

KiCad Multi-User Setup (in Bearbeitung)

Bei [Team14](#) verwenden wir [KiCad](#) auf mehreren PCs mit sogar unterschiedlichen Betriebssystemen: Windows 10 und Linux Mint 18. In diesem Artikel beschreibe ich unser Setup und worauf geachtet werden muss. ¹⁾

Synchronisation

Zur Synchronisation unserer Daten verwenden wir einerseits Seafile, und andererseits git-versionierte Projekte mit GitLab.

fp-lib-table und sym-lib-table

Ab KiCad 5 werden neben den Footprints nun auch die Schaltplansymbole über zentrale bzw. projektspezifische Tabellen verwaltet. Das sind einfache Textdateien, die die Bibliotheken auflisten. Hierbei sollten fixe Pfade in den Einträgen unbedingt vermieden werden.

projektspezifisch

Bei den projektspezifischen Tabellen passiert dies automatisch durch den Präfix `${KIPRJMOD}`.

systemweit

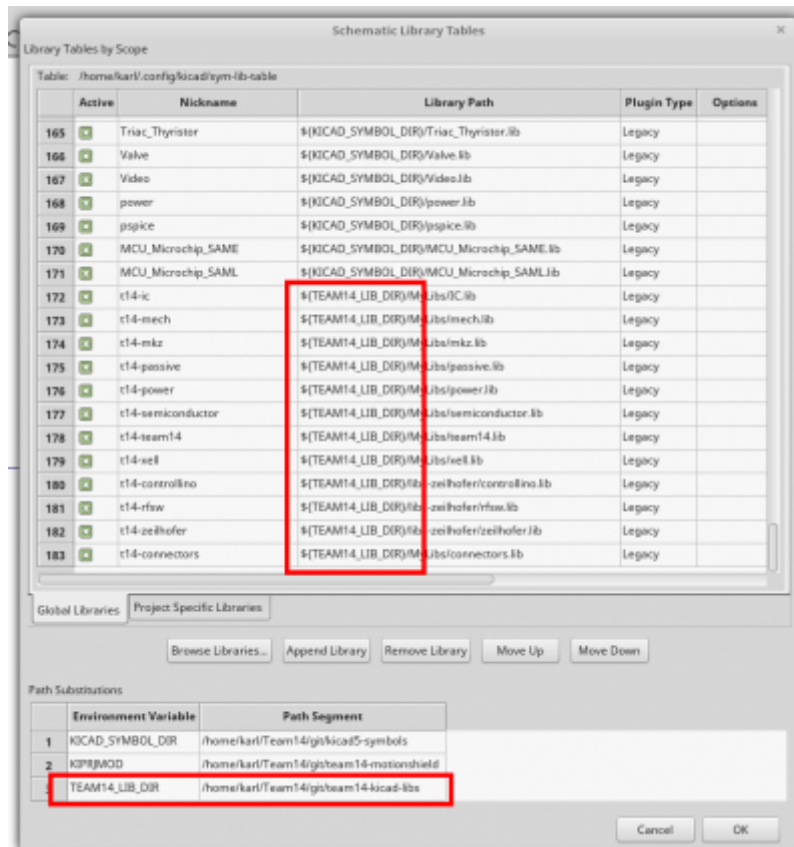
Bei den systemweiten Tabellen, die unter Linux im Ordner `~/.config/kicad` zu finden sind, sollten auch unbedingt fixe Pfade vermieden werden. Für die offiziellen Libs von GitHub werden planmäßig entsprechende Umgebungsvariablen in der Datei `~/.config/kicad/kicad_common` angelegt, siehe auch [KiCad 5: Projektportierung von KiCad4](#).

Hier kann man aber auch firmeneigene Variablen anlegen. Bei uns ist die Firmenbibliothek ein einziges Git-Repository, in dem sich Symbole, Footprints und 3D-Modelle befinden.

Durch den Eintrag in die **kicad_common**

```
# Team14 library path:  
TEAM14_LIB_DIR=/home/karl/Team14/git/team14-kicad-libs
```

kann man dann die neu geschaffene Umgebungsvariable in den beiden systemweiten Tabellen verwenden, und somit ganz auf fixe Pfade in den beiden Tabellen verzichten.



Was ist auf jedem System zu machen?

kicad_common

In der kicad_common müssen die Pfade auf jedem System richtig eingerichtet werden. Das sollte die einzige Datei sein, in der systemspezifische Pfade stehen.

Synchronisierung der Tabellen

Da die systemweiten Tabellen an einer speziellen Stelle im Dateisystem liegen müssen, stellt sich die Frage, wie man die am besten synchronisiert.

Git Repository

in dem KiCad Config-Ordnung befinden sich einige Dateien, wobei eigentlich nur die **fp-lib-table** und **sym-lib-table** synchronisiert werden sollen. D.h. man könnte diesen Ordner per Git verwalten, wobei fast alle Dateien in der .gitignore gelistet sind.

- Vorteile: Versionierung und Diff macht Änderungen sichtbar
- Nachteil: manuelles Triggern (push/pull) notwendig

Seafile mit Symlinks

Man könnte diese Dateien auch per Seafile synchronisieren, wobei sich diese dann irgendwo im „Firmen-Ordnung“ befinden. Per Symlink können die Dateien dann in den KiCad Config-Ordner verlinkt werden.

- Vorteile: vollautomatisch und versioniert
- Nachteil: Symlinks sind auf Windows zwar möglich, aber unüblich.

Seafile mit manuellem Kopieren

Um keine Symlinks verwenden zu müssen, könnten die Dateien auf Aufforderung z.B. per eMail, vom Benutzer manuell aktualisiert.

- Vorteil: bewusstes Update
- Nachteil: manueller, lästiger Vorgang

Da ein Update dieser Tabellen nach anfänglicher Einrichtung nur selten aktualisiert werden müssen, ist ein manueller Eingriff vermutlich akzeptabel.

KiCad 4

Für KiCad 4 existiert die neue sym-lib-table noch nicht, und somit stehen die Pfade der verwendeten Symbol Bibliotheken in der .pro Datei.

Ziel ist es somit

- überflüssige Bibliotheken aus der .pro Datei entfernen
- sicherstellen, dass alle Bibliotheken sich in Pfaden der Umgebungsvariablen befinden

Projektdatei bereinigen

Die Projekt Datei (*.pro) sollte der Übersicht halber so gut wie möglich bereinigt sein. Um herauszufinden, welche Bibliotheken im Projekt überhaupt in Verwendung sind, bietet es sich an, die Netzliste zu analysieren.

```
cat my-kicad4-project.net | grep -i [.]lib
```

Dies liefert z.B. folgende Ausgabe:

```
karl@karl:~$ cd ~/Team14/gst/; cat cc-board.kicad_pcb | grep -i [.]lib
[ur] /home/karl/Team14/gst/kicad4-symbols/transistor.lib))
[ur] /home/karl/Team14/gst/team14-kicad-libs/MyLibs/pz1hofer.lib))
[ur] /home/karl/Team14/gst/kicad4-symbols/device.lib))
[ur] /home/karl/Team14/gst/...hardware/kicad/...cc-board/dsd-cc.lib))
[ur] /home/karl/Team14/gst/kicad4-symbols/connector.lib))
[ur] /home/karl/Team14/gst/team14-kicad-libs/MyLibs/IC.lib))
[ur] /home/karl/Team14/gst/kicad4-symbols/linear.lib))
[ur] /home/karl/Team14/gst/...cc-hardware/kicad/...cc-board/tpm.lib))
[ur] /home/karl/Team14/gst/team14-kicad-libs/MyLibs/connectors.lib))
[ur] /home/karl/Team14/gst/team14-kicad-libs/MyLibs/semiconductor.lib))
[ur] /home/karl/Team14/gst/team14-kicad-libs/MyLibs/mech.lib))
[ur] /home/karl/Team14/gst/kicad4-symbols/switch.lib))
```

Hier kann auch gleichzeitig überprüft werden, ob Bibliotheken in Verwendung sind, die sich nicht in

den Pfaden der Variablen von kicad_common befinden. Dies sollte unbedingt behoben werden, da es ansonsten zu Problemen auf den Unterschiedlichen Systemen der Benutzer kommen kann.

Umgebungsvariablen in der Projektdatei

KiCad 4 ersetzt manuell eingestellte Pfade im **Component Libraries Dialog** leider nicht durch die verfügbaren Umgebungsvariablen. Das muss daher per Text-Editor gemacht werden. Hier ein Auszug einer optimalen Projektdatei, die nur Pfade aus den Umgebungsvariablen verwendet:

[my-kicad4-project.pro](#)

```
...
LibDir=${TEAM14_SYMBOLS_DIR};${KICAD4_SYMBOLS_DIR}
[eeschema/libraries]
LibName1=dsd-cc
LibName2=t14_zeilhofer
LibName3=device
LibName4=Connector
LibName5=Transistor
LibName6=linear
LibName7=t14_ic
LibName8=BMS
LibName9=t14_semiconductor
LibName10=t14_mech
LibName11=t14_connectors
LibName12=Switch
...
```

Leider fehlt hier völlig der Zusammenhang, welche Lib in welchem Pfad enthalten ist. Das ist ja einer der Gründe, warum mit KiCad 5 die sym-lib-table eingeführt wurde. Dies kann wie oben beschrieben jedoch mit der Netzliste analysiert werden.

Wir haben die Situation mit einem Präfix für unsere Team14 Libs etwas verbessert.

Hinweise

Auf meinem System sind unter /usr/share/kicad/library die Bibliotheken installiert, die ich (irgendwann) mit KiCad mitinstalliert habe. Obwohl ich KiCad 4.0.7 per PPA auf meinem System installiert hab, sind diese Bibliotheken nicht identisch mit den offiziellen Libs für V4.0.7, die man sich hier herunterladen kann: <http://downloads.kicad-pcb.org/libraries/>
Für die Zusammenarbeit mit KiCad 4 über mehrere Systeme hinweg verwenden wir die Lib V4.0.7 als gemeinsamen Nenner.

Verwendete Umgebungsvariablen

In der kicad_common für KiCad 4 habe ich z.B. folgende Bibliothekspfade eingetragen:

KiCad 4 Lib

```
KICAD4_3D_DIR=/home/karl/Team14/git/kicad4-packages3D  
KICAD4_FOOTPRINTS_DIR=/home/karl/Team14/git/kicad4-footprints  
KICAD4_SYMBOLS_DIR=/home/karl/Team14/git/kicad4-symbols
```

Team14 Lib

```
TEAM14_3D_DIR=/home/karl/Team14/git/team14-kicad-libs/modelle3D  
TEAM14_SYMBOLS_DIR=/home/karl/Team14/git/team14-kicad-libs/MyLibs  
TEAM14_FOOTPRINTS_DIR=/home/karl/Team14/git/team14-kicad-libs/MyMods
```

Links

- <https://forum.kicad.info/t/kicad-multi-user-capability/7096>

[kicad](#), [software](#), [deutsch](#), [artikel](#)

¹⁾

bezieht sich auf KiCad 5, inwieweit dies auch für KiCad 4 gilt, muss noch geprüft werden

From:

<http://www.zeilhofer.co.at/wiki/> - **Verschiedenste Artikel von Karl Zeilhofer**

Permanent link:

http://www.zeilhofer.co.at/wiki/doku.php?id=kicad_multiuser_setup&rev=1524573054

Last update: **2018/04/24 14:30**

