KiCad
Open Sorce EDA Software

Carsten Schönert

TroLUG Troisdorf

11.2.2016

Themenübersicht

About me

- *6.12.71
- Bornheim, Bonn
- erste Linuxversuche 1996 mit Suse, 1998 Wechsel zu Debian
- Debian Maintainer (Icedove, Sogo-Connector, libcoap, ...), Mitarbeit seit 2011
- FSFE Fellow
- Interesse an offener (freier!) Software und Hardware

Ein ganz kurzer Überblick

- KiCad ist eine Open Source EDA Software[1] für Windows, Linux und Mac
- Erstellung elektronischer Schaltpläne mittels Bibliotheken von Symbolen
- Planung und Erstellung von PCB auf Basis von Footprint Bibliotheken
- Kontrolle, Visualisierung durch 3D-Modelle
- Multilayer Platinen mit bis 32 (64) Lagen
- Schnittstelle an MCAD Systeme durch IDF[2] Export
- Ausgabe von Platinen und Bohrdaten durch Gerberformat, aber auch SVG, PDF, DXF
- Möglichkeit Footprints und Platinenlayout aus Eagle zu importieren
- Python Skripting Interface

[1] EDA Electronic Design Automation

[2] IDF Intermediate Data Format

Einige chronologische Daten I

- erste Veröffentlichung 1992 durch Jean-Piere Charras
- einzelner Entwickler bis 2006, auch diverse Studenten
- keine Versionsverwaltung, nur Veröffentlichung von Tarballs
- 2007 erster Upload auf Sourceforge, keine ML
- Dick Hollenbeck und Wayne Stambaugh stoßen hinzu, 3 Developer full time?
- Wechsel zu CMake, Umstellen aller Kommentare auf Englisch
- seit dem Rework des kompletten C++ Codes, starkes Refrakturieren
- Erstellung der Developer/Source Dokumentation mit Doxygen

Einige chronologische Daten II

- CleanUp, Konsolidierung und Vereinheitlichen der Menüs
- Wechsel, Benutzen der Boost Bibliothek für Pointer und Geometriefunktionen
- Einbau eines Design Regelmanager
- Hinzufügen Specctra Export (Autorouter) und Freerouter Funktion
- Einbau der Python Skripting Konsole
- weitere Import und Exportfunktionen (SVG, PDF, externe Bauteilbibliotheken, DXF)
- Benutzung von wxWidget Framework für GUI, aktuell wxWidget 3.0
- Aufbau Template Mechanismus für neue Projekte (z.B. Raspberry Shields)

Einige chronologische Daten III

- 2012 CERN beteiligt KiCad an der Open Hardware Initiative (finanzielle Mittel!)
- Cern beteiligt sich zusätzlich durch Programmierer (Einbau OpenGL Rendering, Push und Shove Routing)
- Erstellung neues Leiterplattenformat $\rightarrow$ Rel. 4.0
- Umstellung auf Single Process Applikation, Eagle PCB und Footprint Import
- GEDA Footprint Import, DXF Import und IDF Export
- GitHub Plugin für Footprint Bibliotheken
- Ausgliederung von Dokumentation, der Footprint und Bauteilbibliotheken auf GitHub
- Neue Webseite seit 2015 auf Hugo basierend

Wo will man hin?

- Modernisierung der UI, editierbare und veränderbare Toolbars (vergleichbar Gimp)
- Verbesserung Usability der Editoren
- Umstellung (Portierung) der Tools in Pcbnew in das neue Tool Framework
- Einfügen weiter Eigenschaften in Schaltplansymbole für Third Party Tools (primär für Simulatoren)
- Verbesserung 3D Modelling, Support aller Dateiformate von OpenCascade
- Verbesserung Vorwärts und Rückwärts Annotation
- Benutzung der Zwischenablage für Pcbnew
- Erweiterung des DRC Umfangs

Wo will man in fernerer Zukunft hin?

- Integration Analog/Digital Simulator?
- Move von Launchpad zu GitHub (Git vs. BZR)?
- Kollaborieren mit anderen Softwareprojekten?

Installation

- KiCad ist erhältlich für Windows, Linux und Mac, auch Pakete für BSD Derivate
- nur ein supportetes Release, Rolling Release, aktuell Version 4.0.1
- tagesaktuelle Pakete für Ubuntu und Fedora 23 durch CI nach Integration der Paketquellen, andere Distris ”hängen” immer etwas hinterher

Manuelle Installation (Debian)

Installation der Abhängigkeiten

```
sudo apt-get install libwxbase3.0-dev libwxgtk3.0-dev \
libgl1-mesa-dev libglew-dev libglm-dev \
libcurl4-openssl-dev libboost-dev libboost-thread-dev \
libboost-system-dev libboost-context-dev libssl-dev \
wx-common cmake
```

Klonen der Sourcen

```
git clone https://github.com/KiCad/kicad-source-mirror.git \
kicad-source
```

Bauen, Installieren

```
cd kicad-source
mkdir build && cd build
cmake -DCMAKE_INSTALL_PREFIX=/wo/auch/immerhin/kicad ..
make -j[x] install
```

Benutzung

Noch etwas Theorie!

- KiCad besteht aus einem Hauptfenster, auch KiCad Manager genannt
- KiCad ist in einzelne Programmteile zergliedert
  1. Eeschema
  2. Pcbnew
  3. Gerbview
  4. Bitmap2Component
  5. Zusatztools: Bauteileditor, Footprinteditor, Pcb Kalkulator und Pl Editor
- Schaltplansymbole und Footprints sind getrennt!
- Alle Dateien sind Textdateien die aus Datenreihen bestehen

Benutzung

Noch etwas Theorie!

- KiCad besteht aus einem Hauptfenster, auch KiCad Manager genannt
- KiCad ist in einzelne Programmteile zergliedert
  1. Eeschema
  2. Pcbnew
  3. Gerbview
  4. Bitmap2Component
  5. Zusatztools: Bauteileditor, Footprinteditor, Pcb Kalkulator und Pl Editor
- Schaltplansymbole und Footprints sind getrennt!
- Alle Dateien sind Textdateien die aus Datenreihen bestehen
- **Livedemo!**

Wie arbeiten?

- Einstellung von Texteditor und PDF-Betrachter (einmalig)
- Einrichtung Arbeitsblatt
- Anlegen von benötigten Netzklassen
- Einrichten/Einbinden lokale(r) Bauteilbibliothek(en)
- Erstellen/Kopieren von Schaltplansymbolen
- Einrichten/Einbinden lokale(r) Footprint(s)
- Erstellen neuer (lokaler) Footprints
- Übung, Übung, Übung

Zusammenfassung

Folgende Punkte sind allgemein gültig:

- Benutzung von lokalen Bibliotheken (für Bauteile wie Footprints)
- Welche Lizenz soll mein Werk haben?
- Gute Vorbereitung ist die halbe Arbeit (Datenblätter, Symbole)
- Benutzung von hierarchischen Schaltplänblättern
- Prüft die Produktionsbedingungen der PCB Manufaktur (getrennte Bohr[loch]dateien?, Minimalmaße?, Kosten)
- Erstellung und Benutzung von Netzklassen, erleichtert das nachträgliche globale Ändern von Eigenschaften

Wo Hilfe erhalten?

- Projekt Webseite! `http://kicad-pcb.org/`
- Handbücher der Tools `http://docs.kicad-pcb.org/de/`
- es gibt ein KiCad Forum
- diverse Video HowTos auf Contextual Electronics
- Tutorials von Timo Gruss
- Zusammenstellung auf mikrocontroller.net
- Youtube!
- diverse Online Ressourcen für Bauteile und Footprints auf kicadlib, miniwatt (beides teilweise veraltet!), bytelabs, udemy, Node7

Verbesserungspotential

- kein Mailinglisten für Benutzer vorhanden, nur eine Benutzergruppe auf Yahoo!
- Projekt liegt auf Launchpad (Git vs.Bazaar)
- Organisation der Dokumentationsarbeit läuft ausschließlich über GitHub
- Vollständigkeit und Gleichheit innerhalb der Handbücher fehlt, gibt kein QS Team
- Übersetzungen der Handbücher, Screenshots von verschiedenen BS
- Migration aus anderen Formaten (Eagle, Target, ...) fehlt teilweise
- zahlreiche fehlende Bauteilsymbole und Footprints

©2016 Carsten Schoenert c.schoenert [at] t-online [dot] de
GPLv3+ or CC BY-SA 4.0.

Sie dürfen: Das vorliegende Werk teilen! Das Material darf in jedwedem Format oder Medium vervielfältigt und weiterverbreitet werden. Sie dürfen das Material bearbeiten, das Material remixen, verändern und darauf aufbauen und zwar für beliebige Zwecke, sogar kommerziell.