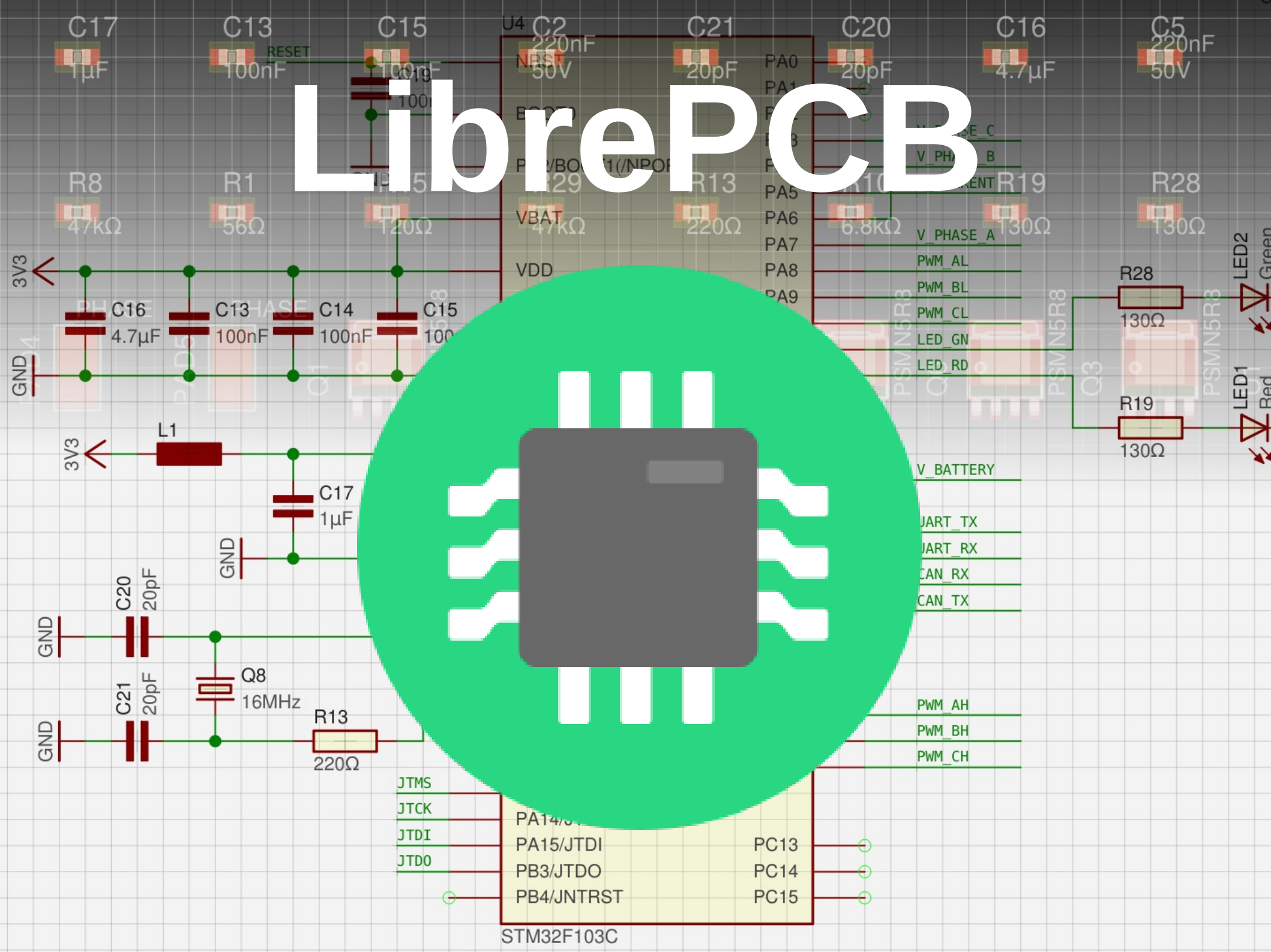
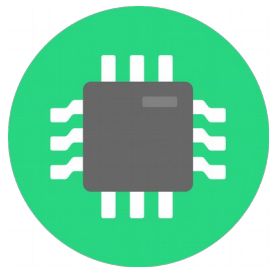


LibrePCB





1. Was ist LibrePCB?

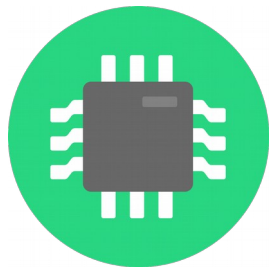
2. Motivation

3. Ziele

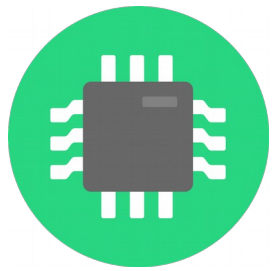
4. Aktueller Stand

5. Live Demo

Was ist LibrePCB?

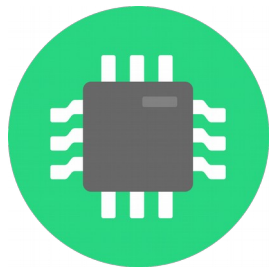


- **EDA Software (Electronic Design Automation)**
- **Initiant: Urban Bruhin**
- **Start: Februar 2013**
- **6 Contributors** (<https://github.com/LibrePCB/LibrePCB/graphs/contributors>)
- **Webseite: <http://librepcb.org>**
- **GitHub: <https://github.com/LibrePCB/LibrePCB>**



• Probleme vieler EDA Tools

- Nur für Windows
- Hohe Lizenzkosten, kostenpflichtige Updates
- Gratisversionen bzw. Hobby-Lizenzen nicht kommerziell nutzbar
- Massive Einschränkungen (PCB Grösse, Anzahl Schemaseiten/Pads/Vias)
- Cloud-Zwang, PCB Hersteller-Zwang, Gerber-Export unmöglich usw.
- Altmodisch, nicht intuitiv, unübliche Shortcuts usw.
- Proprietäre, binäre Dateiformate => Versionsverwaltung unbrauchbar
- Bauteilbibliotheks-System unflexibel und mühsam
- ...



- **Freie Software (Open-Source)**

- GPLv3 Lizenz

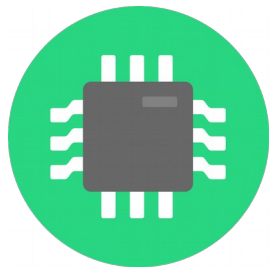
- **Multiplattform** 

- Entwickelt mit C++/Qt

- **Mehrsprachig**

- Software (GUI)
- Bauteilbibliothek





- **All-In-One Applikation**

- Kein Import/Export zwischen verschiedenen Programmen nötig

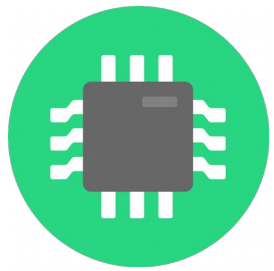
- **Automatische Forwärts/Rückwärts Annotation**

- Schema und Board sind jederzeit synchron

- **Footprint-Zuweisung direkt im Board-Editor**

- Optional auch im Schema-Editor möglich

Einfaches Dateiformat



- **“Alles ist ein Verzeichnis”**

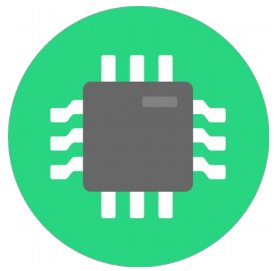
- PCB Projekte
- Bauteilbibliotheken
- Bibliotheks-Elemente (Symbole, Footprints, ...)

- **XML als primäres Dateiformat**

- Ermöglicht Versionsverwaltung von Projekten & Bibliotheken



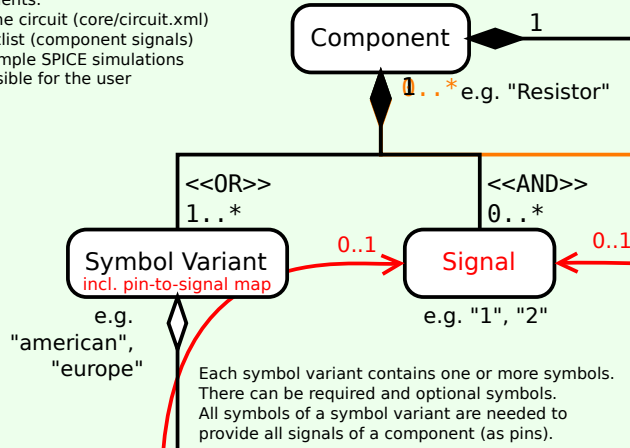
Innovatives Bibliotheks-Konzept



- **Flexibel**
- **Hohe Wiederverwendbarkeit**
- **Mehrsprachig**
- **Kategorisierbar**
- **Identifikation der Elemente per UUID**
- **Bibliotheksübergreifende Referenzen möglich**
- **Unterstützung für 3D- und SPICE Modelle**
- **Inklusive Lieferanten und Bestellnummern**

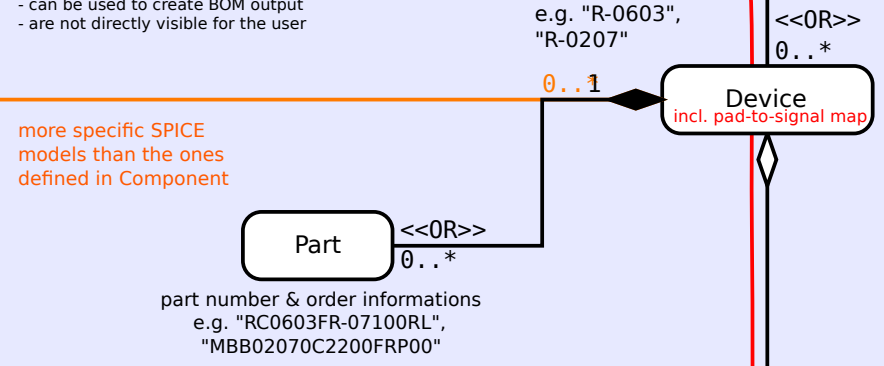
Component Directory: UUID.cmp

In projects, components:
 - will be added to the circuit (core/circuit.xml)
 - are part of the netlist (component signals)
 - can be used for simple SPICE simulations
 - are not directly visible for the user



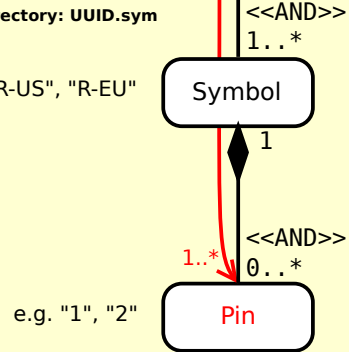
Device Directory: UUID.dev

In projects, devices:
 - will be added to boards (boards/*.xml)
 - can be used for SPICE simulations
 - can be used to create BOM output
 - are not directly visible for the user



Symbol Directory: UUID.sym

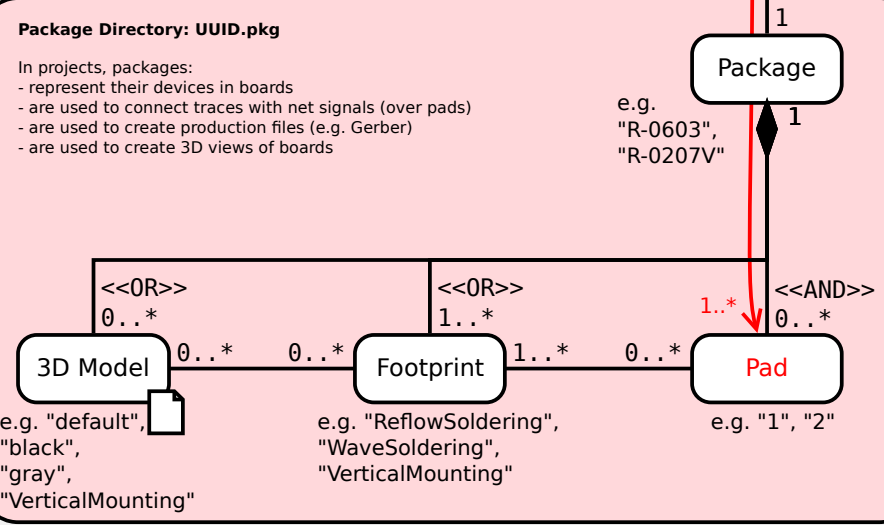
e.g. "R-US", "R-EU"



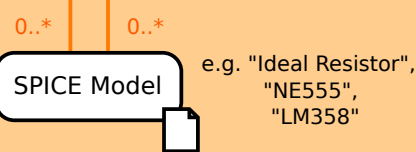
In projects, symbols:
 - will be added to schematic pages (schematics/*.xml)
 - represent their components (which are not visible otherwise)
 - are used to connect net signals with component signals (over pins)

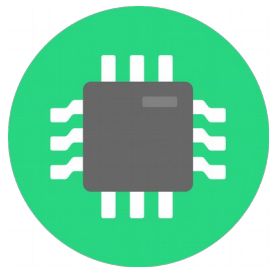
Package Directory: UUID.pkg

In projects, packages:
 - represent their devices in boards
 - are used to connect traces with net signals (over pads)
 - are used to create production files (e.g. Gerber)
 - are used to create 3D views of boards



SPICE Model Directory: UUID.spc





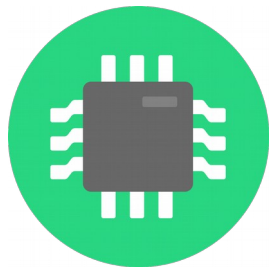
- **Grosse Bauteile-Bibliothek auf GitHub**

- <https://github.com/LibrePCB-Libraries>
- Inklusive 3D Modelle, SPICE Modelle, Lieferanten, Bestellnummern, ...

- **LibrePCB als etablierte EDA Software**

- Insbesondere bei Open-Hardware Projekten

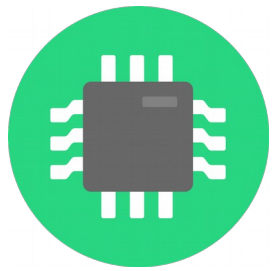
Aktueller Stand



- + Es können einfache Schemas erstellt werden
- + Es können einfache PCBs designt werden
- + Es können Gerber-Files exportiert werden

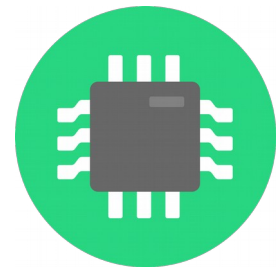
- **Noch kein Bibliotheks-Editor vorhanden**
- **Features sehr rudimentär und unvollständig**
- **Dateiformat noch nicht stabil** (breaking changes möglich!)

Unterstützung erwünscht!



- <https://github.com/LibrePCB/LibrePCB/issues>
 - Forken und Pull Requests erstellen!
- **Contributing guidelines kommen bald ;)**
 - <https://github.com/LibrePCB/LibrePCB/blob/master/CONTRIBUTING.md>

Live Demo



The screenshot displays the LibrePCB software interface. On the left, the 'Workspace Projects' panel shows a tree view for 'Demo Brushless Controller' with subfolders like 'boards', 'core', 'description', 'library', 'meta', 'panels', 'schematics', 'simulations', and 'user'. The main workspace is divided into three overlapping windows:

- Board Editor:** Shows a top-down view of a PCB layout with various components and traces.
- Schematic Editor:** Displays a circuit schematic for a 'Brushless Controller'. It includes a motor driver IC (U1), a USB-to-UART bridge (U2), and various passive components like resistors (R1-R10), capacitors (C1, C2, C3), and diodes (D1, D2). The schematic is connected to a grid with power rails (VCC, GND) and phase labels (PHASE A, B, C).
- ERC (Error Checker):** Shows a list of errors and warnings. All counts are zero, indicating a clean design.

At the bottom of the schematic editor, a metadata box contains the following information:

Brushless Controller	16.01.16 23:22
Demo Brushless Controller Project	U. Bruhin
	Page 1/1