Schaltplan- und Layout-Erstellung mit freier Software

Easterhack 2008

Chaos Computer Club Cologne

Stefan Schürmans, BlinkenArea

stefan@blinkenarea.org

Version 1.0.1

Inhalt

- Schaltplaneditor „gEDA-gschem“
- Netzlisten-Erzeuger „gEDA-gnetlist“
- Layouteditor „PCB“
- Text-basierte Dateiformate
  - Schaltplan-Dateiformat „*.sch“
  - Layout-Dateiformat „*.pcb“
  - kreativer Umgang mit den Dateiformaten
- Weiter gemäß Ideen der Teilnehmer

Fragen bitte jederzeit stellen. Wir haben genug Zeit...

gEDA-gschem

- freier Schaltplaneditor (OpenSource)
  - incl. vieler Standard-Symbole
  - Erstellung von neuen Symbolen
  - Export in Bildformate und PostScript
- Bedienung
  - Zeigen und Klicken mit der Maus
  - Befehle über Menüs oder mit Tastendruck
    - Tastendruck-Bedienung ähnlich effizient wie in „vi“
- (noch) kein Profi-Tool, aber gut benutzbar

gEDA-gschem - Schaltplan

- Symbole
  - Name = Typ Nummer (z.B. R1, C1, IC1)
  - Wert (z.B. 10kE, 100nF, ATtiny2313, ...)
- spezielle Symbole
  - Potentiale: Masse, Betriebsspannung, ...
- Drähte
  - verbinden Pins der Symbole
  - Draht-Draht-Verbindungen mit Punkt
  - rotes Viereck zeigt offene Enden

gEDA-gschem – Tasten

ESC	Befehl abbrechen
z	Ansicht vergrößern (zoom in)
Z	Ansicht verkleinern (zoom out)
x	Ansicht zentrieren
w	Fenster vergrößern (window)
v e	alles anzeigen (view everything)
n	Draht (net) zeichnen
i	Symbol einfügen (insert)
e S	Bauteil-Stück wählen (edit slot)
U	Rückgängig (undo)
R	Wiederherstellen (redo)

DEL	löschen
c	kopieren (copy)
m	verschieben (m)
e r	drehen (edit rotate)
e i	spiegeln (edit mirror)
e e	Eigenschaften (edit edit)
e x	Text-Eigenschaften (edit text)
y u	Ausschneiden (cut)
y c	Kopieren (copy)
y p	Einfügen (paste)

Aufgabe: Schaltplan erstellen

- Elektronischer Würfel aus AVR-Workshop
  - Ausgangspunkt
    - Schaltplan als PDF
    - Symbole
  - Ziel
    - gEDA-gschem Schaltplan
    - erst einmal ohne Batterie
      - Symbol existiert noch nicht
  - Tipp
    - Bauteile einbetten
      - Änderungen an Symbolen zerstören sonst Schaltplan

gEDA-gschem - Symbole

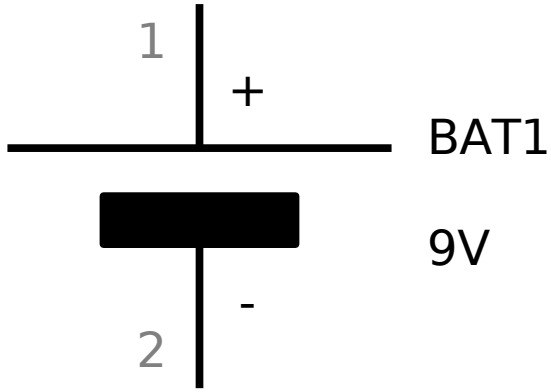
- Vektorgrafik
  - Linien,Rechtecke, Kreise, ...
- Pins
  - Nummer
  - Funktion
    - Eingang (in), Ausgang (out), Ein-/Ausgang (io)
    - Stromversorgung (pwr), passiv (pas)
- Bauteil = 1..N gleiche Symbole
  - 2..N: Probleme mit Pins zur Stromversorgung

gEDA-gschem – Tasten (Symb.)

H s	Symbol bearbeiten (hierarchy symbol)
H u	zurück zum Schaltplan (hierarchy up)
l	Linie zeichnen (line)
b	Kasten zeichnen (box)
ai	Kreis zeichnen
ap	Pin einfügen (add pin)
at	Text einfügen (add text)
aa	Attribut (z.B. Name, Wert) einfügen (add attribute)

Aufgabe: Symbol erstellen

- Batteriesymbol erstellen
  - Ziel
    - gEDA-gschem Symbol
    - zwei Pins vom Typ „passiv“



- Schaltplan vervollständigen
  - Batterie einfügen und verdrahten

gEDA-gnetlist

- Netzliste
  - Verbindungen zwischen Bauelementen
  - Netz = Liste verbundener Pins
    - Netz hat Name
      - automatisch
      - benutzerdefiniert (durch Attribute im Schaltplan)
- Erstellung einer Netzliste
  - `gnetlist -g PCB -o edice.net edice.sch`

PCB

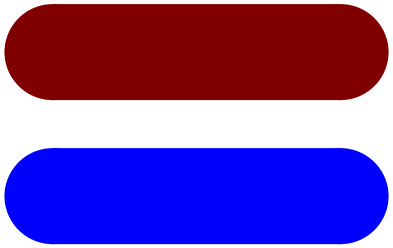
- freier Layout-Editor (OpenSource)
  - incl. vieler Standard-Bauelemente
  - Erstellung von neuen Bauelementen
  - Export in Gerber/RS-274X und PostScript
- Bedienung
  - Zeigen und Klicken mit der Maus
  - Befehle über Menüs oder mit Tastendruck
    - Tastendruck-Bedienung ähnlich effizient wie in „vi“
- (noch) kein Profi-Tool, aber gut benutzbar

PCB - Layout

- Bauelemente
  - Name = Typ Nummer (z.B. R1, C1, IC1)
  - Gehäuse (TO92, DIP16, SO16, 0805, ...)
    - Position der Pins
- Verbindungen
  - Luftlinien (rat lines)
    - noch fehlende Verbindungen aus Netzliste
  - Leiterbahnen (lines)
  - Dukos = Durchkontaktierungen (vias)

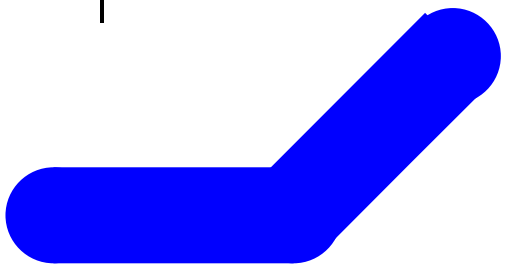
PCB – Leiterbahnen (lines)

- Ebene (layer)
  - Bestückungsseite (component)
  - Lötseite (solder)



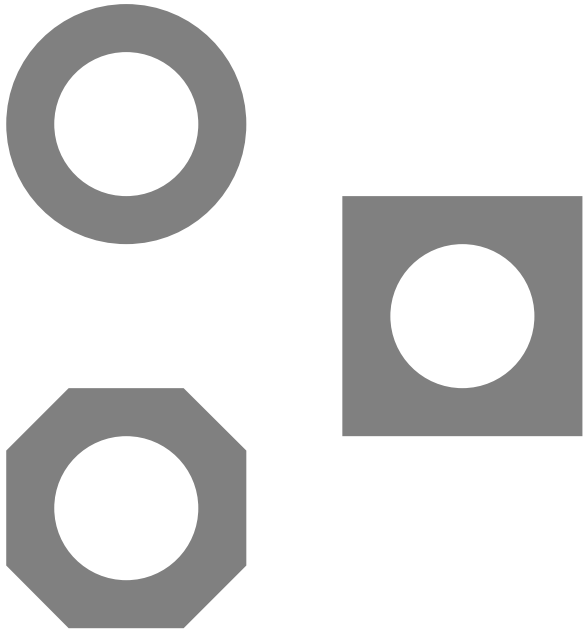
- Breite (width), Abstand (clearance)

- Verlegung
  - keine scharfen Ecken
  - ideal: nur in 45°-Richtungen

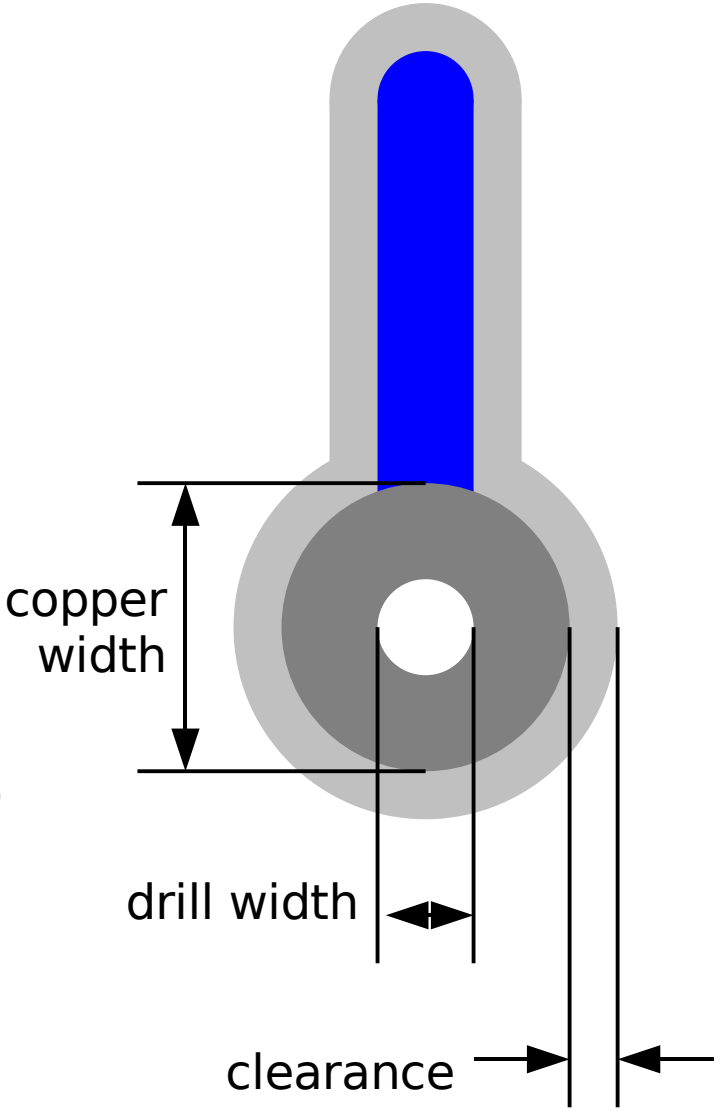


PCB – Dukos (vias)

- Form
  - rund
  - quadratisch
  - acht-eckig

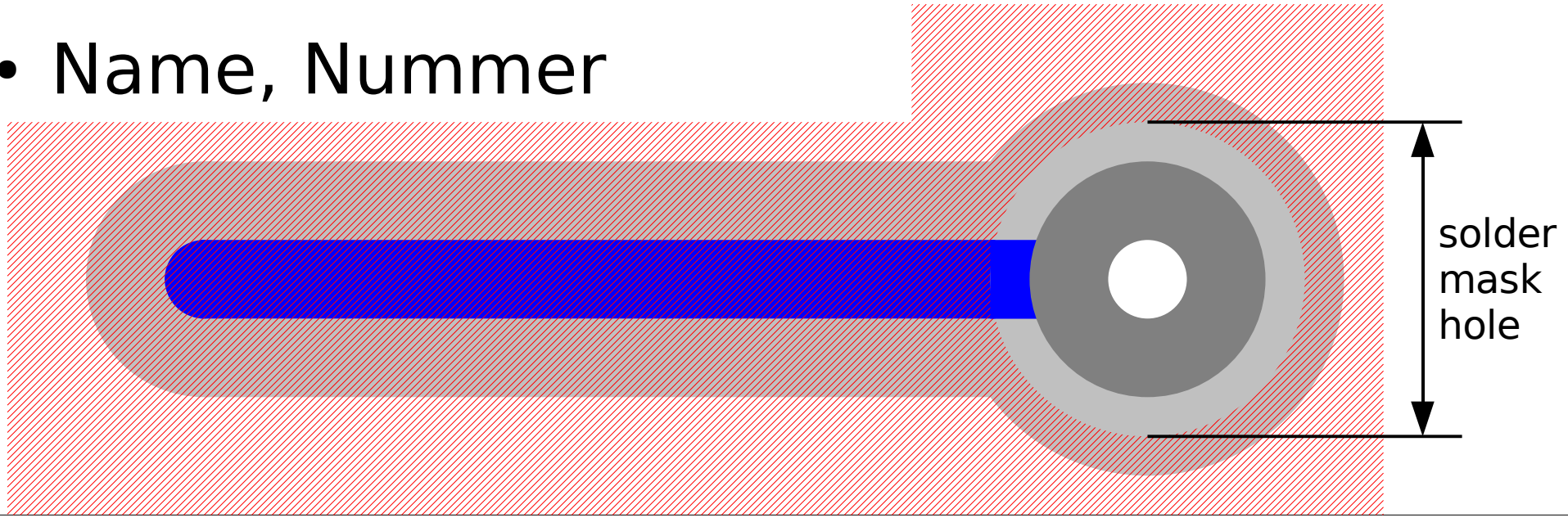


- Größe
  - Kupfer-Größe (copper width)
  - Bohrungs-Größe (drill width)
  - Abstand (clearance)



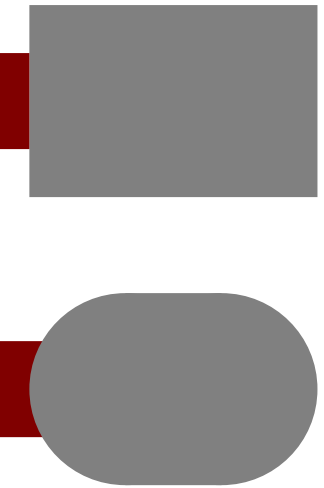
PCB – Pins (pins)

- Form und Größe wie bei Dukos
  - Pin 1 hat oft andere Form
- Lötstoppmasken-Loch (solder mask hole)
  - Bereich ohne Lötstopplack
    - Pins müssen lötbar sein
- Name, Nummer



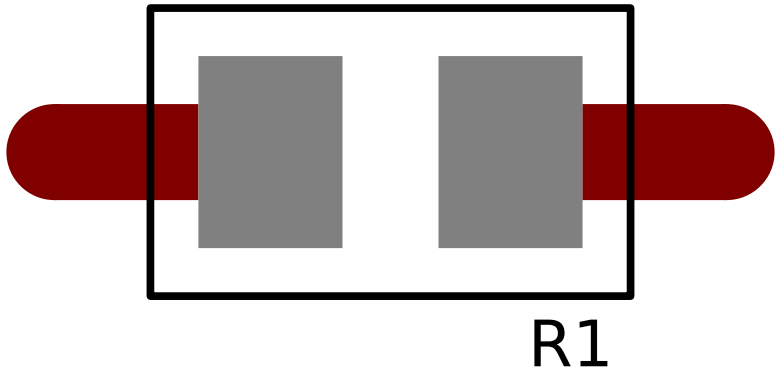
PCB – Lötfelder (pads)

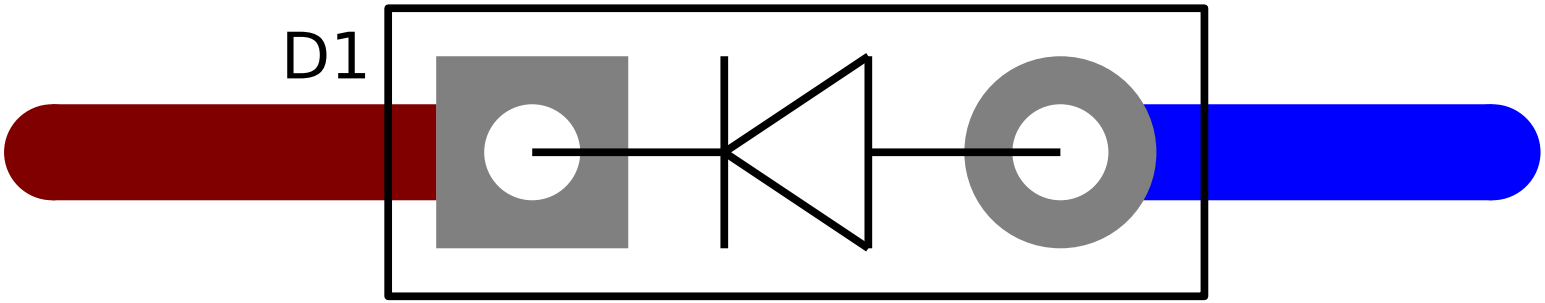
- Form
  - rechteckig, abgerundet
- Größe
  - Breite (width)
  - Anfangspunkt (begin)
  - Endpunkt (end)
- Lötstoppmasken-Breite (solder mask width)
- Name, Nummer



PCB – Bauelemente (elements)

- Pins, Pads
  - mit Name, Nummer
    - unsichtbar
- Bestückungsdruck (silk)
  - Kennzeichnung Bauteilumriss
- Name = Typ Nummer (z.B. R1, C1, IC1)





PCB – Tasten 1/2

ESC	Befehl abbrechen
z	Ansicht vergrößern (zoom in)
Z	Ansicht verkleinern (zoom out)
c	Ansicht zentrieren (center)
w	Fenster vergrößern (window)
v	alles anzeigen (view)
DEL	Auswahl löschen
^X	Auswahl ausschneiden
^x	Auswahl kopieren
TAB	Ansicht von anderer Seite (von oben, von unten)

F1	Duko
F2	gerade Leiterbahn
F3	gebogene Leiterbahn
F4	Text
F5	Rechteck
F6	Polygon
F7	Einfügen
F8	Löschen
F9	Drehen
F11	Auswählen
u	Rückgängig (undo)
R	Wiederherstellen (redo)

PCB – Tasten 2/2

|	normal / dünn darstellen
.	alle Richtungen / nur 45°
/	Linien-Modus umschalten
1	Bestückungsseite auswähen
2	Lötseite auswählen
3..8	Zwischen-Ebenen auswählen
m	Ebene ändern
M	Ebene in Auswahl ändern
q	rund / quadratisch
^O	rund / acht-eckig
a	Eigensch. des Leiters verwenden
n	Name ändern
h	Name sichtbar / unsichtbar
H	Namen in Auswahl (un)sichtbar
b	Bauteil auf andere Seite
B	Auswahl auf andere Seite
e	Luftlinien löschen
o	Luftlinien einfügen / optimieren
f	Verbindung hervorheben
F	Verbindungs-Hervorhebung aus

Aufgabe: Layout erstellen

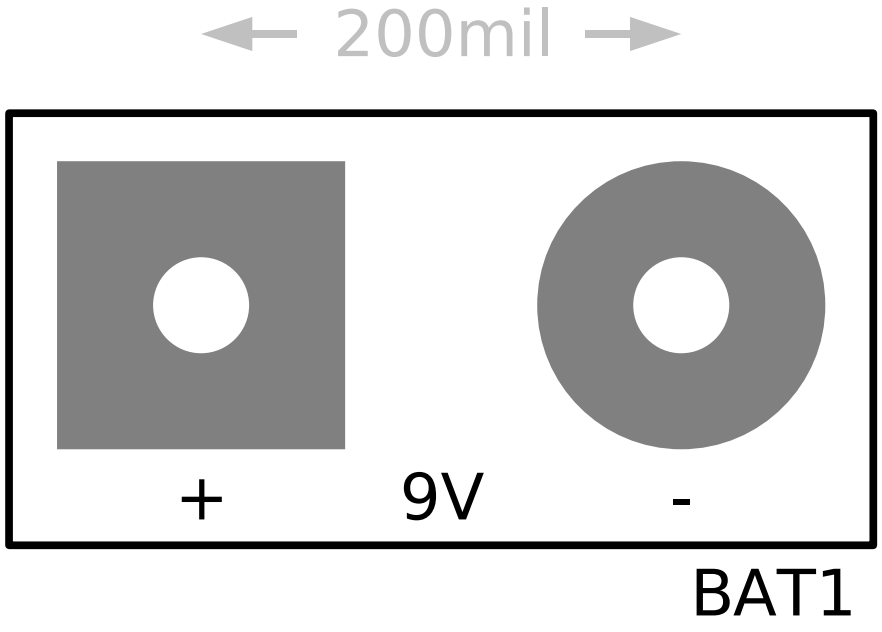
- Elektronischer Würfel aus AVR-Workshop
  - Ausgangspunkt
    - Netzliste
      - aus Schaltplan erzeugt
    - Bauteile
  - Ziel
    - PCB Layout
    - erst einmal ohne Batterie, Taster
      - Bauteile existieren noch nicht

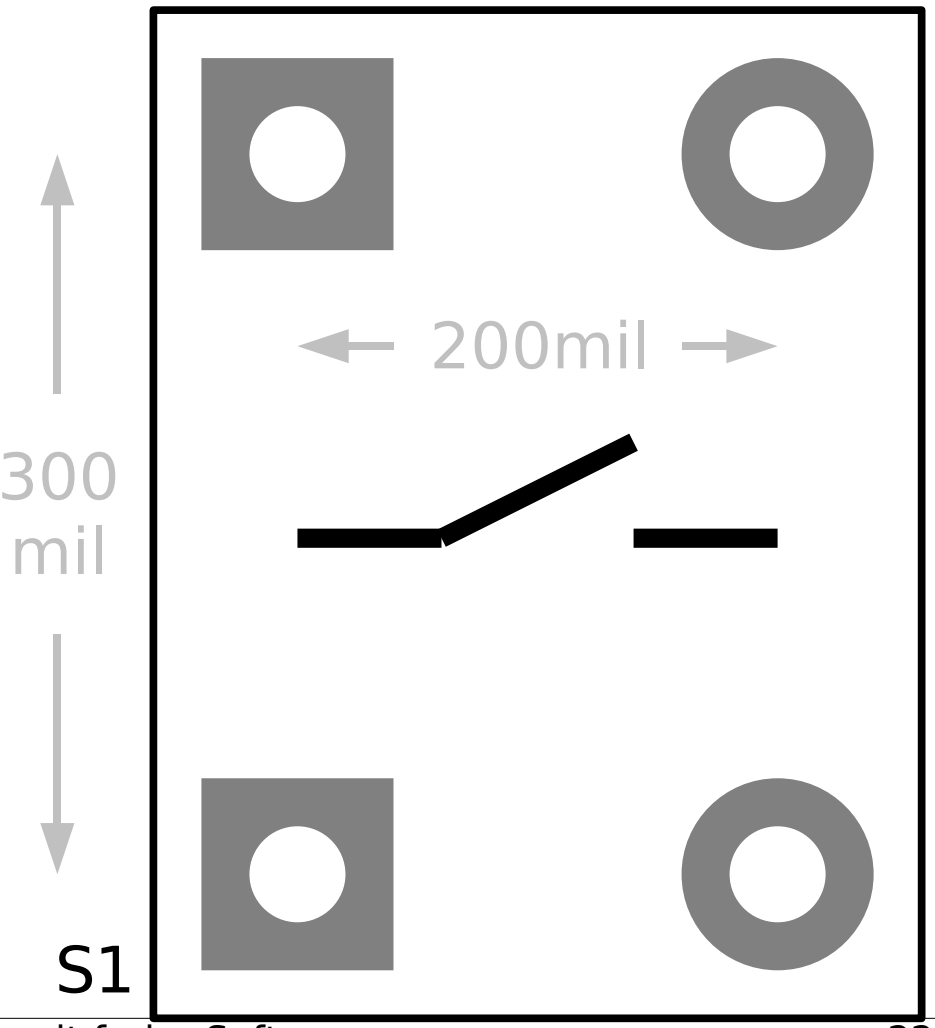
PCB – Bauelement erstellen

- Bauelement als Layout zeichnen
  - Pins als Vias, Pads als Leiter
    - in Reihenfolge der Pin-Nummem
    - mit Namen
  - Bestückungsdruck für Bauteilumriss
- Objekte des Bauelements ausschneiden
  - Referenzpunkt wird Ursprung des Bauteils
- Buffer in Bauelement konvertieren
  - Menü: Buffer, Convert buffer to element

Aufgabe: Bauelemente erstellen

- fehlende Bauelemente erstellen
  - für Batterie-Anschluss
  - für Taster
- Layout komplettieren





Schaltplan-Dateiformat „*.sch“

- Text-basiertes Format
  - dokumentiert
    - `/usr/share/doc/geda-doc/wiki/geda_file_format_spec.html`
  - Beispiel-Zeilen
    - Linie
      - `L <x> <y> <param> ...`
    - Netz
      - `N <x1> <y1> <x2> <y2> ...`
    - Text-Attribut
      - `T <x> <y> <param> ... 1 <name>=<value>`
    - Pin
      - `P <x> <y> <param> ...`
        `{`
        `...`
        `}`
    - Symbol
      - `C <x> <y> <param> ...`
        `[`
        `...`
        `]`

Schaltplan-Dateiformat „*.sch“

```
V 20061020 1
C 14300 8100 1 90 0 EMBEDDEDres.sym
[
P 14000 9000 14000 8850 1 0 0
{
T 13950 8900 5 8 0 1 90 0 1
pinnumber=2
T 14000 9000 5 10 0 0 90 0 1
pintype=pas
}
P 14000 8100 14000 8252 1 0 0
...
B 13900 8250 200 600 3 0 0 0 -1 -1 0 -1 -1 -1 -1 -1
T 13950 8500 5 10 0 0 90 0 1
device=resistor
T 13800 8550 8 10 0 1 90 3 1
refdes=R?
T 14200 8550 8 10 0 1 90 5 1
value=?E
]
N 10300 3300 10300 3100 4
...
```

Netzlisten-Dateiformat „*.net“

- beschreibt verbundene Pins
- eine Zeile pro Netz
  - Name Bauteil-Pin Bauteil-Pin ...
    - Zeilenfortsetzung mit Backslash am Ende
- Beispiel
    - `GND  BAT1-2 C1-2 IC1-2 C2-2 LED1-1 J1-1 \`
    - `  J2-1 J3-1 J4-1`
    - `VIN  BAT1-1 V1-1 IC1-3`
    - `VCC  IC1-1 C2-1 R1-1`
    - `unnamed_net1 LED1-2 R1-2`

Layout-Dateiformat „*.pcb“

- Text-basiertes Format
  - dokumentiert
    - `info pcb 'File Formats'`
  - Aufbau
    - Header
    - Zeichensatz
    - Dukos
    - Bauelemente
      - Pins, Pads, Umriss
    - Ebenen
      - Leiter, Flächen
    - Netze

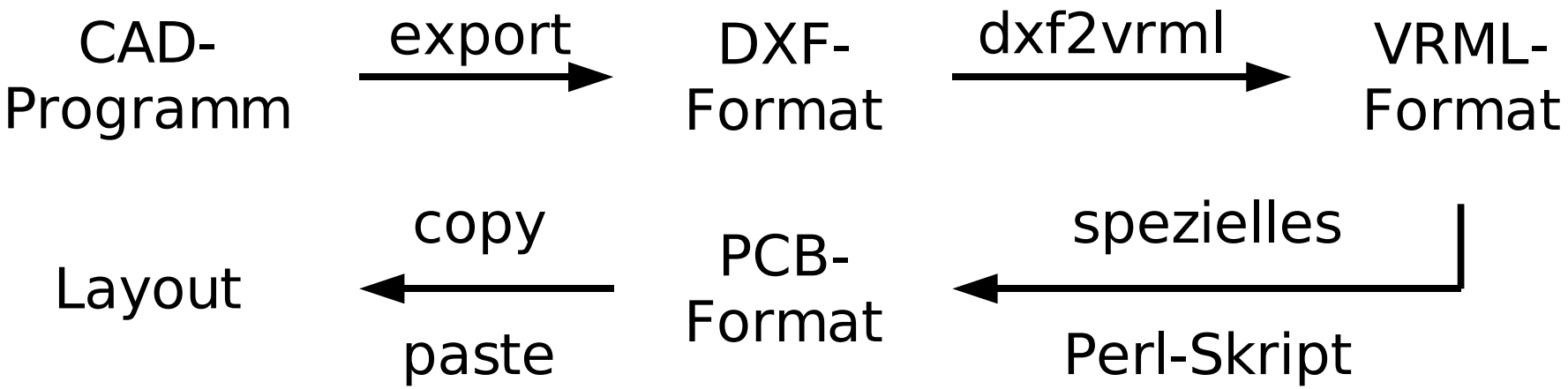
Layout-Dateiformat „*.pcb“

```
PCB["mypcb" 200000 100000]

Grid[2500.000000 0 0 1]
Cursor[107158 78182 1.731727]
PolyArea[200000000.000000]
Thermal[0.500000]
...
Groups("1,c:2,s:3:4:5:6:7:8")
Styles["Signal,2000,4000,2000,1150:Power,3000,5000,2500,1150:...]
...
Via[47500 72500 5000 2300 0 2500 "" ""]
...
Element["" "SMD" "R8" "0805" 110000 80000 -7500 -10500 0 100 ""]
(
        Pad[3200 -1700 3200 1700 3600 2300 4200 "1" "1" "square"]
        Pad[-3200 -1700 -3200 1700 3600 2300 4200 "2" "2" "square"]
        ElementLine [6000 -4500 6000 4500 1000]
        ElementLine [-6000 -4500 6000 -4500 1000]
        ElementLine [-6000 -4500 -6000 4500 1000]
        ElementLine [-6000 4500 6000 4500 1000]
        )
...
```

kreativer Umgang mit Dateiformaten

- Clearance per Skript vereinheitlichen
- LED-Positionierung bei GroggyClock
  - Uhr mit 60 LEDs als „Sekunden“
  - Problem: Positionierung der LEDs im Layout
  - Lösung: Übernahme aus mech. Zeichnung



Ideen der Teilnehmer

- Fragen?
- Layout für Würfel in SMD-Technik?
  - evtl. direkte Umsetzung
- Gerber/RS-274X-Format?
  - Export aus PCB
  - Ansehen mit „gerbv“
- sonstiges?