

HOWTO setup a professional design environment for gEDA

Peter Kaiser, peter@easy-asic.de

This document is released under GFDL

<http://www.gnu.org/copyleft/fdl.html>

24. January 2005

Who should read this document

After you have successfully installed gEDA software on your computer, you might want to start immediately working. Do so, if you are a hacker. You will have fun and come to good results.

But if you have requirements like:

- Several people work on the same design. May be one person draws the symbols, the next one creates the schematic and does simulation and a third one is responsible for the layout.
- Someday, somebody else will take your design and needs to make changes.
- Regular automatic backups

Then a design environment with scripts and different libraries is needed. This HOWTO describes the way to set it up and gives examples to make it easy for you.

Schematic Libraries

At least one symbol, necessary for a design, is not in the standard library. That's the reason why you will have more directories, containing symbols for your design. We can distinguish between four different kind of libraries:

System library: This library comes with gschem distribution. It's located in `<geda-install>/share/gEDA/sym`. It contains hundred of symbols and is the first choice for new users.

Technology library: All users in a network have access to this library. In a semiconductor design, this is the vendor symbol library for a specific technology.

User library: This library is somewhere in your home directory and contains your private symbols.

I suggest ~/gEDA.

Project library: This contains symbols for a single project. A typical example is a block symbol in a hierarchical design. I suggest to create a ./sym directory in the project directory.

Setup files

To get access from gschem to the libraries, you need to tell the program during the startup, where it can find those libraries. Most of the gaf tools read in 6 setup files. The first three are for the common setup and read by most programs (gschem, gnetlist, gattrib, ..). The second three are program specific. Each triple has the following order: System, user and project rc file. This results in the following order:

1. <geda-install>/share/gEDA/system-gafrc *
2. ~/.gEDA/gafrc *
3. ./gafrc
4. <geda-install>/share/gEDA/system-gschemrc
5. ~/.gEDA/gschemrc
6. ./gschemrc

* This is valid for the next versions after 20041228

The scripting language for the setup files is guile, which is a special version of scheme. If you want to write your own scripts and become a guru, you can get more information from:

<www>

or

The Scheme Programming Language; R. Kent Dybvig; The MIT Press, Cambridge Massachusetts; ISBN 0-262-54148-3

Useful commands in a rc file

Comments do start with “;”

Display messages to the console window:

```
(display "gschemrc: loading private design environment \n")
```

Add a new library:

```
(component-library "./sym")
```

Underlying schematics for hierarchical designs:

```
(source-library ".")
```

Copy all invisible attributes from the symbol library to the schematic:

```
(promote-invisible "enabled")
```

Running different versions as the same user

I always use prefix during the installation. The stable version is in:

```
~/geda-install_20041228
```

the CVS version is in:

```
~/geda-install_20050109
```

I switch between both with the PATH and LD_LIBRARY_PATH variable in the .profile.

Backups

< tbd >

Make files

< tbd >

Example user gafrc file

```
(display "gafrc: \n")
```

```
(display "gafrc: loading DHOMAY design environment \n")
```

```
(display "gafrc: \n")
```

```
:: User symbol libraries
```

```
(component-library "/home/peter/gEDA/libraries_pg/power")
```

```
(component-library "/home/peter/gEDA/libraries_pg/primitive_spice")
```

```
(component-library "/home/peter/gEDA/libraries_pg/cells_analog")
```

```
(component-library "/home/peter/gEDA/libraries_pg/cells_verilog_1")
```

```
:: Project symbol library
```

```
:: Hierarchical designs, symbols
```

```
(component-library "./sym")
```

```
(component-library ".")
```

```
:: Hierarchical designs, underlying schematics
```

```
(source-library ".")
```

Document Revision History

07. January 2005	First release

Comments and suggestions about this document are welcome. Please send me an email.