

Hinweise zum Python 3 Skript „PyKiCad-CaseSensitiveLibCure_RevD_13Apr2015.py“

Ab Januar/Februar (BZR4646) wird bei Symbolnamen in KiCad-Eeschema nach Groß- und Kleinschreibung unterschieden. Früher wurden in Schaltplandateien für die Symbolnamen nur Großbuchstaben verwendet. Da dieses überrestriktive Vorgehen aber viele Nachteile hat, wurde nach und nach davon abgewichen. Symbolnamen durften später in der Bibliothek Groß- und Kleinbuchstaben enthalten. Aus Kompatibilitätsgründen wurden diese aber bei der Übernahme in einen Schaltplan in Großbuchstaben umgewandelt. Noch später konnte dann auch Eeschema sowohl Groß- als auch Kleinbuchstaben verwenden, musste aber dabei darauf achten, ob nicht noch eventuell in alten Schaltplänen großgeschriebene Symbolnamen existieren, die zu groß- und kleingeschriebenen Symbolnamen in der Bibliothek gehören. Dadurch ergab sich eine unangenehme Mehrdeutigkeit. Um diese zu beseitigen, wurde ab BZR4646 die Option, dass für nur großgeschriebene Symbolnamen im Schaltplan auch nach passenden klein- oder gemischtgeschriebenen Symbolnamen in der Bibliothek gesucht wird, abgeschaltet. Ab KiCad BZR4646 müssen darum die Symbolnamen in den Schaltplandateien den Originalnamen in der Library exakt entsprechen. Leider werden dadurch bei alten Schaltplandateien mit großgeschriebenen Symbolnamen diese nicht mehr in den Bibliotheksdateien gefunden. Das gilt auch für die "-cache.lib" Bibliothek. Allerdings speicherte KiCad schon eine ganze Weile die Schaltpläne mit den der Bibliothek exakt entsprechenden Symbolnamen, so dass jemand, der mit aktuellen Programmen aktuelle Projekte bearbeitet den Übergang kaum bemerkt. Leider tritt das Problem trotzdem bei alten Schaltplänen auf, die sehr lange nicht verwendet wurden. Um die Symbolnamen in alten Schaltplandateien anzupassen, existiert das Python 3 Skript "PyKiCad-CaseSensitiveLibCure_RevC_21Mar2014.py". Es ist ein "Stand alone" Python 3 Skript, das nicht in das KiCad interne Python Skripting eingebunden ist (Das beruht auf Python 2).

Anwendung:

- 1) PyKiCad-CaseSensitiveLibCure_RevD_13Apr2015.py starten. Entweder direkt oder indem Sie zuerst "Idle" (für Python 3) starten und darin dann PyKiCad-CaseSensitiveLibCure_RevD_13apr2015.py laden und starten. Falls sie eine Fehlermeldung erhalten, die lautet: „Import Error: No module named tkinter“ haben Sie höchstwahrscheinlich fälschlich Python 2 (Idle 2) verwendet.
- 2) Die zu konvertierende Schaltplandatei auswählen. Die entsprechende Schaltfläche verwenden und das Menü benutzen, um die Schaltplandatei zu wählen und zu öffnen (endet auf .sch). Es kann immer nur ein Schaltplan verwendet werden.
- 3) Die dabei zu verwendenden Bibliotheken auswählen. Die entsprechende Schaltfläche verwenden und das Menü benutzen, um eine oder mehrere KiCad Symbolbibliotheken (enden auf .lib)

- auszuwählen.
- 4) Die Schaltfläche „Umwandeln“ verwenden, um die Konvertierung zu starten. Die konvertierte Schaltplandatei wird mit der zusätzlichen Endung „-Converted.sch“ gespeichert.
 - 5) Hinweise zur Funktion: Das Skript legt eine Liste aller verfügbaren Symbolnamen aus den Bibliotheken und deren in pure Großschrift gewandelten Entsprechungen an. Dann wird jeder Symbolname aus dem Schaltplan genommen, und überprüft, ob er einem der in Großschrift gewandelten aus der Liste entspricht. Ist das der Fall, wird er durch das Original ersetzt. Wird keine Entsprechung gefunden, wird der ursprüngliche Name weiterverwendet. Enthält ein Symbolname, für den keine Entsprechung gefunden wurde, auch nur einen Buchstaben in Kleinschrift, wird er als „neu“ erkannt und nichts weiter unternommen. Enthält er aber außer Sonderzeichen und Zahlen nur Großbuchstaben, wird er in einer Liste erfasst, die Sie anschließend in einem Meldungsfenster mit Hilfe einer Schaltfläche in die Zwischenablage Ihres PCs kopieren können, um sie dann in einem Editor Ihrer Wahl betrachten und bearbeiten zu können. Entweder Sie machen mit der umgewandelten Datei und passenden Bibliotheken einen neuen Umwandlungsdurchgang, oder, in Einzelfällen, bearbeiten Sie die Schaltplandatei per Hand in einem Editor.
 - 6) Warnungen: Wenn mehrere (in umgewandelter Form) gleich lautende Symbolnamen in den Bibliotheken auftauchen, wird das Original des zuerst gefundenen verwendet. Unterschaltpläne müssen einzeln separat bearbeitet werden. Versuchen Sie zuerst mit der „-cache.lib“ der Schaltplandatei zu arbeiten. Benutzen Sie in Eeschema NIEMALS die Funktion „Projekt speichern“, solange Sie noch unbekannte (Kästchen mit Fragezeichen) Symbole haben. Die „-cache.lib“ würde dabei überschrieben, und falls dort nicht erkannte Symbole gelagert wären, würden diese zerstört.
 - 7) Achtung: Das Programm erkennt keine Namen, welche in der Schaltplandatei oder in der Bibliothek mit einer Tilde „~“ markiert sind. Bei diesen muss die Anpassung von Hand gemacht werden.

Das Skript „PyKiCad-CaseSensitiveLibCure_RevD_13Mar2015.py“ steht unter der GNU-GPL und darf frei, auch für kommerzielle Zwecke, verwendet werden.

Bei Fragen und Vorschlägen zu dem Skript schreiben Sie bitte eine E-Mail an: bernd.wiebus@gmx.de

Autor: Bernd Wiebus
Uedem, den 13. April 2015