linSmith

John Coppens
ON6JC/LW3HAZ
john<at>jcoppens.com

August 21, 2008

1 Introduction

1.1 History

The Smith chart is one of the most useful tools to respresent transmission line problems and adaptation in general. As useful as the tool is, at least as frustrating is trying to use it in documentation.

Of course it is possible to use existing tools like general-purpose drawing programs, and, even better, some vector-based drawing programs (like Skencil, Inkscape and others). But trying to get a more or less complicated Smith chart problem to look nice is a nightmare. All these programs are thought out for a Cartesian axis system, and trying to do something not even really polar is - at least - very time-consuming.

I needed a tool like this for my classes at the University, and started the project while still in my 'Windows period' in 1997, as first steps in graphics programming using Delphi (version 1!).

At about the same time, due to a few very spectacular computer crashes and data losses, and my reluctance to base my class material on a mixture of commercial packages I couldn't really afford, I switched to Linux. Never looked back.

Then finally, around 2004, the Smith chart documentation problem re-appeared and I decided to convert the (very basic) Delphi version into Linux. A large undertaking, considering I didn't have that much graphics experience using GTK.

`linsmith` continues to be developed, and regularly new ideas (some mine, many from users) appear that make it more useful.

1.2 Tools

I had GTK+ 2 installed, together with Glade2, so I based everything on that. I didn't use any special graph libraries - maybe that's an error, but I didn't think the program's (lack of) complexity justified it.

Most, if not all, glyphs and graphics were done with the GIMP.

The gnome-print library solved many of the problems for Postscript output in a fairly simple way. It would be nice to have an option there to output Encapsulated Postscript directly - in fact that would only involve reducing the gnome-print output.

2 License

This program is (c) 1997- John Coppens. It has been released under the General Public License.

Here's the official GPL introduction (The complete text is distributed with this package):

> "This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA. "

3 Starting `linSmith`

Please read the installation instructions in the `INSTALL` file in the distribution if you never compiled and installed a program under Linux before. There are no special tricks to perform though, installation should be painless.

Once installed, you can start the program simply with `linsmith`. The first screen that appears is the main window with a splash screen instead of the chart:

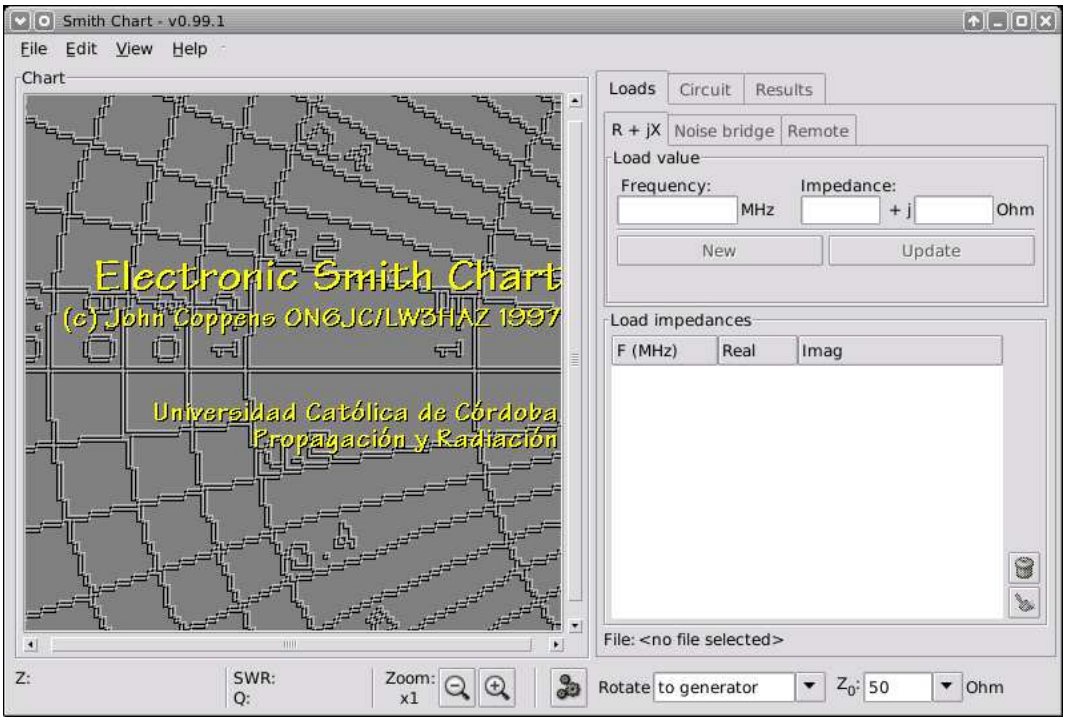


Figure 1: Initial `linSmith` screen

When the cursor is moved over the chart window at the left, the splash image disappears and the regular chart is drawn. (This also happens when either a load file or an element file is loaded)

Then, the screen looks like this:

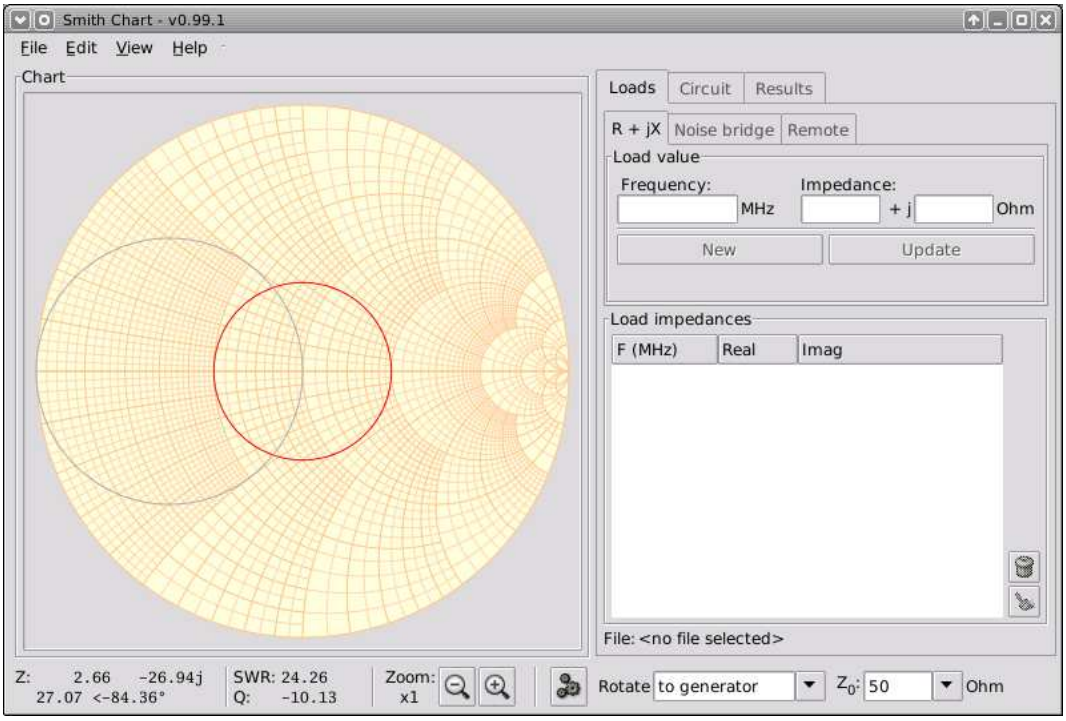


Figure 2: `linSmith` in operating mode

4 Setting up `linSmith`

When recently installed, linSmith will use the preferences shipped with the program. Of course those are mine, and tastes differ. There are so many customizable parameters in `linSmith`, you might get lost... (I did - in more than one occasion). You can always use the values of the screenshots below to return to a workable situation.

There are two clients for the graphical part: the screen, and Postscript output. As these are quite different media, with distinct requirements, I've made most parameters separately configurable. In fact, the configuration part may well be the most complicated section of the program.

A typical example of different requirements: on paper colors are generally much darker than on the screen, so normally, one would select lighter colors to make them easier to distinguish. Note that colors for screen use have the 'alpha' parameter accesible in `Preferences`. If ever some graphic element is missing (an arc, for example), check in preferences if the 'alpha' value is non-zero (max corresponds to opaque).

Screen hardware makes it difficult to represent half-pixels, so the linewidth requirements are different too (even the 'units' are different).

Note that many of the parameters are *not* instantaneous - it may be necessary to restart the program (this is not necessary for printing - printing parameters will be active immediately).

As usual with GTK programs, a configuration file will be generated (in this case in your home directory) as `'~/.gnome-2/linsmith'`. If you wish, you can copy this file to other users.

4.1 The 'General' parameter page

Here, you can control the showing of the SWR circle, and its diameter. Also, it's possible to enable an auxiliary circle which represents the image of the R=1 circle. This is a mainly didactic tool.

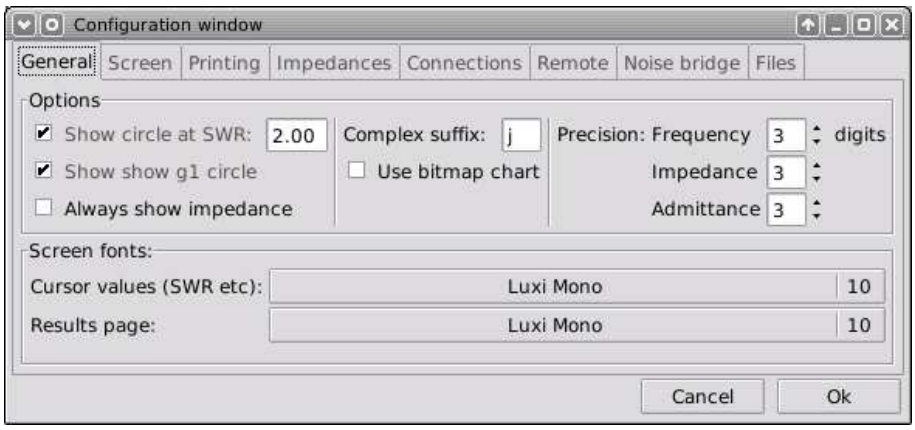


Figure 3: General parameters

If you prefer to use another letter for the imaginary part of complex numbers, change 'Complex suffix'.

Normally you will not want to change the files section, which represents the paths to the lastly loaded loads and elements files, and the last output file.

Show circle at SWR: Show a n SWR-circle in the center, with the indicated radius

Show g1 circle: I like to use the image of the R=1 circle for adapting.

Always show impedance: This forces the Results page to always use impedance for the circuit values. If unchecked, values will be impedance or admittance, according to the Z/Y conversion blocks inserted in the elements.

Complex suffix: Some prefer 'i' instead of 'j'.

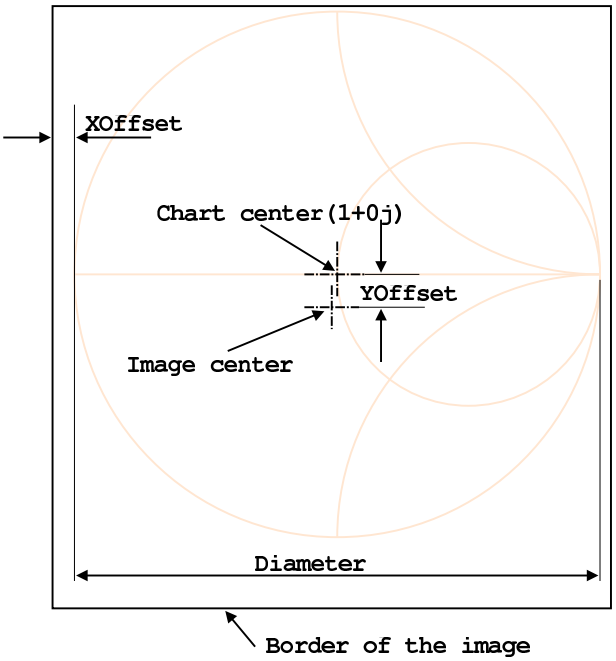
Use bitmap chart: For the moment, this is the only option, but I plan to generate a vector based chart in a future version, for better on-screen scaling. This has no influence on the Postscript output.

Precision: Determines the number of digits of precision for frequency, impedance and admittance on the Results page and the results export.

Screen fonts: 'Cursor values' modifies the font of the values under the chart, 'Results page' modifies the font on the Results page.

4.2 Screen parameters

On the screen, a bitmapped or a vector-based background image is used, according to the selection on the 'General' tab. On this page, it's possible to change the parameters of the background image. Please note that changes only take effect after a program restart.



The background chart is installed in the default `pixmaps` directory, the location of which may vary on your machine. To avoid an excessively large archive, I've included only one (smallish) chart. If there is interest, I have a few larger charts, or, you can even generate your own, printing a blank chart with `linsmith`, and then converting it to a bitmap (using eg. `convert` or `Gimp`).

Figure 5: Adjusting a new chart image

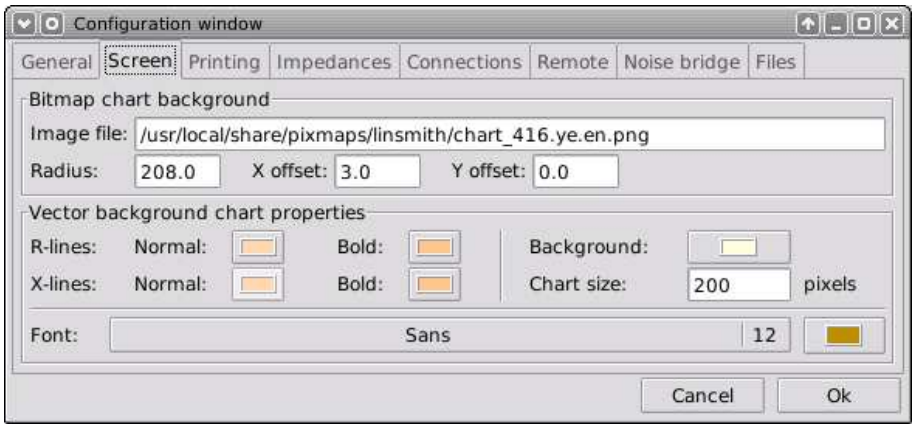


Figure 4: Screen parameters

For the Bitmap chart:

Image file specifies a bitmap to be used as background. Most image formats are supported. An image size of about 200-300 pixels on each side is best.

To adapt the program to a new background image, it is necessary to specify a few parameters:

Field	Description	Unit	Example
Radius	Distance from $R = 0.0$ to $R = \infty$	pixels	341.0
X offset	Distance from the left border of the image to $R = 0.0$	pixels	5.0
Y offset	Offset from the center of the image to the $X = 0.0$ line	pixels	0.0

For the vector chart:

To improve the aspect of the chart, you will probably want to edit the default colors of the screen chart in 'Preferences'. Though I'm sure this is a matter of personal taste, these are the colors I've used for the screen shots and the manual (on the 'Screen' tab of Preferences):

		Hue	Saturation	Value
R-lines and X-lines	'Normal'	30	30	100
	'Bold'	30	45	100
Background color		58	13	100

The initial size of the vector chart can be set on the same Preferences page. 200 pixels will almost fill the space of the startup screen.

Note: The font definition is not used yet!

4.3 Printing

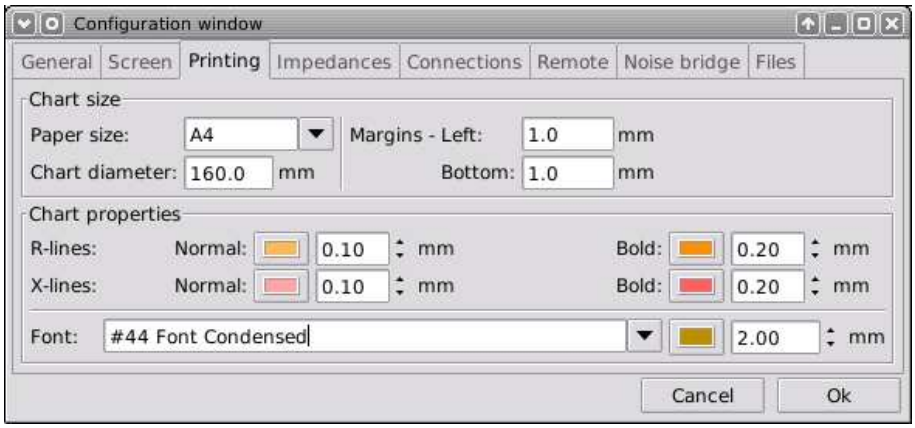


Figure 6: Printing setup

These are parameters specifically related to the Postscript output of the program:

Field	Description	Unit	Example
Paper size	Selectable		A4
Chart diameter	Output size of the Smith chart	mm	160.0
Margin left	Moves the chart to the right	mm	1.0
Margin bottom	Moves the chart up	mm	1.0

Notes:

- The paper size is not actually used, but must be sent to the generating routines.
- Margin left and bottom should probably be set small, so that a later conversion to EPS (Encapsulated Postscript) doesn't cause problems. Some converters leave lots of white space, or get completely lost trying to determine the 'Bounding box' that contains the chart.
  I had good results with 'convert' (which is probably on all Unix machines anyway):
  `convert mychart.ps EPS:mychart.eps`

With the chart properties list, it is possible to determine the colors of the background chart. Note that inkjet printers do not cope very well with non-primary colors (primary colors are Cyan/Magenta/Yellow/Black) and very narrow lines, as they have to be simulated using dithering.

4.4 Impedances

Here, the colors of the 'dots' can be selected. On the screen, the dots will have solid colors if they are mobile (can be dragged with the mouse). And they will have a black border if they are fixed (undragable).

Note: This is (still) a feature on the TO-DO list.

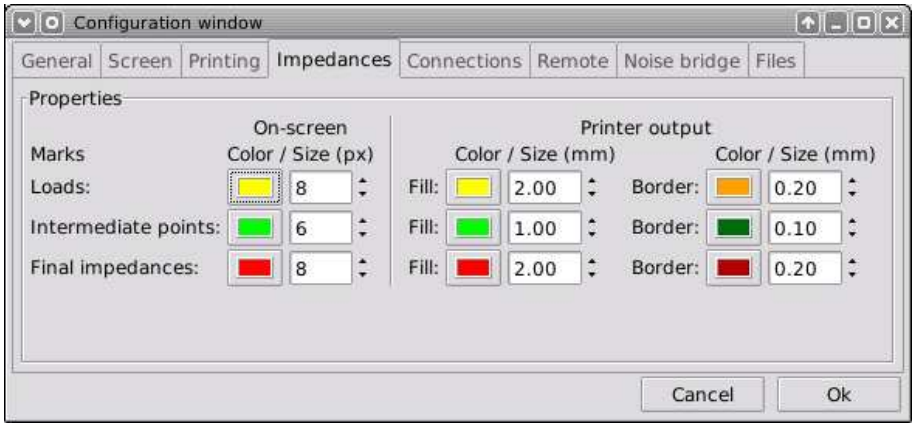


Figure 7: Impedance plotting

On the Postscript output, both border and fill color can be defined, as well as the diameter, and border linewidth. Using a darker version of the same color as border gives a nice touch to the graph.

4.5 Connections

In a similar way, this page permits setting up the color for the connecting arcs and lines, and for the SWR and g=1 circles.

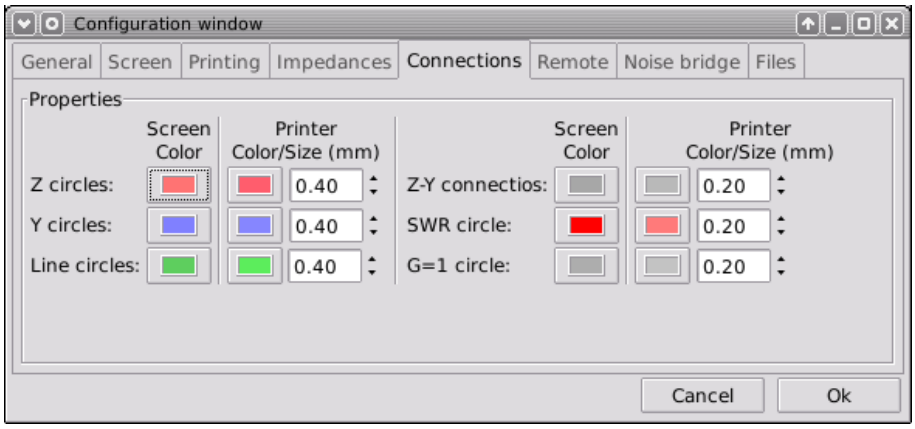


Figure 8: Connections

I'm ambivalent about showing a connecting line for the transformer... Haven't done so yet.

4.6 Remote

The remote tab permits defining some parameters of the remote instrument interface. This is work in progress, and may change (completely) over the coming versions if the actual approach doesn't seem appropiate.

For the moment, I've planned communications using pipes or using the serial port.

4.6.1 Pipes

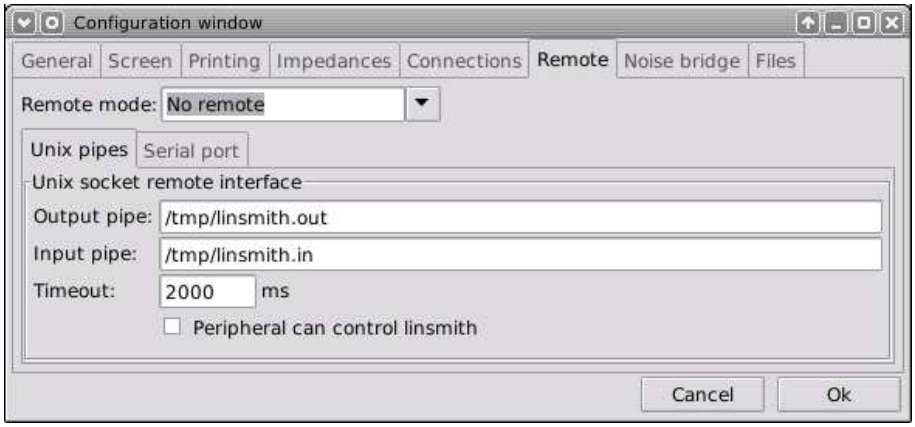


Figure 9: The `Remote` control tab for pipes

Input pipe and Output pipe are two files that should exist only during the run of `linsmith`. If the program is exited correctly, those files will be deleted at the end of the execution. 'Input' and 'Output' are as seen from linsmith. An instrument should write to 'Input' and read from 'Output'.

Typical names for these files are `/tmp/linsmith.in` and `/tmp/linsmith.out`.

When '`Peripheral can control linsmith`' is enabled, linsmith will accept the load values from the instrument with prompting, and some extra commands to control the load list.

For more information on remote control, check section 6 on page 13.

4.6.2 Serial communications

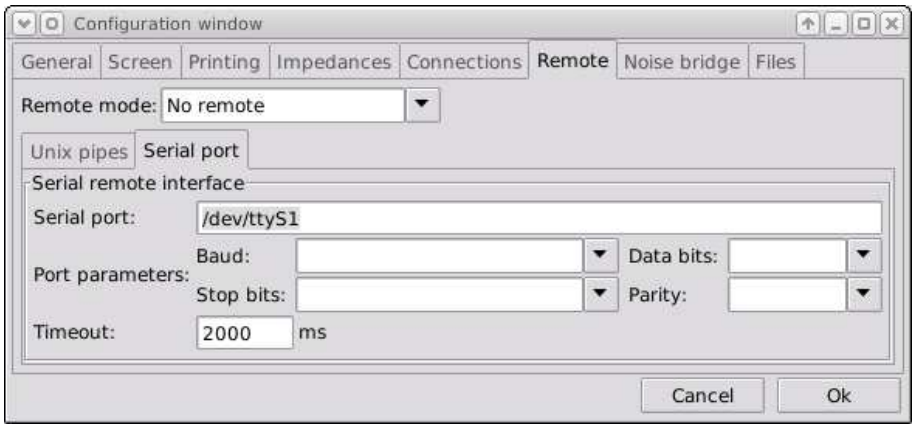


Figure 10: The `Remote` control tab for serial communications

This is a somewhat simpler interface, but needs configuration of parameters such as baudrate, number of bits, port, etc.

4.7 Noise bridge

Many variations of noise bridges are in use. Not only change the parameters (components used for construction), but there are quite a few different configurations too. The one I constructed a long time ago, was the circuit from the Feb 1977 issue of Ham Radio, by W8BXI. It uses a parallel combination of a potentiometer and capacitor, instead of the more traditional series configuration.

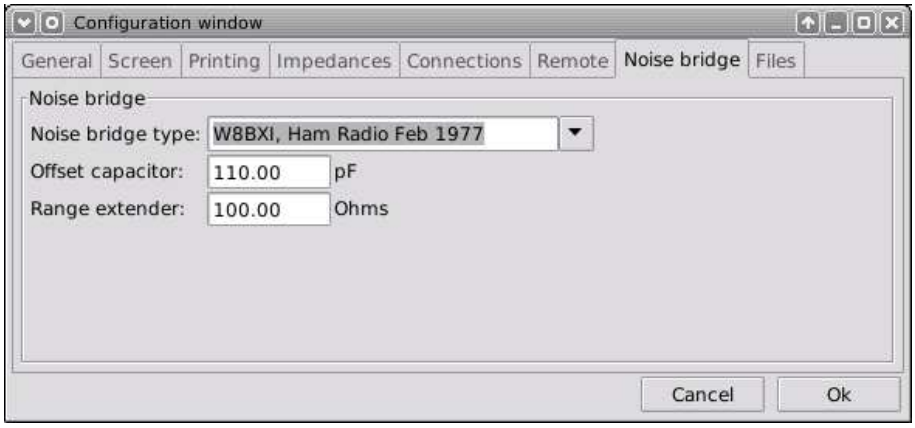


Figure 11: The `Noise bridge` setup tab

The values to be configured are the offset capacitor (the capacitor in parallel with the unknown impedance), and, if needed the value of the resistor used in the extender.

If there is interest in other configurations, please e-mail me.

4.8 Files

This is more of an 'informational' tab: it shows the names of the most recently used files.

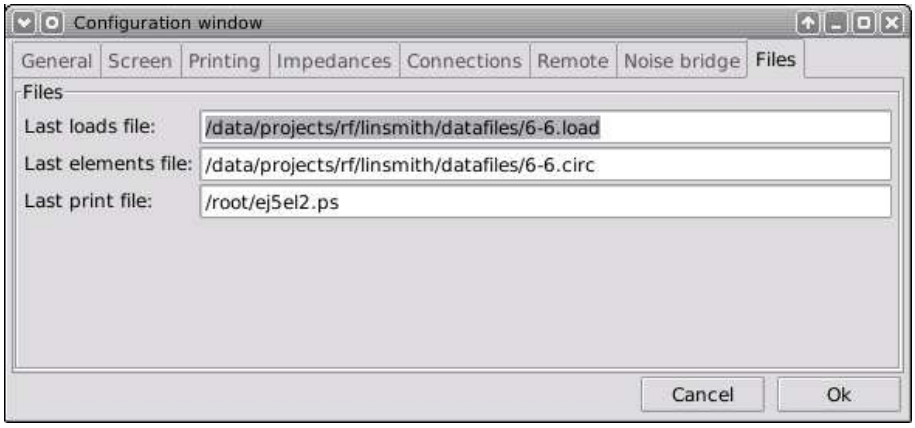


Figure 12: The `Files` tab

5 Using the program

5.1 Inputting the Loads

Inputting the load impedances is logically the first step to do.

Loads can be entered as $R + jX$ impedances, or as R/C pairs from a noise bridge.

5.1.1 R+jX impedances

To add a new load, simply enter the values - frequency (in MHz), R and X in Ohms. Click on `New` button will transfer the value to the load list. Since version 0.99.6, the input routine checks if the three values are present, and if the frequency is different from 0.

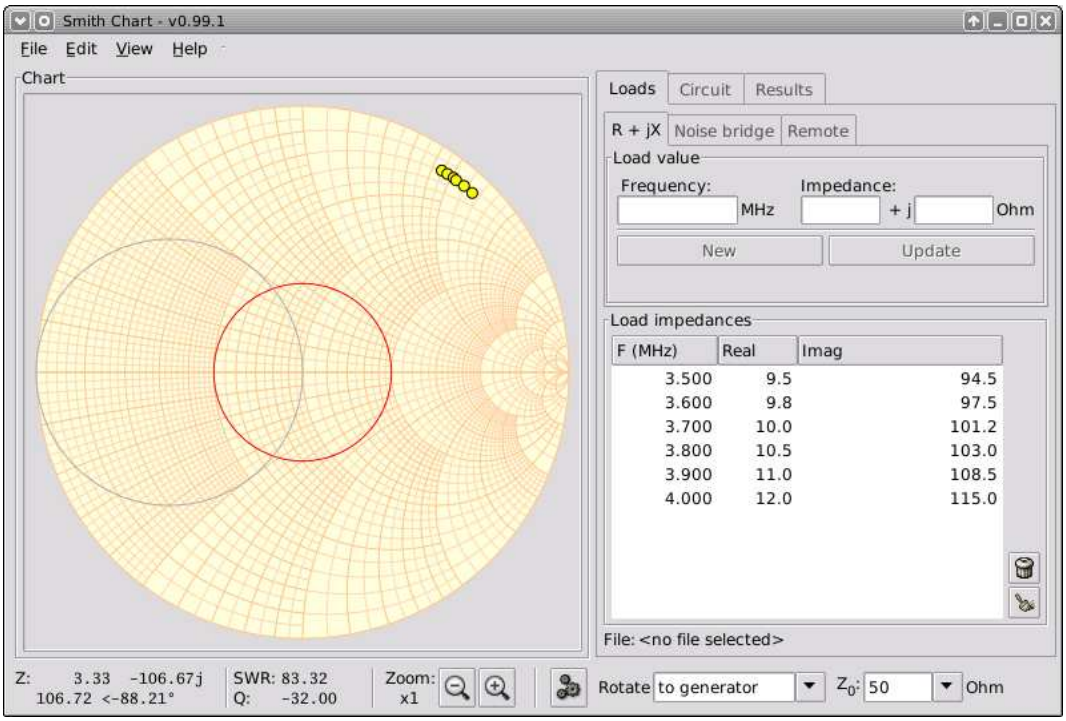


Figure 13: Entering loads as R + jX values

After selecting a line in the load list, it's possible to click on the trash can to delete that load from the list, or to edit its value and the click on `Update` to update the list's value. The clear button under the trash can empties the list completely.

5.1.2 R/C measurements

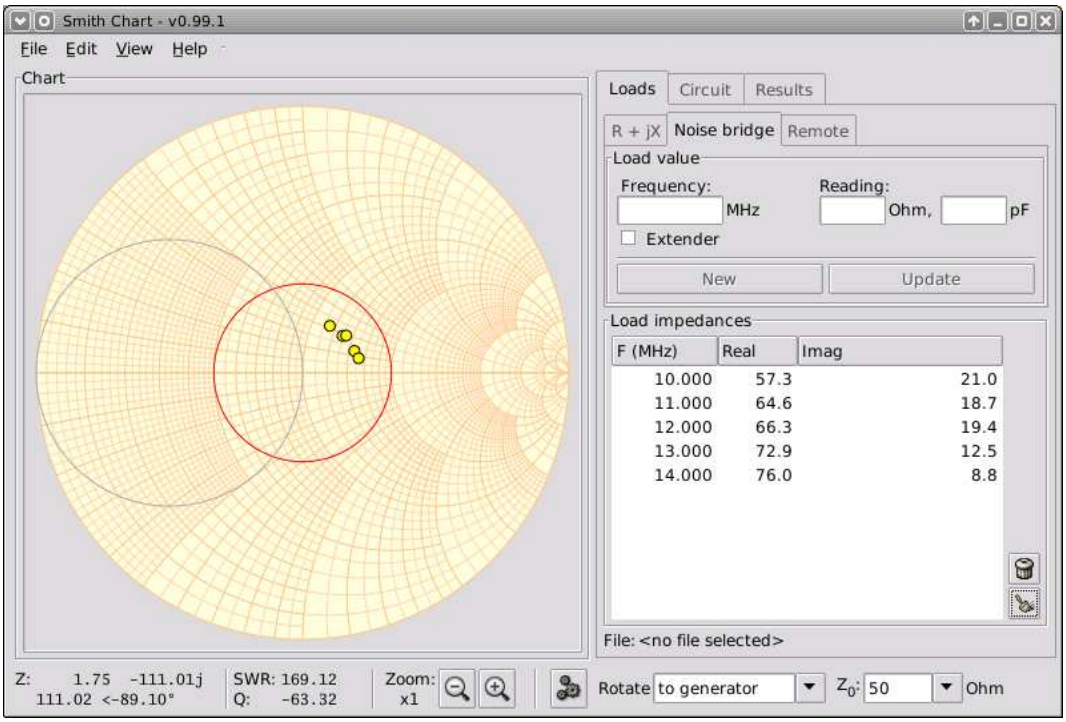


Figure 14: Entering loads noise bridge R/C pairs

Using a noise bridge, we obtain readings from the bridge's potentiometer and variable capacitors. For the moment, the only bridge type supported by `linsmith` is the circuit published by W8BXI, in the Feb 1977 issue of Ham Radio. But even then, I'm sure there are variations between the bridges in use, so, using the '`noise bridge`' tab in `Preferences`, it's possible to configure the particular values for your bridge.

When using noise bridge R/C pairs, the values in the table will still be impedances (R+jX), but selecting a line in the table will show the original R/C values for editing.

The R/C pairs will be saved together with the R+jX values.

Note: It's even possible to switch to R+jX during entry, but once there, the R/C values will not be updated!

5.1.3 Remote data

This input method is not working (well) for the moment. The code may be updated in the future, so look for the next versions.

5.1.4 Import data

Comma separated values (CSV) Through the `File` menu, it is possible to import comma separated values. This is quite handy, as they can be generated easily. Please note that if you are in a locale which uses a decimal comma (most European countries), you will find that conversion routines will be confused if you use commas to separate values too. In `Edit|Preferences`, it is possible to change the separator to a semicolon (;), or other character to avoid that conflict.

The file should contain only frequency,R,X triplets, like the following example. The conversion routine permits a fair amount of freedom, such as leading spaces, and missing fractional parts.

```
100,12.5,22.3
110, 13.4, 25.0
120,15,-33
```

s2p Two-port parameters From version 0.99.9 on, it is possible to directly import Touchstone (R) two-port data files. For the moment, only Version 1.0 files are imported (though in some cases Version 2.0 files might be read correctly).

When the datafile is read correctly, a window will appear showing the imported records. In this window, a selection can be made of the range (or elements) to be brought into linSmith.

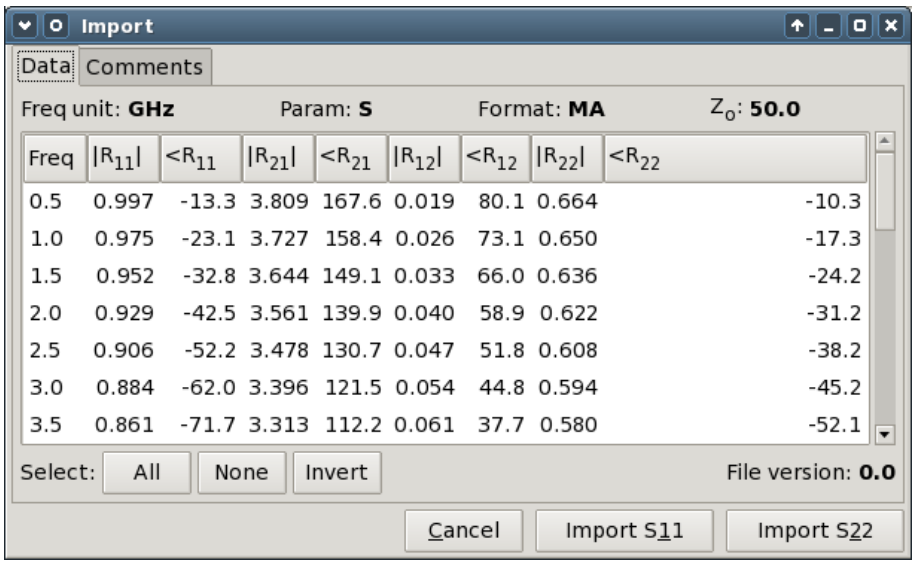


Figure 15: Importing s2p files

On the second page ('Comments') is a list of the comment lines read:

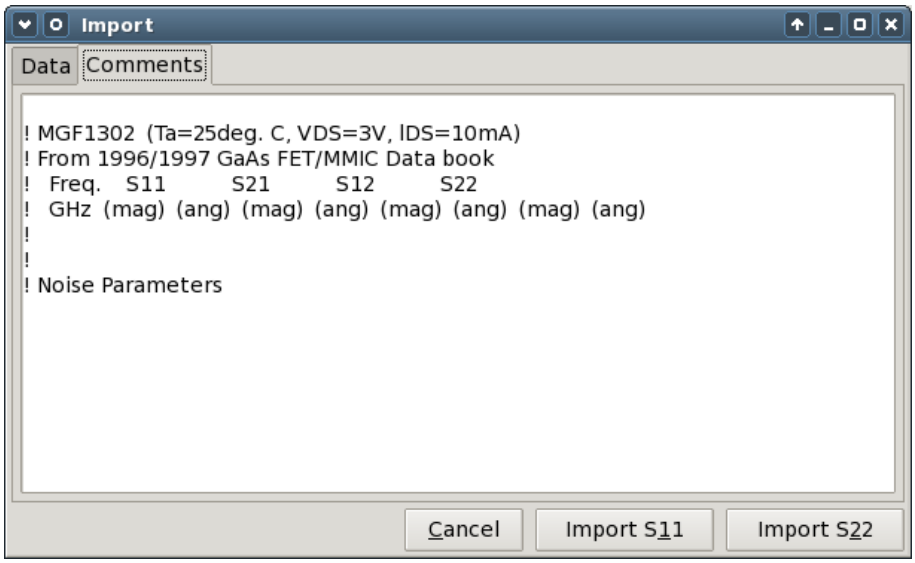


Figure 16: Comment cards

Finally (after selecting a range - or all) of data lines, select if you want to use the input or output impedances. After clicking one of the two options, the data will be imported as loads:

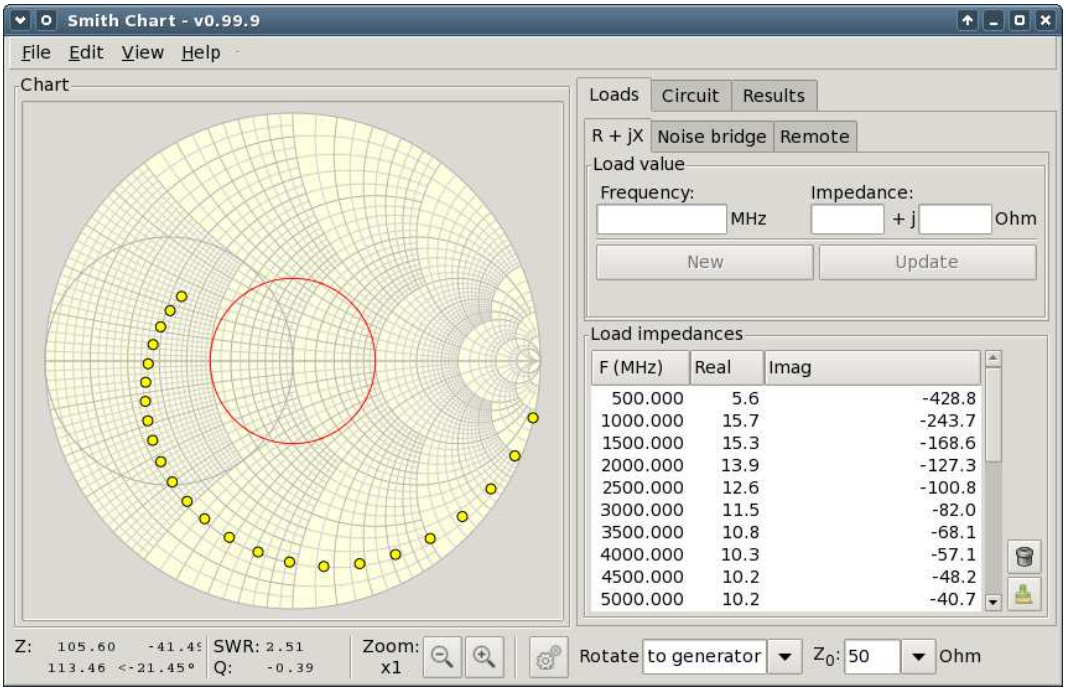


Figure 17: Imported data (MGF1302)

5.2 Entering the circuit elements

This is a little more involved. For most components, it is necessary to select how the component will be connected in the circuit. Some are exceptions: a serial line section can hardly be connected in parallel!

To enter a value, select the entry box, and modify its contents to the new value.

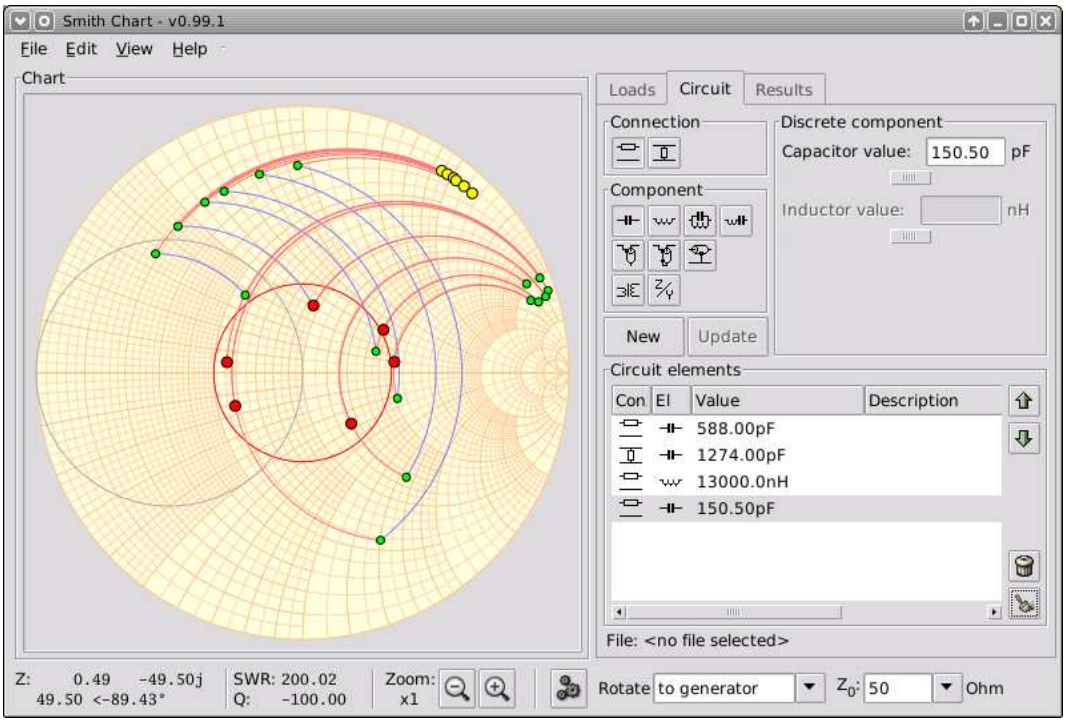


Figure 18: Elements

These are the possible circuit elements (between []'s, I've marked the possible connections):

5.2.1 Discrete capacitor [P/S]

5.2.2 Discrete inductor [P/S]

5.2.3 Parallel capacitor/inductor combination [P/S]

5.2.4 Series capacitor/inductor combination [P/S]

Discrete components are first defined as capacitors and/or inductors, then their value can be entered as pF and nH. A scrollbar under the component value will permit 'experimenting' with the component value more easily. From version 0.8.1 on, the effect of the scrollbars is visible immediately.

5.2.5 Stubs [P/S]

Both open and closed stubs can be added to the circuit. Their configuration is defined by the stub length, the characteristic impedance, and the velocity factor

Though loss factors can be programmed, they are not taken into account yet.

Scrollbars under the length and characteristic impedance facilitate experimenting. Again, from version 0.8.1 on, the effect of the scrollbars is visible immediately.

5.2.6 Line sections [S]

Sections of transmission lines can be connected in series.

5.2.7 Transformer [S]

A simple (ideal) transformer can be simulated too. No model is used - just simple impedance transformation.

5.2.8 Z/Y transform [n/a]

The Z/Y transform is really a pseudo-component, which switches between impedance and admittance charts. This was include principally as a didactic tool to help students comprehend the effect of parallel and series circuits. Internally all calculations proceed with impedances.

The Log list is not changed by this.

5.3 The Log page

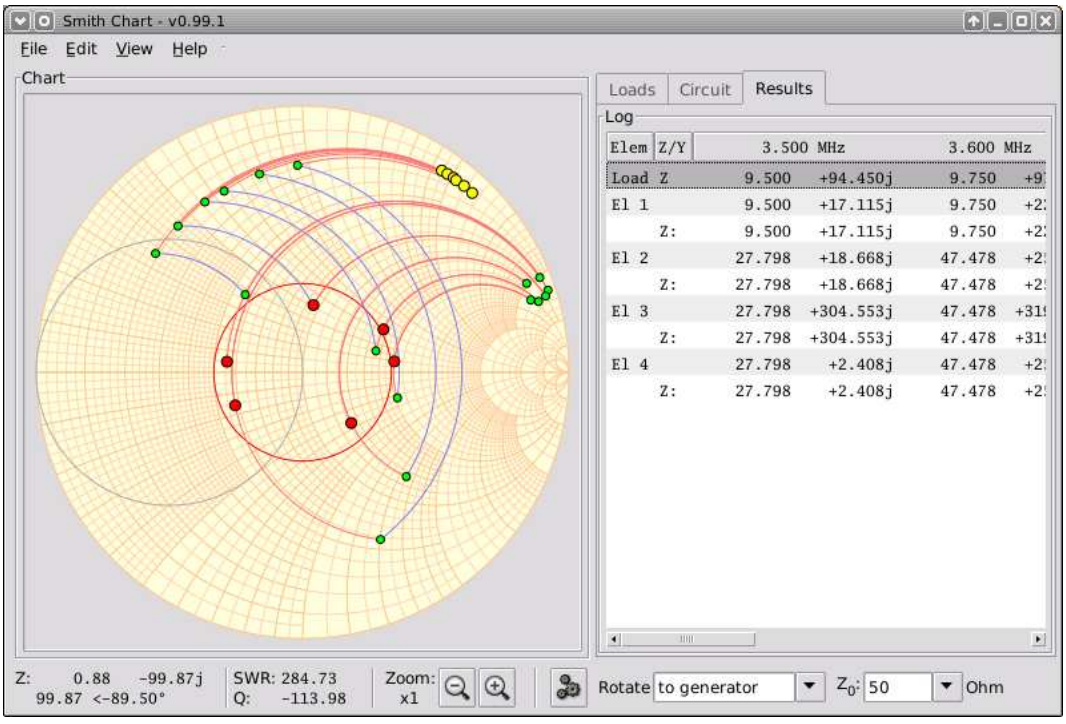


Figure 19: Numerical results

The Log page shows the intermediate and final results in table form. Of each element, its value value is shown, and then the combined circuit impedance with the component included.

The Z/Y column shows the kind of result (impedance or admittance). This depends on two factors:

- It is possible to force the use of impedances in all cases by setting 'Always use impedances' in Preferences,
- or the value will be shown according to the number of Z/Y blocks inserted in the circuit.

5.4 Using the value scrollbars

The behaviour of the scrollbars has been modified in the 0.9.x series of linSmith.

Using the value scrollbars is really a convenient form to experiment with values (value in discrete components, length and Z_0 in transmission lines, and transformer ratio).

From version 0.9.0 on, the scrollbar will re-center itself 1 second after the scrollbar button has been released. This is much more intuitive, and permits experimenting with improved precision (the scrollbar is actually logarithmic - more precision in the center).

The scrollbar now modifies the associated edit-box only. The component list remains the same.

Clicking on the component list will restore the original element value (before scrollbar modifications). Clicking on the Update button will update the list to the new value. Both the Update button and the scrollbar are disabled if the value in the edit-box is not valid.

The scrollbar permits changing between 0.5 and 2 times the actual value, to maintain a reasonable resolution.

5.5 Printing

To output a plot, click on the 'Export as Postscript...' option in the 'File' menu. A dialog will appear, permitting to enter a filename for the Postscript output file. If the file exists, a warning will appear.

6 Remote instruments

On pressing the Update from instrument button on the loads page, linsmith will open a small dialog, asking for the limits, and send commands to the output pipe to interrogate the instrument:

```
getload <freq>
```

linsmith then waits for a reply from the instrument in the form of:

```
load <freq> <real> <imaginary>
```

This process will repeat until the range specified in the dialog box is completely covered.

Note: <freq>, <real> and <imaginary> should be decimal values, and frequency expressed in kHz.

If 'Peripheral can control linsmith' is enabled, linsmith will additionally recognise the

```
load clear
```

command from the peripheral, indicating elimitating all the previously defined loads. More commands may be added later...

Also note that you can actually test this without any external program. In one terminal window do (while linsmith is running, of course):

```
cat < /tmp/linsmith.out
```

and, in another terminal, you can enter the values manually (after seeing the prompt in the other window):

```
echo ''load 14305.2 0.45 0.23'' > /tmp/linsmith.in
```

If you'd like to input an entire file with values, enable the 'Peripheral can control linsmith' option, and do

```
cat loadfile > /tmp/linsmith.in
```

7 Internationalization

From version 0.7.6 on, linSmith can be translated to other languages, and, in fact received translations to Chinese, Spanish and German.

I you would like to translate the program, here are the steps to follow:

- Unpack the tar.gz source package as usual.
- Go to the po/ directory
- Copy linsmith.pot to a file consisting of the desired language and .po, eg.: nl.po for Dutch.
- Edit this file with an editor capable of editing your language using Unicode - UTF-8 (This is not as obvious as it sounds. See the note below.) For each translatable term, the .po file contains a comment, and two lines. Eg:
  ```
  #:  src/interface.c:91
  msgid "Circuit"
  msgstr ""
  ```
  Translate the term after msgid, and insert the translation between the quotes after msgstr. Do leave all lines in the destination file - they are needed to make the translation, and to automatically update the files when a new version comes along.
- To test, do a 'make install' and run linsmith again. If the language corresponds to the one of your system, the program should now start in your language! If you want to test linsmith in one of the other languages, you can do this:
  ```
  LANG=zh_CN linsmith
  ```
  Of course, you need to have a font installed, capable of representing the language, to see the result.
- Note: if you test other languages with the above method, remember that your window manager was probably started in *your* language, so the title of the main window may not be translated correctly!
- And a last note: If you would like to contribute the translation to the package, please do so. I would appreciate it very much. If you would like to know more about the fields to be completed at the top of the .po-file, or other details about the internationalization process, please consult http://www.gnu.org/software/gettext/manual/, the official documentation for the gnu gettext project.

NOTE on UTF-8 editors: Most simple editors recognize only 7 bit ASCII. Others are compatible with UTF-8 and UTF-16, but for the GTK+ texts UTF must be the explicitly the 8-bit variant. Not all terminal emulators are UTF8 capable, and even if they are, some must be explicitly enable in their config file or the command line. So don't necessarily blame the editor! An excellent terminal is rxvt-unicode (sometines called urxvt).

Editors JOE (a (not-so-)simple Borland IDE-like editor) received full UTF-8 support. To enable it press ^T E and enter UTF8, or modify joerc. If you prefer a graphic interface. a powerful editor with character set selection is Nedit.

Yudit is a very capable multi-character-set-able editor - the GUI is not entirely of my liking, but that's personal.

I understand AbiWord is also UTF-8 capable.

8 Thanks

My sincere thanks to many people... In particular for the patience of the gnome-print people whom I falsely accused of bugs.

Thanks to Georg Baum for encouraging me to publish the program, and for all the patient help in preparing and testing this package. Mind - any error in the packaging is mine - don't blame Georg!

To SourceForge for hosting linsmith - and so many other projects!

And a heartfelt to the entire Linux society for all the excellent programs available. I hope this program counts as a (minor) retribution.

Please check the Changelog for the list of people who reported bugs or suggested improvements.

9 linSmith in Linux distributions

Of course I encourage the distribution of linSmith, and I welcome the inclusion of the package in several distributions. There is a problem though: none of them warn me that they have done so, and in several cases, I have had complaints about out-of date versions and bugs, which were already solved in the original (sourceforge) versions but never made it to the distros.

Please guys and gals, if you include linSmith in your distro, either subscribe to the news feed at Sourceforge, or send me a mail, so I can keep you advised of newer versions.

Debian now includes a package for linSmith, well-maintained by Margarita. Thanks!

10 Bugs, feature requests, patches and such

I'm quite sure there is a lot to be improved and I wouldn't be surprised to hear about bugs. Please use the forum at http://sourceforge.net/projects/linsmith for comments and such.

I'd like your feedback, but be understanding - I am neither a professional programmer, nor do I have that much time on hand. If you need to contact me personally, use the address at the top of this document, or the mailform on my site: http://www.jcoppens.com/misc/mail.en.php.

Have fun,
John

11 References