30 October 2015

INFO-0064 Embedded systems

Exercise session 6

Electronic design

Kevin Hogan

kevin.hogan@ulg.ac.be

Planning for today

- Electronic design
- Electronic schematics
- Printed Circuit Board
- gEDA
- Best practices
- Example of what not to do
- Debugging procedure

Electronic design

- What not to do

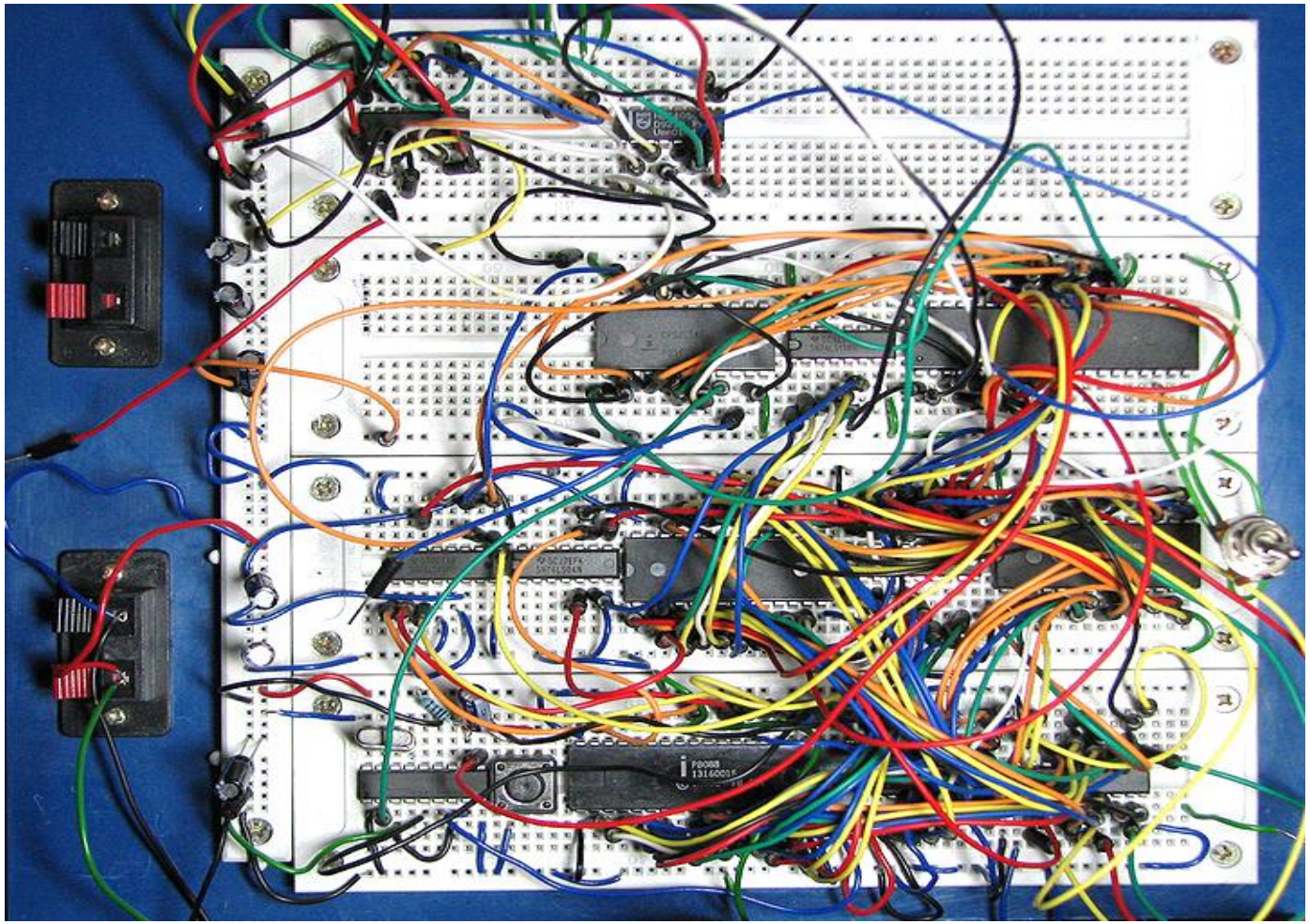


Image source : wikimedia commons

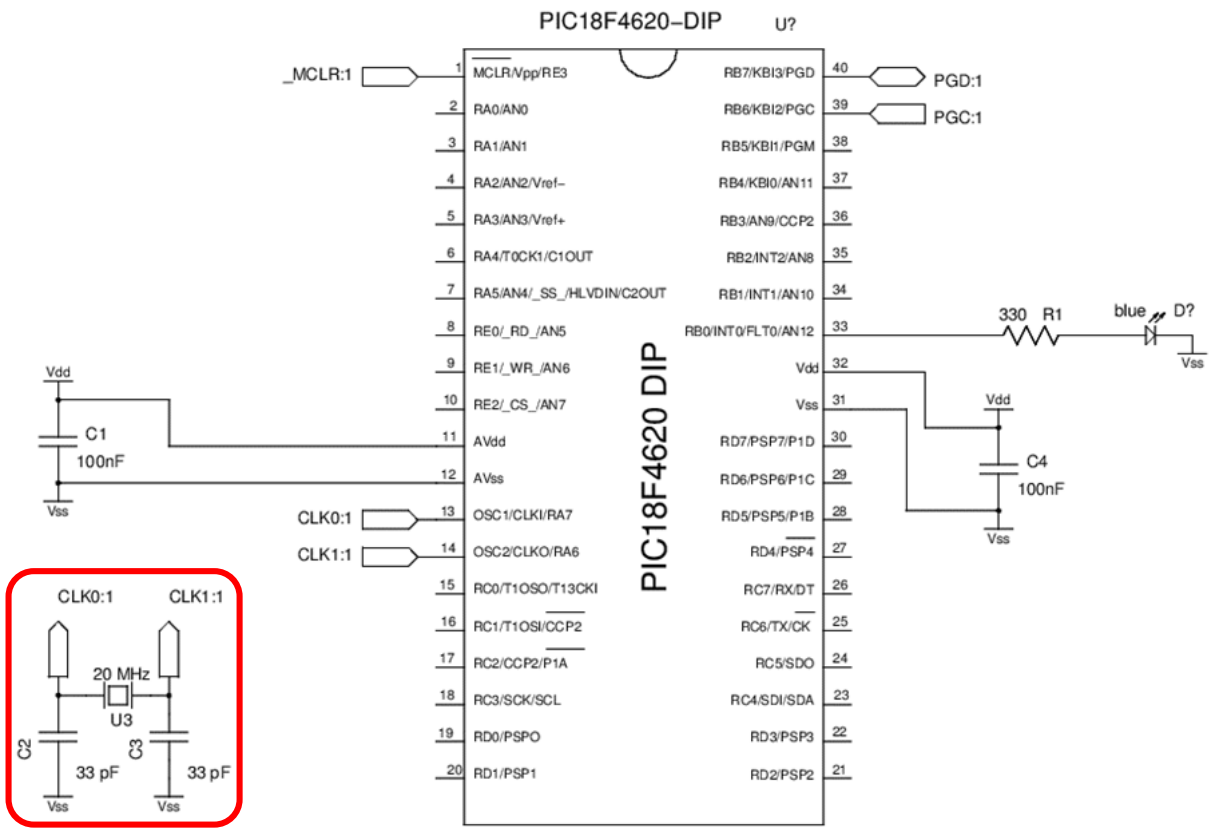
Electronic design

- Electronic circuit design

1. System specifications & constraints
2. Define a global solution
3. Conceive an electronic schematic that meets specifications
4. Prototype realization & tests (return to 2 if needed)
5. Design & realization of a **printed circuit board** (pcb)

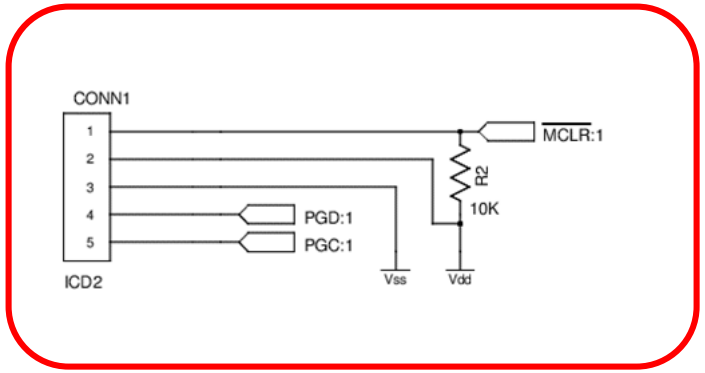
Electronic schematic

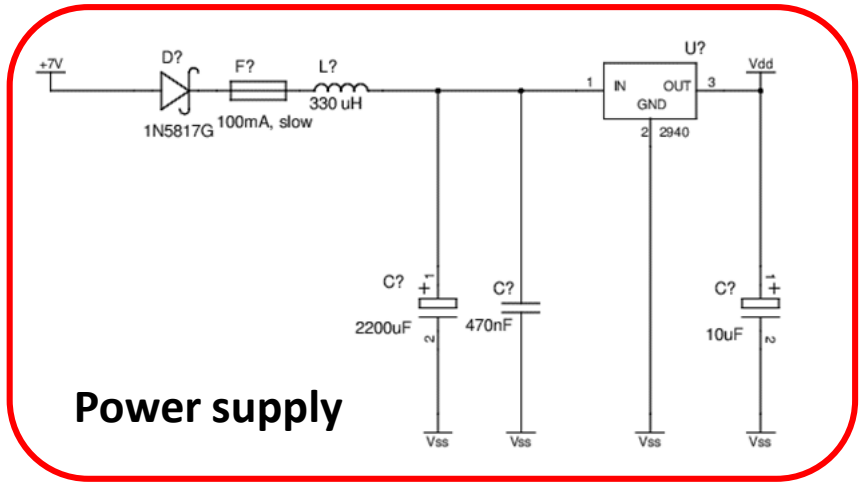
- Example: Blinky



Oscillator

Programming





Power supply

Electronic schematic

Schematic = elements + connections (*netlist*)

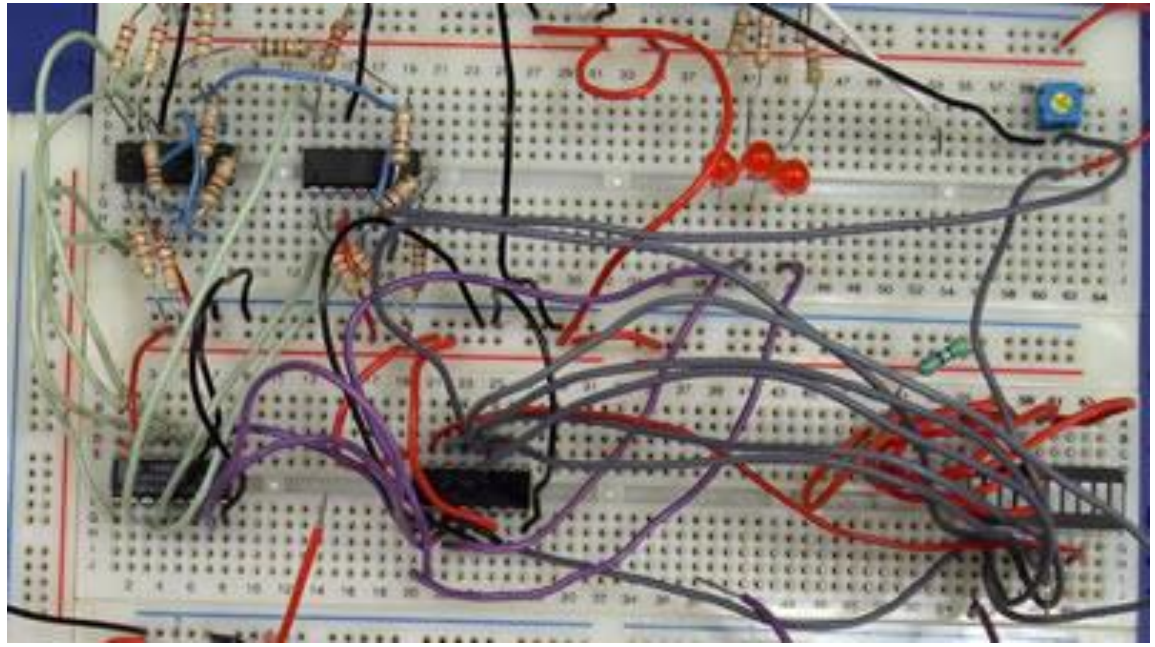
➔Purpose: Clear <u>understanding</u> of the circuit

- Displays all the elements
- Displays all the connections between elements
- Does not display the physical locations
- Divided in functional blocs

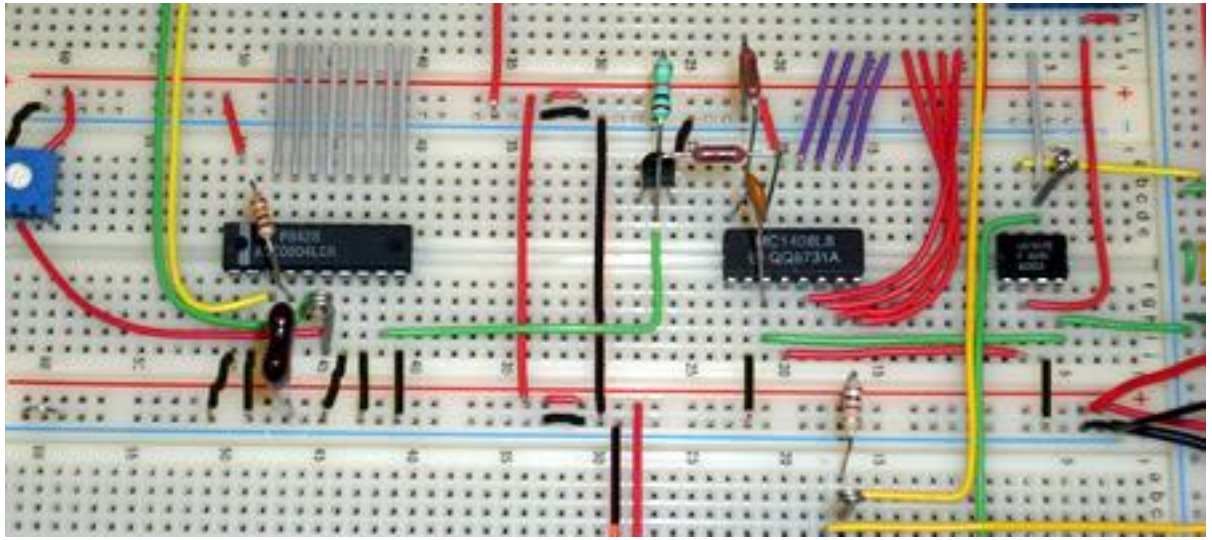
Prototyping

- Use a breadboard
- Try to make it as clear and clean as possible

Avoid that

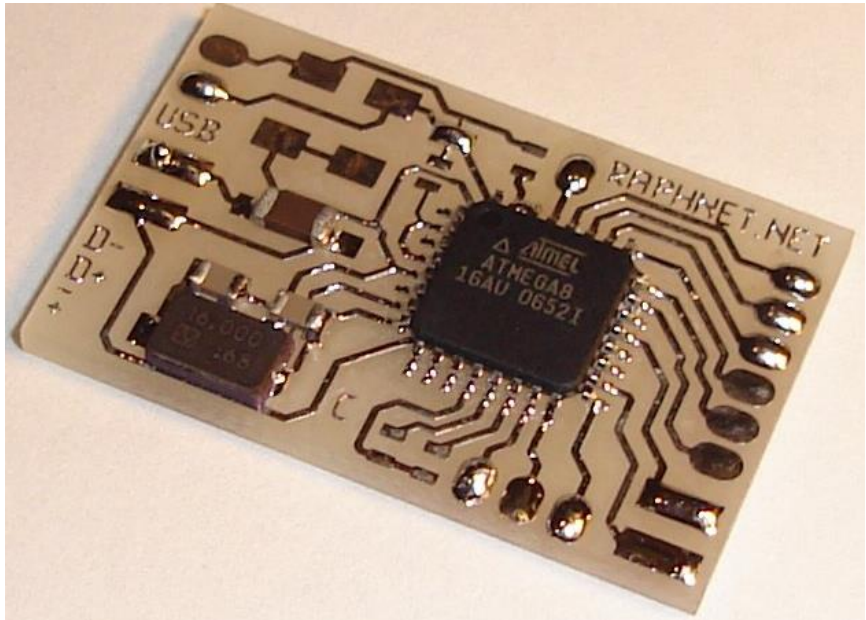


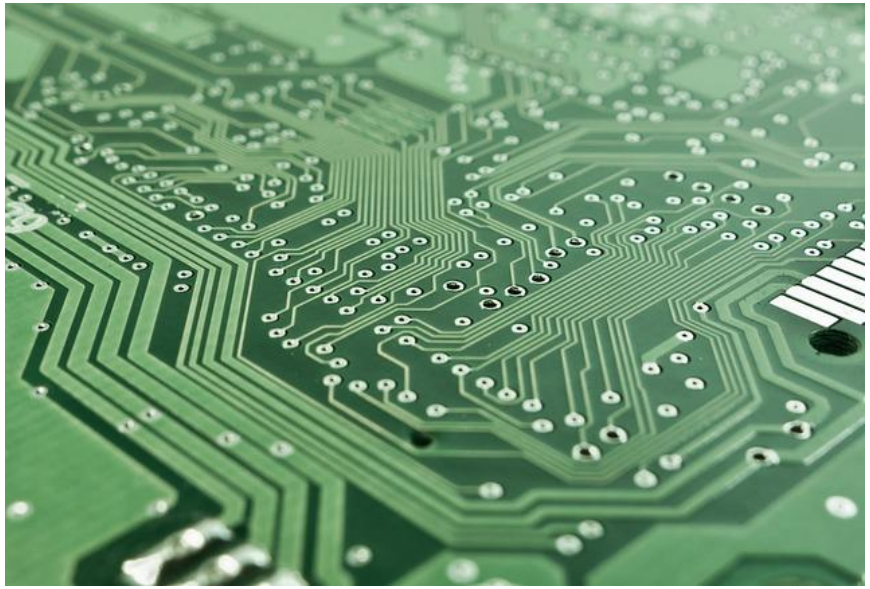
That’s what we want



Printed Circuit Board

- Holds mechanically all the components
- Performs all the electrical connections
- May perform an active role (antenna, ground plane, …)
- Composed of several layers
- Easy and cost-efficient production

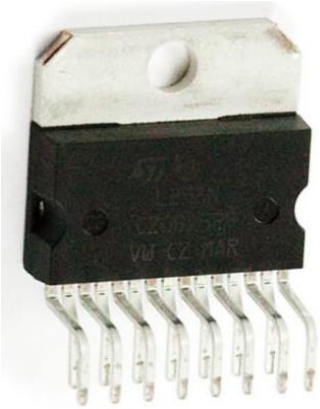


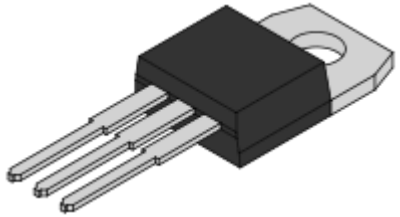


Packaging types

Through-hole technology

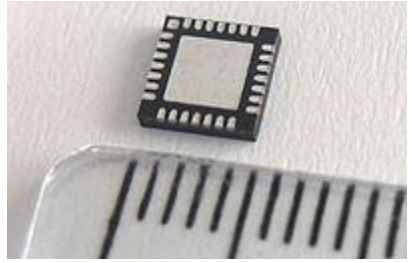
Dual In-line Package (**DIP**)





Transistor Outline (**TO-220**)

Surface-mount technology



Dual/Quad-Flat No-leads (**DFN/QFN**)

Quad Flat Package (**QFP**)

Small Outline Integrated Circuit (**SOIC**)

Surface-mount components (**SMC)**

gEDA

gpl **E**lectronic **D**esign **A**utomation

- Open-source software
- Manages all the process from schematics to pcb
- Runs on Linux, Mac and (a bit) on Windows
- Full suite and toolkit

gEDA

- Full suite and toolkits:
  - gschem: Schematics design
  - pcb: PCB design
  - gerbv: Gerber file viewer
  - ngspice: SPICE circuit simulator
  - Etc.

You can also check out KiCad, another open source EDA software

gEDA

- Three steps process

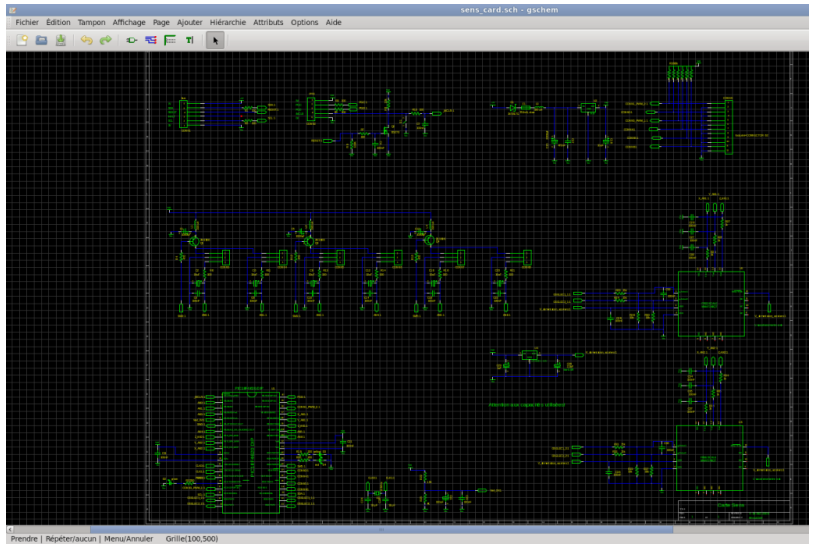
Schematics design
gschem

→

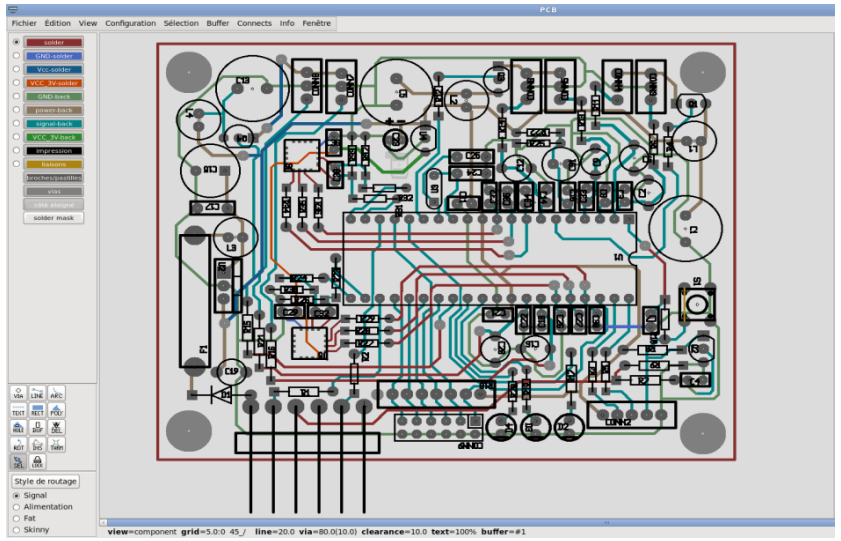
Conversion from schematics to PCB
gsch2pcb

→

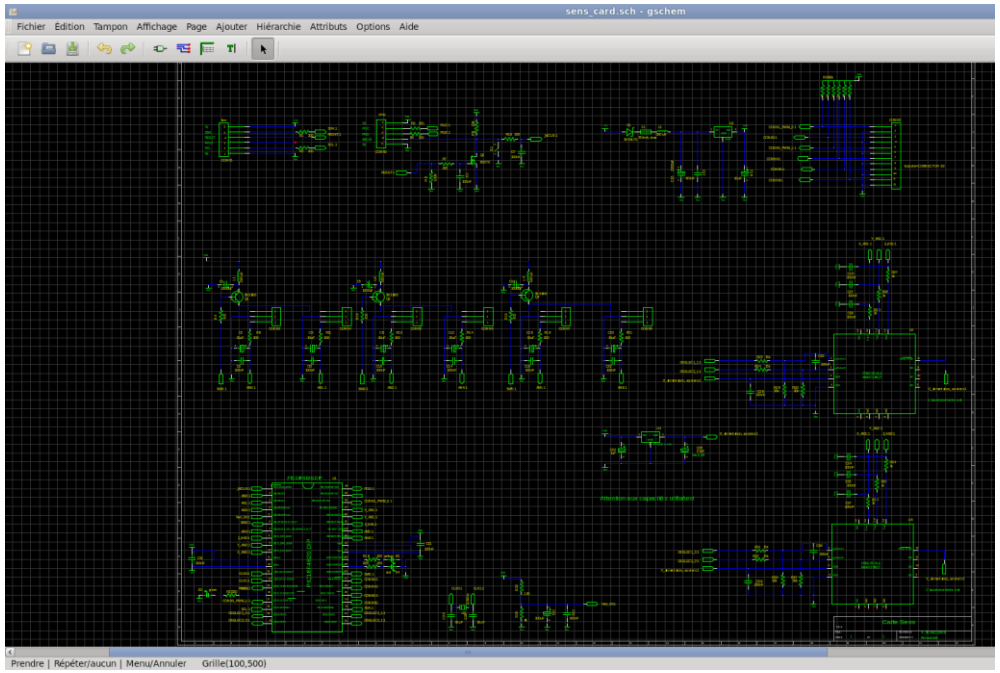
PCB design
pcb



```
$ gsch2pcb myfile.sch
```



gEDA



1. Schematic design - *gschem*

- An element = 1 (or more) symbols
- Already contains many elements and symbols libraries
- For more ➔ http://www.gedasymbols.org/
- To create your own library with your own symbols:
  1. Create a text file named "gafrc" in /home/.gEDA/
  2. Inside “gafrc”:

     `(component-library-search "directory_address" "library_name")`
  3. Place your .sym files in *directory_address*

gEDA

1. Schematic design - *gschem*

- Create your own symbols
  1. From an existing symbol: modify an already existing .sym file
  2. From scratch
- !! Pinout !! Be sure to be in agreement with the datasheet!

gEDA

1. Schematic design - *gschem*

- Each element must have a refdes

➔Leave the “?” (R?, C?, …) till the end and use `autonumber`

- **Always specify the element footprint** (edit attribute footprint)

➔It associates the element to its physical package

gEDA

2. From schematics to PCB – *gsch2pcb*

- In a terminal: run `$ gsch2pcb myfile.sch`

➔Creates the netlist, the .pcb file with all the footprints

- Check the instructions given in the terminal (missing footprint, …)

gEDA

2. From schematics to PCB – *gsch2pcb*

- You can also use a project file

➔Add your own footprints, mix different schematics, …

1. Create a text file `myproject` and place the following inside:

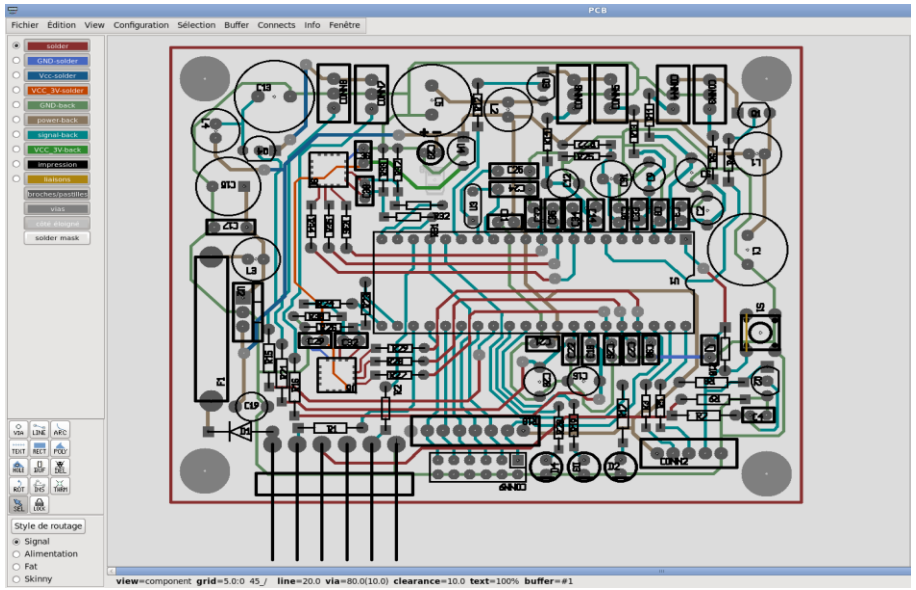
```
schematics myfile1.sch myfile2.sch
elements-dir /footprint_directory1_address/
elements-dir /footprint_directory2_address/
output-name mypcb
```

2. In a terminal: run `$ gsch2pcb myproject`

➔Creates a `mypcb.pcb` file using `myfile{1,2}` and footprints from `/footprint_directory{1,2}_address/`

gEDA

3. PCB design – *pcb*



- Define elements & connections physical locations
- Commonly used unit: **mil**

  1 mil = 1/1000 inch = 0,0254 mm
- Create your own footprint
  - !! Use element datasheet !!
  - !! Pin/pads numeration !!
  - ALWAYS print a 1:1 paper version to check dimensions
  - **One mistake = all the board is useless**

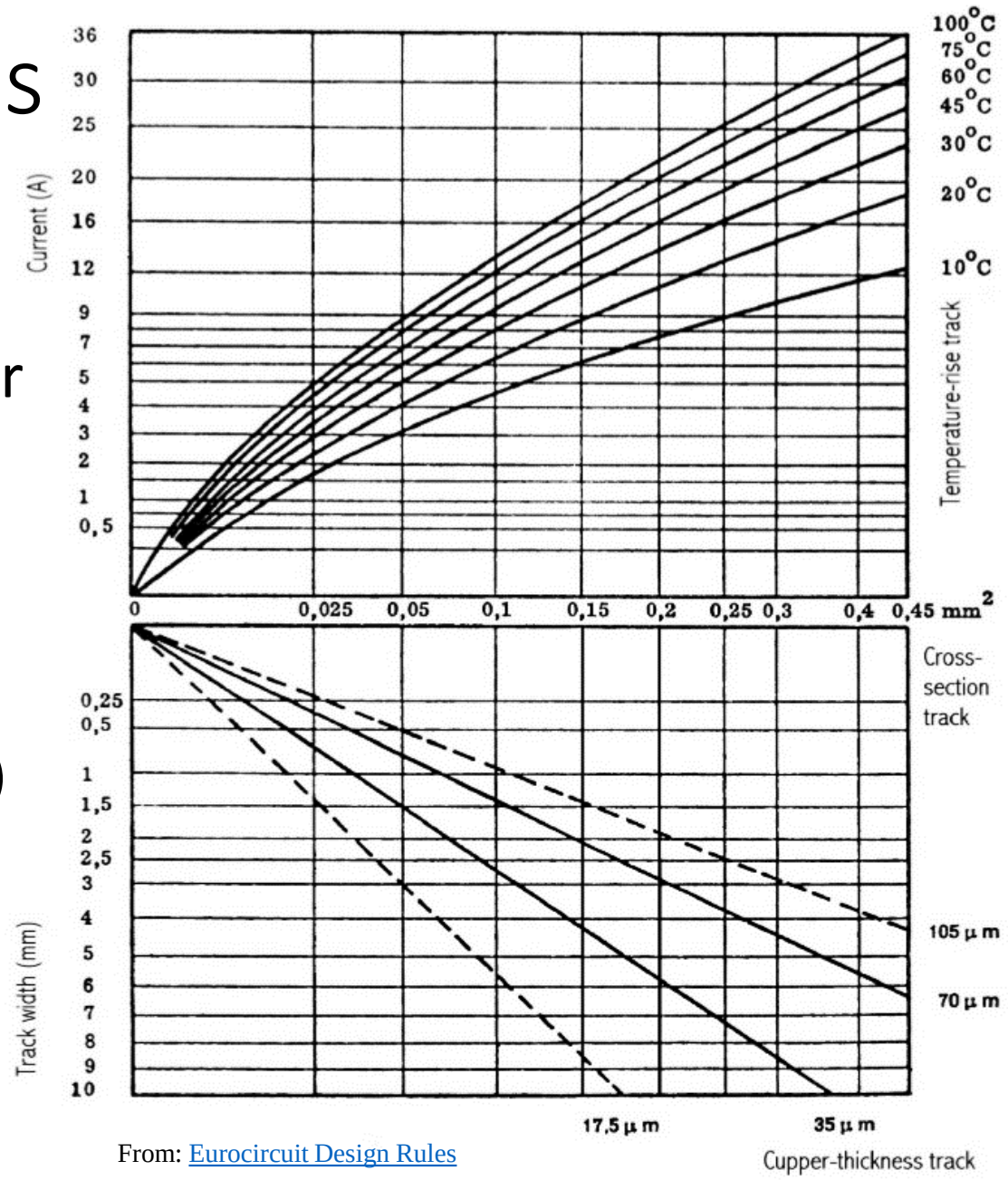
PCB Design – best practices

General

- Always write PCB & author name on the board
- Size tracks width to the current
- Space elements
- Place several LEDs or buttons for debugging
- Print a 1:1 paper version to check that dimensions are correct

PCB Design – best practices

- Size tracks width to the current
- It may also depend on the manufacturer design rules
- If realization in Montefiore [mils]
  - Normal track width: 24 (min 16)
  - Power track width: 30
  - Via: 82x82 (min 70x70)
  - Via hole: 18
  - Clearance: 12



From: [Eurocircuit Design Rules]

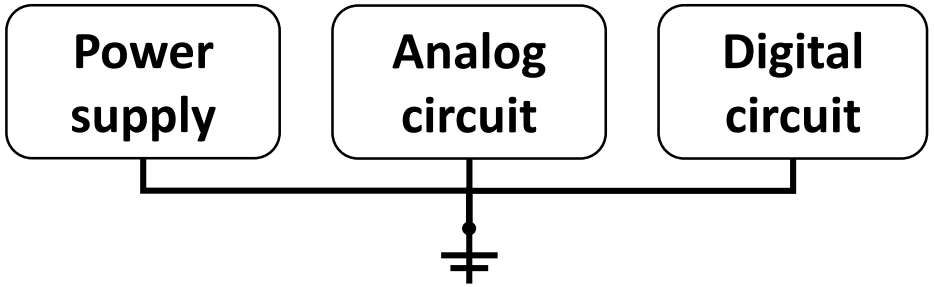
PCB Design – best practices

- Leave a 3mm spacing between circuit and board outline
- Hole in every corner (i.e. place the PCB on screws)
- Decoupling capacitors have to be as close as possible to the component they protect
- !! Ground loop = current going through the ground between two points ➔ potential difference
- Ground star topology

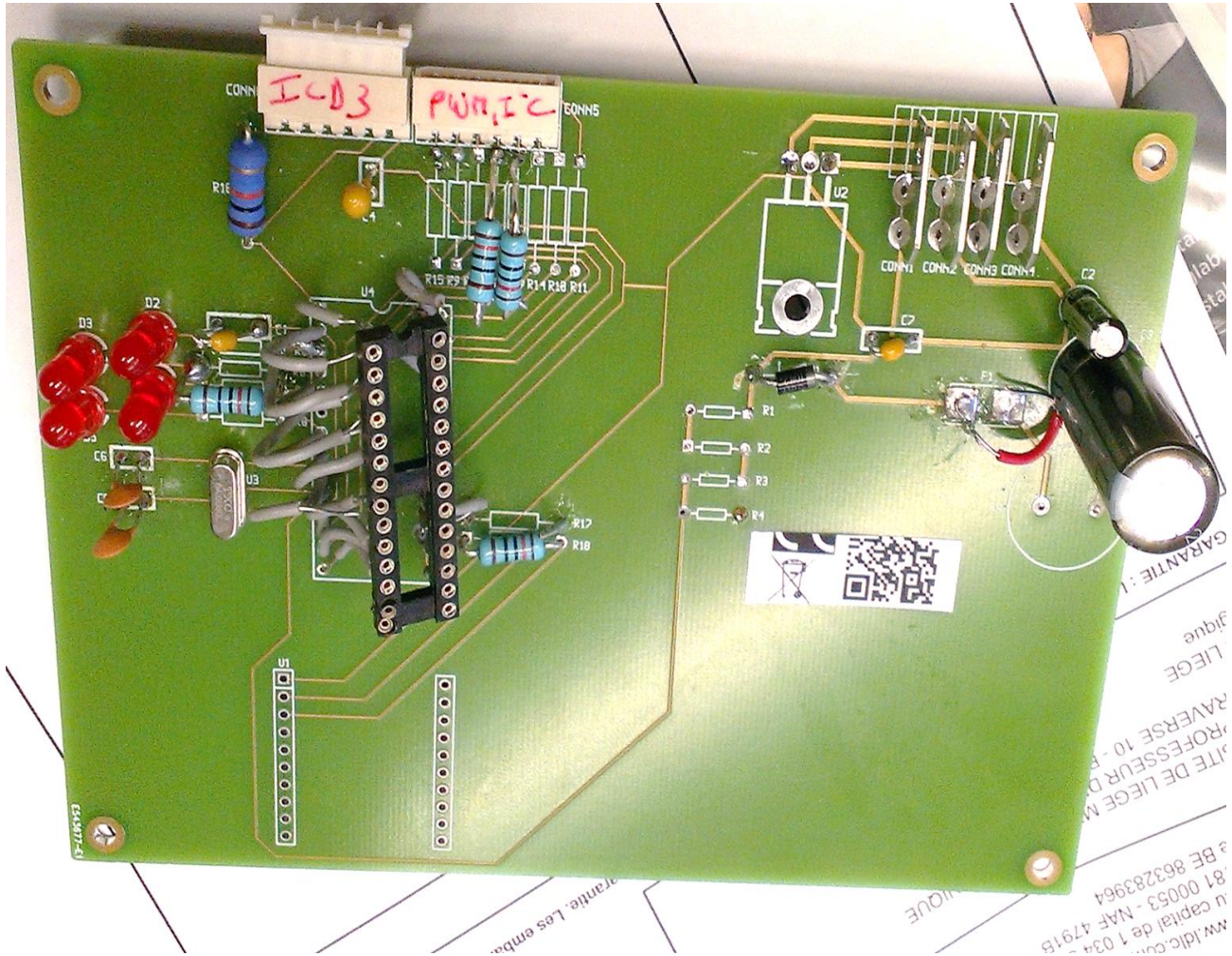
NO YES YES

TO AVOID

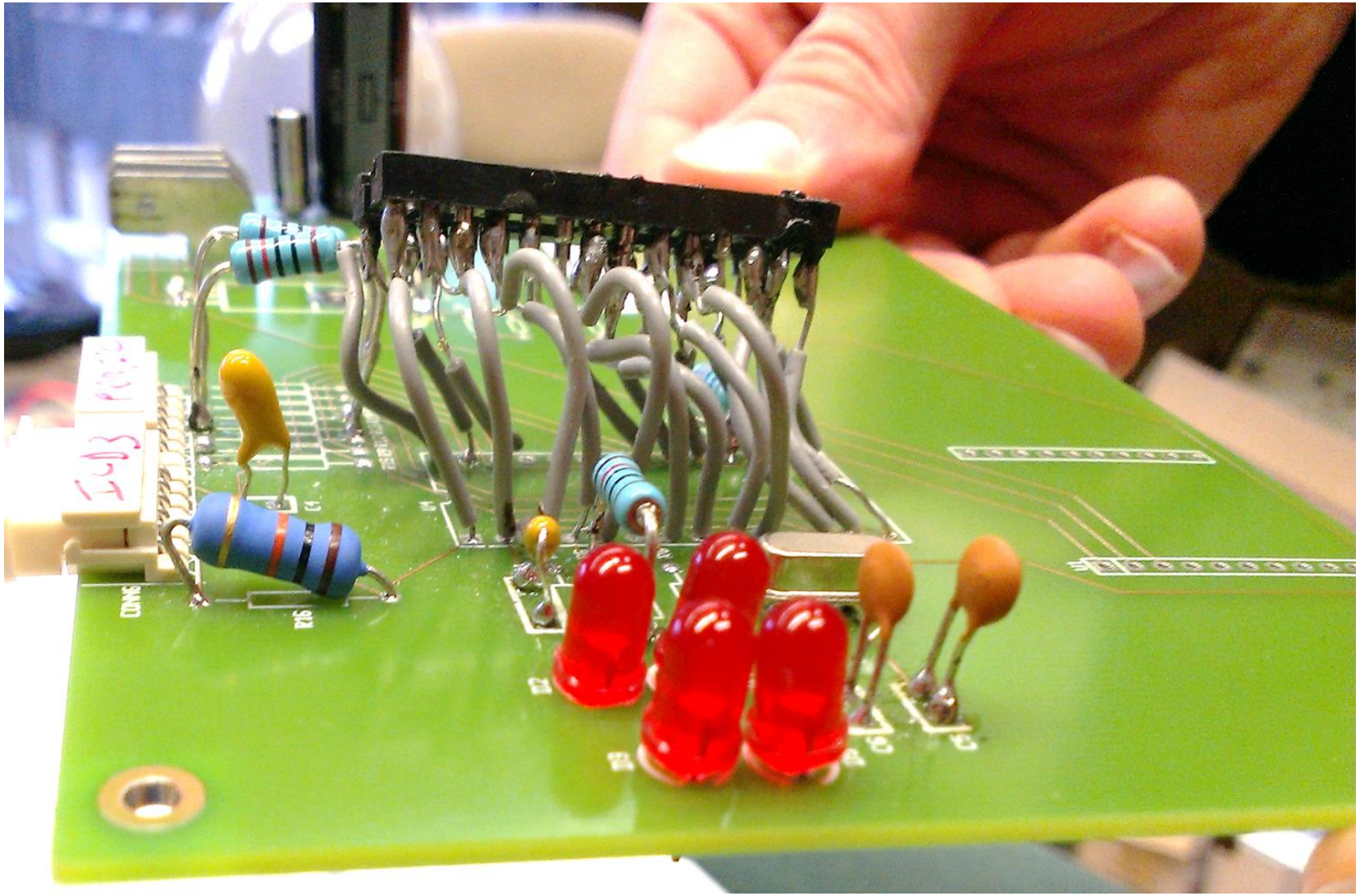
➔ USE THE 'GRID'



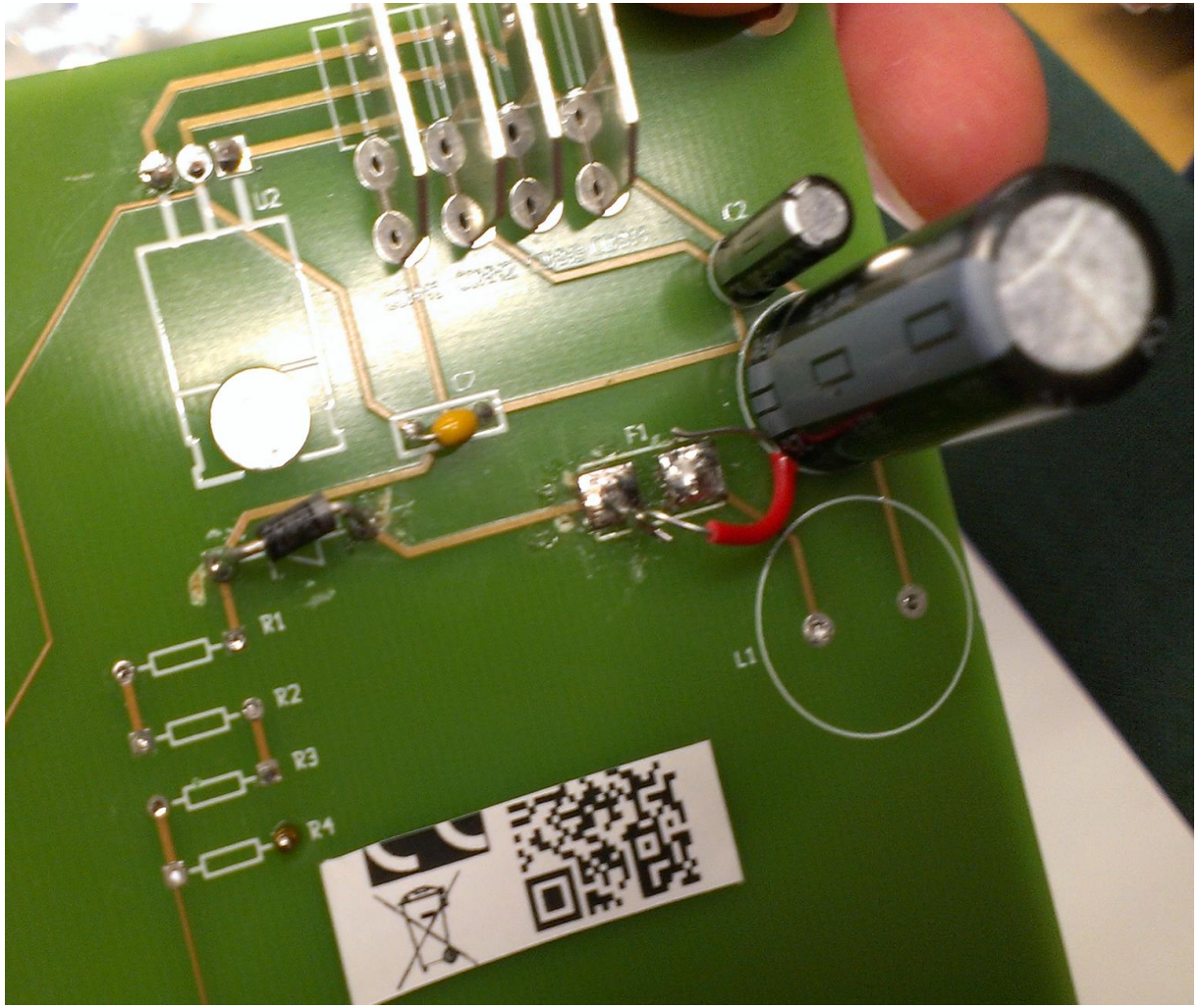
Example: What not to do



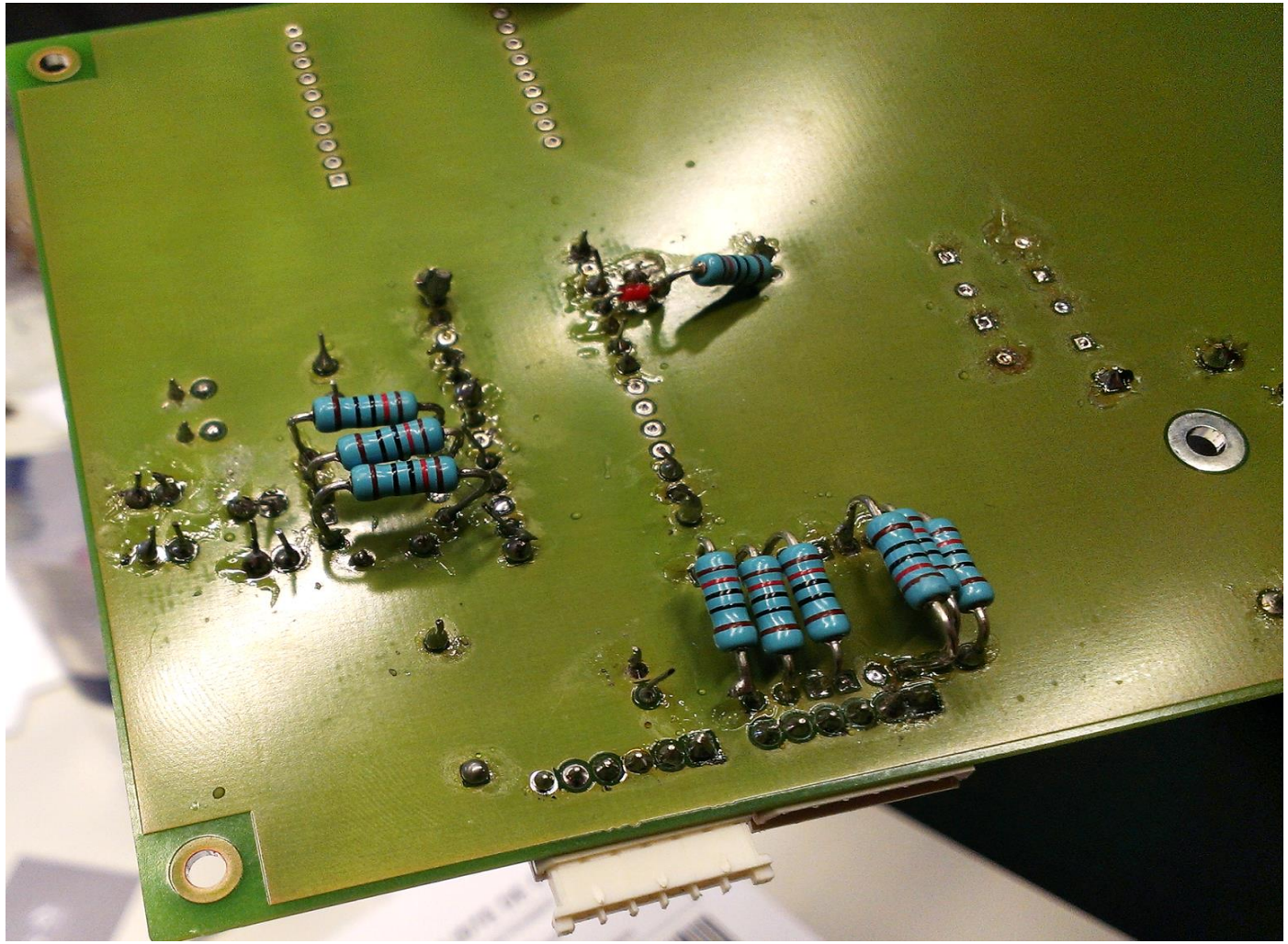
Example: What not to do



Example: What not to do



Example: What not to do



Debugging procedure

Trust no one!

- Keep calm
- Use the oscilloscope
- Check **elements connections & pinout**
- Check soldering
- Check the power voltage
- Check pin voltages
- Process step-by-step and repeat this procedure