available.

Address of variables:

In some commands, the number of a buffer, StringBuffers or array must be specified. So for example, instead of b30 must be specified here 30
Here, for simplicity also be specified & b30 which is considerably easier with aliases:
counter: 7 = 0
& counter then supplies 7

Operators:

Addition: +
Subtraction: -
Multiplication: *
Division: /
Remainder:%
Logical and: &
Logical or: |

Logical exclusive or ^
Logical Shift Left: <<
Arithmetic shift right: >>
Logical shift right:> (unsigned 32-bit)

+, The operators -, *, / can also be used in a simplified form:
b1 + = 1 instead of b1 = b1 + 1
b3 / = 5 instead of b3 = b3 / 5
adds the number 1 on b1, b3 and divided by the number 5

There is no mathematical hierarchy. Expressions are evaluated from left to right, but can be clamped arbitrarily.

The operator + can also be applied to the string.

Control Structures:

if, then, else, endif leads program components under certain conditions:

```
if b1 == 5
then
PRINT "b5 is equal to 5"
else
PRINT "b5 is not equal to 5"
endif
```

If b1 5 is the part behind then run otherwise the part behind else

Several conditions can be linked with "or" and "and":

```
if b1 == 5
and b2 == 3
then
PRINT "b1 and b2 = 5 = 3"
endif
```

If b2 is 5 and b2 3, the PRINT statement is executed but **if** can also stand alone without **then,** then allowed to follow only one line:

if *b1 == 5*

    PRINT *"b1 = 5"*

Grind:

for variable FROM TO STEP SIZE **next** executes a block of N Anwesungen of times:

for *b1 1 10 1*
PRINT *B1 LF*
next

counts the variable b1 starting with 1 up to 10 in increments of Figure 1.
With **break** the loop can be terminated prematurely at any time.

Jump instructions:

goto: label jumps to the specified label (as **loop)**
One must never made **for next** loops or **switch ends** blocks pop out with **goto** or **return.**

GOSUB: label the subroutine executes label. The subroutine must be terminated with **return.**

ongoto var 1: label1 2: label2 3-5: label3 branches depending on the value of var to the appropriate brands

ongosub var 1: label1 2: label2 3-5: label3 results depending on the value of var
the relevant subroutines from

Distributor:

switch var **ends** into Abhängikeit the value in the corresponding rows in the **case** var block:

```
Switch b1
Case 1
PRINT "1"
Breakfast
Case 2
PRINT "2"
Breakfast
default
PRINT "neither of them"
ends
```

Graphical coordinate system

The graphic coordinate system has its zero point on the top left. Coordinates are given in pixels.

Macros:

To voice simplification multiple commands can be combined to macros:

```
MACRO "print string" S1 S2 S3
PRINT TAB s1 s2 s3 TAB TAB LF
ENDM
```

This Macro is like "" It's "is called, for example, with print string" Hello ""
There is already an extensive collection of macros for different applications.

Such Macrodefintionen can eg be imported from external files and are then included with **INCLUDE:**

```
        INCLUDE "/glob_macros.tcl"
```

Adds the file glob_macros.tcl from the same target directory.

Macro Directory

"Glob_macros.tcl" => Global Macros

for some of the following macros must first few variables are set
CXP: the X position of the element
CYP: the Y position of the element
cxs: the width of the element
Cys: the height of the element
CCS: the text size of the item text
CDX: the distance to the next element in the X direction
CDY: the distance to the next element in the Y direction

Button *"text" code key*
Creates a button with the text *"text"* the result code *code* and the key code *key.* The Y position is automatically advanced to the next element.

xButton *"text" code key*
Creates a button with the text *"text"* the result code *code* and the key code *key.* The X position is automatically advanced to the next element.

Standard pop *"text" dbufn offset Num code*
Creates a PopUpButton with the text *"text"*, the number of buffers for transferring values *dbufn,* the offset of display values into String array *offset,* the number of entries *Qty,* and the result code *code.* The Y position is automatically advanced to the next element.

Easy Popup *"text" dbufn code*
Creates a PopUpButton with the text *"text"*, the number of buffers for

transferring values *dbufn,* and the result code *code.* The Y position is automatically advanced to the next element. The elements are passed here in the text by separation with a colon:

Easy Popup *"What is your gender? Male: Female" 1 5*

CheckBox *"text" dbufn code*

Creates a checkbox button with the text *"text",* the number of buffers for transferring values *dbufn,* and the result code *code.* The Y position is automatically advanced to the next element.

Radio *"text" dbufn code*

Creates a radio button with the text *"text",* the number of buffers for transferring values *dbufn,* and the result code *code.* The Y position is automatically advanced to the next element. If several radio buttons to each other, so can always only one of them can be selected. You must use the same buffer (dbufn must be equal)

Slider *dbufn maxval code*

Creates a slider with the number of the buffer for transfer of values *dbufn,* the maximum value at full scale *maxval,* and the result code *code.* The Y position is automatically advanced to the next element.

Segment *"text"* *dbufn code*

Creates a segment selection with the elements in "text" by Dopp e lpunkt are passed separately, the *code* number of the buffer for transfer of values *dbufn,* and the result code. The Y position is automatically advanced to the next item, for example,

Segment *"1: 2: 3: 4: 5" 1 5*

Select Serial Port *"text" dbufn code*

Creates a PopUpButton with the text *"text",* the number of buffers for transferring values *dbufn,* and the result code *code.* It sets all available serial ports.

The Y position is automatically advanced to the next element.

SelectVPDOnl *"text" dbufn code*

Creates a PopUpButton with the text *"text",* the number of buffers for transferring values *dbufn,* and the result code *code.* It sets all available channels from a previously loaded Varioport definition file. The Y position is automatically advanced to the next element.

SelectVPDChn *"text" dbufn code*

Creates a PopUpButton with the text *"text",* the number of buffers for transferring values *dbufn,* and the result code *code.* It sets all available channels from a previously loaded Varioport data file. The Y position is automatically advanced to the next element.

RangeOfNumbers *"text" dbufn min max code*

Creates a PopUpButton with the text *"text",* the number of buffers for transferring values *dbufn,* the minimal number of *minutes* displayed, the maximum number displayed *max,* and the result code *code.* It sets all the numbers including *min* and *max.*

The Y position is automatically advanced to the next element.

TInput *"text" "dbufn code*

Creates a Te x tfenster with the text *"text",* the number of StringBuffers for passing values *dbufn,* and the result code *code.* It can be typed free text. The Y position is automatically advanced to the next element.

Text *"text"*

Prints the text *"text".* The Y position is not continued.

Value *"text" val_chan*
Displays the text *"text"* and the value of the channel or the buffer *val_chan* on.
The X position is automatically advanced to the next element.

DText *"text"*
Defines an entry in the string array with the text *"text"*
Prior to the desired position, and the value of the associated numerical value must be set:
CTP = array index (1 .. 512)
CTI = associated numerical value
CTP and CTI are automatically increased by. 1

TSlider *xp yp xs ys dbufn code max inc*
Creates a special slide with Endpfeilen at the position *xp, yp* with the size *xs, ys dbufn* with the number of the buffer for transfer of values, the result code *code,* the maximum value at full scale *max,* as well as the value of *inc,* the click on the Endpfeile to be subtracted or added.

Opendir *"path", "match"*
Reads all entries of the folder "path" in the Stringarray from index 300 on. With *"match",* a selection of the entries are read that contain the text *"match".*
The variable *dir_entries* then contains the number of scanned file names.
It can be read as directories to 200 entries

NextDirEntry
Reads the next entry of the previously read with opendir folder.
The name of the next file is now in the string variable next_file ,

OpenLDir "Path", "match"

Opens the directory of the folder "path". With " match "You can choose a selection of messages that contain the text" match "Included.
The Variable dir_entries then contains the number of scanned file names.
The variable contains the first entry next_file

LNextDirEntry
Reads the next entry of the previously opened with OpenLDir folder. The name of the next file is now in the string variable next_file , OpenLDir and LNextDirEntry only use the first entry of the string array DEFINE TEXT The directories can contain any number of entries.

Force2Screens
Tests the number of available screens and displays an error message when only 1 screen is found.

ClrScreen2
2. Deletes the screen to black

Present Picture *"path"*
Displays the image of image path *"path"* at the 2nd screen centered in original size.

Play Sound "path"
Plays the sound of audio path *"path".*

StopSound
Stops the sound that was started with **PlaySound.**

Copy Screen
Copies the screen 2 in the first screen to the coordinates *s2cs_xp, s2cs_yp* in size *s2cs_xs, s2cs_ys*

Instruction2 "text"
Displays the text "text" in an instruction window on the 2. screen.

Tlog "text"
Displays the text "text" in a log text window. Previously, the position of the window must be defined: position => *tlxp, tlyp,* size *tlxs, tlys,* and character size *TLCS.* The timer is used to enforce a minimum *tlogt* display time.

ResetTlog
Sets the LogTimer *tlogt* back.

StartITimer *time*
Starts the timer *exp_itimer* with the timing *time* in milliseconds.

StopITimer
Stops the timer *exp_itimer.*

Vpd_Chn *"ChName" Khnum chncol nxticks nyticks*
Creates a channel representation named *"ChName"* the channel number in the channel color *chncol Chnum* with the number of X ticks *nxticks* and the number Y ticks *nyticks.* Previously, the position must gxp, gyp and size GXS, gys be set.
gdy denotes the distance to the next channel. dtime is the display time in seconds, DStart the start time shown in milliseconds, XFLAGS displaying flags are (see VPDGRAPH)
A VPD file must have been previously loaded. glnum is a serial number starting with 1 (automatically incremented)

ChnArrayGraph "ChName" Khnum chncol chnum2 chncol2 nxticks nyticks
Creates a channel representation with markers with the name " ChName "The number of the channel arrays Khnum in the channel color chncol the number of the channel arrays chnum2 in the channel color chncol2 with the number of X Ticks nxticks and the

number Y Ticks nyticks , Previously, the position must gxp, gyp and size GXS, gys be set. gdy denotes the distance to the next channel. dtime the display time is based on the array length in seconds, DStart the start time shown in milliseconds, XFLAGS displaying flags are (see VPDGRAPH)
The two arrays must be of equal length and be initialized. glnum is a serial number starting with 1 (automatically incremented)

Vpd _ CRDW *num*
Draws the graph *num* new.

DGraph *min max abn*
Draw a line graph with amplitude of *min to max* of the complete array Buffers *abn* with the number. The position variables are gxp, gyp, GXS, GYS.

OnlChan *ymin ymax XFLAGS nxticks nyticks xtime Khnum*
Creates an online channel display with the name with the scaling of *ymin* to *ymax* the display options *XFLAGS* with the number of X ticks *nxticks* and the number Y ticks *nyticks,* the speed in Mi k Rose Customer *xtime* and the channel number *Khnum.*
Previously, the position must gxp, gyp and size GXS, gys be set.
gdy denotes the distance to the next channel. glnum is a serial number starting with 1 (automatically incremented)
A VPD definition file must have been previously loaded.

Init_Experiment *"name"*
Initializes the necessary for an experiment variables, loads the associated configuration file and the Excel test control files, as well as the definition file.

Exp_Cond_PopUp *"text" code*
Creates a pop-up with the text *"text"* the IDCODE automatically *code* the to **Init_Experiment** its options displays.

Next Line
Loads the next parameter from the Excel control file. trial_par1 to trial_par6, and string variables trial_par7 and trial_par8

ExpHeader *"text"*
Clears the screen and produces a heading with the name *"text"*

"Tmacros.tcl" => Macros for questionnaires

The position variables for the questionnaire items can be preset with 2 macros:

Uleft
Begins on the top left *with the query and uses exactly half the screen.*
Uright
Starts right *up* with *the query and uses exactly half the screen.*
UCenter
Begins centered above *the query and uses exactly half the screen.*

Buffnum The variables and *sbuffnum* must be set to the desired starting positions of the result buffer.

MITEM *"Question" "scale"*
Creates an item with several possible answers. "Question" is the text of the question, "Scale" is the scale where the entries are separated by a colon:
MITEM *"Wi* e Feel "?" badly: medium: good "

MRadios *"Question: Anwort1: answer2 ...: AntwortN"*
Creates an item with several possible answers. "Question" is the text of the question and the possible answers which the entries are separated by a colon:

MRadios *"Gender: Male, Female"*
The answers are here all visible to one another.

PopUp *"Question: Anwort1: answer2 ...: AntwortN"*
Creates a popup item with several possible answers. "Question" is the text of the question and the possible answers which the entries are separated by a colon:
PopUp *"Gender: Male, Female"*
The answers fold here at a mouse click.

Numbers *"Question" of up*
Creates a popup Item numbers as answers. "Question" is the text of the question. The numbers go *from* from to *to.*
Numbers *"How old are you?" 18 99*

Free Text *"question"*
Creates a text box for free text *input.*

Continue
Produced in the lower right corner of a "Next" button and waits for *input.*

Instruction "text"
Displays the text "text" in an instruction window on the 1. screen.

iScale "bad: medium: good" H Öhe
Shows a scale in the right half of the screen to the top.
The answers are separated by a colon. *Height is the height* of the scale in pixels.

Item "question"
Displays an item in the left half of the screen match the scale to the right of it and the right to scale response Buttons

SetZeros *up*

Sets the buffer of the numbers of *of* to *to zero.*

CheckNonZeros *up*
Tests the buffer of the numbers of *of* to *to zero.* If one of the Buffer zero, the variable is b1 = 0, otherwise 1

Plot_macros.tcl => Macros to output graphics and tables

For more details on these macros see gplot.tcl sample file.

FFTPlot " header "xp yp xs ys bgcol ymax yticks srate
Draws a FFT of the array 32 with the headline " header "To the position xp, yp with the size xs, ys and the background color bgcol , Ymax is the maximum y value, yticks the number of divisions of the y axis. Is ymax = 0, the y axis is scaled automatically. Srate is the sampling rate of the data in the array.

Histogram *"header" xp yp xs ys bgcol ymax yticks flow index*
Draws a histogram with the heading *"header"* of the position *xp, yp* with the size *xs, ys* and the background color r BE *bgcol.* Ymax is the maximum y value *yticks* the number of divisions of the y axis. Is *flow index* = 0 The X axis is labeled with numbers, otherwise with texts from the string array with the index *flow index*
The data is passed in the array. 1 Is *ymax* = 0 the Y scaling is set automatically. If ymax is less than zero from the raw data in the histogram array 2 is calculated (in Array 1) and automatically scales in the x and y axis. Ymax is the desired number of levels of the histogram.

PieChart *"header" xp yp xs ys bgcol flow index*
Draws a pie chart with the heading *"header"* of the position *xp, yp*

with the size *xs, ys* and the background color r BE *bgcol.* Is *flow index*> 0 a legend with texts from the string array with the index is drawn *flow index*

The data is passed in the array. 1

Line *Chart "header"* *xp yp xs ys bgcol ymin ymax yticks flow index*

Draw a line graph with error bars with the heading *"header"* of the position *xp, yp* with the size *xs, ys* and the background color r BE *bgcol,* the Y axis range from *ymin* to *ymax* and the number Y ticks *yticks.* Is *flow index* = 0 The X axis is labeled with numbers, otherwise *flow index* with texts from the string array with the *index.* If *ymax* = 0 the Y scaling is set automatically.

The data is passed in the array. 1 The values for the error bars are a block behind those for the mean values.

Scattergram *"header" xp yp xs ys bgcol ymin ymax xmax xmin yticks xticks*

Draws a cloud chart with the heading *"header"* of the position *xp, yp* with the size *xs, ys* and the background color r BE *bgcol,* the Y axis ticks *yticks* Y range from *ymin* to *ymax,* the Y axis range *xmin to* xmax, and the number and the number X ticks *yticks.* Is *ymax* = 0 the Y scaling is set automatically. The data is passed in the array 1 (x) and Array 2 (y).

Eda_Graph *"header" xp yp xs ys bgcol p1_eda p2_eda p3_eda sr_eda*

Draw a line graph EDA with the heading *"header"* of the position *xp, yp* with the size *xs, ys* and the background color r BE *bgcol,* with 3 marks in milliseconds distance from the start for 3 standard measurement points an EDA SCR reaction. sr_eda is the sampling rate of the EDA. The data is passed in the array. 1

Table *"header" xp yp xs ys gbcol xzellen yzellen flow index lhflag* Draws a table headed *"header"* of the position *xp, yp* with the size *xs, ys* and the background color r BE *bgcol,* with the number of cells in the X direction and *yzellen xzellen* the number of cells in Y direction. Is *flow index*> 0 are labeled *flow index* column headings with texts from the string array with the index. Is *lhflag = 0,* all cells in the X direction are drawn the same size, otherwise column 1 half of xs and therefore the other spread the other half. The data is passed in Array 1, the values are continuously distributed in all cells, from top left to bottom right.

cnv3to2 xp yp zp convert 3D coordinates into 2D coordinates in. The resulting 2D x and y positions in Sthen X2D and Y2D.

Move3D xp yp zp sets the starting point of a 3D line on x, y, z.

Line3d xp yp zp col FLG draws a 3D line from last point to x, y, z in color col. Flg see at LINE.

Frame3D xp yp zp xs ys zs framecol fillcolor draws a 3D frame and the position x, y, z fillcolor with the size xs, xs, zs the frame color framecol and the blend color.

FFT_Waterfall "header" xp yp xs ys bgcol pymax fft_srate fwscale Vtime
Draws a 3D FFT diagram available (Waterfall) at the position x, y with the size xs and ys, the Hintergrundfrabe bgcol, pymax not used, the sampling rate of the input data (in array a2) of the scaling and the start time in milliseconds (since beginning of the file) (The start time of a loaded VPD file is used)

Bioanalyze.tcl => Macros for parameterization of biodata

NotchInt srate
filters the array a50 with a 50 Hz notch filter (srate is the sampling rate of the data). Thereafter, a high-pass filter is calculated and converted the array in absolute values. Finally, the data are smoothed with a low pass. The arrays 51 and 52 are used for intermediate values.

Find_SCR srate
place in the array a50 DFA SCRs (srate is the sampling rate of the data).
The arrays A51 and A52 are used for intermediate results.
A50 array is not changed. The results are displayed in array 53rd
Always 4 values per SCR found:
1. Start time in milliseconds
2. Length of the SCR in milliseconds
3. Amount of SCR in microsiemens
4. time from maximum to drop to 50% of the maximum

TCL keywords in alphabetical order:

ASORT NUM sort the array with the number NUM ascending

ACLR NUM deletes the array with the number NUM

AABS NUM converts the array with the number NUM in absolute values.

AHP ANUM FAC, IIR (infinite impulse response) Array highpass with ArrayBuffer ANUM and the factor fac 1-20

ALP ANUM NUM, Array moving average low-pass with ArrayBuffer ANUM and the number values NUM

AMC ANUM1 ANUM2 DANUM LEN Array Moving Correlation calculates the sliding correlation with the length LEN of the two arrays and ANUM1 ANUM2. The result is stored the DANUM.
when LEN is equal to the length of the input array, the total correlation of Array1 and Array2 is calculated. The results are then returned to the first cell of Array3.
1 = correlation (* 100), 2 = MW of 1, 3 = MW of 2, 4 = STD of 1, 5 = STD of 2, 6 = TND of 1, 7 = TND 2

AMT SANUM DANUM LEN Array Moving Trend calculates the sliding trend with the length LEN of the input array SANUM and stores the result in DANUM.

AHIST SANUM DANUM MINBNUM MAXBNUM calculates the histogram of the array SANUM and stores the result into the array DANUM. In the buffers with the number MINBNUM and MAXBNUM are the minimum and maximum values of the target histogram. If the value in MAXBNUM = 0 min and maximum values of the array are automatically calculated and returned in the two buffers.

AHR SANUM DANUM FREQ (FLAG) Array ECG to HR calculated from Array1 (ECG) heart rate in pulses / min and sets the pulses in the target array from DANUM (data rate 4 Hz). The sampling rate of the ECG FREQ must be specified. Is SANUM negative it denotes a channel number of a previously loaded VPD file. This allows very long ECG leads are calculated without large memory consumption. If FLAG = 1 intervals are stored (in milliseconds) of the target array Interbeat instead pulses / min. If FLAG = 2,3,4 are for debugging purposes between values of detection A lgorithmus displayed.
The scaling of the ECG input is by default in microvolts. If the ECG scaled in millivolts must FLAG bits are set 8 FLAG + 0x100
With FLAG Bit 9 of the command generated in ArrayBuffer 100 additional markers at each R-wave. This can be marked with the special display mode of VPDGRAPH the R-wave. At the top of the Word FLAG parameter then the desired Markerkode must stand. (eg

FLAG + 0x00040200 sets a line marker in each R-wave)
If FLAG = 0x400, the heart rate of a pulse curve (Pleti) are calculated. The sampling rate should be 128 Hz here.

AimPort PATH ANUM ELNUM imported numerical values from a table with ASCI PATH in the ArrayBuffer ANUM. The element ELNUM from the line is read , The array length is automatically set to hand the rows found.
Is ANUM negative binary data in WORD (16 bit) format are read into the array.
The array length is defined with reference to the file length.

ACOPY SRCN SRCI DSTN DSTI NUM: NUM items copied the array of SRCN INDEX SRCI in the target array DSTN and the index DSTI Index goes from 0 to here ArraySize-1

ABSTATS NUM BNUM: calculates the statistics number NUM (1..32) of the complete array buffer with the number BNUM. The statistics must be reset with RESET STATS once. Results see ADDSTATS.

ADDSTATS NUM VAL: the individual adds value VAL statistics number NUM (1..32) added. The statistics must be reset with RESET STATS once.
The results can be accessed using the functions AVG, STD, NUM, MIN, MAX, TND, INT. The values must be retrieved with negative indices. These are the same statistics buffers as ABSTATS.

ADump NUM XP YP XS YS FC BC TXTSIZE: draws a hex dump of the array buffer number NUM (for debugging purposes, only bytes per cell)

and expression1 = expression2: linking and after if statement

ANOVA PATH SAN1 SAN2 SAN3 SVN HT DAN calculated an ANOVA with the factors of the array SAN1 SAN2 and SAN3

(maximum 3 factors) and the data array SVN. Is HT> 1 being the DTEXT names of up to 3 factors (otherwise they are called A, B, C) When HT> 1 the names of the rows of the matrix are the 3 variables DTEXT following items will be retained as an output
Dan zero, the results are written in the print window, otherwise the finish line array DAN. The target array always contains 6 columns, the number of rows depends on the factors.
Here PATH must be an empty string. If PATH is a file path, the data matrix of these (cvs text) file is read and written, the results from this print window ; all other parameters are ignored.

APPEND "path": Opens an output file and appends all the output data into an existing file. See also **OUTPUT**

bx = ASK "text" creates a query box with the message text and the buttons "Cancel" and "OK". If Cancel is pressed, returns 0 if OK is pressed. 1

AUTO NUM BNUM INC MIN MAX TIME FLAGS (GRAVITY) SID1 SID2: install an automatic change of the buffer BNUM. He is all TIME milliseconds increased by the value INC and if the value MAX is reached backwards to MIN lowered etc. Are SID1 and / or SID2 equal to 0 so they call sound Ids of the sounds to be played when reaching MIN or MAX ,
FLAGS:

1 = ignore the MIN / MAX values and count continues
2 = set when reaching the MAX value back to MIN and counting further upwards
4 = hear when reaching the MAX value at counting
8 = hear when reaching the MIN value to be counted
16 = increment the value INC to GRAVITY a non-linear increase to reach (gravitational effect, for example, balls)
32 = humiliate the INC value to GRAVITY at a non-linear decrease to reach (drag effect)
64 = each time reaching a min or Maxwertes the value INC is

incremented by a random value from 0-GRAVITY.

128 = TIME and GRAVITY are key codes. The variable is changed when the corresponding buttons are pressed. The Time button is added to the buffer INC which GRAVITY key subtracts from Buffer INC.

256 = INC is a mathematical operator (1 = +, 2 = -, = * 3, 4 = /) The Buffer No. MIN is calculated with this operator with the number MAX Buffer and stored in buffer number BNUM.

Is SID1> 0, this is another operator (as INC) and SID2 then another buffer number for an additional operation.

With the numeric variables AUTOx can be queried whether the buffer value has been updated.

BEEP N: Creates N Peeptöne (System Warning)

b1 ... 1000: = expression sets the Buffer 1 ... 1000 to the value of expression

BUFF [bx]: = expression sets the buffer with the number in bx to the value expression

BITS N : Controls a USB to Parallel Converter FTDIChip. (FT245RL when finished board available) This results in a quasi-parallel interface with 8 IO lines.

BITS 256 opens the port
BITS 512 closes the port
BITS 1024 + 0-255 sets the IO Data direction = 1 0 = Input Output
BITS 2048 reads the level of the data lines. Result in Buffer 2
BITS 0 - BITS 255 sets the 8 outputs of the port to 0 or 1

BITS 0x100 opens the port
BITS 0x40f sets the line 0-3 to 4-7 and output to input
BITS 1 sets as the first output of the ports on 1
BITS 2 sets the second output at 1
BITS 3 sets the 1st and 2nd output at 1

BITS 0x800 reads the level in B2
BITS 0x200 closes the port

BLOCK = 0 VAL allows all background processes (default state), 1 switches off all background processes.

CASCADE NUM SPORT BAUD NCHANS FAST DCHN ... : Defines a data source with the number NUM (1-16), the associated serial port and baud rate BAUD SPORT and the number of channels that data source NCHANS the source channel FAST, and the target channel of the previously loaded definition file. It always must be specified for each channel, a source and a destination channel. The source channel is always the relative number of the channel of the source device and the destination channel is usually the name of the destination channel of the definition file. All defined sources should preferably use the same baud rate.

CHANPS CNUM SELECTOR VAL: sets channel parameters in the definition file, which must have been previously loaded. CNUM the channel.
Is CNUM positive, the channel number refers to the number of pre-defined with START ONL -1 channel order.
Is CNUM negative, the channel number refers to the number of the definition file.

SELECTOR:
0 = Set SCANFRQ
1 = Set FLTMSK
2 = Set TMUX
3 = Set RESVD
4 = Set SIZE (only the channel ON / OFF Bit 0 = Off 1 = On)

CPARS CNUM SELECTOR: provides channel parameters from the

definition file must be loaded in advance. CNUM the channel.
Is CNUM positive, the channel number refers to the number of pre-defined with START ONL -1 channel order.
Is CNUM negative, the channel number refers to the number of the definition file.

SELECTOR:
0 = returns SCANFRQ
1 = provides FLTMSK
2 = provides TMUX
3 = delivers RESVD
4 = delivers SIZE (only the channel ON / OFF Bit 0 = Off 1 = On)

CLR SCREEN COLOR: clears the screen with color

CLOSE SCREEN (NUM) : Closes the screen with the number NUM ,Will not really needed because screen when **END** is automatically closed

CLOSE OUTPUT: closes the output file.

CLOSE SPORT: closes the serial port. 1

CLOSE sport2: closes the serial port. 2

CHSET XP YP MAXCHANS TYPE opens a window to configure the channel sets at position XP, YP with the maximum number MAXCHANS per display window. If TYPE = 0, the channel sets a previously loaded VPD file be edited Otherwise, the channel sets a previously loaded definition file.

RCHSET ABNUM TYPE reads all channel sets in an array with the number ABNUM.
TYPE see CHSET ,

WCHSET ABNUM TYPE stores the array ABNUM in Kanalset.
TYPE see CHSET ,

CHAIN "path" FLAG terminates the program and start the program path. If FLAG = 1 remain BUFFER AND SBUFFER obtained.

CONTROL "text" XP YP XS YS TSIZ FC BC BNUM VALUE TYPE KEYCODE (PUSTART PUANZ (PUINC)):
Create a Control with the text "text" at the position XP, YP the size XS YS and the text size and the color TSIZ FC (*) and BC (*).
BNUM is the buffer number in the preset or the value of the returned Controls (if Type = 1 and 2), value is the output code (if Type = 0 or 3).
TYPE:
0 = Button (text here can be a picture path "_Pfad"
1 = Checkbox (returns 0 or 1 in the corresponding buffer)
2 = Radio Button (1..N provides in relevant Buffer)
3 = Text field BNUM the SBUFF serving as input and output, in value represents the maximum length of text "text" is an optional heading or is about to enter text if option 4 =

Option in normal text input:
4 = optional header is just around the text input, above else
8 = input field is active immediately, not just when clicking
16 = Return key is not executed in the box, so can be used to terminate the Controls the Return key, for example, if, for example, an end button with the Return key is seized as a key code.

KEYCODE is a key code with the control with the keyboard can be accessed.

A button is to flag 128 to a PopUpButton. Here still a start value (PUSTART) and a number of elements (PUANZ) for PopBox must be specified.
The starting value refers to the start number of DEFINE TEXT. The

Control provides as a result the weight of the selected item.
Is PUANZ -1 all channels of a previously loaded Defdatei be displayed at -2 all serial interfaces, at -3 all channels loaded with a LOAD VPDFILE Varioport file at -4 TMS all files and at -5 with 3 additional parameters a series of numbers (Number, Start, Incr). The selected option is returned as number BNUM and as a string in SBUFF No. PUSTART.

For TYPE = 512 a scroll with 2 arrow keys is generated. In Buffer No. BNUM the start value is passed and returned. In keycode of the maximum possible value is passed (number of buffers)

For TYPE = 1024 A sensor surface is generated. This provides a mouse in X position to the face value proportional in BNUM, KEYCODE = BUFFER WITH MAX VALUE

The controls are visible when with bx = **CONTROLS** the result is called. You must always use **KILL CONTROLS** will be deleted if they are no longer needed

The numeric variable **CREF** delivers to each CONTROL definition a reference number with can be accessed later in the Control with COPT command.

COPT REF SEL (TEXT) (XP) (YP) changed the existing control with the CREF REF as follows:
SEL = 0 sets the title of a control with the text TEXT
SEL = 1 enabled the Control
SEL = 2 disabled the control (button is dimmed and switched off)
SEL = 3 deletes the last defined CONTROL again (ignoring REF)
SEL = 5 sets the position of a slider (position as a string) NUMSTR ("% f" heading)
SEL = 6 sets the button text color (color as a string) NUMSTR ("% 3.0f" color)
SEL = 7 hides the Control

SEL = 8 makes the control visible again
SEL = 9 shifts the control to position XP, YP
SEL = 10 sets the increment of TSliders (value as a string) NUMSTR ("% f" inc)
SEL = 11 sets the current value of a TSliders (value as a string) NUMSTR ("% f" inc)
SEL = 12 ensures that a text field receives focus (key input)

CFIR TYPE SFREQ CFREQ BWID TAPS ABN: calculated FIR (Finite Impulse Response) filter coefficients of the characteristic TYPE ("H" = High-pass, "L" = Low Pass, "B" = bandpass, "N" = Notch) the sample frequency SFREQ, the cutoff frequency or corner frequency CFREQ, the bandwidth BWID, the number taps (<= 1024) in the array ABN. This array is required for the calculation of FIR filters, see above.

CIRCLE XP YP DM FC BC FLAG (SA EA): draws a circle on the position XP, YP with the diameter DM and the colors FC and BC. Is Flag 1, the circle is not filled., When FLAG 2 and starting angle and end angle SA SE is a circular section marked (ARC)

DRAW CONTROLS records all new CONTROLS.

DRAW GRAPH XP YP XS YS FGCOL BGCOL TICKCOL XTICKS YTICKS YMIN YMAX FLAGS NUMCHN ABUFF1 CHNCOL1 ...: draws a line graph of the position xp, yp with the size xs, ys and the frame color FGCOL, the background color FGCOL who Tickfa r BE TICKCOL, the number X and Y ticks, the minimum and maximum Y deflection, the number of channels NUMCHN, the number of the displayed ArrayBuffer ABUFF the channel color CHNCOL etc for all channels. Up to 8 channels can be drawn in a graph.

FLAGS:
#define SKIP 0x10000
#define Gflags_sclf1 32

#define Gflags_sclf2 64
#define ??? 32768
0x10000 = with very large arrays, not all points are plotted
32 = the label be 2 before and 3 decimal places displayed
64 = the label are 3 Before and 1 decimal shown
default is 5 ago and no decimal
0x40000000 = There we a FTFT graphics drawn (combinded features must be passed in Array) YMAX here is a scaling factor. (the larger, the smaller the display)

DRAW PICT "path" XP YP XS YS TIME PICBUF:

Note: The time parameter is no longer supported. (Always 0)

Direct Mode: PICBUF = 0
paints a picture of the path file. Is XS = 0, it is adjusted on the full screen, XS -1 it is shown in its original size. Is TIME = 0, the image is displayed permanently, TIME> 0 so it appears TIME milliseconds and deleted afterwards.

Buffer Mode: PICBUF = 0!
The image is loaded on the first call in a buffer PICBUF (1-32), the second and subsequent calls the image from this buffer is drawn. If the command with -PICBUF the screen is called again deleted from the memory.
Is TIME = 0, the image is displayed permanently, TIME> 0 so it appears TIME milliseconds and deleted afterwards in correspondence of the last drawn background. Was previously shown a different image, it will re-drawn (masking), otherwise the screen deleted with the last background color.
In this masking mode, all background processes to be stopped in order to ensure the most accurate presentation time. Biodata are cached only up to about 1 second in the interrupt. The presentation times can be only multiples of the refresh rate of the connected monitor systemic always.

Note: Large images consume a lot of memory!
The loading of images can take some time when compressed images!
The paths are used only when charging, otherwise an empty string is passed.

Is Time <0 and PICBUF> 0 the rectangle is copied as a cut directly from a pre-charged buffer in the screen.

```
// Loads 2 images side-frame in the image memory
DRAWPIC "/Bild1.jpg" 0 0 0 0 0 1 1 // Loading image in memory 1
DRAWPIC "/Bild2.jpg" 0 0 0 0 0 2 2 // invites image in memory 2
// Figure 2 is masked by image 1 displayed for 20 ms
DRAWPIC "" 0 0 0 0 0 1 // shows image 1
TICKS 1000 // waits 1 sec
DRAWPIC "" 0 0 0 0 0 2 // shows image 2
TICKS 20 // waits 20ms
DRAWPIC "" 0 0 0 0 0 1 // shows again image 1
Remove DRAWPIC "" 0 0 0 0 0 -1 // Frame 1 memory
Remove DRAWPIC "" 0 0 0 0 0 -2 2 // image from memory
```

DRAW SCREEN NUM (XP, YP, XS, YS): copies the image buffer NUM in the currently active character background. The frame buffer can be set with SCREEN NUM.
Is xp, yp, etc is specified, the image memory in the coordinates xp, yp etc copied.
Is NUM -1 or -2 is used as the source of the entire screen 1 or 2 and not the image memory. (Thus, for example, can be produced a reduced copy of screen 2 to screen 1)

DEFINE itemType NUM XP YP XS YS TXBC TXTC TXTSIZ FRMFC FRMBC STYLE: defines an item type at position XP, YP in size XS, YS with the text color TXTBC and TXTC and the text size TXTSIZ, the frame color FRMFC and FRMBC and STYLE
STYLE:
1 = 0-> square frame, 1-> round frame

2 = item text xuy centered instead of left-aligned
4 = if option 2 left-aligned text is then centered only x

DEFINE TEXT NUM "text" WEIGHT: defines the text with the number NUM (1 ... N) with "text" and the weight WEIGHT. These texts can for example be accessed using the defined with the pool definition values or with random values (max. 512 texts)

DEFINE COLOR NUM Red Green Blue: defines the color NUM with the RBG values (0-1000 per color)

DEFINE PATH "path": defines the default path "path" for files that start with partial pathname (/ name). At startup, this variable is an empty string. **PATH** is also available as a string variable. This allows all file input and output redirect to other locations. The special drive name "package" is the container of the program itself. Thus files can be bound in the application for "stand alone" operation. Such with the menu command "Auto Start" bound file can be freely copied to other computers and other drives without installation and start. The special drive name "home" is where is the tcl file and is also in the string variable HPATH the directory.

DEFINE PATH "hd / VARIOTEST"
OUTPUT "/results/test.txt"
Creates a file with the path "HD / vario test / results / test.txt"

DEFINE VPATH "path": defines the path name that is used for the output of Varioport raw file. (see START ONL)

DELFIL "path " : deletes the file with the pathname "path".

EBUS FLG initializes a serial interface to an EBUS heating controller. The interface must be previously opened with SOPEN2. FLG = 0 starts communication, 1 they terminated.
EBS [index] is a string variable, then the heater values can be retrieved

using the.

1 = outdoor sensor
2 = DHW
3 = Boiler
4 = collector
5 = solar storage
6 = return
7 = burner
8 = solar pump
9 = Burner ???
10 = water ???
11 = room temperature

END: exit the program

ENDM: terminates a macro definition

else Alternative branch of **if then else endif** statement

endif: Ends an **IF THEN endif** or **if then else endif** statement

ends: Ends a **switch case** statement

bx = EyeTrack FUNC (P1) (P2) (P3) controls an eye tracker the company TheEyeTribe.
FUNC:
Start = 0 Eytracker
Stop 1 = eyetracker
2 = Push Mode P1 = 0.1 Push Mode on or off
3 = version set P1 = Version
4 = last status code
5 = set screen resolution P1 = ArrayBuffer number with the parameters
6 = get screen resolution P1 = ArrayBuffer number with the parameters

7 = get FrameData P1 = ArrayBuffer number with the parameters valid if result = 0
8 = Calibration clear
9 = Calibration start P1 = Number of points
10 = Calibration point start P1 = P2 x pos = y pos
11 = Calibration point end
12 = Calibration abort
13 = Get Calibration Result P1 = ArrayBuffer number with the parameters
14 = Start Recording gaze data to file P1 = Sbuffer number contains the path, P2> 0 are output at point instead of comma floating point numbers.
15 = Stop recording to file

Arrays: (see documentation theeyetribe)
SCREEN (> = 5 elements)

```
        int screen index; // Screen Index
        int screenresw; // Screen resolution width in pixels
        int screenresh; // Screen resolution height in pixels
float screenpsyw; // Screen physical width in meters
float screenpsyh; // Screen height in meters Physical
```

FRAME (> = 22 elements)

```
Long time; // Timestamp ms
bool fix; // Fix gauze
unsigned Long state; // Tracker state
short raw_x; // Raw gauze x
short raw_y; // Raw gauze y
short avg_x; // Avg gauze x
short avg_y; // Avg gauze y
short l_raw_x;
short l_raw_y;
short l_avg_x;
short l_avg_y;
float l_psize;
```

```
short l_pc_x;
short l_pc_y;
short r_raw_x;
short r_raw_y;
short r_avg_x;
short r_avg_y;
float r_psize;
short r_pc_x;
short r_pc_y;

CALIBRESULT (> 230 items)
bool result;
float °
float degl;
float degr;
float num;
struct CALIBPOINTS {
short state;
float cp_x;
float cp_y;
float mecp_x;
float mecp_y;
float acd_ad;
float acd_adl;
float acd_adr;
float mepix_mep;
float mepix_mepl;
float mepix_mepr;
float asdp_asd;
float asdp_asdl;
float asdp_asdr;
} Calibpoints [ 16 ];
```

FADE TIME SB BN (TTIME): copies the screen buffer SB in the visible image storage random pixel by pixel without replacement. If a key is pressed or the mouse stops the process and in RTIME the response time and is passed in the buffer whose number is BN is the percentage amount of copy previously copied pixels. In Time, an additional waiting time 1 percent each of the copied pixels is passed in milliseconds. If TTIME specified (in milliseconds) so the fade process is terminated after TTIME milliseconds. Here no key or mouse is queried.

FFT SNUM DNUM calculates a fast Fourier transform of the ArrayBuffer number SNUM in ArrayBuffer number DNUM.
The size of SNUM must be a power of. 2 (So for example, 128,256,512 etc) The ArrayBuffer DNUM must be exactly half the size.

FTFT SELECTOR ABNUM controls the Fast Time Frequency Transform (W.Martens 1992) The ArrayBuffer No. ABNUM used for input and output parameters, and must have at least 32 members. (must be specified only when SELECTOR 0)

SELECTOR:

0 = init FTFT

Array elements:

1 = Layer => Layer number eg 3 => 8 strips (max 4 => 16 bands)

2 = steepness (typically 1)

3 = bandstop (usually 1)

4 = residu (usually 0)

5 = Number of channels (max 8)

6 = Noise Floor (typically 30)

Calculate 1 = FTFT, each time you call a channel strip is calculated

Array elements

9-16 = channels to be calculated

// Results in array fetch 17-25

2 = Power of channel strips

3 = frequency of the channel strips

4 = bandwidth of the channel strips

5 = Signal of channel strips

7 = Delay calculation in samples

Return 10 = FTFT memory
11 = calculated from the raw data array (array index number 9) for the entire channel, the "combined features" in the output array (array index number in 10) The output array can then DRAW GRAPH (Flag 0x40000000) are displayed.

FIR NUM ABNUM CHSNUM CHDNUM MAVG: calculates an FIR filter (finite impulse response) with the coefficients from the array buffer with the number ABNUM and the online channel CHSNUM. Is CHDNUM = 0 is written back into the channel the filtered result of the input channel CHSNUM. Is CHDNUM> 0, it indicates the number of a buffer in which the result is written. MAVG is an optional rectifier and smoother. If> 0, the result of the filter is rectified and smoothed with a moving average of length MAVG. NUM is a sequential number (1..8) may 8 FIR filters are computed simultaneously. It also multiple filters of the same channel may be calculated.
The ArrayBuffer must scaling the coefficients array contain (usually 32767) and then the filter coefficients as the first element. The number of FIR coefficients is determined by the array size (= Array length -1)

Is NUM <0 a FIR filter is calculated with the coefficients from the ArrayBuffer ABNUM with the array and stored in the array CHDNUM CHSNUM. Here, however, the last data points of the target array remain unfilled.

FLUSH deletes all the data in the event buffer, eg all keystrokes stored until then.

FLUSHSCRN VAR (forcing the update of the display memory)
VAR = 1 => each graphics command is drawn immediately and displayed VAR = 0 => The graphics commands are only drawn, not shown. Only when FLUSHSCRN all previously drawn graphics are also displayed.
FLUSHSCRN 11 => allocates memory for intensive image

manipulation
FLUSHSCRN 12 => releases the memory again freely

for BNUM BEGIN END INC: defines with **next-ending loop.**
BNUM is the number of a buffer, BEGIN is the start value of the buffer, the end value END and INC Increment of the buffer. It must never be popped with goto or loop of a FOR-NEXT loop. A return is also not allowed within a loop. **Break** leaves a FOR NEXT loop.

FRAME XP YP XS YS FC BC STYLE: draws a box with or without filling
STYLE:
1 = round frame, otherwise square frame
Not draw 2 = background, but only frame

FONT = "Name": sets the font "name" as the active character set. All subsequent text will be drawn in this font.
The default character set is "Helvetica"
Note: The character set should also be present in the system. Otherwise a similar displays, which sometimes leads to unsightly effects.

FUNC TYPE NUM INPUT PAR is a numerical function of type TYPE
0 = Moving Average, PAR = Number of values to be averaged (max 1024)
1 = IIR Highpass, PAR = constant ca 1-12
2 = Moving maximum PAR number of values to be tested (max 1024)
3 = Moving median filter, PAR = number of values which are included in the filter (max 512)

All functions must be initialized with Type + 256th There is always up to 4 definitions (NUM = 1..4)

goto: label jumps to the marked label line
Label will be marked by a colon eg: MAIN
it must not command standing in this line.

GOSUB: label jumps to the marked label subroutine. Label will be marked by a colon eg: ISSUE
it must not command standing in this line. Subroutines have to **return** to close. Nesting is allowed up to a depth of 16th

HELPTEXT XP YP XS YS FS: creates a window with a text of heading XP, YP with the window size XS, YS and the font size FS. Text begins with an exclamation point, the text without the exclamation mark is displayed, otherwise text is the path of a file that contains a help text. This file can be either a plain text file (.txt) or rich text file (.rtf, .rtfd), which can then also contain variably formatted text and images.

HIDEC: hides the cursor (mouse) is **SHOWC** visible again.

ITEM SCALE1 ... scales "text" TYPE RSET: draws an item of type TYPE with the text and the text Responseset RSET and expects the corresponding user input. Is TYPE -1 the item is not marked, RSET -1 no user input is executed. The response weight is added to the scale SCALE. Up to 4 scales can be specified here is the value of the inverted negative scale value is added.
If the text "#" (just this sign!) So the text from the text pool is fetched with the current pool number.

INCLUDE "path" binds the script file path in the text.

INSTR (s1 s2) returns the position of the string s2 in s1 (0 ... N) or -1 if the string is not

ifne expression leads to the next line only if expression nonzero
ifeq expression leads to the next line only if expression is equal to zero
ifpl expression leads to the next line only if expression is greater than or equal to zero
IFMI expression leads to the next line only if expression is less than zero

if (expression1) == expression2: executes the next line only if expression1 is equal to expression2. Are also allowed! = (Not equal),> (greater than),> = (greater than or equal), <(less than), and <= (less than or equal). Is on the next line **then** then all parts of the program to be **else** or **endif** executed. There are up to 127 nesting of **if then else endif** statements allowed.

IF sx == text argument leads to the next line only if sx equal to the text in text argument is.

KILL AUTO: deletes all variables or car keys from memory.

KILL CONTROLS: deletes all defined Controls from the memory.

KILL PICTS: deletes all images loaded from memory.

LCD commands refer to the Logitech G13 Gamepad
The numeric variable **G13** is 1 when the G13 game pad is connected, 0 otherwise.
The G13 has a black and white display with 160 by 43 pixels. The buttons of the gamepad can be freely assigned with key codes using the Logitech Utilities.

LCD CLR: deletes all objects from the LCD screen of the G13.
LCD TEXT XP YP TS "text": prints the text "text" at the position XP, YP of the LCD screen of the G13 with the text size TS. If TS <0 is only the text used (from the 2nd call, changing texts for themselves)
LCD BAR XP YP XS YS POS: draws a bar graph with the size XS, YS to the position XP, YP of the LCD screen of the G13 with the Baran show of POS (0-100%)
Is XS = 0 is only used the position (from the 2nd call for changing bar graph)

loop: label as goto

LOAD DEF "path" **FLAG:** loads the definition file "path" in the memory. Even the START REC "Path command" invites t the channel name. If FLAG 1, the definition file is also sent to the recorder.If FLAG 3, the file is permanently stored in the recorder. ("Cold start file")

LOAD SPIL "Pfad.bin" **NUM:** loads the binary Realtimespil "Pfad.bin" in SPIL memory NUM. There are 4 SPIL memory equivalent to Varioport.
The Realtimespilprogramme be performed at START ONL as preprocessing, if defined in the definition file.

LoadText "path" **NUM ("Match") (OFFSET):** loads the text from the "File path" in the memory. There may only ever be a text without quotation marks in a row. A maximum of 512 lines of text are read. (see DEFINE TEXT) In the buffer with the number NUM the number of read lines of text is returned. Displays path to a directory, all file name of this directory is read (also invisible)
Match is an optional string can be loaded with only certain texts in which match occurs.
OFFSET is an optional starting point from which the text lines are stored:

LoadText "path" 1 "picture" 100
stores the text in the file " path " in the DEFINE TEXT buffer starting at position 100. The number of rows found is returned in b1. Only read texts in which "image" occurs.

If path points to a directory and OFFSET is negative, then the directory entry is stored with the IndexOffset in DEFINE TEXT position. 1

LOAD VPDFILE PATH

Loads t a Vitaport, Varioport data file. Then all the channels displayed and the following variables of the file can be queried:

Strings:
VPD _NM (X) channel name, channel number X =
VPD _UN (X) channel unit, X = channel number
VPD _DATE Date of recording
VPD _TIME Start time of recording
VPD _DURATION length of recording (Hour, Min, Sec, msec)

Numerics:
VPD_CHNS Number of channels
VPD_GFRQ Global sampling in heart. If this variable is set, the values indicate a correction factor for the global sample 1.0 = no change> 1 changes to lower rates <1 to higher rates.
VPD_MXMS recording time in milliseconds
VPD_STMS start time in milliseconds since midnight
VPD_SR (X) Sample Rate of the channel X
VPD_CNUM ("CNAME") determines the number of the channel with the name "CNAME"

VPD_CURS supplies in DEFINE CHANNELS the cursor position in milliseconds at K lick in the graph. If this variable is set VPD_CURS = xxx all cursors are set in all graphs on this value and the corresponding values of the graph displayed.

SVPDHD eventually secures modified Varioport header in VPD files

VPDINT START END NUM supplies the Buffer numbers START to begin in the end of END a, e Interval in milliseconds the interval NUM. NUM will be initialized with 0, the number of a, e Intervals in NUM returned. If the result is negative in this case there is an inconsistency in the interval markers. START, END and NUM are the numbers of the corresponding buffer, so VPDINT 1 2 3 delivers in b1 and b2 the beginning and end of the interval whose number was

specified in b3.
If NUM initialized with -1, the A, E are returned Global Marker.

SETMRK TYPE TIME sets a marker of type TYPE to the value TIME in milliseconds.
TYPE:

1 = Global Start marker (A)
2 = Global end marker (E)
3 = start marker (a)
4 = end marker (s)
5 = Delete Marker (A or E)
6 = Delete all markers (a, e)
7 = Save all markers (A, E, A, E)

Interval brands can hold down the mouse button in a channel are set as follows:

A: sets the start of interval brand
e: sets the interval end brand
c: deletes the interval brand under the cursor
r: deletes all brands interval
R: deletes all brands interval and the Global Brands
x: backs up all intervals in the VPD file
A: sets the global Start Marker
E: is the global end marker

The content of the channels can then use the command AX = VPDCHAN (y) OFFSET copied into an array and displayed with DRAW GRAPH. (x = number Array y = channel number, offset is the file position in samples.

a1 = VPDCHAN (1) 256 invites t Channel 1 in the array 1 from the position 256. When the sampling rate would be 128 Hz that then the 3rd second.

The channels are already scaled physically, eg in uV

NVPDF "PATH" GSRATE TOTIME "CNAME" SRATE DSIZE DTYPE OFFSET MFAC DFAC "UNIT" creates a VPD file named PATH, the global sampling, the total time in seconds that the channel CNAME, the sampling SRATE, the data size DSIZE, the data type DTYPE (usually 5), the offset OFFSET, the MulFaktor MFAC, the Divide factor DFAC and the Unit UNIT.

NVPDC generates a channel in an existing VPD file. The parameters are the same as those from NVPDF CNAME

VPDGRAPH "text" XP YP XS YS FGCOL BGCOL TICKCOL XTICKS YTICKS YMIN YMAX FLAGS STARTMS DTIME LNUM NUMCHN CHAN1 CHNCOL1 ...: defines a Varioport Line graph with user interface to the position xp, yp with the size xs, ys and the frame color FGCOL, the background color FGCOL, the Tick Color TICKCOL, the number X and Y ticks, the minimum and maximum Y deflection, the starting time STARTMS and the display time in seconds DTIME. LNUM is a sequential number (1..16), the number of channels NUMCHN, the number of the displayed Varioport (.vpd) channels of the channel color CHNCOL etc. for all channels. Up to 8 channels can be drawn in a graph. Is CHANx negative the corresponding ArrayBuffer is shown. The graph is provided with all of the standard user interface (for example Crosshair Cursor) According to the definition need to GET CONTROLS the control loop can be activated and the count must be deleted with KILL CONTROL last. Text is the text in front of the selection Popup for the channel.

FLAGS:
#define Gflags_sclf1 32
#define Gflags_sclf2 64

32 = the label be 2 before and 3 decimal places displayed

64 = the label are 3 Before and 1 decimal shown
default is 5 ago and no decimal

Other Flags see DRAW GRAPH

VPDGRAPH option flag = 0x80000 and 2 specified array uses the second array as a marker to indicate on the 1st channel brands.
Marker values:
1 = short line marker in the color of 2. Arrays
2 = Cross marker in the color of the second array
3 = small circle in the color of 2. Arrays
4 = long line marker in the color of 2. Arrays
5 = short line marker in red
6 = short line marker in green
7 = short line markers in blue
// Show these markers typical EDA SCR surveying with horizontal and vertical lines in 3 different colors
256 = EDA start SCR indicator (green)
512 = maximum EDA SCR indicator (blue)
1024 = EDA 70% drop from the maximum indicator (orange)

LOCK MSEC: waiting MSEC milliseconds are stopped all background processes. This time measurement is more accurate than TICKS
Attention: biodata are cached in Interruptbuffer only up to about 1 second.

LINE XP YP COL (DICK): Draw a line to the position XP, YP in color COL. The starting point can be set with MOVE. optional line thickness DICK , Is COL bit a dotted line is set 8 drawn.

MONITOR NUM CHNM Smask SBIT emask EBIT DBNUM: Sets a channel bit control monitor (NUM 1..16) of the channel CHNM. The channel will be online tested for bit pattern and returns the distance in milliseconds between 2 bit states.
The bits of the bit mask Smask A timer starts at bit state SBIT and

stops the timekeeping at the bit mask and the bit state emask EBIT. The measured time is returned in the buffer with the number DBNUM.

MOVE XP YP: Sets the starting point of a line on XP, YP

MACRO "NAME" **ARG1 ARG2 ARGN:** starts a macro definition. All commands between MACRO and ENDM be saved under the name NAME and can be replaced with this name in the further program.
This argument may be (only BUFF and SBUFF) passed.
Example:

```
MACRO "test" s1 b1
MESSAGE s1 b1
ENDM

test "The value is:% f" 200

END
```

MESSAGE "text" (expression) generates a Userbox with the message text and the value expression. The text formats the expression after "c" Convention eg message "Value:% f" b1 indicates the value of buffer 1

NEW SCREEN (FLG): creates a new screen. Is only the beginning of the program even necessary.
If a 2nd screen is found can be used with FLG = 2, this screen as a separate graphic window.
256 => creates an OpenGL screen, can be written to the GL but only with special open commands.
1024 => produces 2 adjacent window as Screen 1 and Screen 2 to simulate 2 screens
1024 + 512 => as above, but screens with each other

// Special Options
32768 + 1 generates a screen on the first monitor
32768 + 2 generates a screen on the second monitor
256 is added while an OpenGL screen is generated
Optionally, the window size to be fully specified
NEW SCREEN 32768 + x XP YP XS YS "name"
4 is added to FLG, screen 2 is also displayed on the main screen
If in addition 8 adds, you can make entries only on windows. 2
eg NEW SCREEN 32768 + 2 + 4 + 8 100 100 800 600 "Submenu"
generates a second screen in the foreground on which one can now make entries is until it is closed again.
The window 1 is inactive.

NFormat = VAL1.VAL2: sets the output format of numbers.
VAL1 is the number of digits before the decimal point and **VAL2** is the number of digits after the decimal point.
Communicates between VAL1 and VAL2 a point, so the output point number is printed with the dot as a decimal point, here is a comma, as in the output a comma is used. Default is the point and numbers to 4 decimal places.

NGRAPH XP YP XS YS FC BC CC "text" TIME YMIN YMAX YLHSIZ YTICKS YTC XTICKS XTC FLAGS PBNUM BNUM CNUM: creates a line graph of the size XS, YS at the position XP, YP with the colors of FC (Border Color) BC (background color) and CC (color channel) of all TIME microseconds draws the next value.
YMIN is the smallest and the largest YMAX Displaying YWERT-YLHSIZ is the width of the Y Label field, YTICKS is the number of Y ticks, tick the YTC color. XTICKS X is the number of ticks and XTC the corresponding color. Is PBNUM> 0 as the frame buffer number contained in the corresponding BUFF is displayed as a background. BNUM is a serial number, CNUM is the corresponding channel number, or if <0 the corresponding buffer number.

FLAGS bits:

```
#define Gflags_AutoOffset 1
#define Gflags_AutoScale 2
#define Gflags_DCMode 4
#define Gflags_WaitEvent 8
#define Gflags_Device2 16
#define Gflags_sclf1 32
#define Gflags_sclf2 64
#define Gflags_offset 256
#define Gflags_ABMode 512
#define Gflags_NRIMode 1024
#define Gflags_Ilabel 2048
#define Gflags_Integrate 4096
#define Gflags_UI 8192
#define Gflags_ BarGraph 16384
```

1 = The curve is in excess of the maxima set back in the middle

2 = The curve is autoscaled in option 8192 the channel is set fix, otherwise selectable

8 = Only data drawn when new present, is also automatically set when TIME = 0

16 = the data of the 2nd serial interface display

32 = the label be 2 before and 3 decimal places displayed

64 = the label are 3 Before and 1 decimal shown

256 = the Line Graph raw value is shifted by the value Goffs

512 = at CNUM <0 is the corresponding ArrayBuffer used as a source, otherwise the Buffer

1024 = When a redraw with XS = 0 the graphics position is set to zero

2048 = The label is drawn inside the Count, not outside

4096 = The graph curve is filled to the zero point

8192 = It User Interfaces are displayed on the graph with which the scaling etc. of the curve can be adjusted.

This option must bx = CONTROLS querying Controls are enabled.

If option 2 then no channel popup is shown, but only the area defined by Text

16384 = It draws a Bar Graph. The bar takes up the following colors: If the value is smaller than the YTICKS color channel is used, the value is smaller than the color for YTICKS XTICKS is used, otherwise the color from XTICKS.

If NGRAPH again called with XS = 0 may allow the scaling without redrawing the entire graph to be changed ,
When Flag = 8192 and VPD channel representation (BNUM> 0) be ymax = 0 read the scales and the autozero flag from the definition file at ymin.

NORMA anum normalizes the array anum with the number, that is the newest value is then in Index. 1

OUTPUT "Path" creates a results file name path
The time counter **STIME** is set to zero, so that **STIME** always contains the time after the start time.

OpenCV
Heretofore, various object identifiers are implemented as a test. ("Face Reader")
OCV SELECTOR "path1" "path2" FLG SIZE XP YP XS YS
SELECTOR
0.1 = testing various hair Classifier in path1 and 2 in a webcam window
2 = start a webcam service with window in the middle and returns the value of Path2 classifier in Buffer B800 back.
3 = stops the webcam service.
4 = calculated with a previously run video from the frame just shown a Classifier and displays the results in a window with the position XP, YP, XS, YS

path1 is the path to a hair or LBP Classifier defintions XML File.
eg face recognition
Path2 is the path to a hair or LBP Classifier defintions XML File.

eg smile on previously detected face.

FLG = 4 to show the webcam window calculations.
SIZE is a reduction factor for the webcam resolution.

OPEN GL
So far there are only a few OpenGL commands.
First, be a screen for OPEN GL has produced => NEW SCREEN 256
OGL_SETUP P1 P2 produces an Object P1 with the phase P2
(Object 2 = Checker tunnel, 99 = Cocotron icon)
OGL_PERS left right bottom top near_val far_value constitutes Perspective (glFrustum)
OGL_CLR RG B T clears the screen with the RGB color RGB (glClearColor)
OGL_DRAW XYZ nu Zoom draws the object (X, Y, Z Zoom)
OGL_MOVE XYZ sets the starting point of a line on X, Y, Z
OGL_LINE XYZ COL draws a line to the point X, Y, Z in color COL
OGL_CIRCLE XYZD BGCOL KCOL FLAG draws a circle at the point X, Y, Z having the diameter D, the fill color BGCOL, the circle color KCOL. Is Flag> 0, the circle is filled, otherwise only the outline is drawn.
OGL_TEXT (Parameters see text) draws a text.

OPTIONS = VAL sets the option flag to VAL; (not used in this version)

ongoto VAR VALUE VALUE "label" VALUE VALUE "label" ...
branches depending on the value range on label for VAR is any expression allowed

ongosub VAR VALUE VALUE "label" VALUE VALUE "label" ...
jumps each label on the value range on the subprogram for VAR any expression is allowed

or expression1 == expression2: OR linkage after IF statement

POOL NUM BNUM (RANGE): (random) reads NUM (max 2047) item numbers of a pool of a can extend over several lines BNUM is the index used for Buffer number (Pool Index). If every time you use the pool Strings "#" is increased or the numeric variable pool to. 1 The numbers may be separated by spaces or commas. Is NUM negative, NUM numbers from 1 to - NUM generated randomly. (without replacement, ie each number occurs only once) is also BNUM negative, NUM generated numbers with replacement of 1-RANGE. If the first two parameter negative and Range = -1 so be NUM numbers in numerical order defined (1,2,3,4,5..NUM)
The numeric variable POOL returns the next value from the pool
A maximum of 4 pools are activated simultaneously. The commands are then called up POOL1 pool4 where POOL1 equivalent with pool.
POOL -10 20 generates a lottery drum with 10 lots from 1-10. The number of previously drawn numbers b10 is in Buffer
b1 = POOL draws without replacement a number out of the drum.

PAUSE SEC: SEC seconds pause. Caution: If a movie running in the background, the time measurement is not accurate than a frame rate of the movie

PICSIZ PATH BNUMx BNUMy delivers the image size in x and y pixels of the image in image path PATH in the buffers with the numbers BNUMx, BNUMy

PlaySound "path" according to NUM: plays a sound file "path" with the volume up (0-100%) on the audio channel NUM (1-8) from
This enables up to 8 sounds are played simultaneously.
Is added to 256 NUM, the sound endlessly played (eg background music) in some cases when loading sounds to delays of up to 1 second. To avoid this, you can load a sound first and then play.

PlaySound "test.wav" 1 1
loads the Sound test.wav in Channel 1

PlaySound "" -100 1
plays the sound of channel 1 with 100% volume.

Is loud negative a previously charged sound is played.

Is NUM <0 thus be played predefined sounds, the path is set ignored.

-1 = Ping
-2 = Pop
-3 = Rest
-4 = Tink
-5 = Bloop
-6 = Attractor
-7 = Basso
-8 = Boing
-9 = Sosumi
-10 = Glass
-11 = Correct
-12 = Purr
-13 = Submarine
-14 = Morse
-15 = Incorrect
-16 = Hero

default = Frog

PLAY BEEP VOLUME RATE TYPE: sounds continuously with the frequency (Hz) and the volume up (0-100) If RATE = 0 is an ongoing sound stopped. TYPE = 0 produces a Rechteckton, TYPE = 1 a sine wave and TYPE = 2 noise

PLAY MOVIE "path" XP YP XS YS: plays a movie file is XS = 0 is the whole screen used XS is -1, it is played in its original size.

POLY ABNUM FC BC FLAG: draw a polygon with the colors of FC and BC. Is Flag> 0, the polygon will not be filled.
ABNUM is the number of array buffer (ABUFF) in which are the

elements of the polygon. Element 1 and 2 are the X and Y positions of the start point, then each follow X and Y delta values of the polygon corners.

PRINT works as **SAVE, but the results are printed in a window, which can then be further processed.**

RCARD PATH BUFF: reading a Varioport memory card (Volume VPDATA) or via serial interface the memory card on the recorder. The serial interface must be previously defined with SET SPORT. The VPD file is saved under the name PATH.
BUFF is the number of a buffer in which the progress of the upload is displayed. (In 1kB steps counting backwards) If the value in this buffer after the end of the instruction is less than zero, an error occurred.

RDWG NUM records the VPD graphs NUM new

REQUIRE num generates a warning when the uVariotest version is less than num.

RESET STATS: resets the statistics of Biosignaldatenerfassung.

RRTIME: sets the response time to key to zero (or starting time).

Return: ends a subroutine.

RESTORE "path": loads t the buffer with the number 950-1000 and the StringBuffer with the numbers 90-100 from the file "path" back.

Rwav PATH ABN1 ABN2: reads a WAV file of PATH in the two arrays ABN1 ABN2 (stereo channels)

SAVE PICT PATH BNUM stores the image stored in the image buffer BNUM depending ending from: jpg = JPEG, TIFF = TIFF, bmp = bmp, png = PNG

To be able to adapt for example images in any size and format.

SAVE DEF PATH writes the currently loaded definition file in the PATH

SAVE ARG1 .. argX: writes the arguments ARG1 .. argX in the output file
ARG is either a "text" or an expression.

sx = STRING: sets the StringBuffer sx (1-100) to the string STRING (can again SBUFF or a constant string "text" be strings can use the "+." are composed characters.
SBUFF1 = "test" + "Construction" + "Save" + SBUFF2
All String arguments can be constant strings or SBUFFS
A special string "#" (Only this symbol) He will be replaced by the particular with the DEFINE TEXT AND POOL string.
Additionally, you can give solid SBUFFS a name:
DFS: "Name"
Name = "Meier"
The alias overrides other definitions! If, for example DFS: "USERID" is defined is the predefined variable "USERID" lost.
If names are assigned to a particular SBUFF, the Buffer number must be specified:
DFS: "Name" 1
has the name "name" to the Sbuffer s1
SX [N] provides the N th character from the string as a string. It can also be produced in any character string with SX [N] = NUM: sets the N th (1..n) character in the string to a value of NUM

SBUFF [val] provides the StringBuffer with the index val

SCREEN NUM: Sets the active image memory. Is NUM = 0 all drawing operations are drawn directly into the visible screen. Is NUM> = 1 an invisible image memory is generated. All subsequent drawing operations are written in this case in the invisible image memory. With

SCREEN 0 and SCREEN DRAW NUM can then be copied into the visible screen of the image memory. Is NUM = -1, the graphics output is explicitly = -2 switched to screen number 1, with NUM to screen number.
2
The resolution of the invisible image memory is set at the first call to the resolution of the currently selected screen.

SET ALEN ABUFFx LEN (FLG CHNUM (TRCHN trval TRFLG)):
sets the length of the array x (1-32) to LEN.
Ax = number behaves as follows:
If FLG bit 5 (16) set the array is shifted, so the newest value is always in the final index (in this case must be the entire array to be pushed which is slow for very large arrays), otherwise the new value will be written into the next respective Index and at the end of the array, the index is reset to 1gesetzt (which is extremely fast, because is not pushed here). The array can be "normalized" with NORMA, ie newest value at position last Index

If FLG = 1 then referred CHNUM the number or the name of the channel Varioport then to start recording his FIR filtered data scaled in the corresponding ABUFFer stores.
If FLAG = 3 then the storage in the array will be stopped if the array is filled, otherwise the data will be overwritten repeatedly. The status of the array can be queried with AFLGx (is full when AFLGx & 128 non-zero)
If FLG = 4 then referred CHNUM the number or the name of the channel Varioport then start recording its raw scaled in the corresponding ABUFFer stores.
If FLG = 5 as 4 but stop when full array.
If FLG = 8 must follow the following parameters channel, trigger channel, trigger, trigger flags.
copies the raw data of the channel **CHNUM** in the array when the channel **TRCHN** the value **trval** occurs for the first time. Thereafter, the entire array is filled and the full array flag is set (AFLG) Is **TRFLG** = 1 VAL is ORed with the trigger channel and set at nonzero the trigger.
it must be 1.4 or 8 set only one of the flag bits!

SET SVOL NUM LEFT RIGHT: set the amplitudes of the stereo speakers on the audio channel NUM LEFT, RIGHT. 0 = off, 100 = full volume. Is NUM = 0, the volume setting refers to the PLAY BEEP tone.

SET GSVOL LEFT RIGHT: sets the global amplitude of the stereo speakers on LEFT, RIGHT. 0 = off, 100 = full volume.
This setting affects all other sounds and other programs! (OSX)

SET FREQ BRATE LEFT RIGHT: set the frequency and amplitude of a continuous tone already generated.

SETMRK TYPE TIME: Positions marker in the previously uploaded file.
TYPE = 1 => sets gloablen start marker on TIME in milliseconds
TYPE = 2 => sets global end marker on TIME in milliseconds
TYPE = 3 => Start sets marker on TIME in milliseconds
TYPE = 4 => sets end marker on TIME in milliseconds

SET SCREEN XS YS (XO YO) ("name"): limited or expanded the screen to the size XS, YS. Must be called before NEW SCREEN and is used for example for developing and testing programs are supposed to work on smaller screen resolutions.
Become XO and YO the window is positioned at specified position XO YO, XO is <0, the window is centered on the screen. Name is an optional window Name

SET CMPVAL CHNUM MAX MIN: sets the reference values of Biokanal CHNUM on MAX and MIN. If the values are exceeded or under during Biosignalaufzeichnung, exceedances are counted and can be queried with CMAX and CMIN.

SET CHCOR CHNUM CCNUM SCALE: sets the correction channel of Biokanals CHNUM on CCNUM. CCNUM is multiplied by SCALE and

added to CHNUM.
(Eg EOG correction of the EEG)

SET SIPORT PNAME (BR ABUFFNUM): sets the port name for the Serial Input the Biodatenaufzeichnung on PNAME. If a baud rate and a ArrayBuffer number indicated, a "serial input" is read automatically and continuously in the ArrayBuffer.

SET SPORT PORT MAXBAUD FLAGS: sets the parameters for the serial port to be used.
PORT is the name of the ports to be used (as text) will show "Ignore" has been selected in the menu of the serial port is used. If the string starts with the @ character, the string is interpreted as a URL and the serial connection is example established via the WLAN WiFly module from Roving Networks (MAXBAUD is ignoring).

MAXBAUD is the baud rate to be used (see table)

FLAGS will be ignored , Dynamic baud rate is NOT more supports.

Baud rates:
0 = 9600
1 = 19200
2 = 38400
3 = 57600
4 = 115200
5 = 230400
6 = 460800
7 = 4800

SET TIMER NUM BNUM TIME FLAG (P1 P2 P3 P4): sets the timer number NUM (1-8)
BNUM is the number of the timer used as buffs. Is BNUM = 0 all timers are cleared. The Buffer counts the time and is set for the end of the interval to 0.

TIME is the time in milliseconds. Is TIME = 0, the timer is cleared.
FLAG:
1 = timer runs forward, otherwise reverse
2 = Buffer counts in seconds, otherwise milliseconds
4 = it is in P1 a SBUFF number given in which a path is set to a sound, the sound is started when the timer expires, P2 is a sound channel number, P3 = if> 0 HCHAN (qv), P4 = value HCHAN
8 = it is a BIT issue started after the timer P1 = BIT, P2 = Time in milliseconds, P3 = if> 0 HCHAN (qv), P4 = value HCHAN
16 = there is a serial handshake line output (such SHAND2) started after the timer P1 = RTS, DTR, P2 = Time in milliseconds, P3 = if> 0 HCHAN (qv), P4 = value HCHAN

= 128 timer is cleared (such as TIME = 0).

SFLUSH (NUM) deletes all the data in the Serial Buffer, eg all A and expenditures stored until then. If NUM is omitted or 1 is the first Serial interface meant for 2 to 2. Serial Interface

SBOX "text" XP YP SPBN SPSN BRBN DFBN DFSN opens a setup window with the heading "text" at the position XP, YP for querying serial port, serial baud rate and definition file (located in / DEF) SPBN is the number of buffers to selection and restitution of the serial port. In StringBuffer with the number SPSN the name of the serial port is returned. BRBN is the number of buffers to choose from and return to the baud rate. DFBN is the number of buffers to choose from and return to the definition file. In StringBuffer with the number DFSN the selected definition file is returned.

SHOWC: makes the cursor (mouse) visible again.

SLEEP SECS: sets the "sleep time" during CONTROLS user query. Normally, the processor load in uVariotest is always 100 percent (SECS = 0). This is to ensure an immediate reaction time as possible on all the events. In battery mode (laptop) and simple programs that do not measure response times, it may be useful to adjust the SLEEP value to eg 0.1 (1/10 second). Thus, the reaction time is reduced to 1/10 of a second but the processor load falls very clearly and the runtime with battery is considerably longer.

Sprites:

Sprites are sliding partially transparent objects, eg for simple biofeedback ads.
The parameters of the sprite command for example can be controlled with the AUTO command, so as to move objects on the screen automatically.

Spriten NUM PNUM "Path" PANZ or
Spriten NUM PNUM PID PANZ: creates a sprite with the number NUM (1-64) and the picture number PNUM (1-512) from the picture "Picture Path" or the resource with the number PID and the number of images in PANZ (PNUM + PANZ must be less than or equal 64 !!!)
Shall with Path more images are loaded so the image path with 3-digit numbers and leading zeros must end (eg "Bild000", "Bild001" etc.) and all images must be numbered consecutively. The images must be drawn on a white background. (Object border = white) and the color white may not occur in the object itself or all white picture elements appear transparent.
Sprites NUM XP YP SIZ ANGLE (TEXT) : Sets the position XP and YP and the size SIZ (normal = 1) and the angle ANGLE (0-359 degrees) of the sprite number NUM. ANGLE is the angle or the image of an image sequence number (eg, spacecraft)
In a text sprite, the text can be changed dynamically by specifying TEXT

Spriten and **sprites** are implemented. SPRITEW and SRITEU are no longer necessary.
Sprite Images must be in PNG format.

Spriten invites all located in the folder png when specifying a folder.
Spriten NUM PNUM "Path" -1 XS YS enables loading of PNG Tiles with multiple images in one file. XS and YS is the size of the sub images. Tiles have often 640 * 480 pixels with 50 frames, that is, the sub images are 64 * 64 pixels in size here.

Spriten NUM TCOL "& TEXT " FSIZ XS YS produces a transparent text sprite with the text color and the font size FSIZ TCOL and the overall size XS, YS.

Sprites used as a coordinate system, alternatively, the usual for all objects, or when you specify SIZE as a negative value refers to XP and YP is on the center of the object and the origin of the coordinate system on the bottom left (not top left)

The ANGLE parameter turns now with only one image by the specified angle.
(Otherwise, specify 0)

SPRITE NUM XP YP returns 1 if point xp, yp in Sprite
SPRITE 0 S1 S2 returns 1 when number Sprite s1 and s2 overlap.
KILL sprites away all sprites from the screen and memory

SPEAK "TEXT": speaks the text "TEXT" from the built-in speech output.
works to time only on OSX. The voice acting is also in very good German if they have charged the German spokesperson "Anna" for example. The speaker, the speaker can with eg SPEAK "Markus: I'm Mark" or SPEAK "Anna: My name is Anna" choose.

Sopen 2 "PORT" **BAUD:** opens the serial port with the name "PORT" and the baud rate BAUD. Then you can with SWRITE2 written to the port and read with SREAD2. The string port starts the @ character, the string is interpreted as a URL and the serial connection is established via the WLAN WiFly module from Roving Networks (BAUD here is the TCP port number to use (2000 at Roving)).

CHEQUE "PORT": returns the numeric value of Portdescriptors for the serial port with the name "PORT". Valid values are positive numbers. If this value is -1, the port does not exist.

SREAD2 NSX: reads a string from the extension by sopen 2 open port in the StringBuffer with the number NSX.

The numeric variable SGET2 delivers the next byte of the opened with SOPEN2 ports.

The numeric variable SBAV2 returns the number of available bytes of the opened with SOPEN2 ports.

SWRITE2 "TEXT" (FLG) **:** Sends the string "TEXT" with the **sopen** 2 open port. If FLG> 0, the string is interpreted as a command and ships NMEA correspondingly provided with checksum.

SWRITEN2 "TEXT" **LEN:** sends LEN characters of the string "TEXT" with the **sopen** 2 open port.

SWABF2 ABNUM LEN: LEN sends binary (8-bit) characters of the array with the number at the ABNUM with SOPEN2 open port.

SRABF2 ABNUM LEN: reads up to len binary (8-bit) characters in the array with the number of ABNUM with SOPEN2 open port.

PBPROC2 ABNUM: starts a handler for the device K5 POWERbreathe the company and provides data in the array with the number ABNUM. The serial port must match SOPEN2 to be open. The data set is stored 9 (6 bytes) in the array from the cell.

SHAND2 VAL: sets the handshake line 1 = DTR, 2 = RTS with the **sopen** 2 open ports.

SCLOSE2: closes with **sopen** opened serial port.

SWRITE "TEXT " : sends the string "TEXT" in the open with START ONLINE Port.

START REC "PATH" **FLAG:** starts Varioport data recording
The definition file path. If path is an empty string, the internal definition file is used. If FLAG = 0, the memory card is overwritten, is FLAG = 1, the measurement is attached to an existing file.

START CONL OPT: launches a cascadierte online transmission with the previously loaded definition file and defined with the CASCADE command channels. Is OPT> 0, the file is saved as VPD file.

START SCAN ONL NUM CHAN1 .. CHANX: launches an online transfer with the sampling SCAN and NUM channels. This is followed by the channel number of the desired NUM channels. Here, the original channel name can be specified in quotes if the definition file above with **LOAD DEF** loaded. The results of the data collection can be accessed via special variables and can be stored in the result file.
Is NUM negative, only all preparations are made for the data collection, but do not start really. In particular, the channels defined herein are defined in this order. This is, for example, a prerequisite for the channel representation. Online channel numbers refer to the order as specified in the START command ONL. (1 ... N)
NUM must be a constant number, not a variable and can be up to 35th

Is SCAN = 0 data acquisition with the parameters of the previously with is **LOAD DEF** loaded definition file starts including preprocessing.

There are, however, the sampling, pre-processing, etc. This file uses only the selected channels in the START command. Only WORD channels are used. If specified here in addition to the option -1 at 0 a Rohwertedatei named USERID.vpd or if **DEFINE VPATH** "path" was performed previously with "Pfad.vpd" output, which can then, for example, be evaluated with Vario Graf.

START SCAN XONL NUM CHAN1 .. CHANX: defined up to 35 online channels. Unlike START ONL NUM must be a variable here, and channels must be specified as strings.

START VREC "path" **XP YP XS YS DEVICE COMPR:** start a video recording in the QuickTime file "path" with the image size XS, YS with the camera source DEVICE (1,2 etc) and the compression COMPR:

0 = QTCompressionOptionsLosslessAppleIntermediateVideo
(Resolution here is the native resolution of the camera)
1 = QTCompressionOptionsLosslessAnimationVideo
(Resolution here is the native resolution of the camera)
2 = QTCompressionOptions120SizeH264Video
(160x120 resolution)
3 = QTCompressionOptions240SizeH264Video
(320x240 resolution)
4 = QTCompressionOptionsSD480SizeH264Video
(720x480 resolution)
5 = QTCompressionOptions120SizeMPEG4Video
(160x120 resolution)
6 = QTCompressionOptions240SizeMPEG4Video
(320x240 resolution)
7 = QTCompressionOptionsSD480SizeMPEG4Video
(720x480 resolution)
8 = QTCompressionOptionsHD720SizeH264Video
(1280x720 resolution)
9 = QTCompressionOptionsHD720SizeMPEG4Video
(1280x720 resolution)

Audio is with QTCompressionOptionsHighQualityAACAudio coded

Is XS <0 No image is displayed, otherwise the monitor image at position XP, YP is displayed. If the path is an empty string "" nothing is recorded, but only the image at position XP, YP displayed.

If the function is called before stopping with a valid path, a recording is started, stopped with an empty path again.

The recording must **STOP VREC** be stopped again.

STOP REC: stops Varioport recorder recording

STOP ONL: stops the Online Broadcast

STOP MOVIE: Stops a Movie

STOP SOUND NUM: stops a sound in sound channel NUM

STOP VREC: stops a video recording

STOP GRAPH (NUM) : Stops all graphs and Values or NUM stops Graph No. NUM

STOP TIME: stops SHOWTIME

SOPT "path" "Wait text" WAIT TIME OPTIONS: starts the SOPT test with pictures of the path and the path waiting Text and **WAITING TIME** (sec) between trials.

The sample images have hot Bild01.jpg to Bild04.jpg, the test images Bild1.jpg to Bild12.jpg

SHOWTIME XP YP XS YS SIZE FCOL BCOL FRCOL FLAG

TIMER : ~~"label":~~ indicates a time at position XP, YP and the size XS, YS in the text size and the color SIZE FCOL, BCOL and the frame color FRCOL. Is FRCOL -1 no border is drawn.
Is Timer> 0 (in seconds) a timer is displayed, otherwise the time of day.
FLAG = 1 show time or timer on, otherwise invisible
FLAG = 2 Timer runs forward, otherwise reverse
FLAG = 4 audio signals from 5 seconds before the end of every second
FLAG = 8 show timer time in seconds, instead of hours, Min, Sec
The variable TSTAT is in Timer Mode 0 and becomes 1 when the timer expires.

STORE "path" **(FLAG)** stores buffs, and SBUFFS in the file "path".
If FLAG = 0 Buffs 950-1000 or SBUFFs 90-100 stored
If FLAG = 2 is a ArrayBuffer saved the number in the high word must be passed:
STORE "HD / abuffer12.abf" (2+ (12 * 0x10000)) stores ArrayBuffer 12 from.
RESTORE "HD / abuffer12.abf" reloads the ArrayBuffer

ListBox: starts a dealer for a trigger box, which is connected to USB and sopen 2 must be open. Repeated calling of this command deletes the previous EventBuffer respectively.

SVPDCHN CNUM ABNUM OFFSET LEN writes the ArrayBuffer ABNUM in the VPD channel CNUM from de point OFFSET and with the length LEN. If len = 0 the length of the array buffer is written

STRINGTABLE NUM PATH SEPCHAR LIN COL DSTR ENC reads up to 4 (NUM => 1..4) ASCI tables such as CSV files and reads individual fields.
If PATH is specified, the command reads the file. SEPCHAR is the delimiter, for example, semicolon (";") in Buffer LIN number the number of rows found in and Buffer number col returns the number of columns (the 1st line).

Is Path is an empty string, the command reads the string of line lin and the column col in the Sbuffer No. DSTR. Is this col = 0, the number of columns in this row will be returned in the string.
ENC is an optional parameter that describes the text encoding of the input file:
0 = UTF8 encoding (default)
1 = Isolatin coding
2 = MACOS novel coding
3 = UTF16 coding

TBOX EVENT ITEM: returns the numeric value of the event event in the trigger box.
ITEM:
1 = Event Type
2 = Event On Time
3 = Event Off Time
4 = AD converter 1 result in EVENT = 0 (type = 0x40)
5 = AD converter 2 result at EVENT = 0 (type = 0x40)

TEXT "text" XP YP SIZE FC BC (XS YS FRCOL FLAG): draws a text at position XP, YP with the font size and colors SIZE FC and BC Optionally, a frame size, color and flag (XS, YS, FRCOL, FLAG) to specify. In this context of the text is then entered formatted. If **FLAG** = 1, the text line by line, centered, left justified otherwise.

TEST NUM PATH1 PATH2 pATH3 FLAGS: controls on some experimental features.
NUM:
Convert 1 = IPD files
2 = QDF files Convert (no duplicates in the header)
PATH1 = QDF path PATH2 source path, pATH3 target path.
3 = QDF files convert (duplicates in the header)
PATH1 = QDF path PATH2 source path, pATH3 target path.
5 = encoding a text file change:
PATH2 is the source path, pATH3 the Zeilpfad. FLAGS coding the

Zeildatei.
Coding 1 = MACOS Roman, 2 = UTF16, 3 = UTF8

TICKS MSEC: waiting MSEC milliseconds Note: If a movie running in the background, the time measurement is not accurate than a frame rate of the movie. Msec <0 the keyboard are scanned and the first key stored with the reaction time during the waiting period. This button and the time can be retrieved using TKEY AND TTIME. To start RRTIME must be called beforehand necessarily.

TIMEOUT = TIME: sets the user entry timeout CONTROLS TIME in milliseconds. 0 = No timeout (Default)

ToPDF "path" (FLAGS) (SCALE) : Returns the window contents as a PDF file (optional on the printer) from.
path => the name of the PDF file.
FLAGS:
1 = horizontal, otherwise portrait
2 = not centered, otherwise centered on Paper
Show 4 = printer dialog, optional printer or print to PDF

SCALE => scaling factor 1 if not specified. (OSX only getting 1 on WIN32)

UDPO "URL " PORT: opens UPD connection to URL with the port PORT. URL can also "0.0.0.0 " be to be when received from an unknown address data.

UDPC: includes a UDP connection.

UDPW NUM: sends the ArrayBuffer NUM to the UDP URL.

UDPS "Text " : sends the text "Text " the UDP URL.

UDPR NUM: receives the ArrayBuffer NUM of the UDP URL.
Is NUM <0, the data in the SBUFFer number NUM are stored.

UDPN: is a numeric variable that contains the number of received UDP Bytes

UDPA: is a string variable and contains the address of the data has been received or the URL when opening.

USERID = "text": allows you to enter a VP marking eg number or name. **USERID** is then available as a string variable.

USERDATA: opens a box for entering various data subjects.
Then the variables are **USERID** (string) **age, gender** and **SCHOOL** (numerically) available.

VPD2EDF "vpdpath" "edfpath": converts the VPD file vpdpath in the EDF edfpath file

VALUE XP YP XS YS FC BC TC TIME SIZE TEXT BNUM CNUM: draws a numerical value of the size XS, YS at the position XP, YP with the colors of FC (Border Color) BC (background color) and TC (text color) of all TIME milliseconds the next indicative value. The text size is SIZE. TEXT is a formatting command to "C" format (eg "HR:% 3.0f". It has here a float (double) number will be accepted, if the number has to be represented as hexadecimal value, for example, the first character of the string a "§" be. . The system then skips in the display and the number can be formatted as hex (eg "§Binärwert: 0x% 04.4x") BNUM is a sequential number (1-16) Up to 16 values simultaneously be active. CNUM denotes the relative channel number of ONL START command. Here are negative numbers indicated the number refers to the corresponding BUFF number.

WAIT KEY: waits for any key or mouse click, movies and sound continue to run

SOUND WAIT NUM: waits until an audio file on channel NUM has ended

WAIT MOVIE: waiting until a Movie file is complete ,

WAIT VBL: waits until an image change occurs.

WCOPY NUM OPT: copies the currently active window and directs all drawing commands in this new window. However, the new window is still invisible and the former continues to be displayed. There can be up to 4 (NUM) copies.
OPT:
0 = it a new window is created copy.
1 = the window copy is deleted from the memory.
2 = window appears and s can be written to the screen.
3 = the window disappears, but it can also be written to the invisible shield
4 = The Main Window (Screen -1) appears again.
5 = The window on the 2nd screen (Screen -2) reappears.

With WCOPY image content can be exchanged extremely fast be (Tachistoskopie)

WEBVIEW XP YP XS YS "Url" : open a Web browser at the location xp, yp the size xs, ys and the URL "url" (currently only OSX)

WINDOW NUM XP YP XS YS "name" creates a graphics window to issue graphics commands (no UserIO) The window will remain after the completion of the script in the editor and can be saved as a JPEG file or printed either.
The text "name" is the headline of the window or the default filename when saving. All graphics commands are diverted to this command on the window. The output to other surfaces can be set

again with SCREEN.

WROLP: copies the user settings from NGRAPH and writes back a modified definition file.

WWAV PATH ABN1 ABN2: writes a WAV file of PATH from the two arrays ABN1 ABN2 (stereo channels) (only works if previously a WAV with Rwav loaded.

Excel interface reads and writes Excel compatible spreadsheets in .xls format with a maximum of 256 columns and 512 rows.
XLSC NUM: creates an Excel worksheet with NUM worksheets. Then Worksheet 1 is active.
XLSX NUM: switches to the worksheet number NUM
XLSW PATH: stores the Excel previously generated with XLSC Worksheet under the Path Path
XLSR PATH: reads an Excel worksheet with the path PATH
XLSN Name: sets the name of the worksheet on the worksheet NAME
XLSS LINE COL: returns the string of cell LINE, COLUMN a generated with XLSC Excel Worksheet, can also be written
XLSA LINE COL: stores the string in the cell LINE, COLUMN a generated with XLSC Excel Worksheet, The cell formatting string or number is set automatically.
XLSD LINE COL: returns the number of cell LINE, COLUMN a generated with XLSC Excel Worksheet, can also be written
XLSE LINE COL: deletes the cell LINE, COLUMN a generated with XLSC Excel Worksheet. Is LINE = 0, all cells are erased.
XLSM R : Returns the number of existing rows in Excel Matrix
XLSMC: returns number of existing columns of a matrix Excel
XLSMS: provides the number of worksheets
XLSCN : Returns the name of the active worksheet

Comments:
Audio files can be mp3 or wav format in AIFF. (OSX any QT compliant)
Image files must be in jpg or png format. (OSX any QT compliant)

Fastest pictures are marked if they have the same color resolution as the screen and are drawn in the original size.
Movie files can be in any format when the corresponding decoder is installed.

Item centered text output only up to max. 2 lines supported

The temporal resolution of the keyboard and mouse response time is about 10-12 ms in OSX USB.
Higher temporal resolutions of response times in the microsecond range can be reached by external Spezialtastaturen. (eg "trigger box" GM 2011)

The resolution of the time registration (TIME) is 1 millisecond (the internal timer works in microseconds (UTIME))

Strings:

Pathname are separated by "/". May drive names do not include a "/" at the beginning. Partial path names begin with "/"
The complete path is then calculated from this part of the path and with **DEFINE PATH** formed target path defined.
Begins the text of a Control Buttons with the "_" character of the text is interpreted as path to an image file and displays the appropriate image as a button.

Then starts a text with a "# .." character is an additional Numeric expression expected for the text and formatting instructions in the text itself (for "C" syntax)
example
TEXT "#The value is:% 8.2f" b1 XP YP ....
draws the text "The value is: 1:30" if b1 is 1.3

If the text only from the "#" character, the text is replaced by a text from the text defining the text index from the pool definition.

ENDLINE or **EOL** is the newline character (default = Carriage Return (13)) In a SAVE command, you can use this string as a newline character. This string can also be set, eg for compatible with Windows or DOS spending on CR + LF or UNIX or LINUX on LF.

EOL = "\ r \ n" is the end of the line to DOS format.
EOL = "\ n" is the end of the line on UNIX format.
EOL = "\ r" is the end of the line on old MAC format.

NUMSTR (FORM NUM) = returns the number of conversion (float) Number in NUM using the format strings FORM after "C" Convention eg "% 3.3f" generates a number with 3 digits before and 3 digits. **NUMSTR** ("auto" val) formats the number automatically, ie with small numbers more decimal places than in big ones.

NUMSTRX (FORM NUM) = returns the number of the transformation (long) number in NUM using the format strings FORM after "C" Convention eg "% 04X" produces a hexadecimal 4 digits.

MS2HMS (NUM) = returns the number conversion of NUM in milliseconds in HH: MM: SEC: MS

MS2SR (CHNUM MS) converts the time in milliseconds in the sampling of the channel with the number CHNUM.

SR2MS (CHNUM SR) converts the sample rate of the channel with the number CHNUM in milliseconds.

Strtok (STRING "d" NUM) = split the text string into individual components that the with String "d" are separated. NUM indicates the desired N-th sub-string. This allows, for example, decompose with spaces separate sentences into words. Is NUM 99 of the last sub-string is returned. Is NUM = 0, the number of all sub-strings is returned (as a number of text). Is NUM negative, the string is split into 2 parts, with -1 the part before the separator string in -2 the part after the separator string.

STRTOKL (PATH "d" IND NUM) = is a combination of the FILESTR and strtok command and breaks down the text from the line IND in the PATH file into individual components, which are separated by the character "d". NUM indicates the desired N-th sub-string. This allows, for example, decompose with spaces separate sentences into words. Is NUM 999, the last sub-string is returned. Is NUM = 0, the number of all sub-strings is returned (as a number of text).

CHNDEF (CHNUM) returns a string with some parameters of the definition file. CHNUM (1-80) is the channel number. A Defdatei must have been previously loaded.
The string contains separated by a colon:
Channel number, name, unit, Kanalgösse (W, B) sample rate, on / off status.

VPRSTAT calls from a Varioport recorder some status messages to grouping them separated by commas in the string. Previously, the port must be prepared with SET SPORT.
Everytime you use this string the recorder is re-queried.
Position:
1 = Run Status 0 = Idle, 1 = measurement runs
2 = Time Recorder
3 = definition file used
4 = Run Status as text
5 = total duration of the measurement that fits on the map
6 = remaining term of the measure until card full
7 = battery voltage
8 = Error Status
9 = Workload (average and maximum)

Numeric expressions:

Expressions are evaluated from left to right. (No mathematical hierarchy!)
supports +, -, *, /,%, + =, - =, * =, / =, and & (logical AND), | (logical OR) and ^ (logical EXCLUSIVE OR). Klammerverschachtelungen are possible. All parameters can be constants or:
b1 .. b1000 = is a floating point variable max. 1000 Buffer
BUFF [BX] = is the value of the buffer with the number in bx

In addition, you can enter a name buffers
DFB: "ACCU"
ACCU = 5
The alias has priority over other Defintionen! If, for example DFB: "SCRNXS" is defined is the predefined variable "SCRNXS" lost.
If the name Buffer number must be assigned to a specific buffer be specified:
DFB: "ACCU" 1
has the name of the buffer B1 to ACCU

a1 .. 100 = Is a float array of length that is set with SET ALEN ax.
ax [x] is the xth element in the array ax
Here, the zero [0] element is the instantaneous shift index.
The element [-1] gives the array length.
The element [-2] yields the Array Flags (also AFLG).
Arrays can = val are described in Index Mode ax (x) = val or in Shift mode ax. In Shift Mode each new element is stored in the next position of the array, the array is full is the position zero (ring buffer)
Arrays can be offset against each other only with the following operations:
ax = ay
ax + ay =
ax - = ay
ax * = ay
ax / ay =
and the sum of an array:
bx = SUMM (ax)
here all the elements of the array are added to each other

and the PWR of an array:
bx = PWR (ax)
here the sum of all absolute values of the array is calculated
SORT ax
Sort the array in ascending order
CLR ax
clears all elements of the array x

but ArrayBuffer can eg be divided by a number:
ax / bx =
here all the elements of the array are divided by the corresponding variable bx

You can also load an array with numerical entries:
a1 = 4 5 3 1

Additionally, you can give a name to the array buffers
DFA: "array"
Array [1] = 5

ArrayBuffer can be saved and loaded again:
b1 = (1 * 0x10000) +2
STORE "abuff1.abf" b1
RESTORE "abuff1.abf" b1
Wherein the 1 right after the bracket is the number of the arrays.

GETKEY = brings key or mouse (mouse button = 1) and waits
CHKKEY = brings key or mouse (mouse button = 1) and does not wait
MKEY = Status of the Modifier Keys:
256 = Shift key pressed (OSX and PC)
Pressed 512 = Control key (PC Right Ctrl)
1024 = Alt key pressed (OSX and PC)
2048 = Apple key pressed (PC Ctrl left)
3072 = Alt (PC)

SKEY2 = The status of the modem signals the 2nd serial ports. For example, external signals (eg connected keys) can be queried.
1 = CTS, DTR = 2, 4 = RING. The response time could be measured here significantly more accurate than using the keyboard.

CHKFILE "Path" = provides a '1' if the file "path" exists otherwise '0'
RAND (X) = gets a random value from 0-X, X <0 is a number of stock 0-X-1 values created, which can be then drawn with RAND (0). (Random without replacement, Lotto number principle)
TIME = time in milliseconds since the start of the program can also be set.
UTIME = time in microseconds.
RTIME = time in milliseconds between the last image or Itempräsentation and keystroke or mouse click
STIME = time in milliseconds from the time of the **OUTPUT** or **APPEND** command.
TTIME = time in milliseconds between the last RRTIME and one keystroke while TICKS - (msec)
TKEY = key code of the first key pressed after RRTIME during TICKS - (msec)
WIN32 (numeric variable on Windows => 1, otherwise 0)
AGE = age of the user input 0 = no input
GENDER = Gender of user input 1 = female 2 = male 0 = no input
SCRNXS = horizontal screen resolution (Pixel)
SCRNYS = vertical screen resolution (Pixel)
SCRNNUM = number of available screens
SCRNXS2 = horizontal screen resolution of the 2nd screen (Pixel)
SCRNYS2 = vertical screen resolution of the 2nd screen (Pixel)
SCRNXSMM = Resolution screen horizontally (in mm)
SCRNYSMM = Resolution screen vertically (in mm)
SCRNXS2MM = Resolution screen horizontally of the 2nd screen (in mm)
SCRNYS2MM = Resolution screen vertically the 2nd screen (in mm)

SCRNXP (x) = X yields the screen position of the first screen of the window position x

SCRNYP (y) = Gives the y screen position of the first screen of the window position y

Be SCRNFREQ 10 = start a DisplayLink service (OSX) or DirectDraw (WIN) has started before WAIT VBL)

SCRNFREQ 11 = stops a DisplayLink service (OSX) or DirectDraw (WIN)

SCRNFREQ 1 = determines the screen refresh rate of the connected monitor.

SCHOOL = education of user input 1 = Primary 2 = Secondary School 3 = School 0 = no input

CHAN [x] = instantaneous value of channel x (x is the relative number of the channel in the **START ONL** command, ie 1-8), can also be set to, for example, external trigger to put in a canal. The set value then replaces exactly one sample of the original channel with the new value.

VPDRC = Result of Varioport commands if> 0 then error

VPD_CNUM ("name") returns the number of the channel "name" if not present 0

VPD_OF [x] = channel offset, VPD_MF [x] = channel mulfac, VPD_DF [x] = channel divfac

VPD_YSCL [x] = num sets the rank of VPD channel

VPD_YTICKS [x] = num sets the Y tick intervals of a VPD channel

VPD_XTICKS [x] = num sets the X tick intervals of a VPD channel

VPD_CFLGS [x] = num sets flags of VPD channel (eg auto zero, etc)

VPD_CMFLAGS chan flags mask sets the flags (flags) of the channel chan though the mask is set. All bits that are not available on 1 in mask are not affected.

VPO_NM [x] returns the channel name of the online channel x (if enabled)

VPO_NMI ("name") returns the channel number of the online channel with the name "name" "if not available -1

HCHAN [x] = VAL sets the high byte of channel x to the value VAL. The low byte remains unchanged. The high byte is permanently set to the value VAL as long as is set to VAL = 0th

Pchan [x] = VAL sets the value of the channel x to the value VAL The channel is permanently set to the value VAL.

Results of statistical calculations are automatically calculated at each Biodatenerfassung:
AVG [x] = mean value of channel x
STD [x] = standard deviation of channel x
MIN [x] = minimum of channel x
MAX [x] = maximum of channel x
TND [x] = Trend channel x
INT [x] = mean value of the absolute values of channel x
NUM [x] = Number of values of channel x
MED [x] = Median of array x (only for array statistics ABSTATS)
If x is negative, the results of the calculated with ADDSTAT statistics are retrieved

CMAX [x] = If the number of exceedances of MAX value of channel x
CMIN [x] = If the number of value falling below MIN of channel x
(MAX and MIN must have been set with SET MAX MIN CMPVAL CHN)

SIN (X) = sine of X
ASIN (X) = Arc sine of X
COS (X) = cosine of X
TAN (X) = Tangent of X

LOG (X) = the logarithm of X (Pseudologarithmus according Myrtek)
LOGN (X) = logarithm of X naturlalis
FLOOR (X) = rounding down of X
CEIL (X) = rounding upwards of X

ABS (X) = absolute value X
SQRT (X) = root X

GXP [NUM] = X is the last position of the graph NUM
GYP [NUM] = Y is the last position of the graph NUM
GFLG [NUM] (can also be set) = are special options of graphs NUM
// Graph has reached the end
#define GENDFLG_ended 1
// And resets at the end
#define GENDFLG_reset 16
// Only the position is reset
#define GENDFLG_resetpos 32
// Graph stop at the end
#define GENDFLG_autostop 64
// Graph is not drawn
#define GENDFLG_suspend 128
// But is not drawn graph simulated (pointers are valid)
#define GENDFLG_simulate 256

GYSCL [NUM] = is the scaling of the graph NUM (To consult the geänderteren user values from NGRAPH)
GIFLG [NUM] = is the flag setting of the graph NUM (To consult the geänderteren user values from NGRAPH)

Goffs [NUM] (can also be set) = is the optional offset the graph NUM

GCHAN [NUM] = is the channel displayed on the graph NUM (can only be set)

RECSTAT = provides 1 when a previously polled Varioport recorder collects data, or 0 if the data collected is not started.

STRNUM (STRING) = returns the number conversion of a string

SOUND (NUM) = provides one audio channel when the NUM playing a sound, zero otherwise.

SPRITE NUM XP YP = returns one if the sprite with the number NUM collides with the position XP, YP

MOUSX, MOUSY, MOUSB provides the x, y position of the mouse, as well as the state of the mouse button.

JOYX, JoyY, Joyz, JoyV, JOYB provides the axes and buttons of a connected joystick.

AutoX = supplies a flag (0 or 1) whether the corresponding auto variable (X = 1 -...) was carried out. The flag is cleared by this call.

bx = **CONTROLS** displays the predefined controls and expects user input. The returned value is the code value of the key pressed.

POOL = the respective next value from a pool. POOL1 to pool4 returns the respective value from the appropriate pool.

WEIGHT = the weight of the last Itemantwort.

RTRIG = is one when an ECG R-wave is detected. Must then be reset to zero.
Mrate = sets the playback rate of the video (0 = stop).
MTIME = is since the start of a video elapsed time in milliseconds.
MOVIE = 1 when a movie is played, otherwise 0
MOVSIZ "path" xbnum ybnum determines the size of the movie "path" in the buffers with the numbers xNum, ybnum.

FNUM "text" XP YP INPUT generates an input box for inputting numbers with the heading "text" at the position XP and YP and INPUT the default value.

String variables:

USERID: returns the userid of USERDATA
PATH: provides the default path (must be set with the DEFINE PATH)
HPATH: returns the path of the V erzeichnisses active TCL file.
DIRATH: is the directory that is selected in the SELECT function first. If dirpath not set the operating system takes each to previously used path.

SELECT NUM ("name") **(heading): Select 0 shows a Dateiladebox and returns the path name of the selected file.** Select 1 "name" is a Dateispeicherbox the default name "name" and returns the path name of the selected file.
Select 2 shows a Ordnerauswahlbox and returns the path of the selected folder.
Optionally, a text are given standing as a headline above the box.
With the string variable FEXTENS , a file extension to specify that only files with the specified extension allowed for the selection of files with SELECT. (otherwise on the empty string translated)

sx: returns a StringBuffer (x => 1 ... 100)
"#": Returns the next string pool (only from the pool or POOL1)
"#text% f" NUM: returns the text without the number sign but with the formatted according to "C" number NUM convention.
"text": is the string "text"
DTEXT [NUM]: is the text with the number NUM (1..N) from the DEFINE TEXT Definition List. Then WEIGHT provides the corresponding text weight. The next text from Poolx can be drawn with DTEXT [Poolx].

CTIME provides the time of day as a string
CDATE returns the date as a string

TAB is the tab character
LF is the line feed character
CR is the Carriage Return character

FILESTR PATH INDEX: returns the text line with the number INDEX from the PATH file
Stands in the file in the 1st line S01, so the text is read in segments separated by S01, S02 etc.
If INDEX = 0 the entire file is loaded.
provides for INDEX = 999999 the number of rows returned as text

Constants:

If x and y positions can be specified also -1 for top or left quadrant, -2 for medium Quadrant and -3 for right or lower quadrant are indicated. Example -2 for xp and yp -1 for centered entering the top center, -2, -2 centered in the middle of the screen

in pictures and movies
If you specify for xs = 0, the object is scaled to full screen, at xs = -1, the object is shown in its original size, at -2 in duplicate original size, etc. Otherwise, it is scaled to the specified xs and ys values

Colour codes can be specified as constants rather than numbers:

Black = C_BK
White = C_WH1 - C_WH8
Red = C_RD1 - C_RD8
Green = C_GN1 - C_GN8
Blue = C_BL1 - C_BL8
Cyan = C_CY1 - C_CY8
Magenta = C_MG1 - C_MG8
Yellow = C_YE1 - C_YE8

Appendix:

Typical structure of a program:

```
// Optionally, the program can only run in a window.
// If SET SCREEN is omitted, the entire screen is used
// This command creates a window with 1024 x 768 pixels at point x = 50, y = 50
SET SCREEN 1024 768 50 50 "test program"
// A new white screen is generated
NEW SCREEN

// Set the path to the folder where the script is located
DEFINE PATH "home"
// Bind a global Macros
INCLUDE "/glob_macros.tcl"

// Predefinition of positions for Macros
// Control x size in pixels
CXS = 200
// Control y size in pixels
Cys = 30
// Control x position => the constant -2 here means middle of the screen
CXP = -2
// Control y position in pixels
CYP = 500
// Text Size
ccs = 16
// Distance between the buttons in the Y direction
CDY = 10

// Create a button with the text "Start" the return value is 1 and the key => 32 => code for Spacebar
Button "Start" 1 32
// Create a button with the text "Exit" the return value 99 and the key => 0 => undefined
"Exit" button 99 0

// Ask user interaction and branch-off
: M loop
```

```
// At code = 1> Start button Go to Label Home
// At code 99 => Exile button Go to Label end
ongoto CONTROLS 1: 99 Start: End
loop: M loop

// Here the program ends when you press Start
:begin
BEEP 1
loop: M loop


:end
// Terminate the program
END
```

Special considerations of the Windows installation:

uVariotest for Windows has the same folder structure as a Max OSX application. (Folder with suffix .app) and is installed as follows:

a) create a folder uVariotest eg on drive C.
b) put uVariotest.app in this folder. The folder structure of uVariotest.app may be changed under any circumstances.
c) Creating a shortcut for uVariotest.app/Contents/Windows/ uVariotest.exe for example on the desktop. Best put the folder with the programs such as EDA also in the uVariotest folder.

Programs that use a second screen _require that the 2nd screen in Windows is defined as "Extended desktop" and the 2nd screen is set up with the top right defined on the 1. screen. (In OSX positioning the 2nd screen is not relevant)

General Installation Instructions:

When you start programs with uVariotest to the setting of paths think (DEFINE PATH) The easiest way here is "home"

For using Varioport or USB8k hardware necessary driver of the serial interfaces must be installed eg FTDIChip or trip Lite driver. In the uVariotest programs then you have the appropriate COM port and baud rate can be set.

Some programs use the gamepad Logitech G13. This also required the driver must be installed and then the key must be defined. Standard definition used for many experiments at the Institute is as follows:

G1-G7: 1,3,4,5,6,7,9
G8 G14: 1,2,3,4,5,6,7,8
G15-G19: 3,4,5,6,7
G20-G22: 4,5,6
and 0 for the left button of the joystick (Next button)

Autostart function:

If necessary, you can convert a TCL program in a standalone application.
These can then be started directly by double-clicking.
For this one duplicates the application uVariotest and renames it to a meaningful name of the new application. Then you start this renamed uVariotest and loads the desired TCL script. Now you call the menu command "Auto Start" and to demand the script is transformed into an application. After completion and restart the TCL script will now be executed directly. Optionally you can take necessary auxiliary files in the application by being copied into the application folder. (OSX => Show Package Contents) To access these help files must be set with the DEFINE PATH global path "package".