

gwCalc Manual

Gerhard Wanderer
gwCalc@wanderer-gw.de

Contents

1	Introduction	2
2	Main Screen	3
2.1	Function Keypad	5
2.2	Basic Keys	6
2.3	Special Characters	6
2.4	Special Keys	7
2.4.1	User Defined Functions	7
2.4.2	Result History	9
2.4.3	Variables	10
2.4.4	Command History	11
2.4.5	Configuration Dialog	12
2.4.6	Unit Conversion	13
3	Notes On Build-In Functions	15

1 Introduction

gwCalc is a easy to use scientific pocket calculator for the Zaurus 5xxx family. gwCalc has two types of history lists, the result history and the command history. The command history lets you easily correct typos in formulas or recalculated formulas with changes. With the result history it is easy to user older results in new formulas.

gwCalc has all basic scientific functions build in, but you can also define your own functions.

gwCalc takes care for the mathematical rules for the operator (“+”, “-”, “*” “/”) hierarchy. Brackets can be used to modify this hierarchy. Brackets are also needed to specify the arguments to functions like sine. Missing closing brackets are automatically added at the end of the command line.

Point (“.”) and coma (“,”) are exchangeable for numeric input.

Blank characters between numbers, variables and functions are interpreted as multiplication:

34 sin(2.3) is equivalent to 34*sin(2.3)

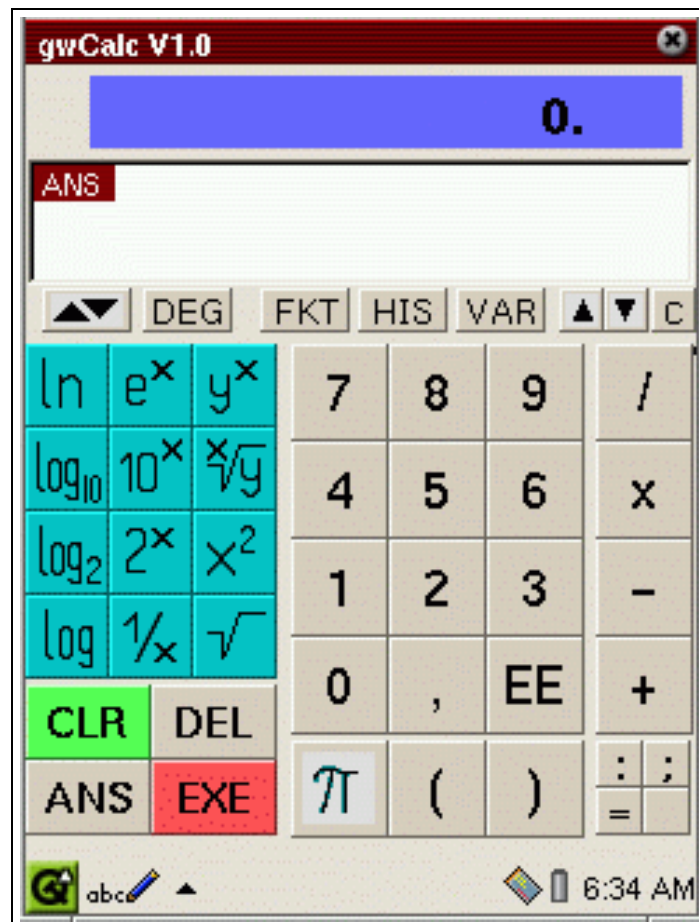


Figure 1: Primary Screen of gwCalc

2 Main Screen

Figure 2 shows the gwCalc main screen. At the top the result display is placed. Left hand of the result display the key status of the “INV” and “HYP” keys is displayed, which are used to select the trigonometric functions (figure 3).

The result display can be configured for “simple number” display or for “engineering” display. If “engineering” is selected the exponent is always a multiple of 3 and the mantissa is divided in blocks of 3 digits for better readability.

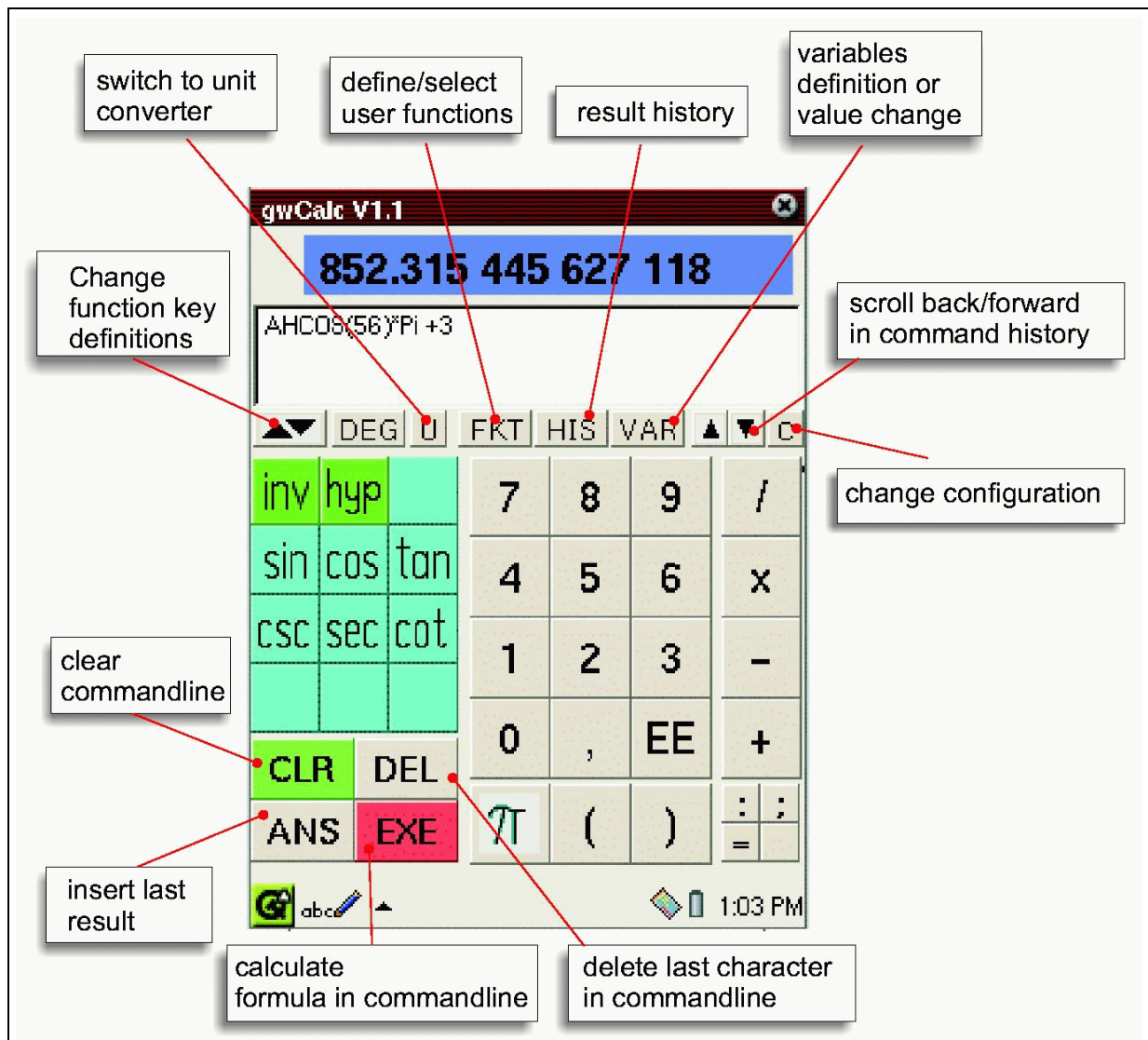


Figure 2: Function Keys

In the command field the formulas are typed in. You can use the keypad for easy and fast formula entry but also all types of zaurus keypads. The function names are compatible to QPLOT, so it is also possible to use the keypad of QPLOT for formula entry.

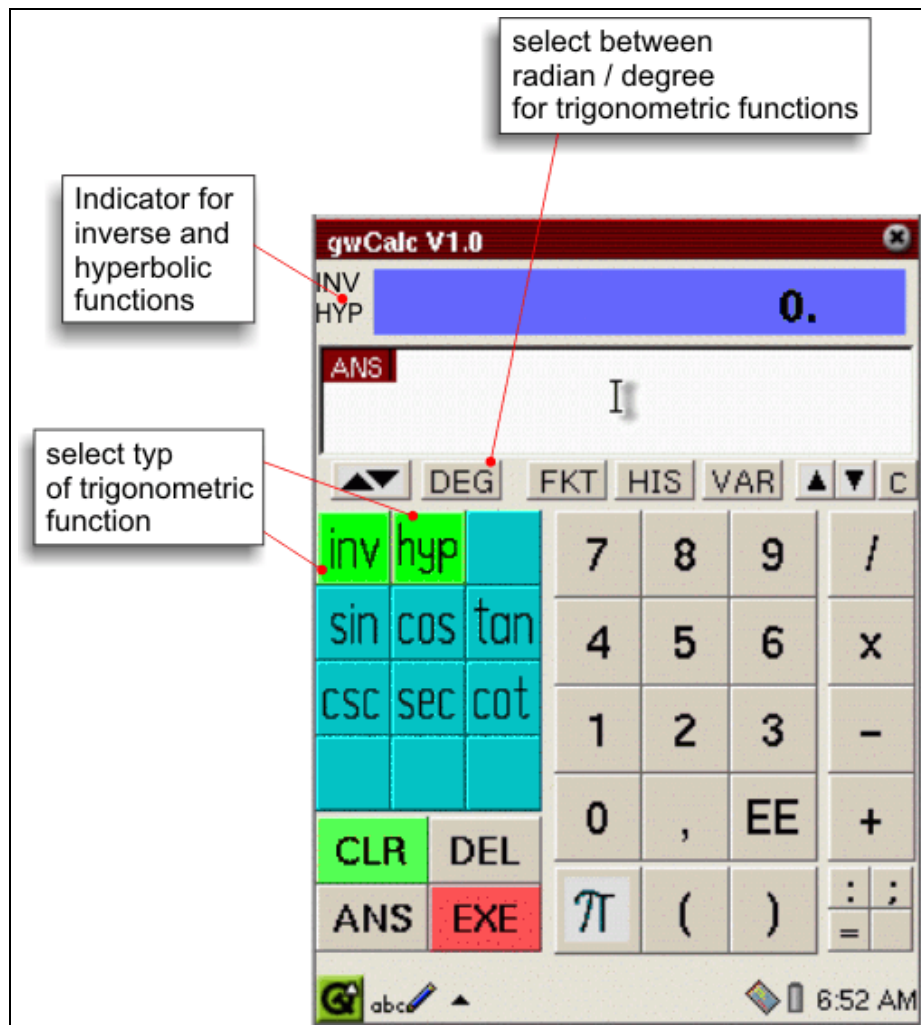



Figure 3: Trigonometric Functions


2.1 Function Keypad


The function keypad has three layers. By typing on the key with up/down arrow you toggle between the layers. The first layer contains all the exponent and logarithmic functions, the second layer all trigonometric functions and the third layer a few special functions.


The key $\boxed{\text{RAD}}$ or $\boxed{\text{RAD}}$ changes the behavior of the trigonometric functions. If $\boxed{\text{RAD}}$ is displayed, all angles are interpreted or output in radians. If $\boxed{\text{DEG}}$ is displayed, all angles are interpreted in degrees.

2.2 Basic Keys






Below the function keypad some editing keys are located. With  the command line can be cleared.

 removes the character left to the cursor from the command line.

 puts the result of the last calculation into the command line at the current cursor position. ANS is an internal variable which is set after each calculation to the calculated result.

 performs the calculation of the formula given in the command line.

2.3 Special Characters

Below the numeric keypad on the right side a few special keys are located (, ,  and ). These keys are needed to define variables and functions via the command line.  is a blank character to separate parts of the formula.

2.4 Special Keys

The special keys above the keypad display pop-ups for variable-, function- or history-editing.

2.4.1 User Defined Functions

The key **[FKT]** opens the functions editor.

In the functions editor you can add, delete or modify the user defined functions.

In the parameter list you define the names of local variables used in the function formula. The values for the variables are set from the parameters of the function in the sequence, in which they appear in this list.

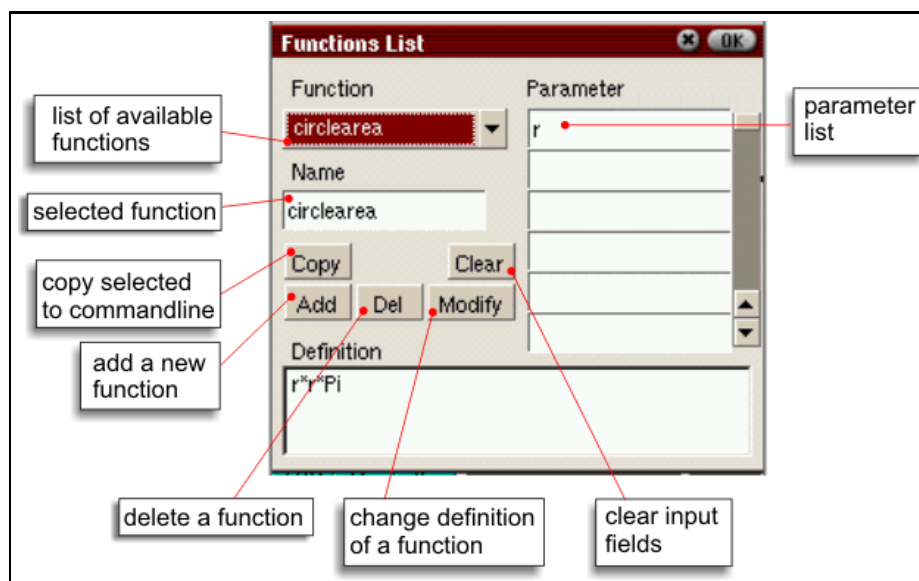


Figure 4: Functions Editor

To define a new function you fill in the parameter list, type a name in the “name” field and type in the function definition in the definition field. Whenever a parameter to the function is needed type the name of one from the parameter list.

By selecting the **[ADD]** button the definition will be added to the list of existing functions. You can now use this function like the build in functions.

To modify a existing function select the function in the function list. The parameter list and the definition will be displayed. After change of the parameter list and/or the definition press the **[modify]** button.

Doc.No.: gwCalc V1.0
Issue: 1 Rev.: -
Date: 28.8.2003
Page: 8

Function definition is also possible in the command line. To define a new function type

```
function(param1;param2;...):=definition
```

Example: to define the formula for calculating the area of a rectangle type in the command line

```
area(H;B):=H*B
```

by executing the formula

```
area(5;7)
```

you get the expected result 35.

Function names are case sensitive. You may not use names of internal functions for your own functions.

If you want to create a new function you can empty all fields in the function editor by the button.

The button puts the selected function into the command line at the current cursor position. Therefore you can give your functions descriptive names without the need of always typing long names.

2.4.2 Result History

The key **HIS** opens the result history. The result history shows the results of the last calculations, the latest at the end of the list.

You can select one of the results. After closing the history list by the **OK** button this value is copied to the command line at the current cursor position.



Figure 5: Result History

2.4.3 Variables

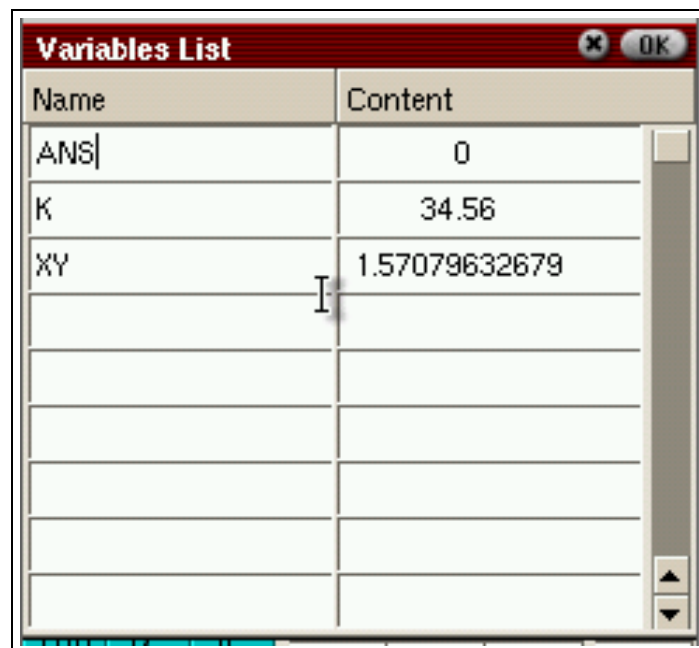
In gwCalc you can create up to 100 variables with arbitrary names. A variable can be easily created, simply by assigning a formula.

$X = \sin(2.)$

assigns the result of “sin(2.)” to the variable “X”. “X” will be created, if it doesn’t already exist.

Variable names are case sensitive. You may not use names of internal functions for variables.

You can see the list of you variables by pressing the **VAR** button.



Name	Content
ANS	0
K	34.56
XY	1.57079632679

Figure 6: Variables List

In this list you can modify the names of the variables, modify the content of a variable and add new variables simply by changing the values in the table.

The changes take place only if you leave this dialog by the **OK** button, otherwise the changes are discarded.

Doc.No.: gwCalc V1.0
Issue: 1 Rev.: -
Date: 28.8.2003
Page: 11

2.4.4 Command History

gwCalc remembers the last typed commandlines. You can scroll through the command history by the up and down keys at the right side of the VAR button.

Command lines, which resulted in a calculation error are not put to the command history.

2.4.5 Configuration Dialog


The button  pops up the configuration dialog.



Figure 7: Configuration Dialog

The result display can be switched between standard display, similar to that found on pocket calculators and an engineering display, which make numbers more readable.

“Auto ANS” selects, whether the string “ANS” is inserted in the command line after execution of a formula. If you often calculate the next formula based on the result of the previous one, switch this to “On”.

“Auto Highlight” has only meaning, if “Auto ANS” is set to “On”. In this case you can determine, whether the string is highlighted, so that it is automatically erased, if you begin typing without setting the cursor.

2.4.6 Unit Conversion

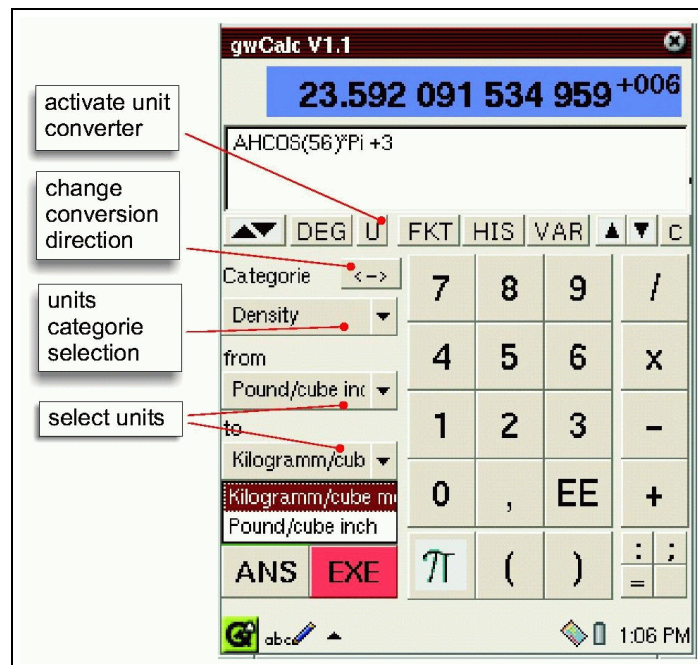


Figure 8: Unit conversion

By tapping the **U** button the unit function keypad is replaced by the units dialog. First select the units categorie from the categorie list. Then select the type of units to convert and type of units you want to have. With the button **<->** the conversion direction can be changed.

The unit conversion is executed *after* solving the formula in the command field by tapping the **EXE** button.

The unit conversion is based on the configuration file *gwcalc.units*, which is placed in the directory *opt/QtPalmtop/apps/Applications/gwCalc*.

In this file is each categorie introduced by the categorie name in brackets *[/]*. The first line after the categorie name is the reference unit, which has a conversion faktor of 1.0 and an offset of 0. The following lines describe the additional known units, each on a separate line.

Each description line consists of the conversion factor (from the reference unit to the actual unit), the conversion offset (from reference unit to the actual unit) and the name of the unit introduced by the character “:”.

Excerpt of configuration file:

[Distance] -----		
1.0	0.0	:Meter
10.	0.	:Dezimeter
100.	0.	:Centimeter
1000.	0.	:Millimeter
39.370078740157	0.	:Inch
3.281	0.	:Foot
[Area] -----		
1.0	0.0	:Square Meter
1.0e6	0.0	:Square Millimeter
1550.0031000062	0.0	:Square Inch
1.e-4	0.0	:Hektar
0.01	0.0	:Ar
[Temperature] -----		
1.0	0.0	:Celsius
1.0	+273.15	:Kelvin
1.8	32.	:Fahrenheit
...		

3 Notes On Build-In Functions

There are two functions, which need more than one parameter: ATAN2 and LOG.

The ATAN2 function calculates the arc tangent of the two variables x and y. It is similar to calculating the arc tangent of y / x , except that the signs of both arguments are used to determine the quadrant of the result.

`atan2(y;x)`

LOG calculates the logarithm to a arbitrary basis.

`log(basis; y)`