

# Introduction to KiCAD & Open Hardware

Katharina Fey

17. March 2018

# Before we get started

## A repository with resources

*<https://github.com/spacekookie/adaconf-kicad-workshop>*  
or <https://spacekookie.de/adaconf> (temp link)

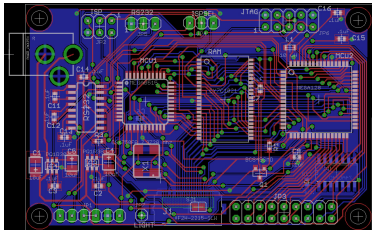
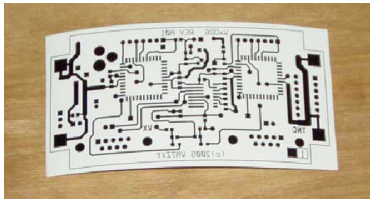
- ▶ PCB design is complicated. KiCAD is complicated
- ▶ Workshop & Slides meant as an introduction
- ▶ You will still have questions afterwards ;)

# Contents

- ▶ Introduction
  - ▶ What is hardware development
  - ▶ What is KiCAD?
- ▶ Workflow
  - ▶ Creating schematic symbols
  - ▶ Creating footprints
  - ▶ Working with schematics and boards
  - ▶ Updating designs
- ▶ Best Practises
  - ▶ Project management
  - ▶ Library/ Parts management
  - ▶ Datasheets
  - ▶ Part Selection

# Introduction

- ▶ Components, Boards and Firmware
- ▶ Board (PCB) design is "drawing"
- ▶ Open Hardware guideline
  - ▶ Schematics, Boards, Symbols, Footprints (as CAD files!)
  - ▶ Bill of material
  - ▶ Firmware (if any)
  - ▶ Mechanical CAD files (if any)
  - ▶ Documentation
  - ▶ Non-commercial license

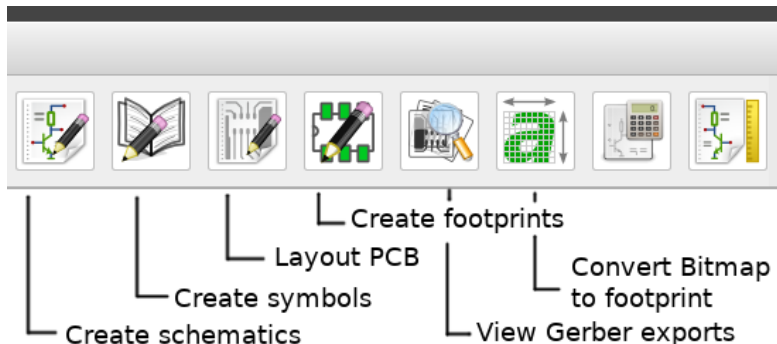


# KiCAD

- ▶ Free software
  - ▶ Developed & funded by CERN
- ▶ Current version 4.0.\*
- ▶ Next version (5.0) just around the corner
- ▶ Open ecosystem around components & footprints

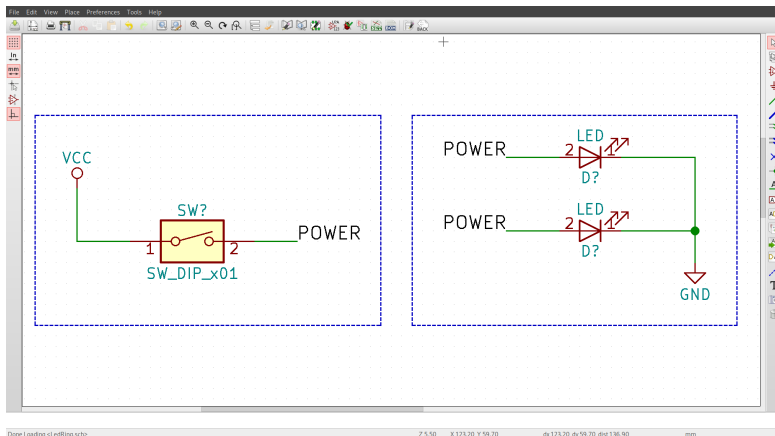
# Workflow

- ▶ Schematics (represent Circuits)
- ▶ Associate Footprints
- ▶ Layout boards & route traces
- ▶ Repeat for iterations



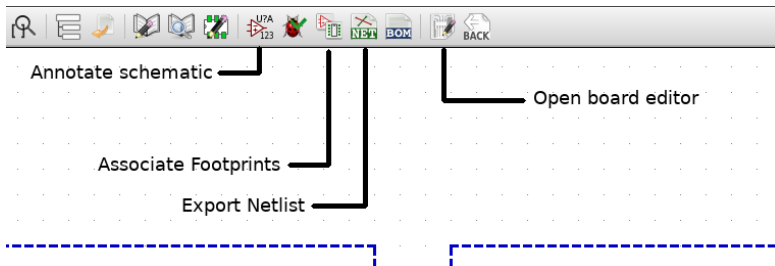
# Schematics

- Symbols (Create missing symbols)
- Labels
- Connections



# Schematics Workflow

- ▶ Create circuit schematic
- ▶ Annotate (Give components unique names)
- ▶ Associate footprints
- ▶ Export Netlist





# Board Layout

- ▶ Import Netlist
  - ▶ Imports component footprints
  - ▶ Can replace/ update/ remove old footprints
  - ▶ Import new Netlist for each revision
- ▶ Layout components
- ▶ Route traces
- ▶ Export
  - ▶ Gerber files (traces)
  - ▶ Drill holes (holes, edge cuts)
  - ▶ (Optional) Bill of Materials

# Best Practises

- ▶ Schematics
  - ▶ Group components into segments (in nice boxes)
  - ▶ Put large segments onto sub-sheets
  - ▶ Use labels to connect segments together
- ▶ Store required libraries in project
- ▶ Include datasheets for components

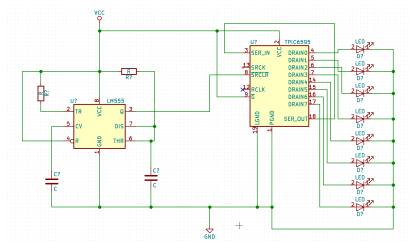


Figure: Avoid doing this!

# Understanding Datasheets

- ▶ Describes a component in detail
- ▶ Not everything always relevant
- ▶ Pick out important information
  - ▶ High-level description
  - ▶ Example usage
  - ▶ Pin-out
  - ▶ Footprint size

# Finding Datasheets & Components

- ▶ Use component categories & search on retail websites
  - ▶ Farnell (Excellent search, medium catalogue)
  - ▶ Digikey (Meh search, excellent catalogue)
  - ▶ Mouser (Good search, good catalogue, expensive!)
  - ▶ Adafruit (Limited catalogue, expensive!)
- ▶ Datasheets provided by retail websites
- ▶ Often includes additional documentation

# Best Practises: Library Management

- ▶ Three types of libraries
  - ▶ Schematic symbols (.lib)
  - ▶ Component footprints (.pretty)
  - ▶ (Optional) 3D models (.3dshapes)
- ▶ Different approaches to library management
  - ▶ Each have their pro's and con's
  - ▶ No "standard" way of doing it
  - ▶ One library per ...
    - ▶ Project
    - ▶ Component type
    - ▶ Manufacturer
    - ▶ Individual component