

## **Hints for using the Python 3 Skript „PyKiCad-CaseSensitiveLibCure\_RevC\_21Mar2014.py“**

From January/February (BZR4646) on , KiCad-Eeschema will discriminate symbolnames for upper and lower case. At former days only upper case letters were used at schematic files.. This over restrictive method has many drawbacks, so it would be changed. Later, symbol names at the symbol library were allowed to have upper and lower case letters. But for compatibility these were transformed to upper case, if used at the schematic. Then Eeschema could use upper and lower case letters, but had to check if there were yet upper case symbol names at old schematics, which fit to upper and lower case names at the library. Because of this there was the possibility of a nasty ambiguity. To get rid of them, from BZR4646 on the option, that Eeschema searches at uppercase names at the schematic for fitting upper and lower case names at the library, was switched of. This is the reason, why from KiCad BZR44646 on the symbolnames at the schematic have match strictly to the symbol names at the librarys.

Since last year, KiCad saves the schematics with exact matching symbolnames. So people, who work with an actual KiCad at actual projects, will not even realize this change.

Unfortunately some old schematic files with only upper case symbolnames cannot find there counterparts at the symbol librarys. Sadly, this is even true for the „-cache.lib“ library. This problem occurs at very old schematics, which are not touched and used for a long time.

For updating such old schematics, there is a Python 3 Skript "PyKiCad-CaseSensitiveLibCure\_RevC\_21Mar2014.py". It is a "Stand alone" Python3 Skript, which does not use the internal KiCad Python Skripting, which uses Python 2.

Use of the Skript:

- 1) Start PyKiCad-CaseSensitiveLibCure\_RevC\_21Mar2014.py This can be done directly via an interpreter or by starting „idle“ (for Python3), then load PyKiCad-CaseSensitiveLibCure\_RevC\_21Mar2014.py and start it, too. If you got an error message like „, ImportError: No module named tkinter“, you will probably have used Python2 (Idle2) instead of Python 3.
- 2) Choose the schematic file for conversion by using the button and the popup menu.. It will have the suffix „.sch“. There can only be used one single schematic at a time.
- 3) Now choose the KiCad librarys for the projekt, by using the button and the popup menu.. It will have the suffix „.lib“. You can use one or more different library files at any time.
- 4) Press button „Convert“, to start the data conversion. The converted schematic file will be saved with an added „-Converted.sch“.
- 5) How it works: The skript creates a list of all available symbolnames from the symbol libraries and their to upper case transformed counterparts. Then a check is done for every symbolname from the schematic wether it

matches to a name transformed to upper case from this list. If it is so, the symbolname from the schematic will be replaced by the original name from the list. If no match found, the old name from the schematic is used further on. If it contains at last only one lower case letter, the name is regarded as „new“, and no actions are furthermore taken on this name. But if there are only upper case letters besides numbers and special signs, it will be added to a list, which can be copied to the PC clipboard by a button at a report window at the end of the process. There you can transfer it to an editor of your choice and read it. Either you make a new converting process with the converted file and a fitting library, or you can edit the schematic file with an editor by hand, which may be easier in some cases with only one or two missing symbols.

6) Warnings: If there are more than one identical names (in converted form) in the libraries, the skript will only use the first found.

Hierarchical subschematics have to be converted one by one for their own. You better try at first to use the „-cache.lib“ matching to the schematic file. NEVER use the function „save schematic project“ of Eeschema, if there are unknown symbols at the schematic (boxes with questionmarks). The „-cache.lib“ file would be overwritten. If there are not yet recognized symbols in this file, they would be destroyed.

7) Attention: The Program will not recognize names, which are marked by a swung dash „~“. Here you have to make the matching by hand.

The Skript „PyKiCad-CaseSensitiveLibCure\_RevC\_21Mar2014.py“ is published under the GNU-GPL. So are allowed to use it freely, even for commercial purposes. There is NO WARRANTY!

For questions and suggestions about this Python 3 skript, you can contact me with e-mail to [bernd.wiebus@gmx.de](mailto:bernd.wiebus@gmx.de)

Author: Bernd Wiebus  
Uedem/Germany, 21. March 2014