

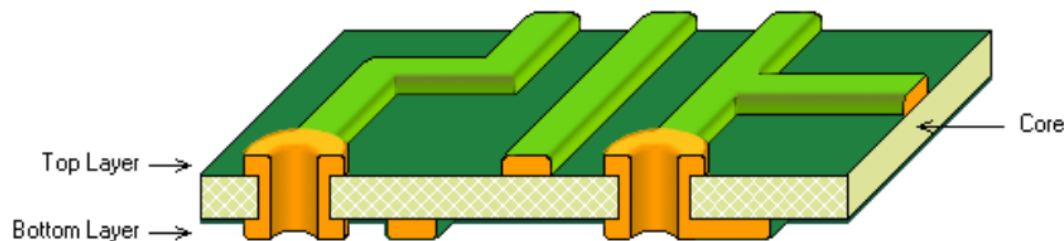
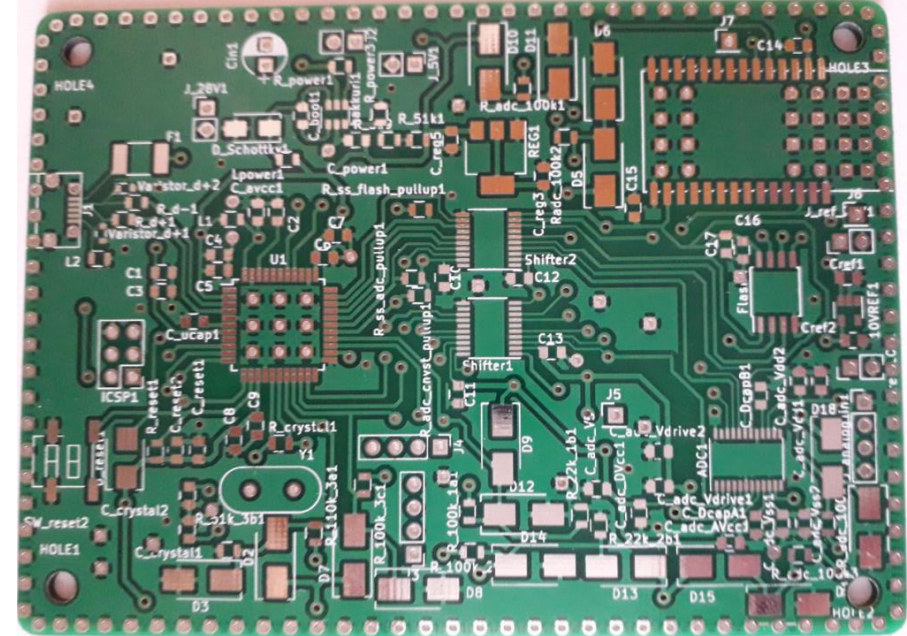
Introduction to PCB design in KiCAD

Where to find these slides and other necessary files

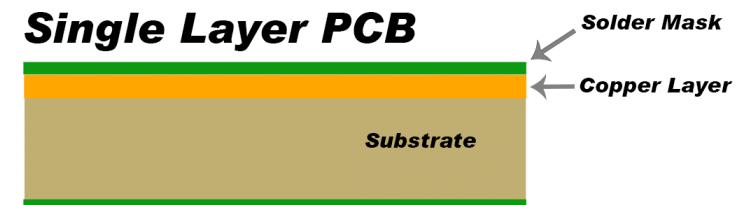
- https://github.com/pvikberg/PCB_lecture
- Download these files now, some of them will be necessary later

What is a PCB?

- PCB = Printed Circuit Board
 - PCB is wiring:
 - Connects components to one another
 - Can be ordered from a factory
- Or
- Can be corroded in the workshop
 - Cheap, fast, more work, variation in quality



Single Layer PCB

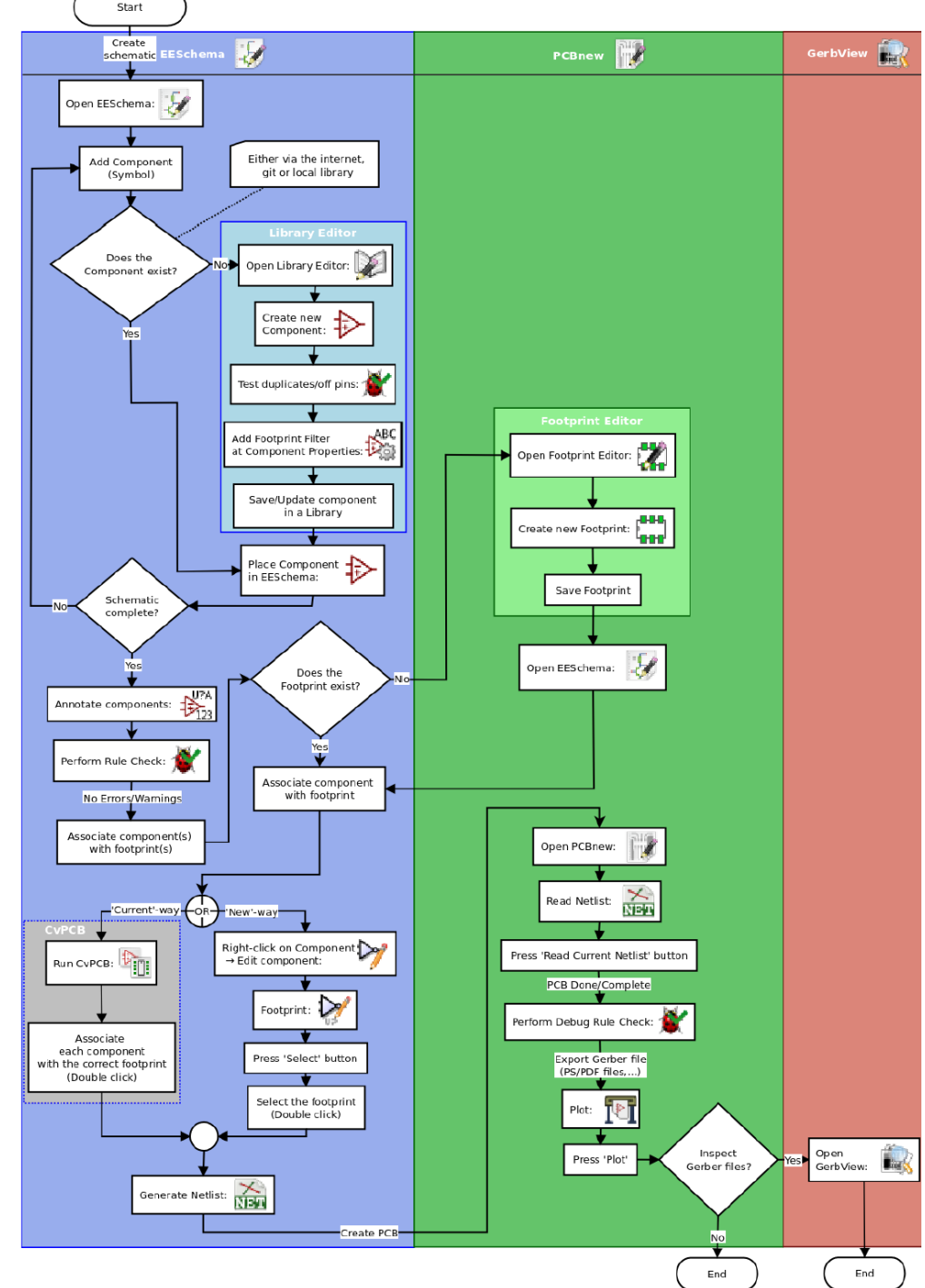


Double Layer PCB



Workflow

- 1: **Decide what is to be done**
- 2: **Choose components**
 - Use datasheets when doing this
 - Make sure components are compatible
- 3: Open KiCAD and start a new project
- 4: **Add the components and make the wirings in EESchema**
 - Use datasheets to find correct pins
- 5: Associate footprints to components and create a netlist
- 6: Open PCBnew, read the netlist and set design rules
- 7: **Place components , wirings, borders, filled areas etc.**
- 8: Create files needed to manufacture PCB
 - In workshop: plot the pdf-files (MIRROR THE FRONT COPPER IMAGE)
 - Ordered from a factory: plot gerbers and drill file



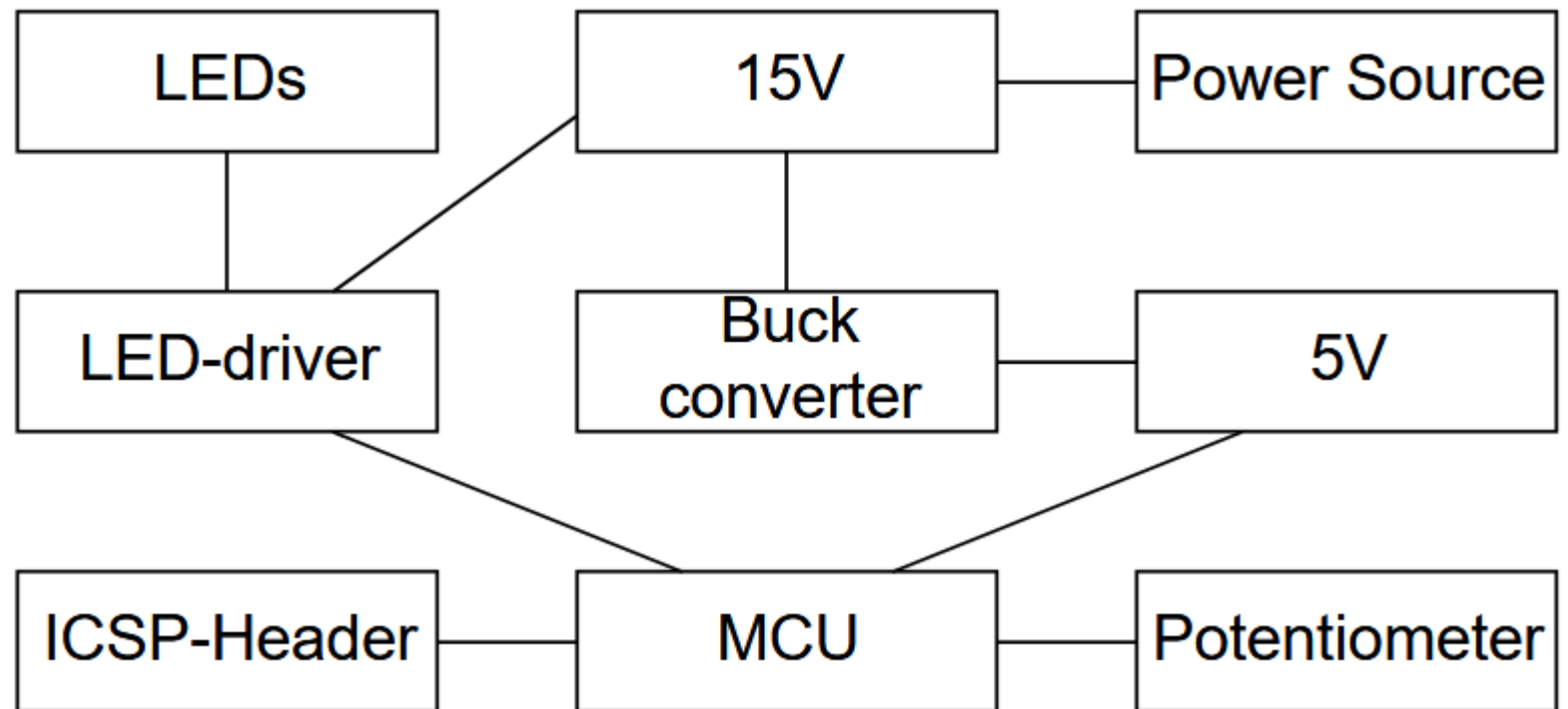
1. Decide what is to be done

- In this example, we wish to use a potentiometer to control the brightness of 4 blue LEDs with a LED-driver and a microcontroller (MCU)
- A very suboptimal (stupid) solution, but should do as an exercise

- Components

needed:

- Potentiometer
- 4 LEDs
- LED-driver
- MCU
- Buck converter

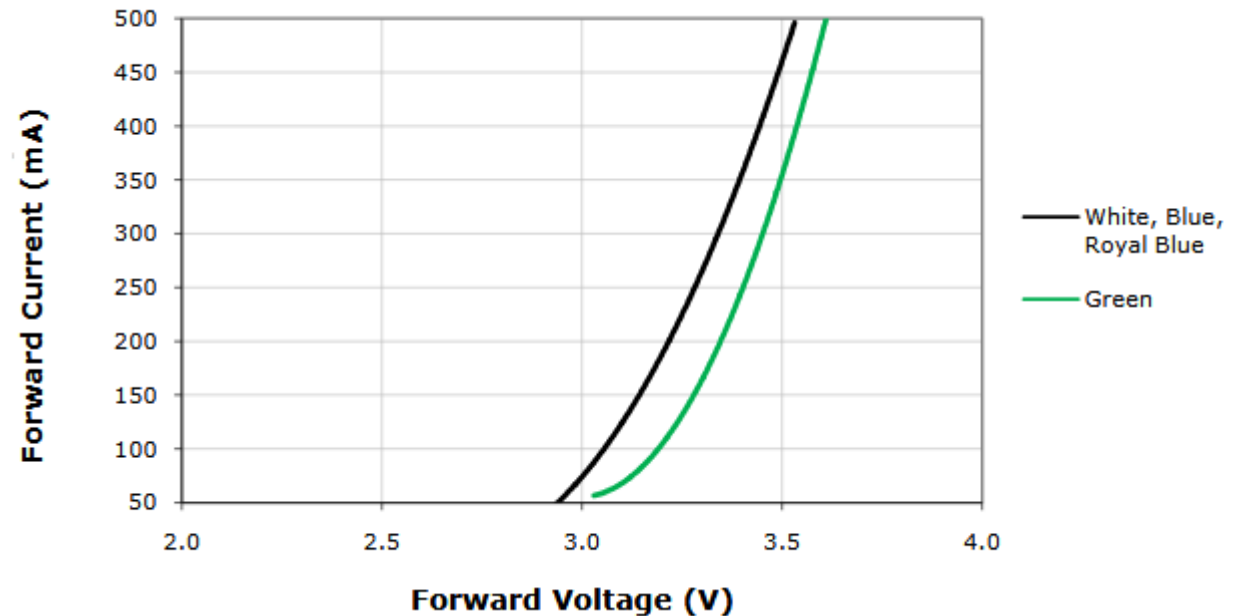
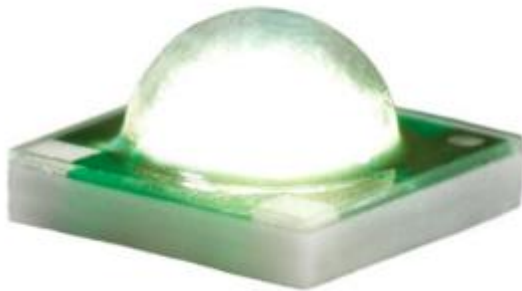


2. Choosing the components

- Time consuming
- See what the workshop has stocked
- Order online:
 - Mouser, Digikey, Farnell etc...
 - Hint: try googling e.g. "led driver farnell" to find the correct category
 - With experience, one could search manufacturer sites (e.g. Texas instruments)
- Almost every component has a datasheet that provides a detailed description of the component
- When ordering online, **pay very close attention to the datasheet** and make decision to purchase based on the datasheet
 - Some important criteria: footprint, casing and soldering difficulty
- For the purposes of this demo, components have already been chosen

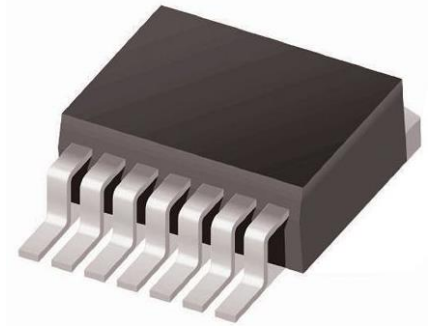
2. Choosing the components: LEDs

- From the datasheet:
- Forward voltage: 3.35V
- Current: ~350mA

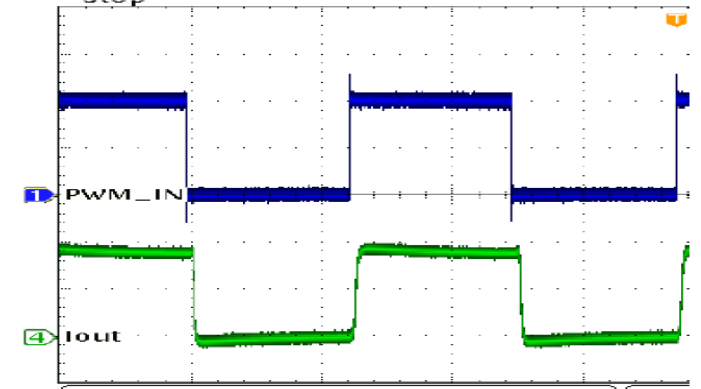


- XPCBLU-L1-0000-00W02 - LED, HB, BLUE, 23.5L
- <http://fi.farnell.com/cree/xpcblu-l1-0000-00w02/led-hb-blue-23-5lm/dp/2419727>

2. Choosing the components LED-driver



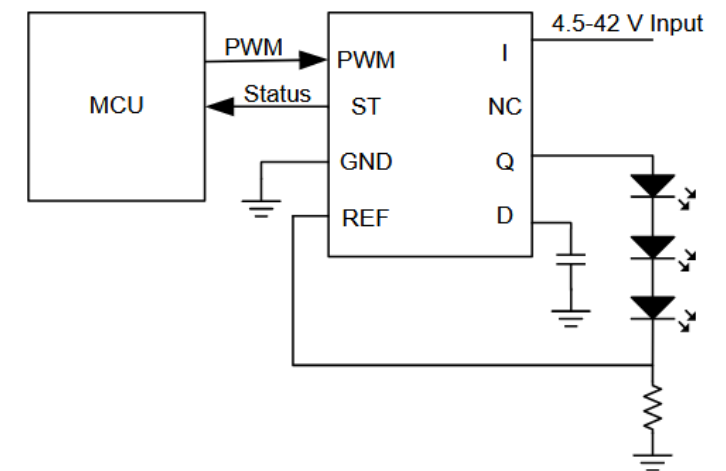
- Led brightness controlled by constant-current PWM-signal
- Pulse Width Modulated (PWM) voltage input controls a constant-current PWM output->
- MCU PWM used to control this LED-driver



- Constant current strength chosen with a reference resistor->

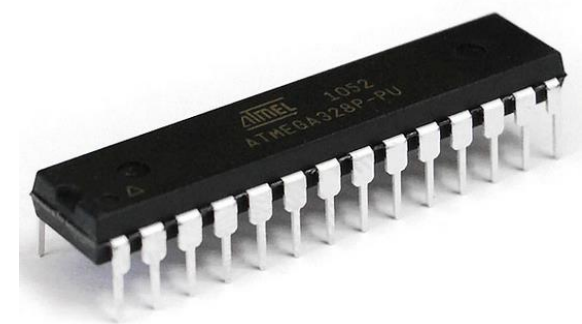
- Texas Instruments, TL4242-Q1 Adjustable LED Driver
- <http://www.ti.com/lit/ds/symlink/tl4242-q1.pdf>

Typical Application Schematic



2. Choosing the components: MCU

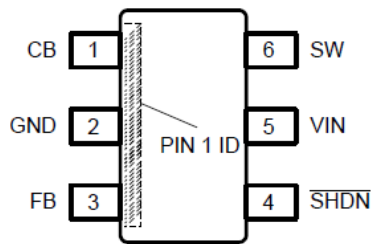
- From the workshop: Atmel-328P: the same as in Arduino UNO
- Study the Arduino UNO schematic to find out what additional components are needed
- ICSP-header needed to program this with an Arduino UNO



2. Choosing the components: Buck Converter

- The input voltage (15V) is too high for the MCU, but it is what the LED-driver needs to drive 4 LEDs (3.5V each)
- -> give the MCU 5V with a buck converter

- Datasheets show a typical application, just what we need ->



9.2 Typical Application

Figure 7 shows typical application where user can adjust output by R1 and R2.

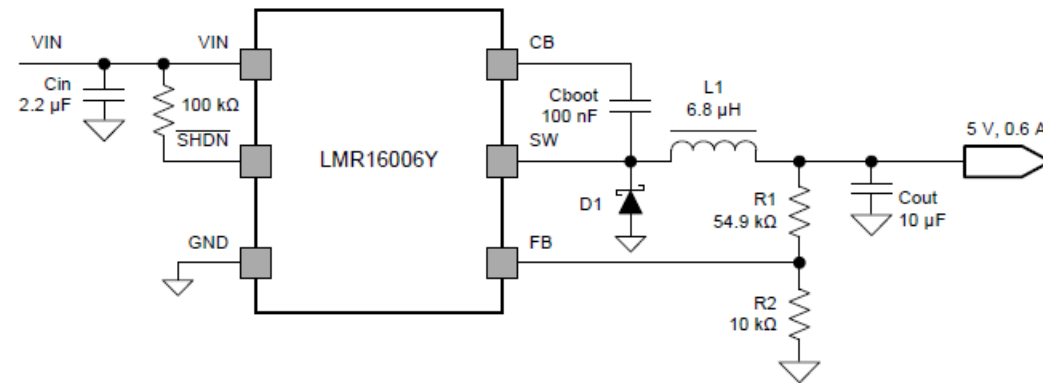
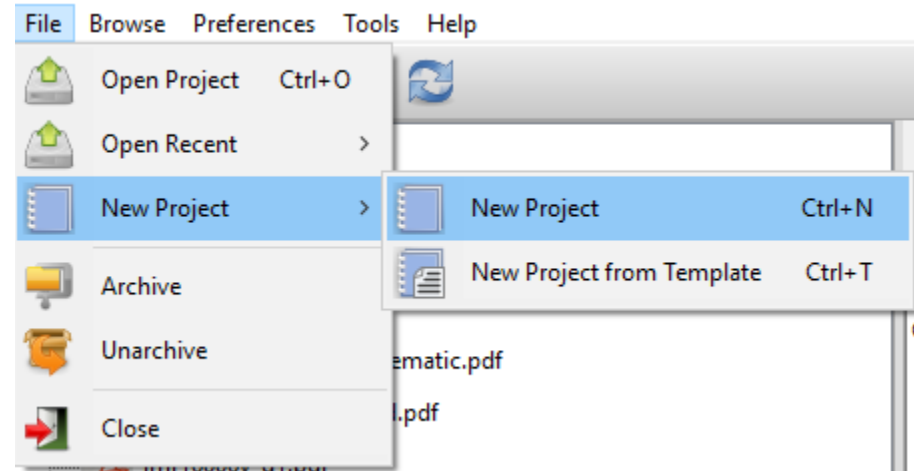
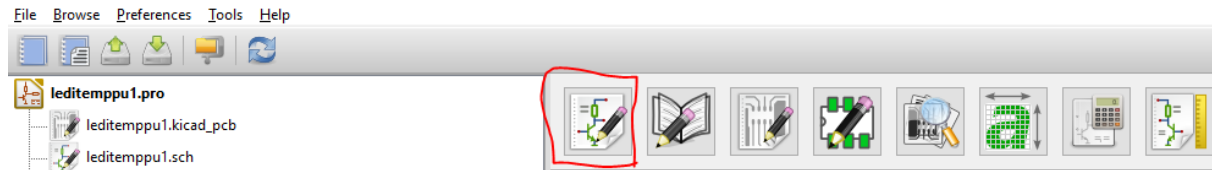


Figure 7. Application Circuit, 5 V Output

- LMR16006Y-Q1 SIMPLE SWITCHER
- <https://eu.mouser.com/ProductDetail/Texas-Instruments/LMR16006YQDDCRQ1?qs=sGAEpiMZZMsMlqGZiACxIYznHxQwyfA9giA2KvrC9rNu6yNWUn0Fsg%3d%3d>

3. Open KiCAD and start new project

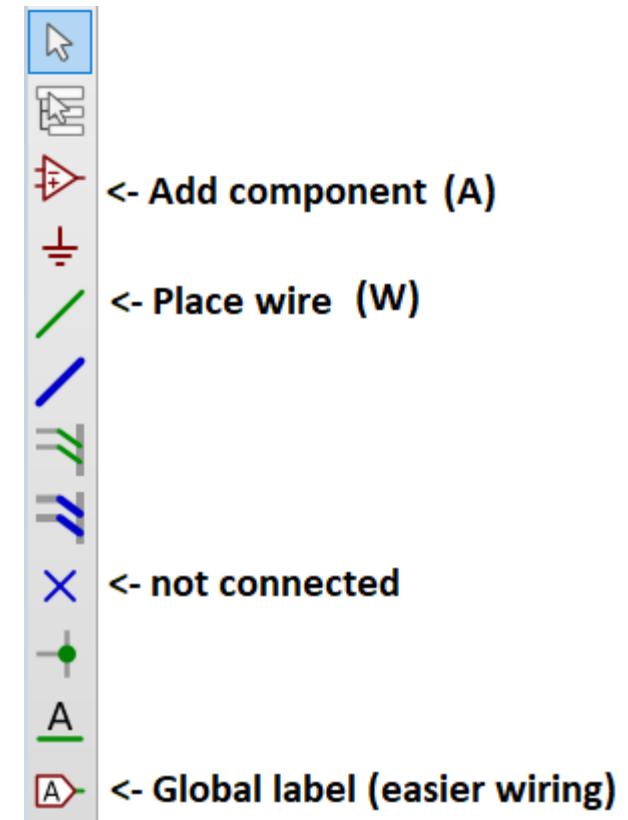
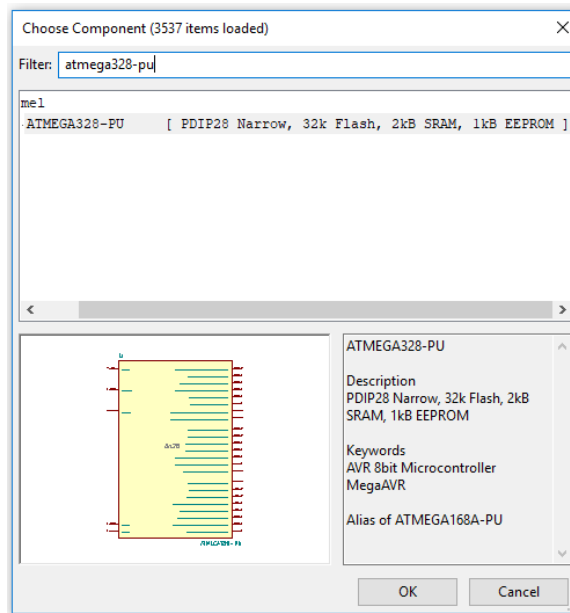
- The following parts are explained more generally in "Getting Started in KiCAD"
http://docs.kicadpcb.org/stable/en/getting_started_in_kicad.pdf
- The KiCAD installer can be found from: <http://kicad-pcb.org/download/>
- Open KiCAD and start new project, name it however you want
- Then, open EESchema



4. Add the components, Using EESchema

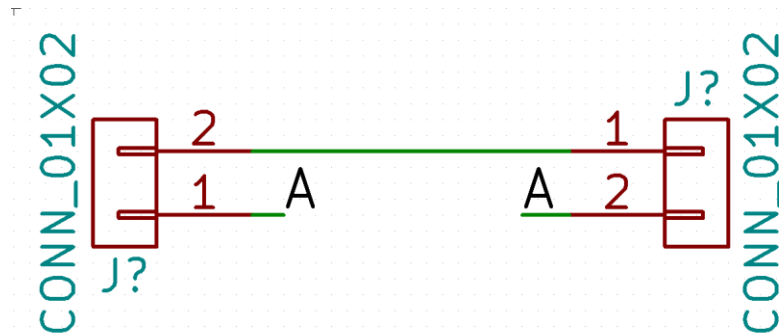
- Move mouse to the big area, and press A
- Search for 'atmega328-pu', press ok
- Then place the component in the lower right corner of the big area
- Hover mouse over a component and press M to move it, R to rotate, click to place

Some explanations

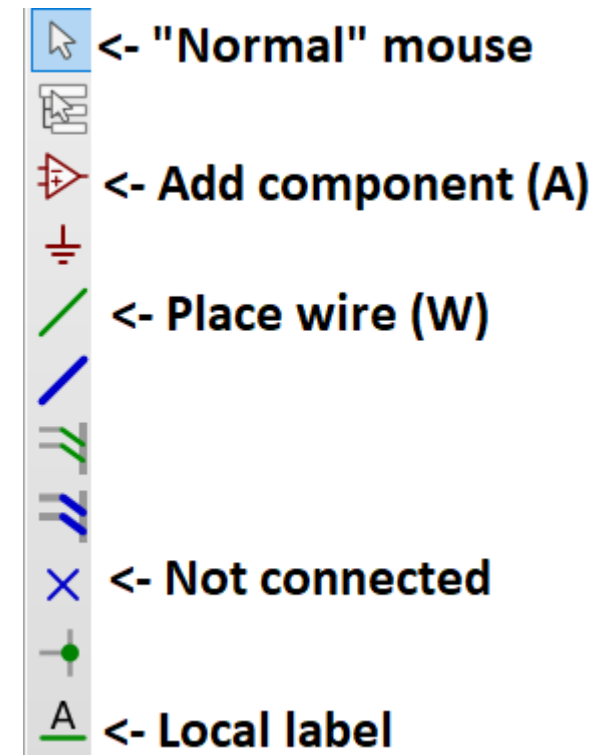


4. Add the components, Using EESchema

- Wires indicate which pins are connected
- Local labels can be used instead of wires, as below
 - Makes rewiring easier

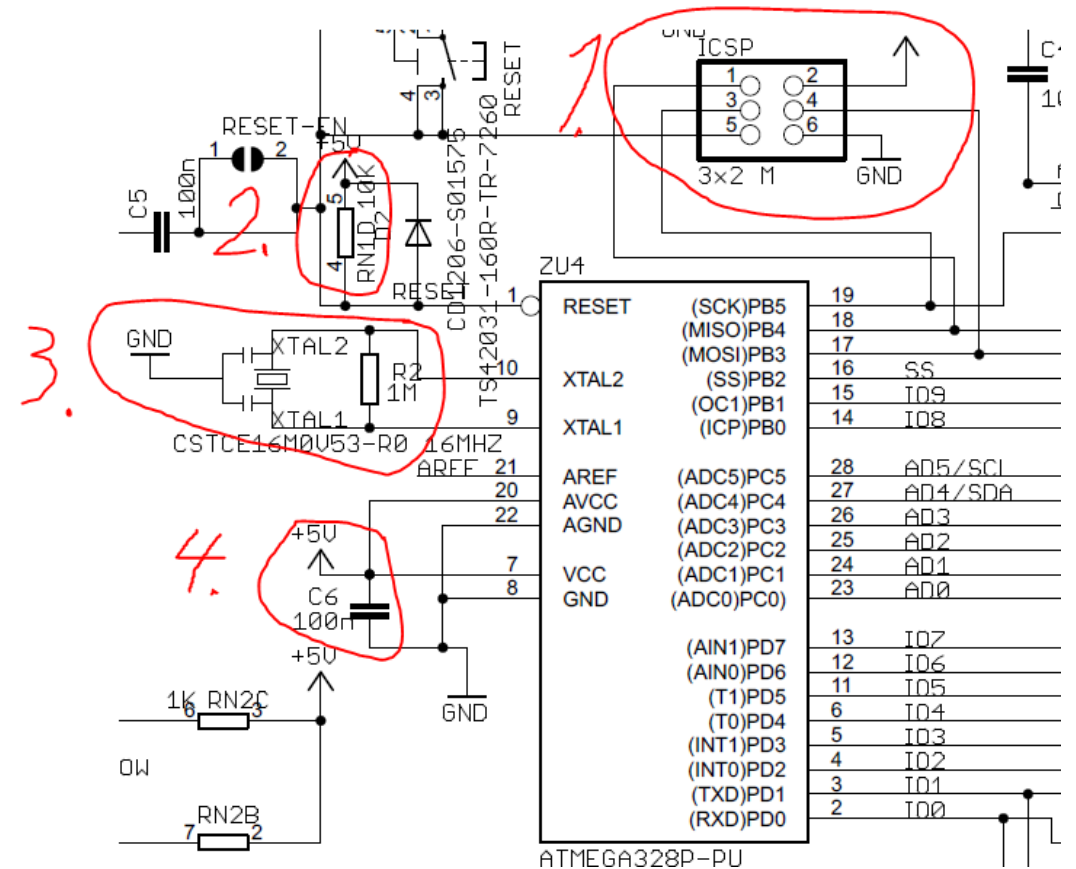


- If a pin is not connected to anything, place a 'not connected' marker on it



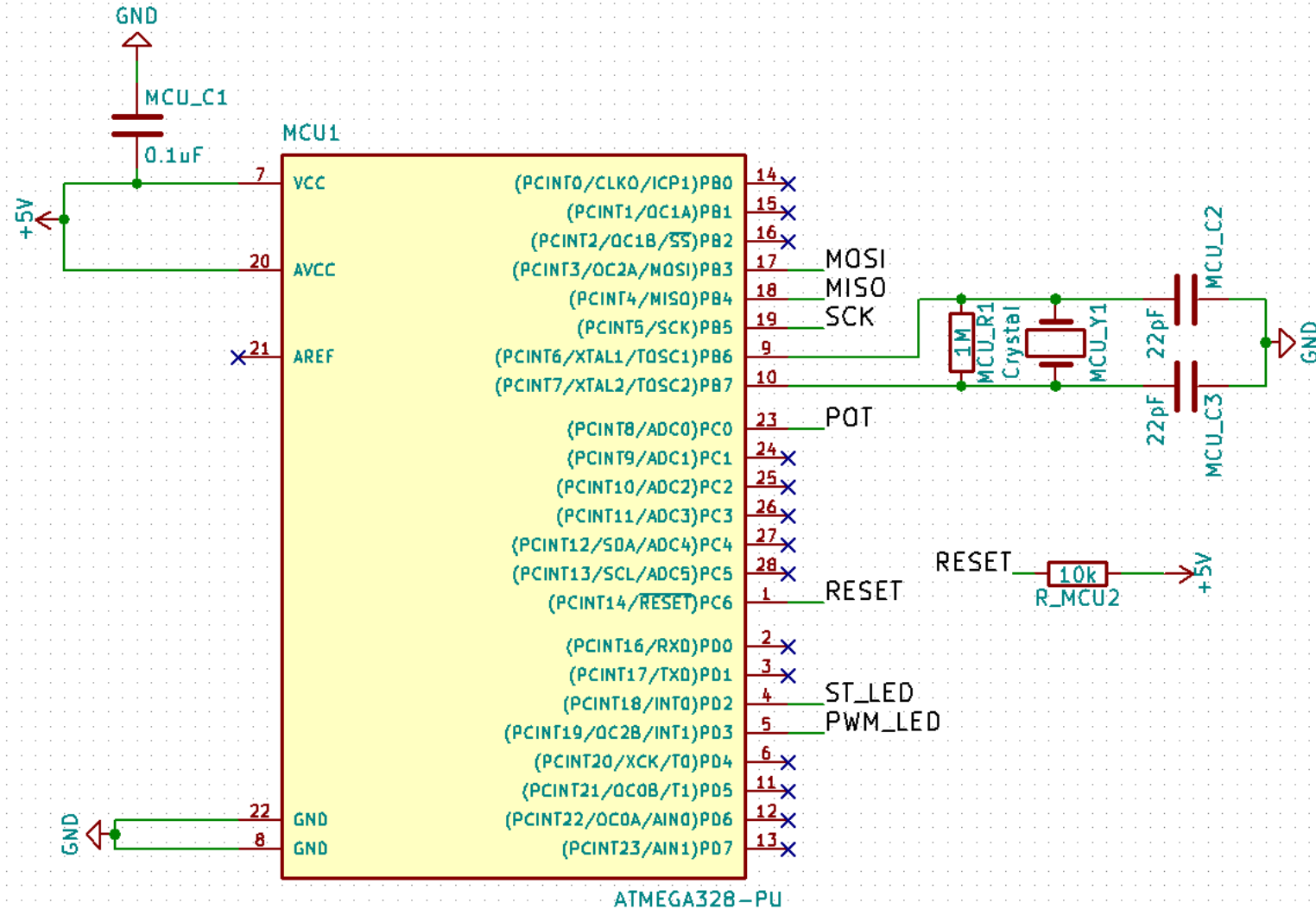
4. Add the components, reading datasheets

- Let's look at the Arduino UNO schematic to see what we need at least
- 1: ICSP-header
- 2: A resistor between RESET & 5V
- 3: A 16MHz crystal, 2 22pF capacitors and a 1M resistor
- 4: A 100nF capacitor



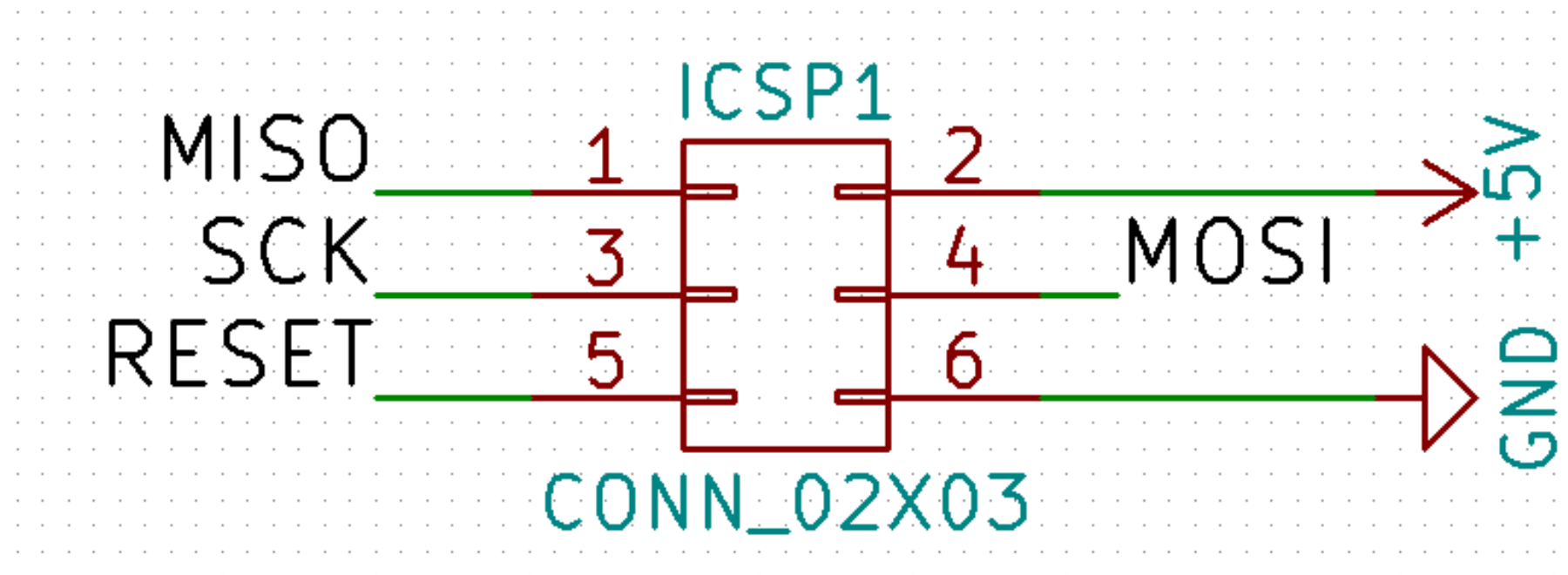
4. Add the components, reading datasheets

- Press A to add a component, use the search bar to find right ones
- R: resistor
- C: capacitor
- Crystal: crystal
- GND: ground
- 5V: +5V
- To move a component after placing, hover mouser over it and press M
- To edit values, hover and press E
- Add components, wires, labels...
like this->
- Remember to annotate (name) the components and add values



4. Add the components, reading datasheets

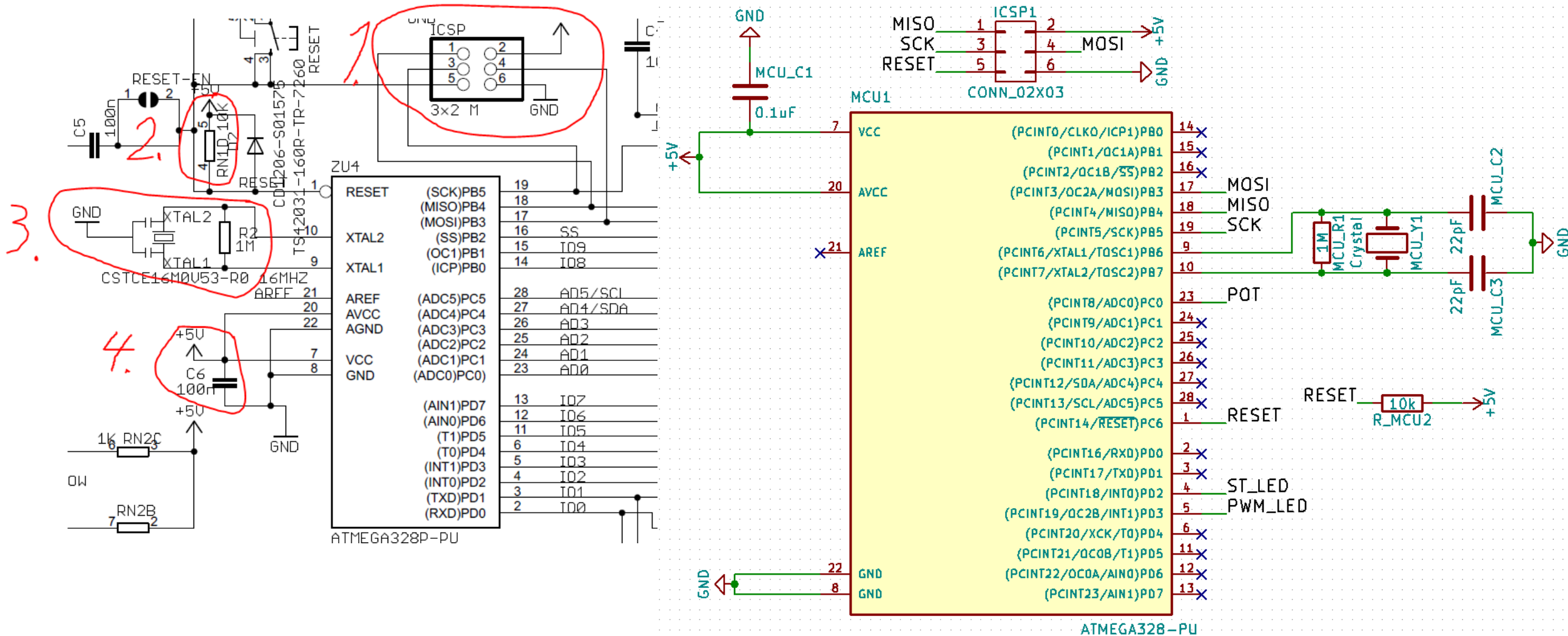
- Now, let's add the ICSP-header
- Press A and search for 'conn_02x03'
- Add the connector, and then place wires and labels like this:



- Remember to annotate the connector

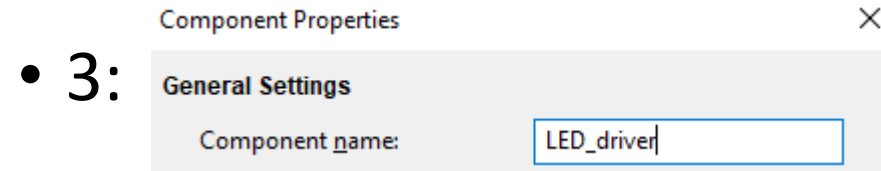
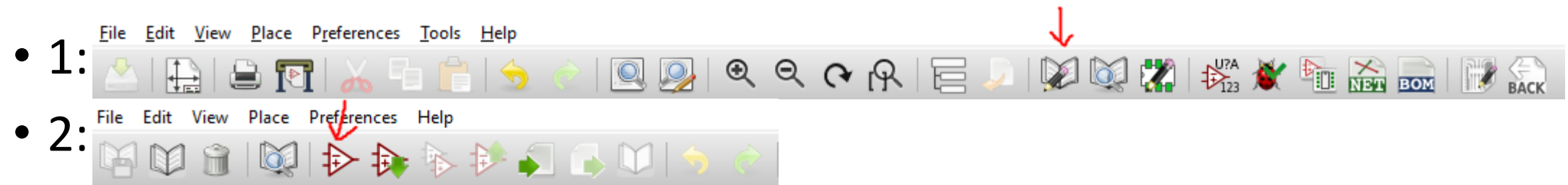
4. Add the components, reading datasheets

- We should now have something that resembles the UNO schematic



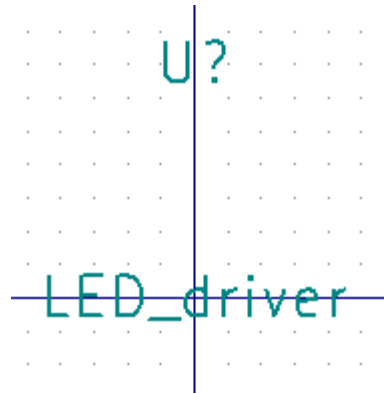
4. Add the components, creating components

- Often, KiCAD does not have the component we intend to use, so we need to create new ones
- Lets create a LED-driver



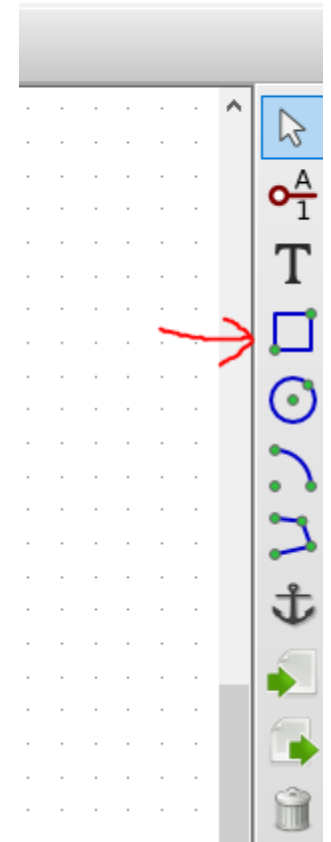
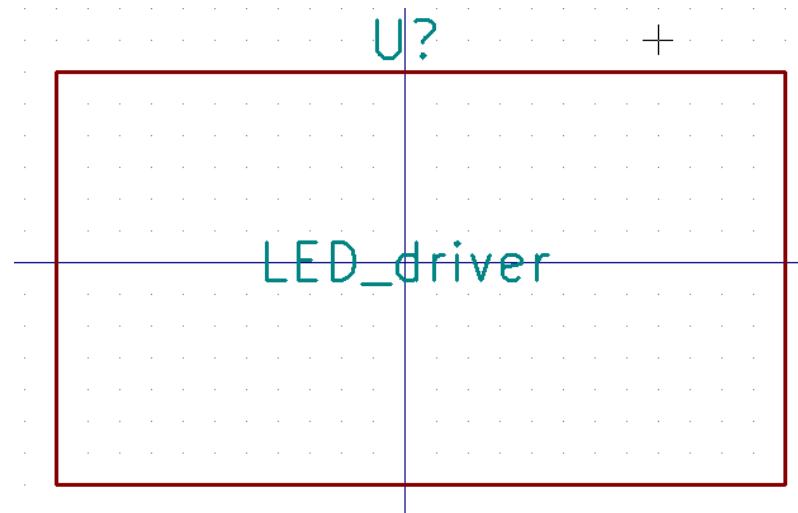
4. Add the components, creating components

- Move the texts (M):



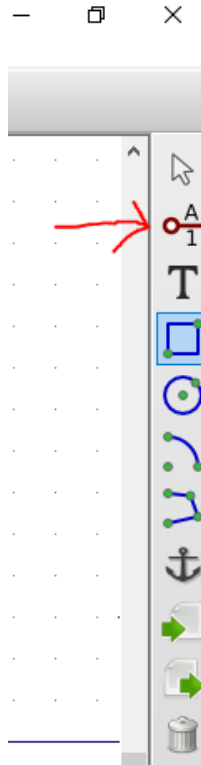
Then create a rectangle:

Draw the rectangle like this:

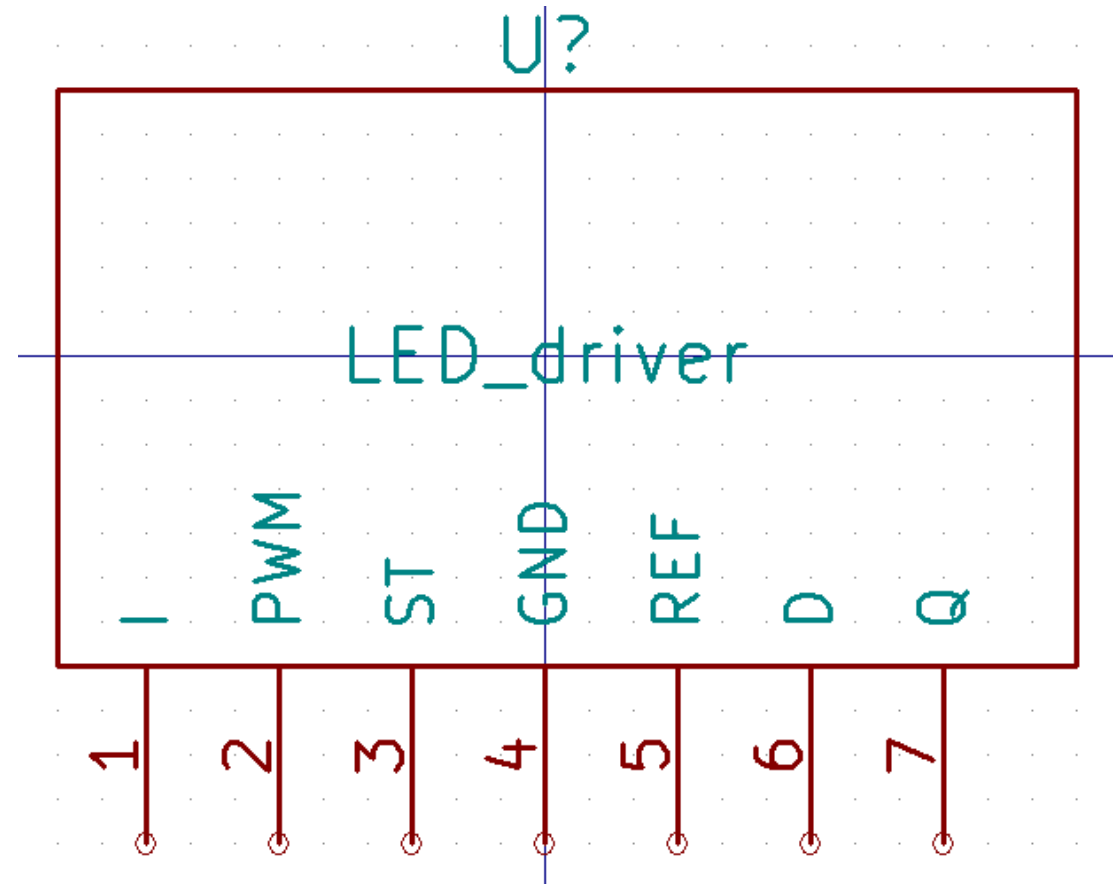


4. Add the components, creating components

- Then Create the pins:

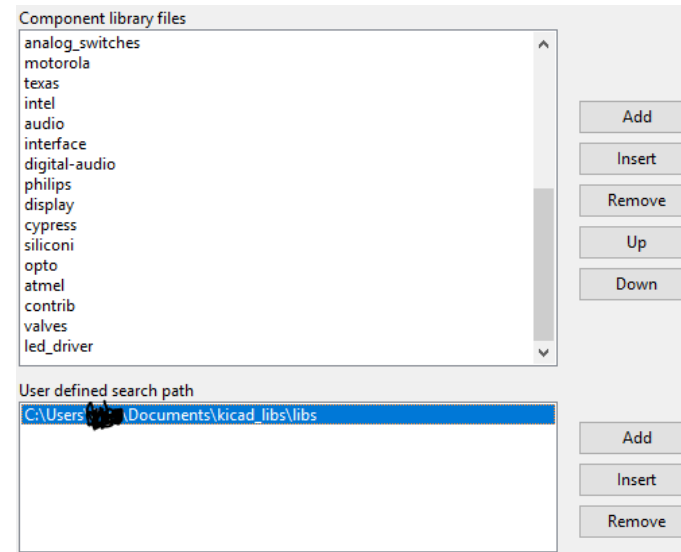
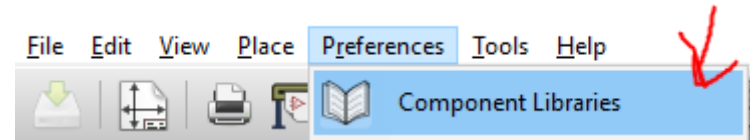


And place and name them like this:



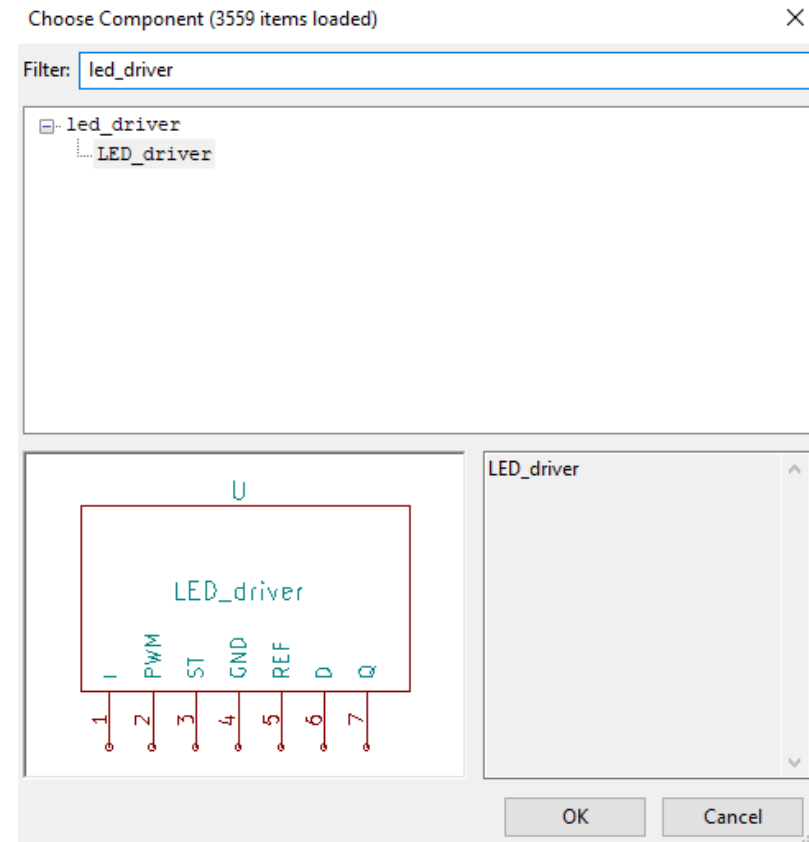
4. Add the components, creating components

- Save the created component:
- Go back to EESchema
- Preferences-> Component libraries
- Add folder of created component to user defined search path
- Add created library file in upper part



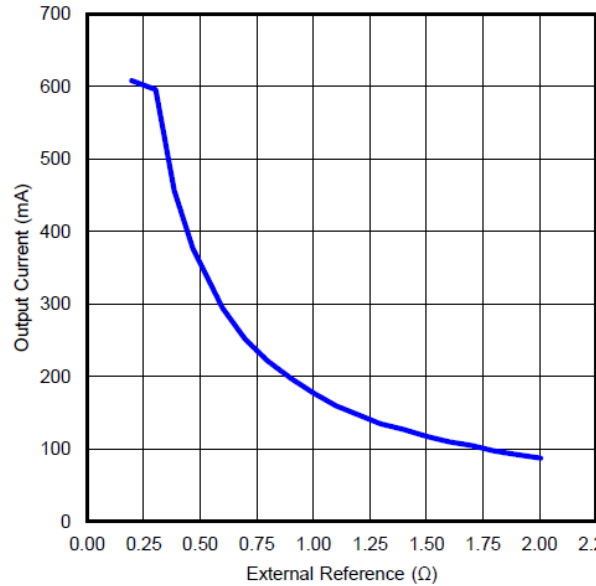
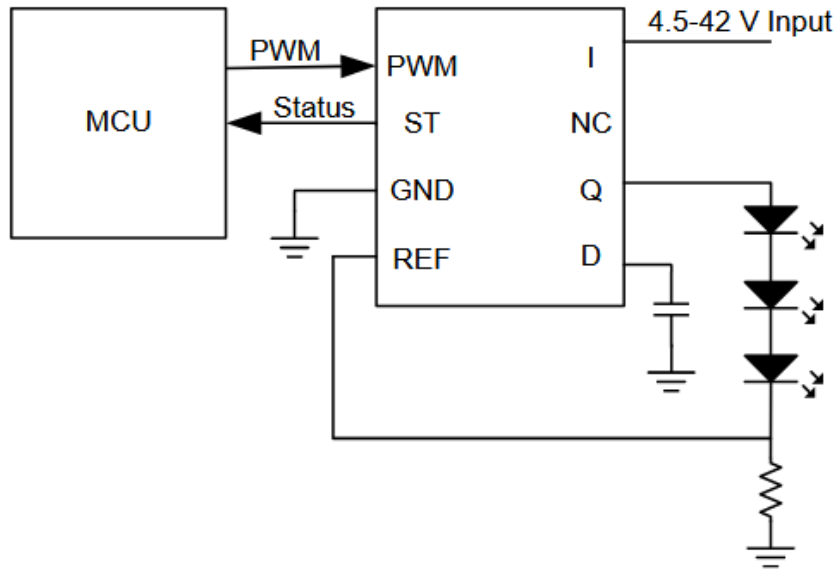
4. Add the components, creating components

- Add the new component in the big area
- Then, let's look at the datasheet

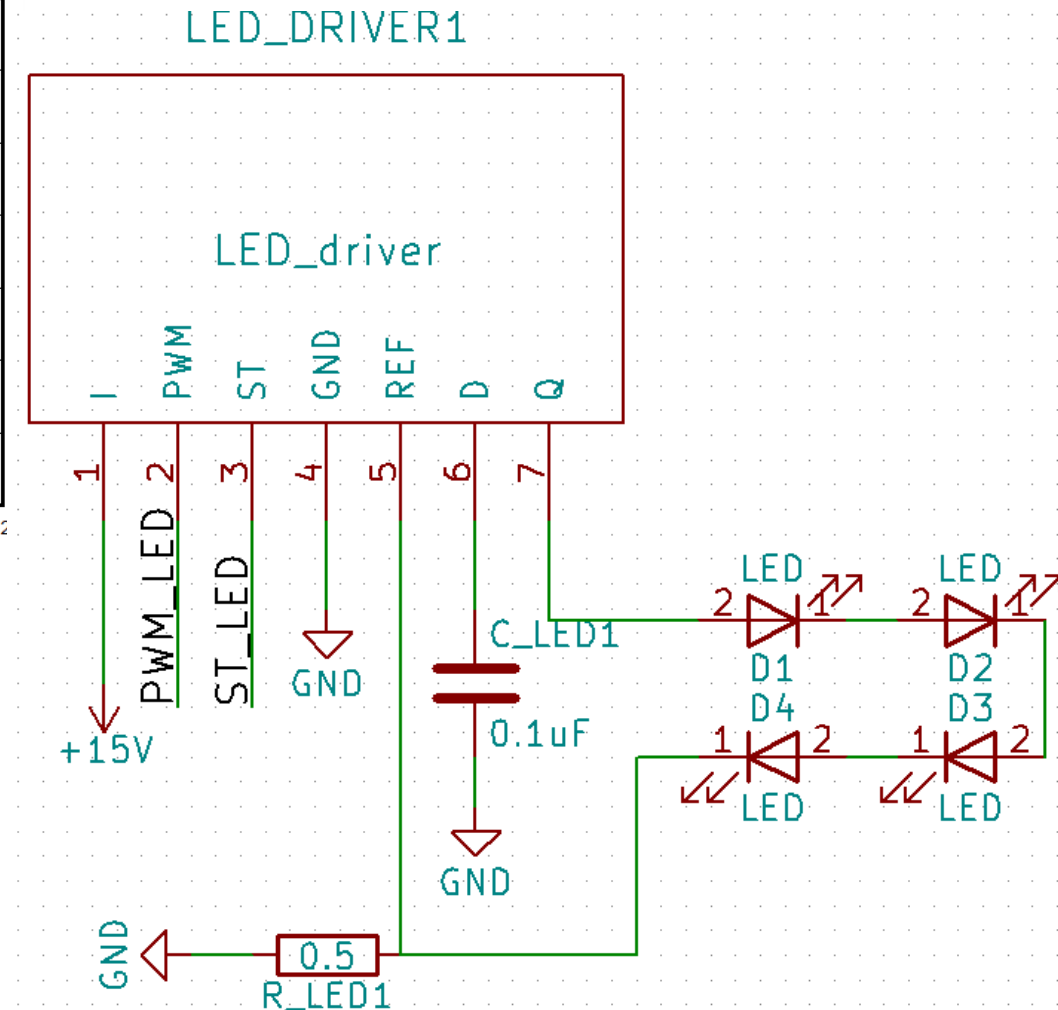


4. Add the components, adding the LED-driver and the leds

Typical Application Schematic

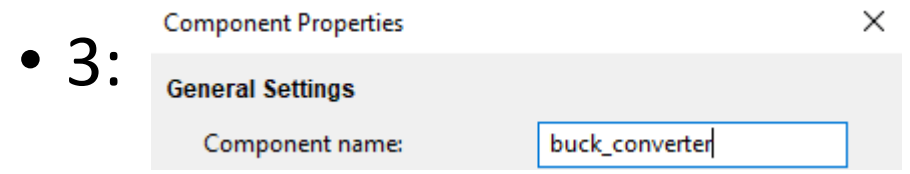
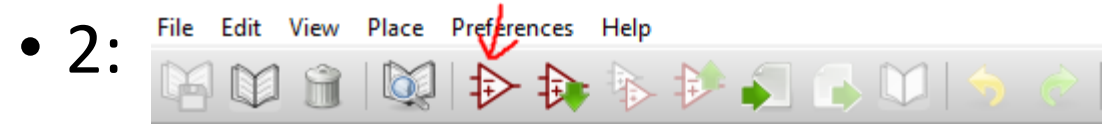


- Desired current: 350mA \rightarrow 0.5 Ohm



4. Add the components, create the buck converter

- Like the LED-driver, the buck converter needs to be created

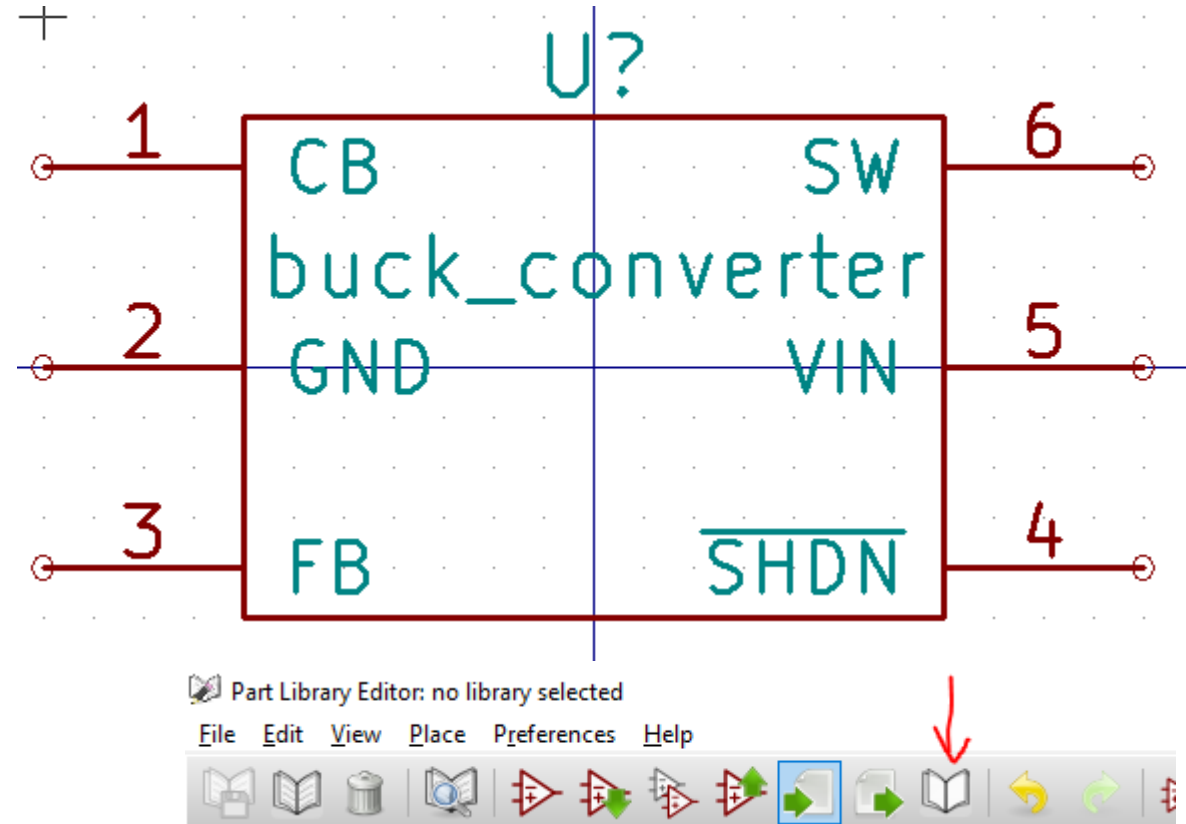


4. Add the components, create the buck converter

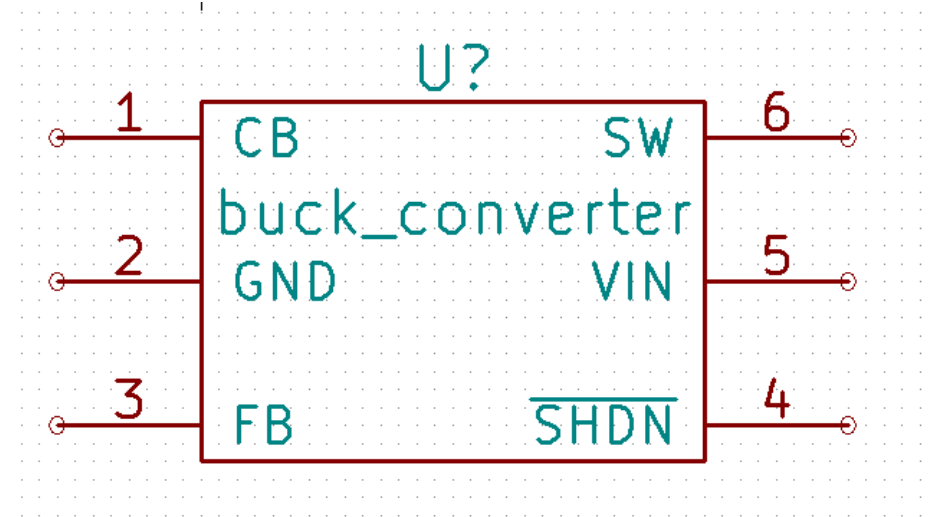
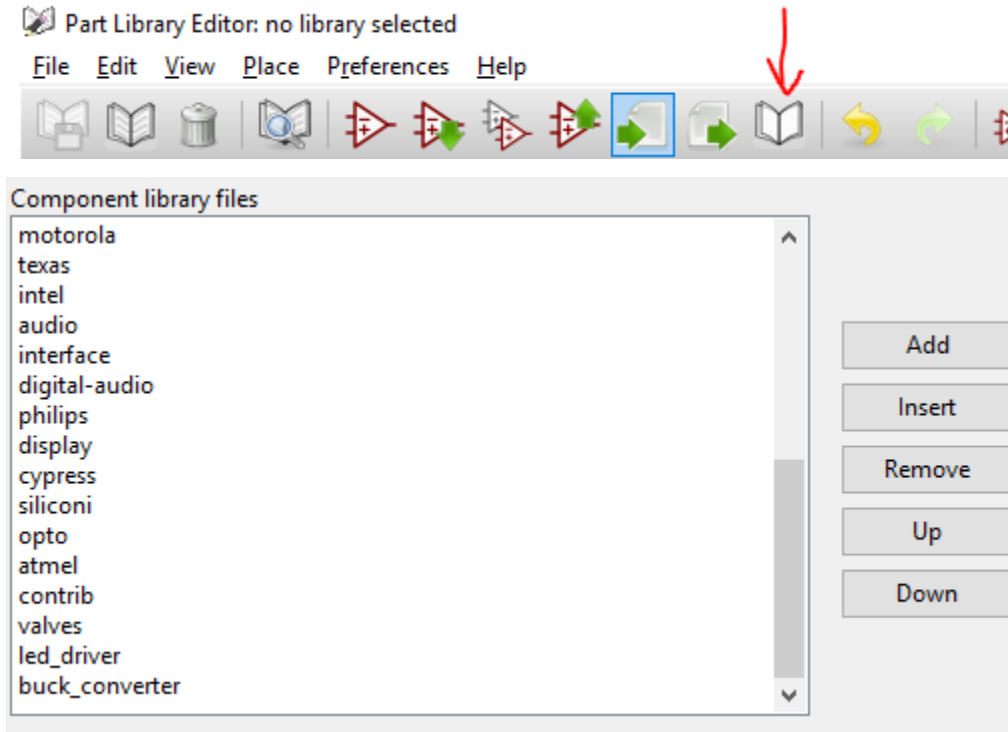
Pin name: SW Name text size: 1,270 millimeters
Pin number: 6 Number text size: 1,270 millimeters
Orientation: Left Length: 5,080 millimeters
Electrical type: Power output
Graphic Style: Line
Sharing
☐ Common to all units in component
☐ Common to all body styles (DeMorgan)
Schematic Properties
☒ Visible

Pin name: VIN Name text size: 1,270 millimeters
Pin number: 5 Number text size: 1,270 millimeters
Orientation: Left Length: 5,080 millimeters
Electrical type: Power input
Graphic Style: Line
Sharing
☐ Common to all units in component
☐ Common to all body styles (DeMorgan)
Schematic Properties
☒ Visible

Pin name: ~SHDN Name text size: 1,270 millimeters
Pin number: 4 Number text size: 1,270 millimeters
Orientation: Left Length: 5,080 millimeters
Electrical type: Input
Graphic Style: Line
Sharing
☐ Common to all units in component
☐ Common to all body styles (DeMorgan)



4. Add the components, create the buck converter



4. Add the components, buck converter datasheet

- Copy and paste the typical application
 - You can find the coil in the component browser using 'L'
 - The diode is a Schottky diode

9.2 Typical Application

Figure 7 shows typical application where user can adjust output by R1 and R2.

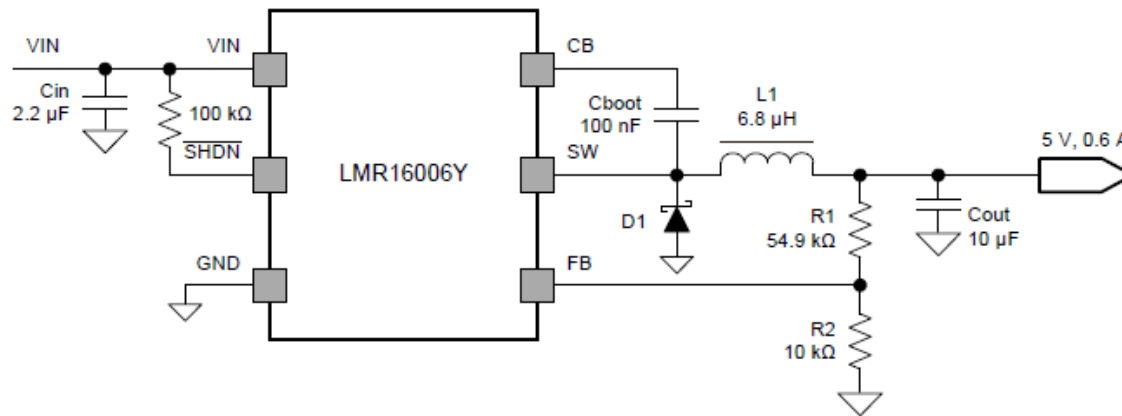


Figure 7. Application Circuit, 5 V Output

4. Add the components, buck converter datasheet

- The 54.9 Ohm resistor broken into 2 resistors because the workshop has 51k and 3.9k Ohm resistors

9.2 Typical Application

Figure 7 shows typical application where user can adjust output by R1 and R2.

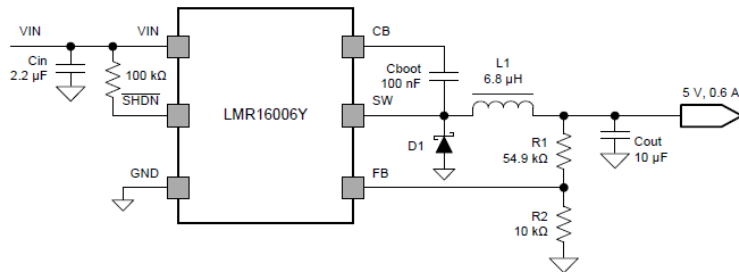
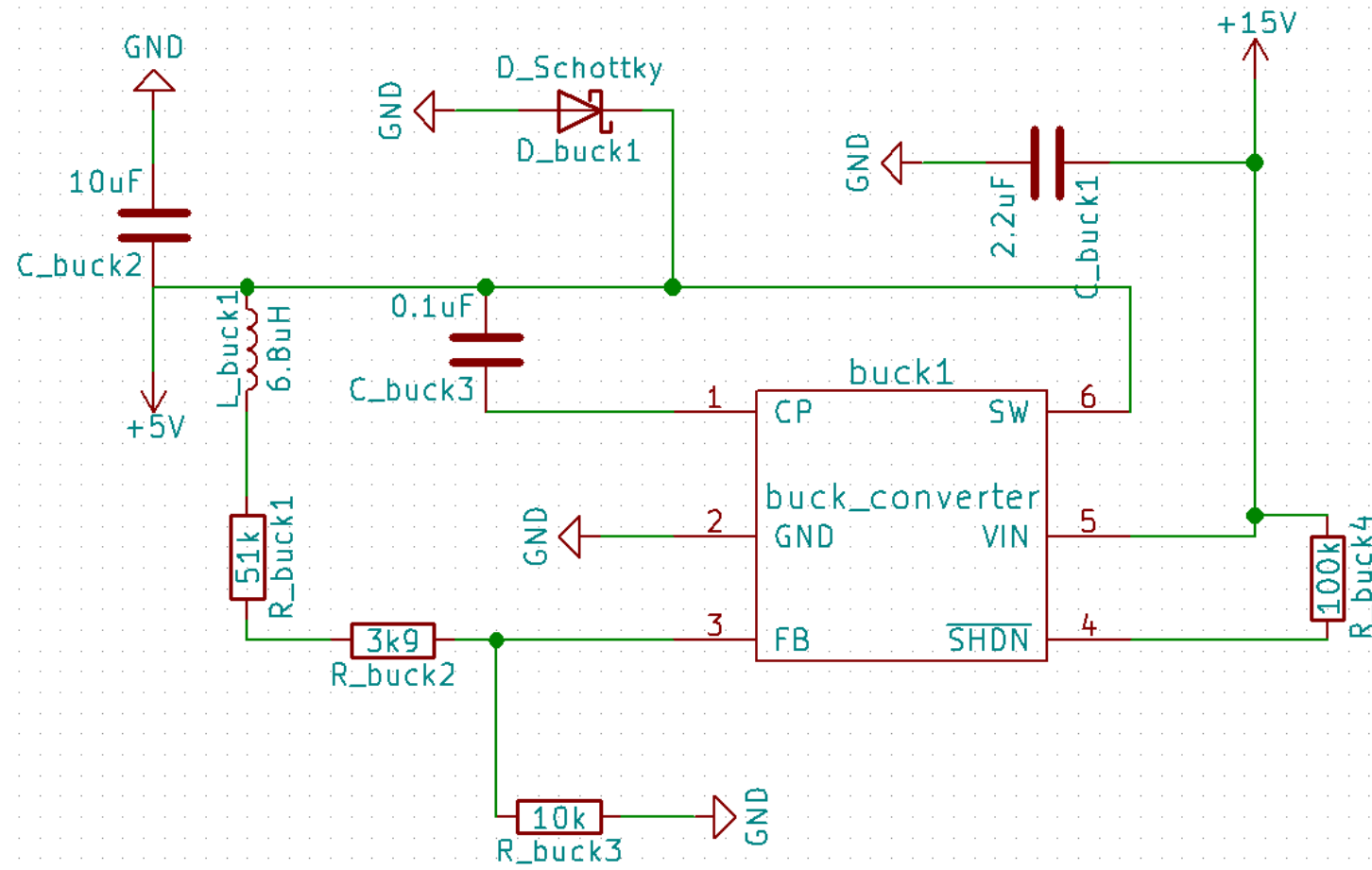
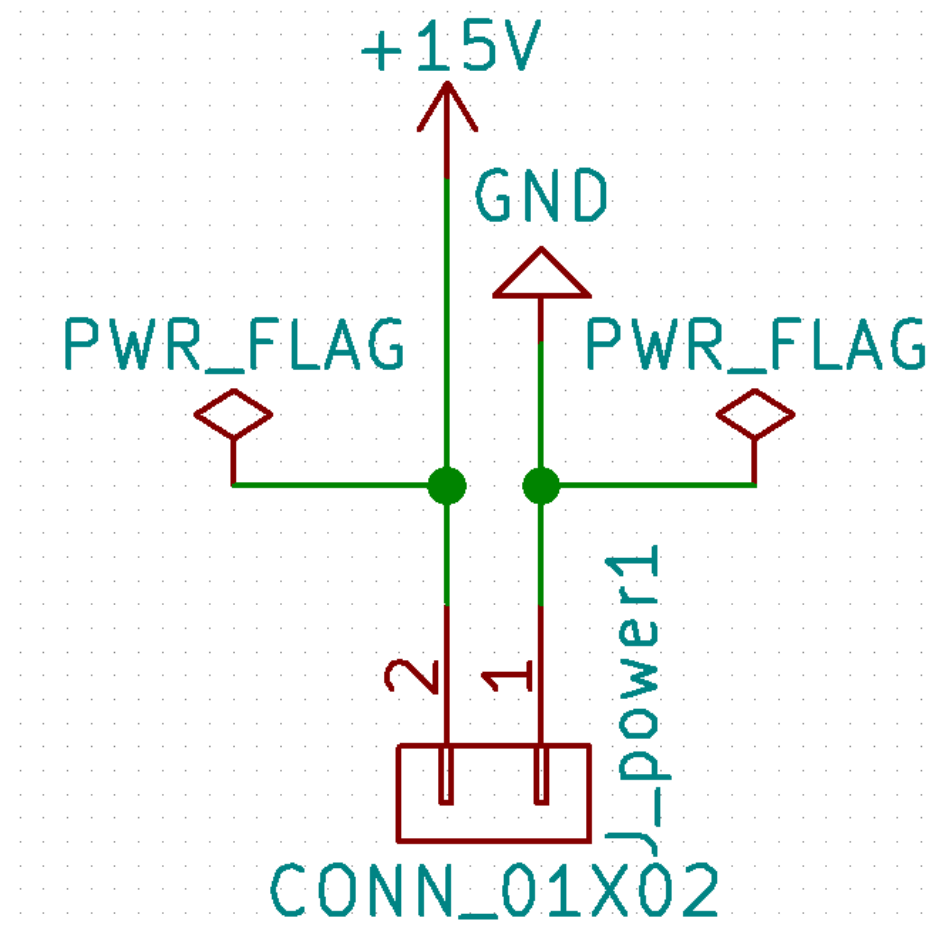
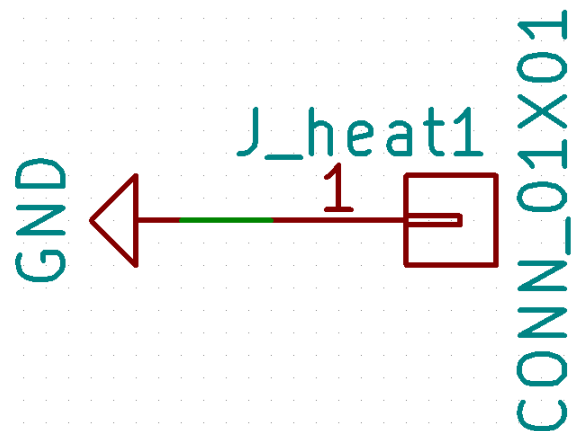


Figure 7. Application Circuit, 5 V Output



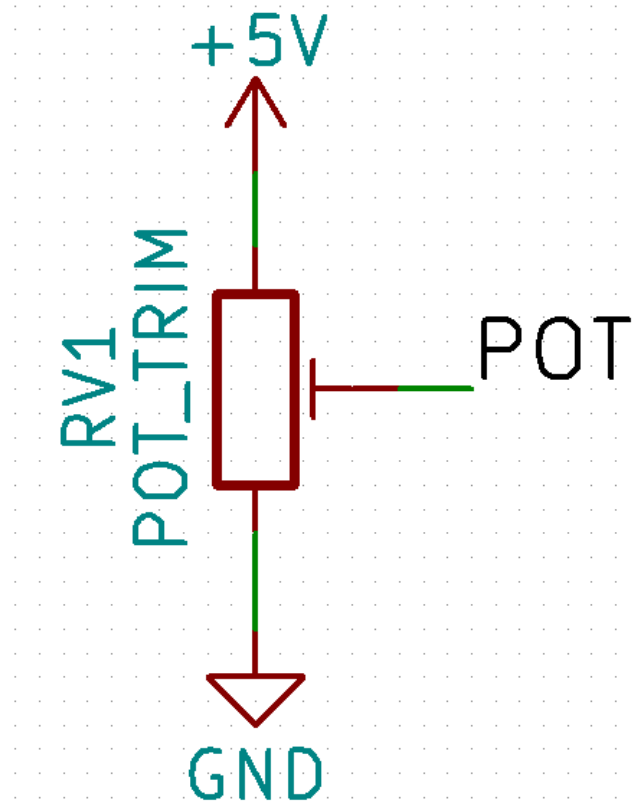
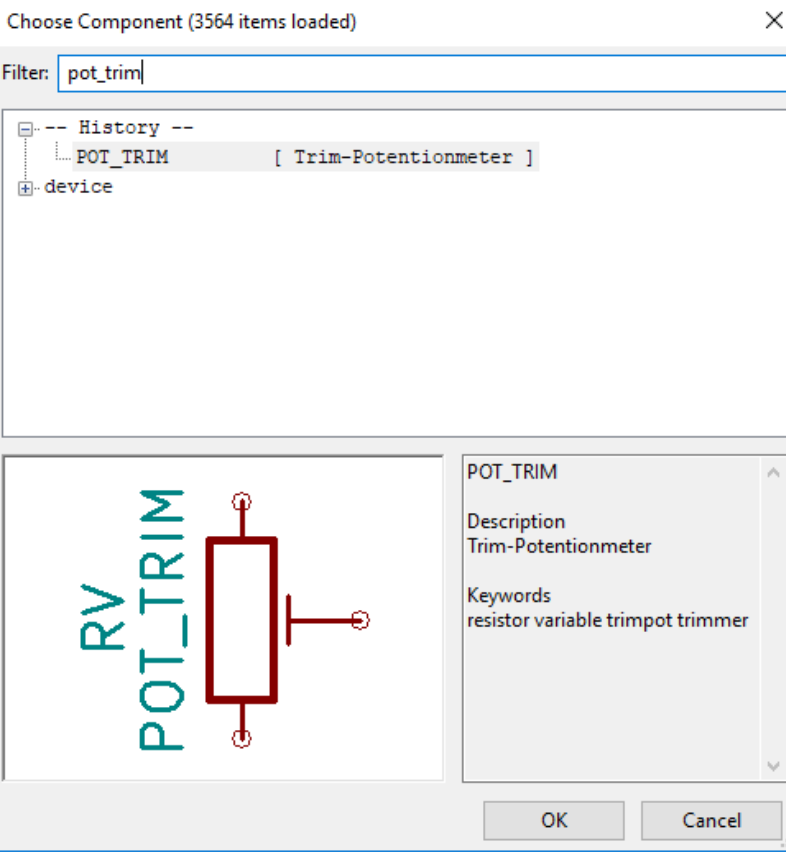
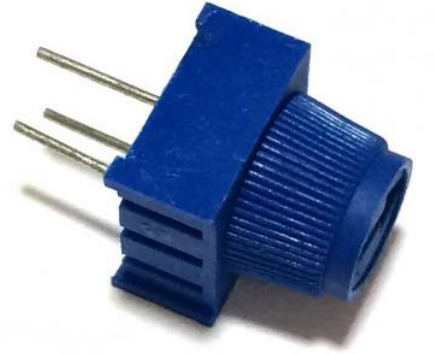
4. Add the components, power pins and heat via

- System has to get power somewhere, add power pins
- Also add power flags to both pins
 - This tell KiCAD where the power comes from
- Also add a heat via



4. Add the components, the potentiometer

- Found in the workshop
- Add component (A), POT_TRIM

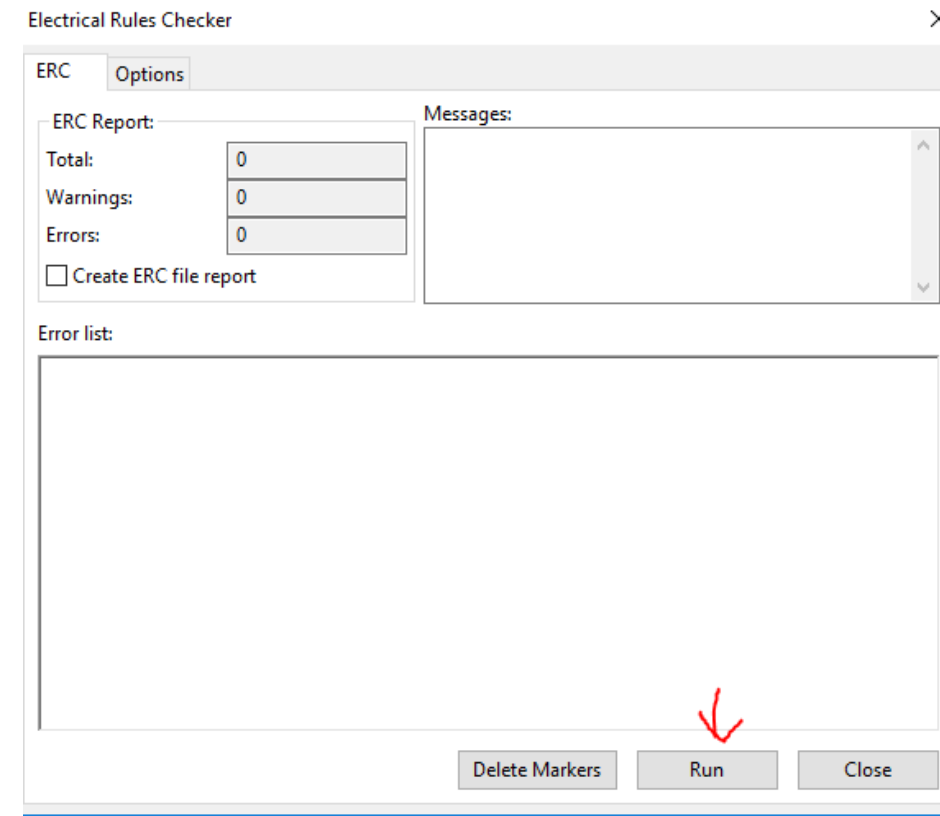


4. Add the components, check for errors

- After making the wirings, you should always check for errors



- If done correctly, you shouldn't have errors



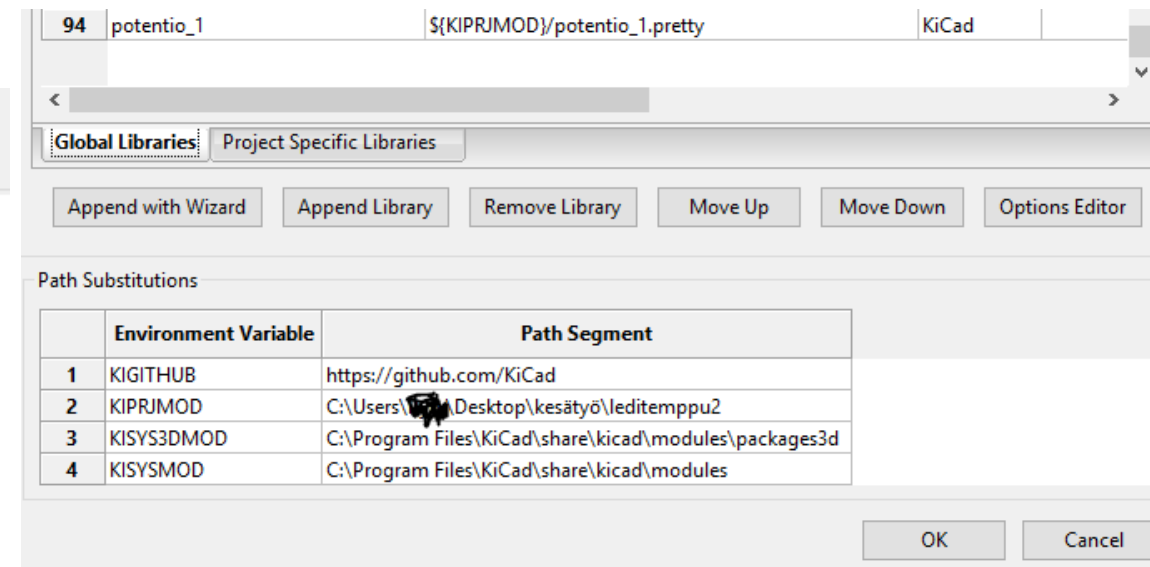
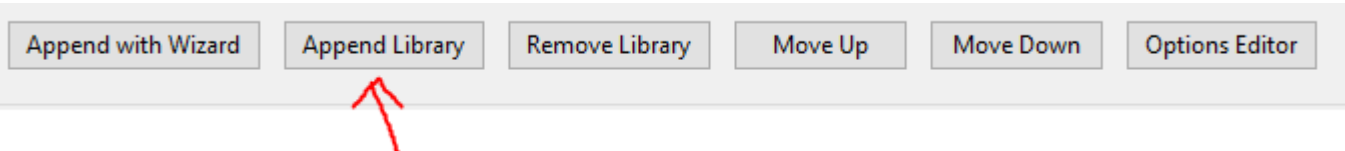
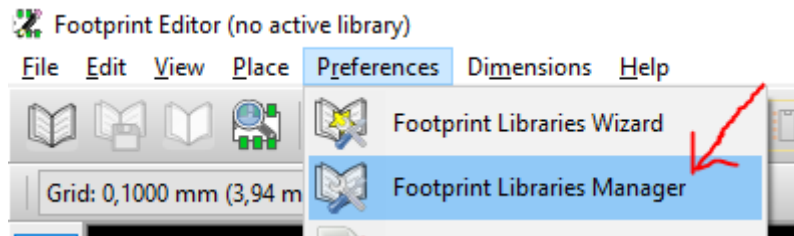
4. Add the components, create a new footprint

- Let's create a new footprint for the potentiometer



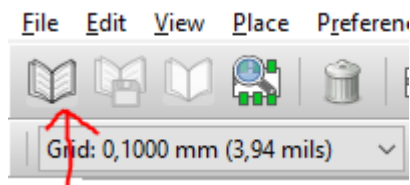
- Create a new folder inside project directory called e.g.

'potentio_1.pretty'



4. Add the components, create a new footprint

• 1:

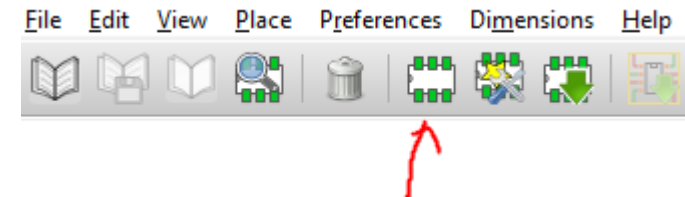


• 2:

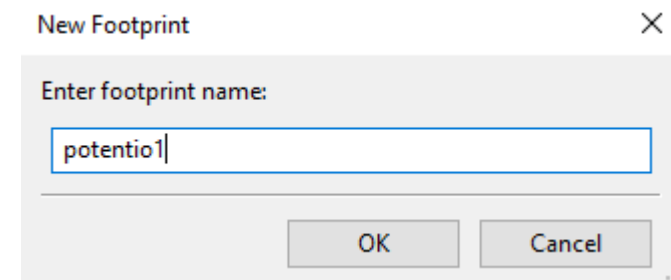
Select Library

Filter:	
poten	
Items:	
Nickname	Description
Potentiometers	Potentiometers / variable resistors
potentio_1	

3:

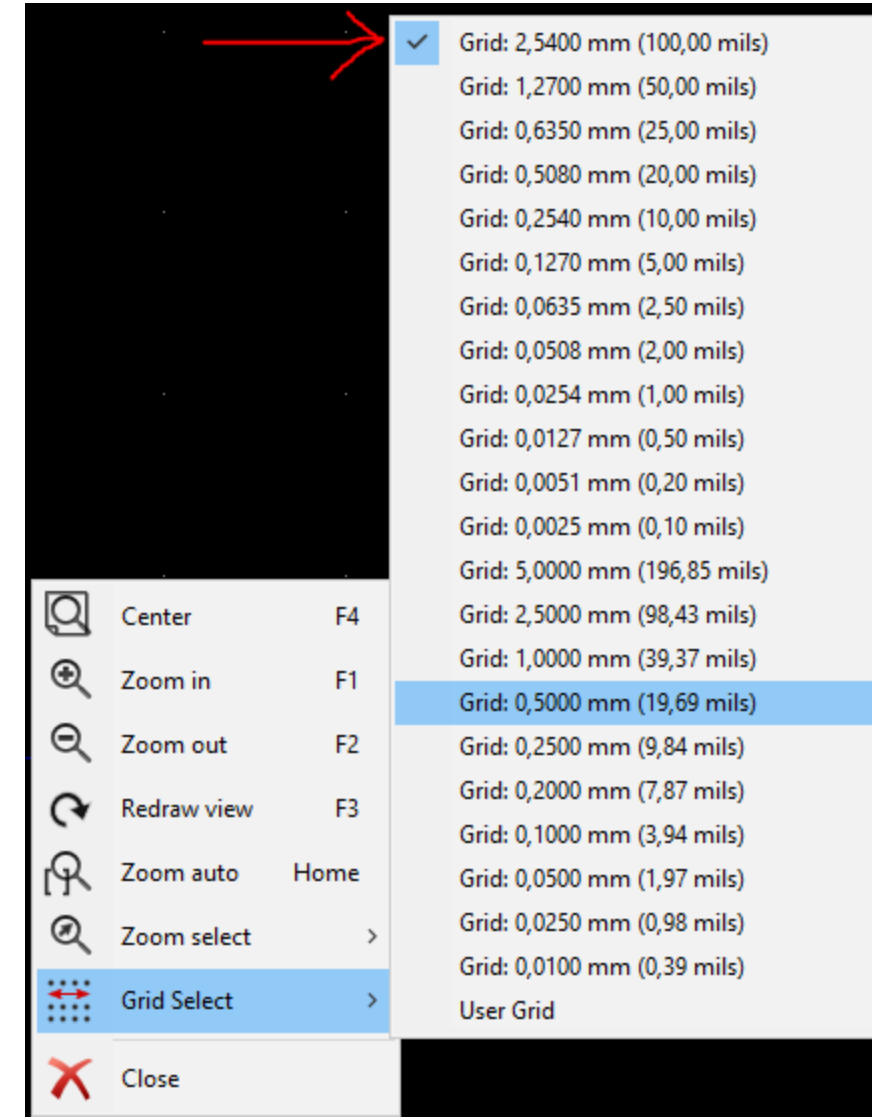
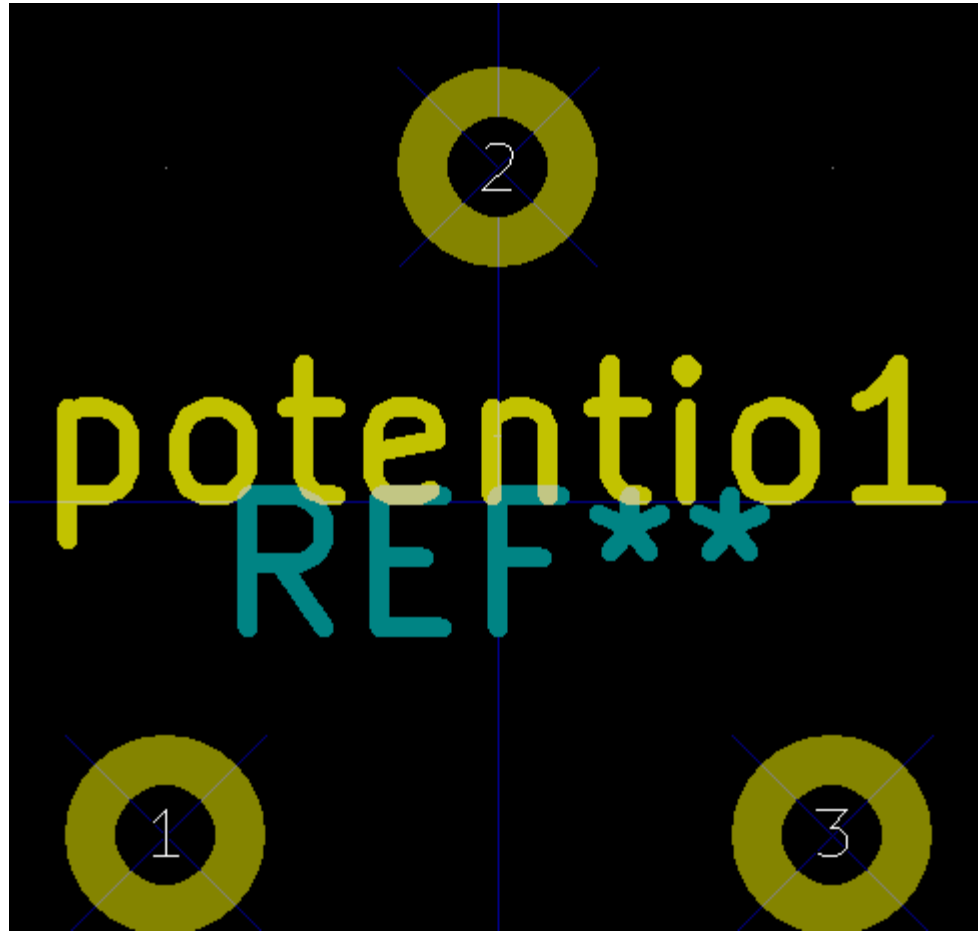
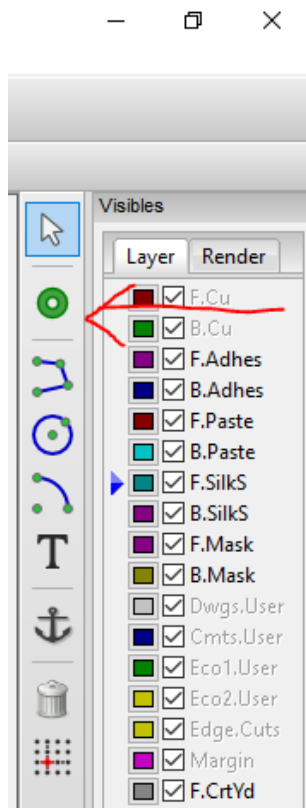


4:



4. Add the components, create a new footprint

- Right click, select grid 2.54mm ->
- Add 3 pads



4. Add the components, create a new footprint

- Save the footprint



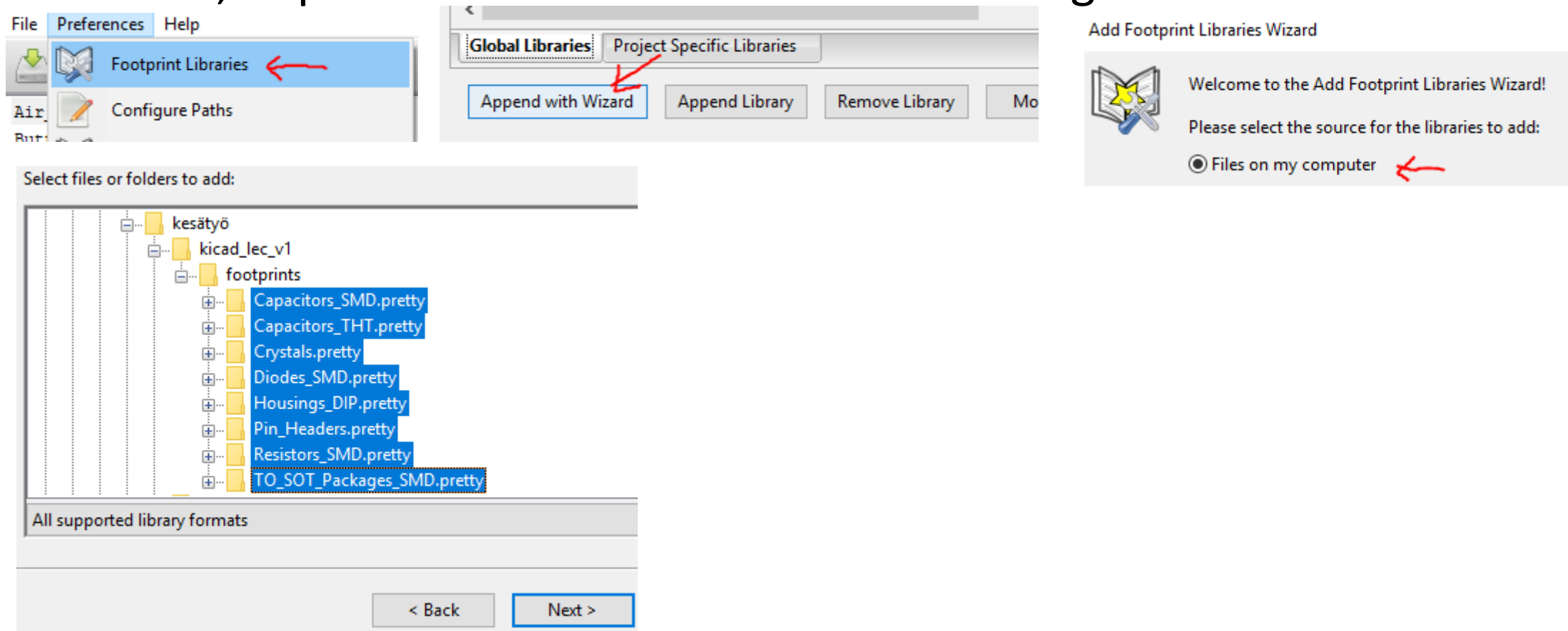
- Close the window and return to EESchema
- Open CvPcb to associate footprints to components



- Press Annotate and Ok
- Wait...

5. Associate footprints to components, importing libraries


- It is possible that the school computers do not have the libraries needed, import the libraries downloaded from github



5. Associate footprints to components

- Now, tell KiCAD what is each component's footprint
- Datasheets and shop product pages of internet suppliers (Mouser etc...) may prove useful

File Preferences Help



	Components to be annotated	Footprints
Air_Coils_SML_N...	1 C1 - 10uF : Capacitors_SMD:C_0805_HandSoldering	2815 Crystals:Resona
Buttons_Switches_S...	2 C2 - 0.1uF : Capacitors_SMD:C_0805_HandSoldering	2816 Crystals:Resona
Buttons_Switches_THT Look at a footprint	3 CONN_LEDGRID1 - CONN_02X04 : Pin_Headers:Pin_Header_Straight_2x04_Pitch2.54mm	2817 Crystals:Resona
Buzzers_Beepers	4 C_22_1 - 22uF : Capacitors_SMD:C_0805_HandSoldering	2818 Crystals:Resona
Capacitors_SMD	5 C_22_2 - 22uF : Capacitors_SMD:C_0805_HandSoldering	2819 Crystals:Resona
Capacitors_THT	6 C_big1 - 2.2uF : Capacitors_THT:CP_Radial_D5.0mm_P2.50mm	2820 Crystals:Resona
Capacitors_Tantalum_SMD	7 C_blue1 - 0.1uF : Capacitors_SMD:C_0805_HandSoldering	2821 Crystals:Resona

5. Associate footprints to components, Spoiler

```
1 buck1 - buck_converter : TO_SOT_Packages_SMD:SOT-23-6_Handsoldering
2 C_buck1 - 2.2uF : Capacitors_THT:CP_Radial_D5.0mm_P2.50mm
3 C_buck2 - 10uF : Capacitors_SMD:C_0805_HandSoldering
4 C_buck3 - 0.1uF : Capacitors_SMD:C_0805_HandSoldering
5 C_LED1 - 0.1uF : Capacitors_SMD:C_0805_HandSoldering
6 D1 - LED : Resistors_SMD:R_1210_HandSoldering
7 D2 - LED : Resistors_SMD:R_1210_HandSoldering
8 D3 - LED : Resistors_SMD:R_1210_HandSoldering
9 D4 - LED : Resistors_SMD:R_1210_HandSoldering
10 D_buck1 - D_Schottky : Diodes_SMD:D_SMA_Handsoldering
11 ICSP1 - CONN_02X03 : Pin_Headers:Pin_Header_Straight_2x03_Pitch2.54mm
12 J_heat1 - CONN_01X01 : Pin_Headers:Pin_Header_Straight_1x01_Pitch2.54mm
13 J_power1 - CONN_01X02 : Pin_Headers:Pin_Header_Straight_1x02_Pitch2.54mm
14 LED_DRIVER1 - LED_driver : TO_SOT_Packages_SMD:TO-263-7_TabPin4
15 L_buck1 - 6.8uH : Resistors_SMD:R_0805_HandSoldering
16 MCU1 - ATMEGA328-PU : Housings_DIP:DIP-28_W7.62mm
17 MCU_C1 - 0.1uF : Capacitors_SMD:C_0805_HandSoldering
18 MCU_C2 - 22pF : Capacitors_SMD:C_0805_HandSoldering
19 MCU_C3 - 22pF : Capacitors_SMD:C_0805_HandSoldering
20 MCU_R1 - 1M : Resistors_SMD:R_0805_HandSoldering
21 MCU_Y1 - Crystal : Crystals:Crystal_HC18-U_Vertical
22 RV1 - POT_TRIM : potentio_1:potential
23 R_buck1 - 51k : Resistors_SMD:R_0805_HandSoldering
24 R_buck2 - 3k9 : Resistors_SMD:R_0805_HandSoldering
25 R_buck3 - 10k : Resistors_SMD:R_0805_HandSoldering
26 R_buck4 - 100k : Resistors_SMD:R_0805_HandSoldering
27 R_LED1 - 0.5 : Resistors_SMD:R_0805_HandSoldering
```

5. Associate footprints to components, create netlist

- Remember to save your changes



- Close this window, return to main window and generate netlist



Intermission, continues tomorrow...

- Today we told KiCAD what components are to be placed on the PCB and how they should be connected
- Tomorrow we shall draw where the components and wires will be physically on the board

Recap, where we are, and intermission

- 1: **Decide what is to be done**
- 2: **Choose components**
 - Use datasheets when doing this
 - Make sure components are compatible
- 3: Open KiCAD and start a new project
- 4: **Add the components and make the wirings in EESchema**
 - Use datasheets to find correct pins
- 5: Associate footprints to components and create a netlist
- <- We are here
- 6: Open PCBnew, read the netlist and set design rules
- 7: **Place components , wirings, borders, filled areas etc.**
- 8: Create files needed to manufacture PCB
 - In workshop: plot the pdf-files (MIRROR THE FRONT COPPER IMAGE)
 - Ordered from a factory: plot gerbers and drill file



• See you tomorrow!

Welcome back!

- 1: **Decide what is to be done**
- 2: **Choose components**
 - Use datasheets when doing this
 - Make sure components are compatible
- 3: Open KiCAD and start a new project
- 4: **Add the components and make the wirings in EESchema**
 - Use datasheets to find correct pins
- 5: Associate footprints to components and create a netlist
- <- We are here
- 6: Open PCBnew, read the netlist and set design rules
- 7: **Place components , wirings, borders, filled areas etc.**
- 8: Create files needed to manufacture PCB
 - In workshop: plot the pdf-files (MIRROR THE FRONT COPPER IMAGE)
 - Ordered from a factory: plot gerbers and drill file

6. Open PCBnew

- Open KiCAD
- Open EESchema



- Then open PCBnew



- Wait...

6. PCBnew, set design rules



Net Classes Editor		Global Design Rules				
Net Classes:						
	Clearance	Track Width	Via Dia	Via Drill	uVia Dia	uVia Drill
Default	0,2	0,3	1,6	0,8	0,3	0,1

Design Rules Editor

Net Classes Editor Global Design Rules

Via Options:

Blind/buried Vias:

☒ Do not allow blind/buried vias
☐ Allow blind/buried vias

Micro Vias:

☒ Do not allow micro vias
☐ Allow micro vias

Minimum Allowed Values:

Min track width (mm): 0,2
Min via diameter (mm): 0,4
Min via drill dia (mm): 0,3
Min uvia diameter (mm): 0,2
Min uvia drill dia (mm): 0,1

Specific via diameters and track widths, which can be used to replace default Netclass values on demand, for arbitrary vias or track segments.

Custom Via Sizes:

Drill value: a blank or 0 => default Netclass value

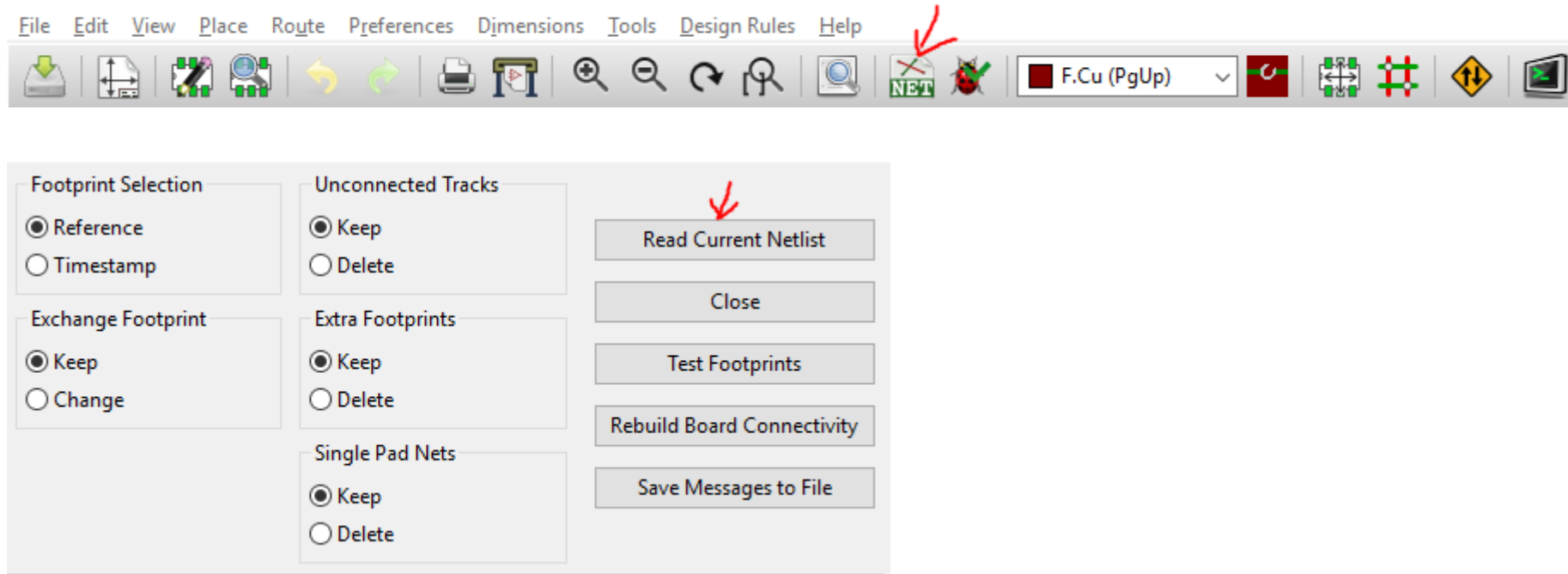
	Diameter	Drill
Via 1		
Via 2		
Via 3		
Via 4		
Via 5		
Via 6		
Via 7		
Via 8		

Custom Track Widths:

	Width
Track 1	0,5
Track 2	0,7
Track 3	2
Track 4	
Track 5	
Track 6	
Track 7	
Track 8	

OK Cancel

6. PCBnew, read netlist

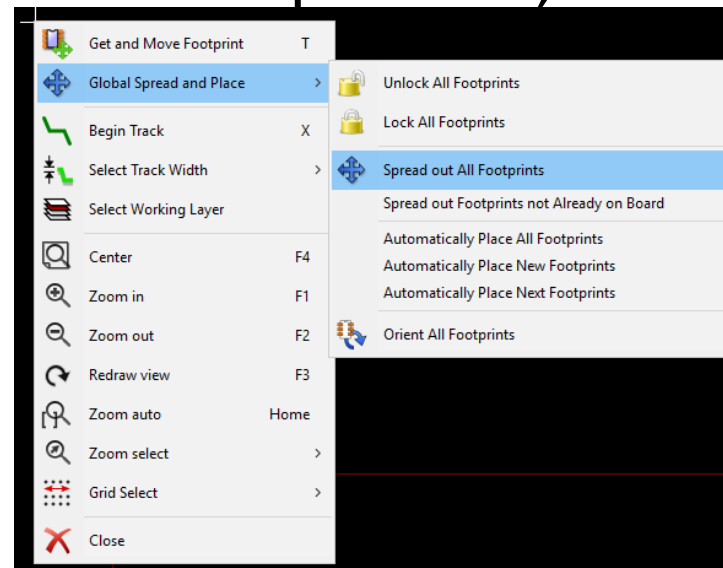


6. PCBnew spread out all components

- Now your components are on top of one another, time to spread them out...

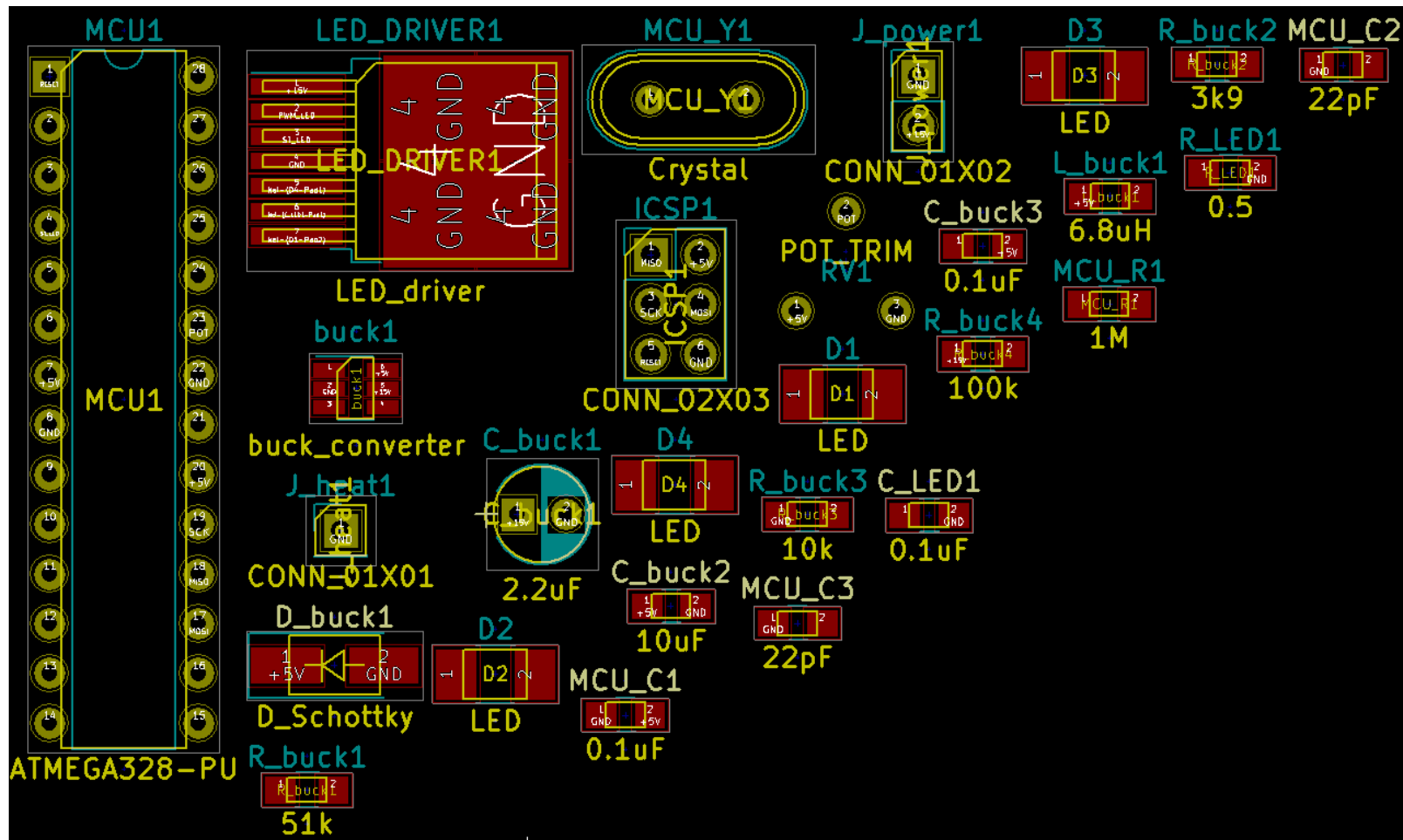


- Right click next to the components, and...



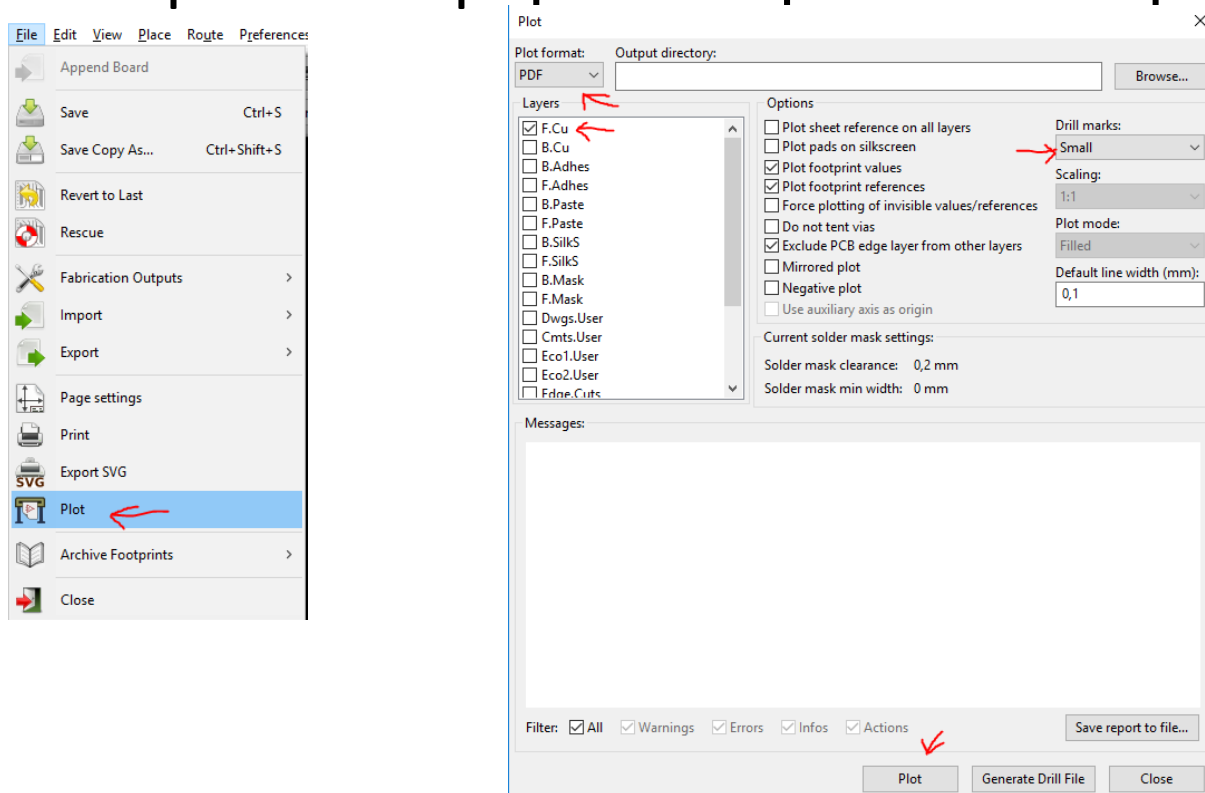
6. PCBnew, spreaded out components

- Now, you should see all the components spread out

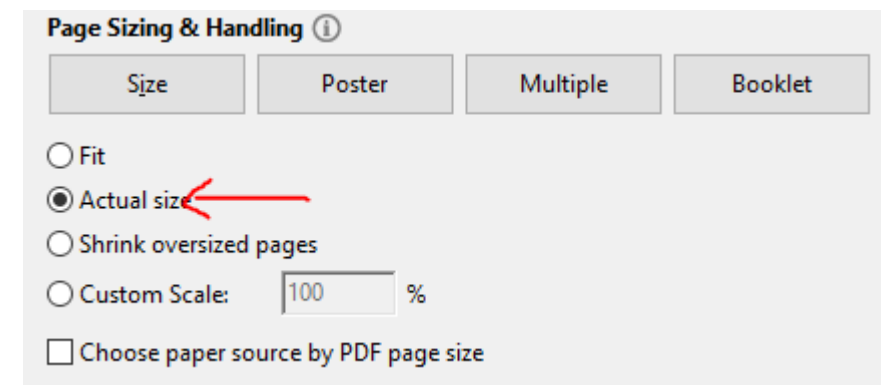


6. PCBnew, verify footprints manually

- To verify that we have chosen the correct footprints, let's print all the footprints on paper and put the components on the paper

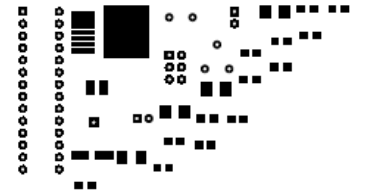


- Print the plotted pdf on a school printer
 - Hint: choose "Actual size" when printing



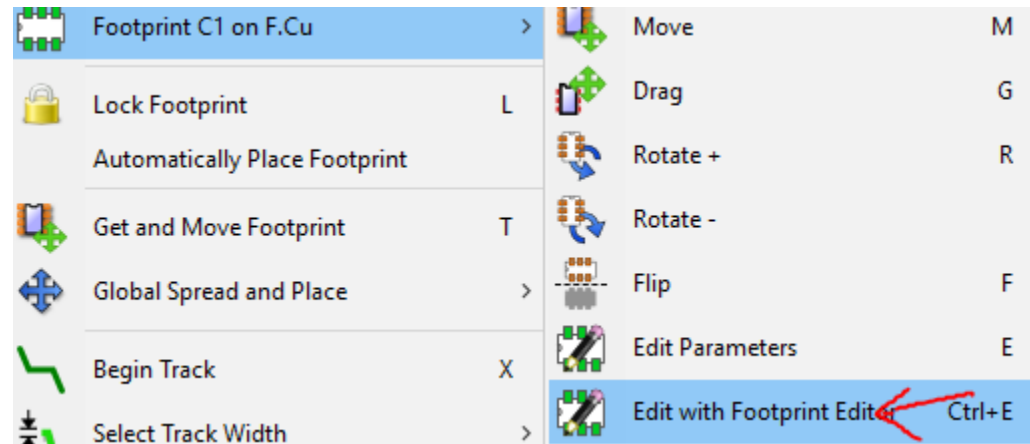
6. PCBnew, verify footprints manually

- Your pdf should look something like this:
- The black parts are the footprints



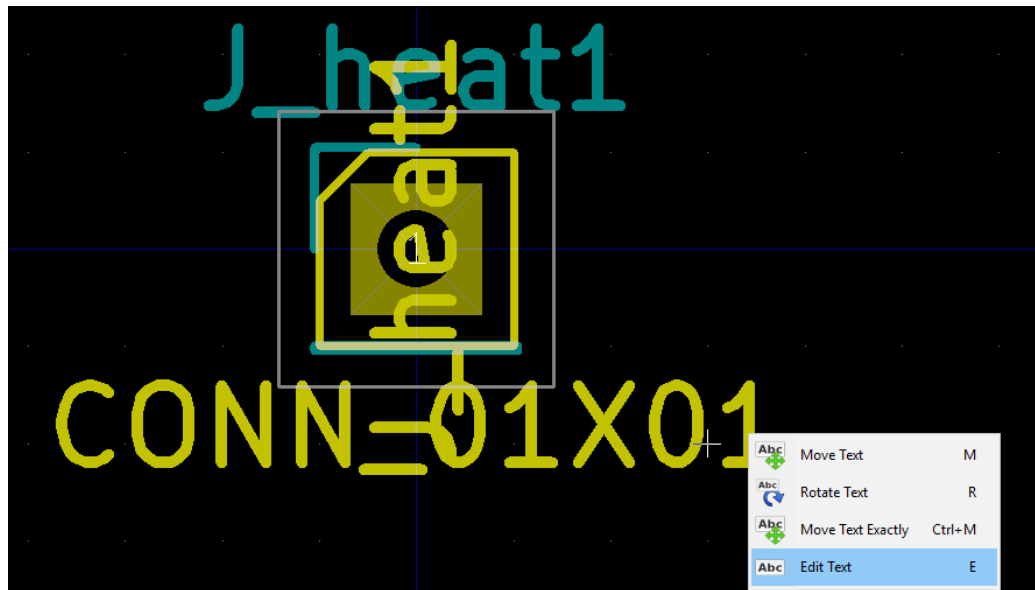
6. Modifying footprints

- If a footprint is completely wrong, you should:
 - 1: Return to EESchema and open CvPcb where the footprint were chosen
 - 2: Choose a better footprint
 - 3: Save changes in CvPcb, and generate a new netlist in EESchema
 - 4: Return to PCBnew, delete the bad footprint, and read the netlist
 - 5: Spread out all the footprints and see if you have had better luck
- If only minor modifications are needed, then right click the footprint, and:

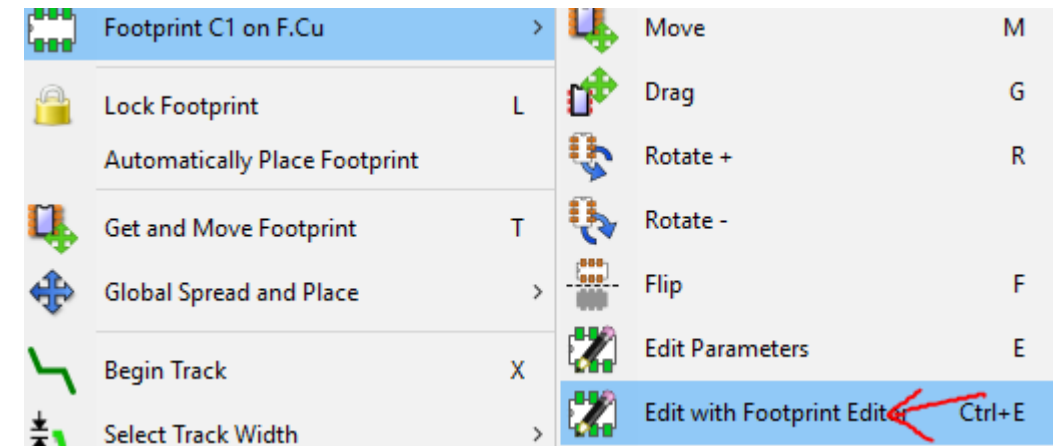


6. Modifying footprints, heat via

- Let's modify the heat via, right click the footprint, and:
- First, let's remove the texts,
- Right click on the text -> Edit Text

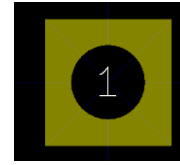


- Remove all text from value box



6. Modifying footprints, heat via

- Then, lets remove the lines (delete):



- Then right click on the pad: -> edit pad

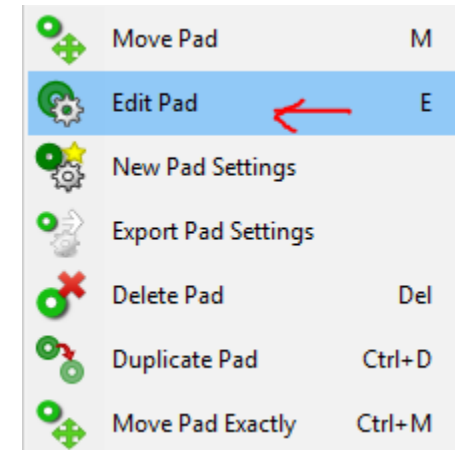
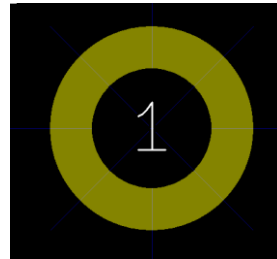
- Then change it to circular, and press ok

Pad number: 1

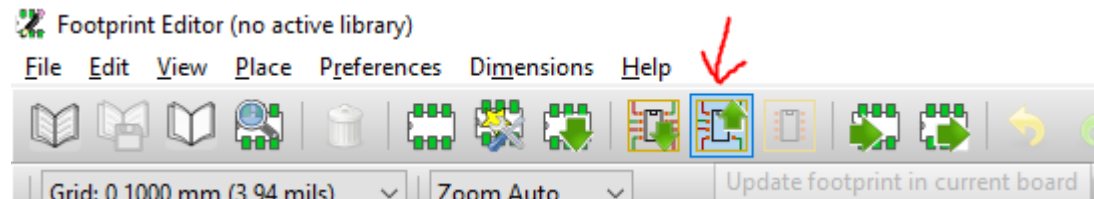
Net name:

Pad type: Through-hole

Shape: Circular



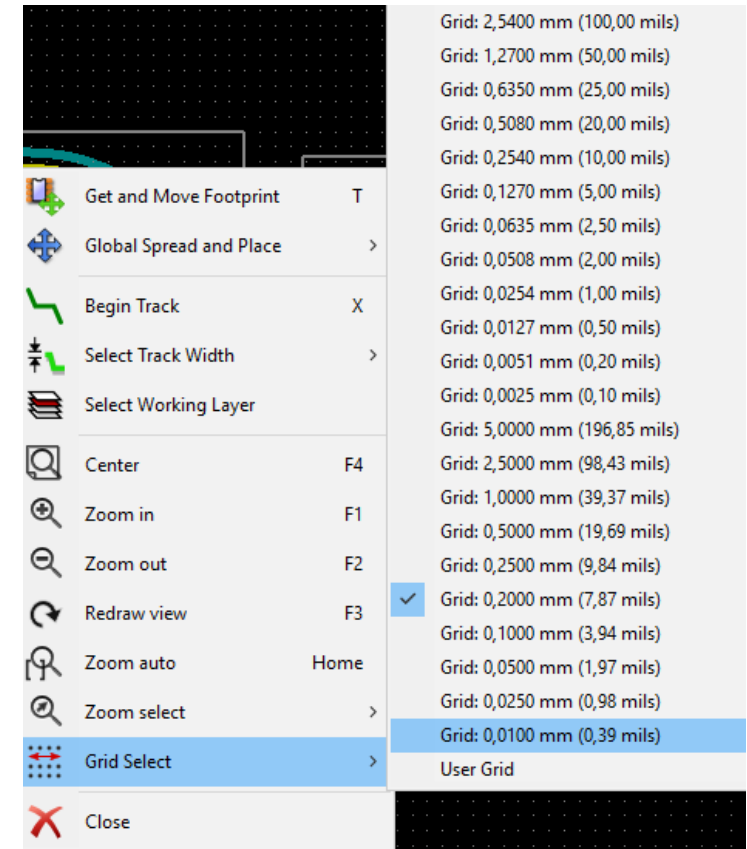
- Finally, update the footprint on the board:



- You can now close the footprint editor

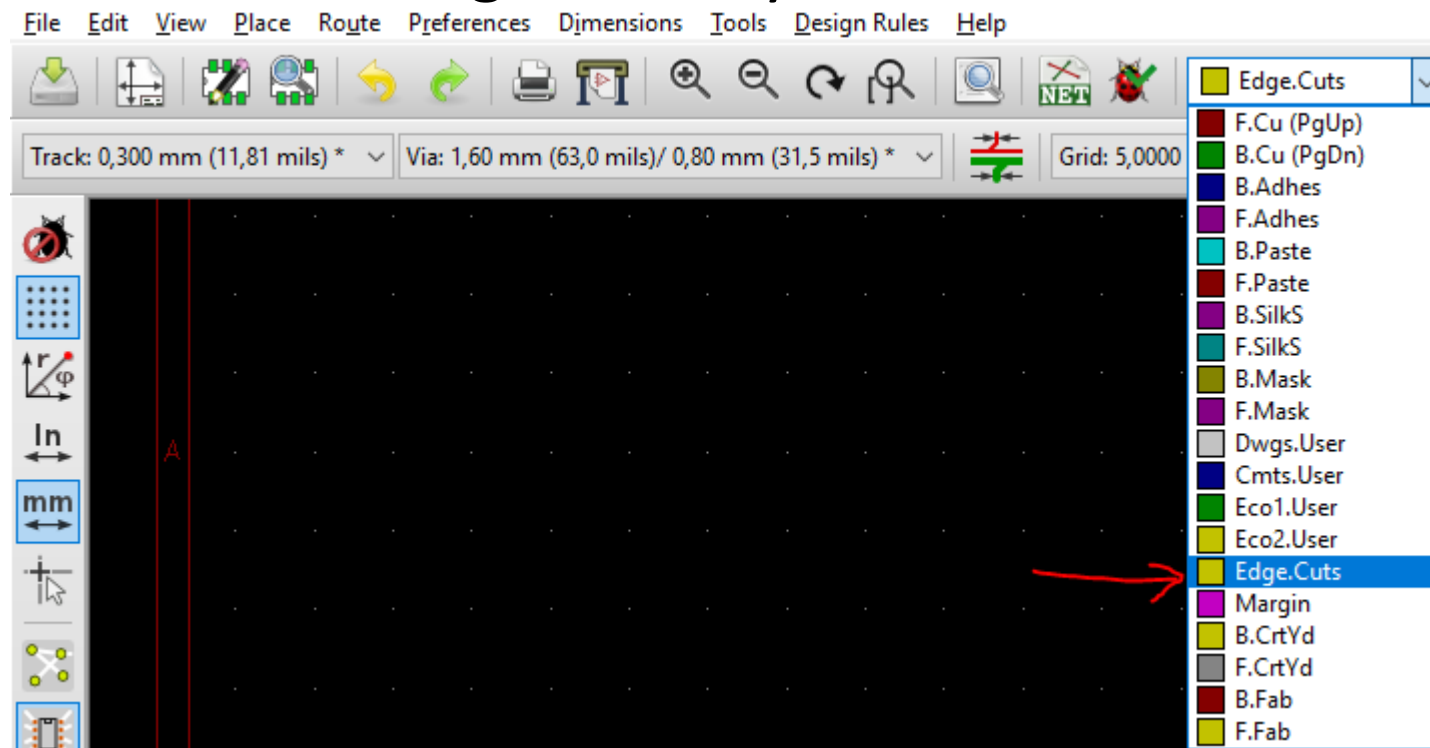
7. Moving footprints

- Footprints are moved by hovering the mouse over them, and pressing M, and can be rotated in 90-degree increments by pressing R, and placed where desired by clicking
- You can change the movement precision by choosing a different grid by right clicking on the background and choosing a different grid



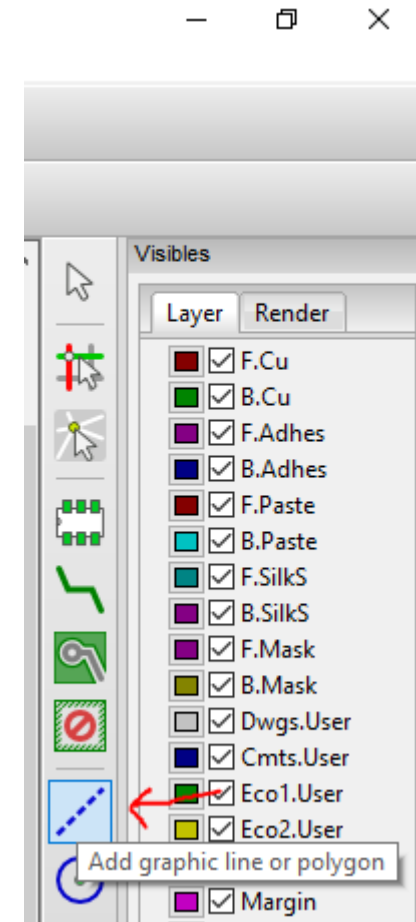
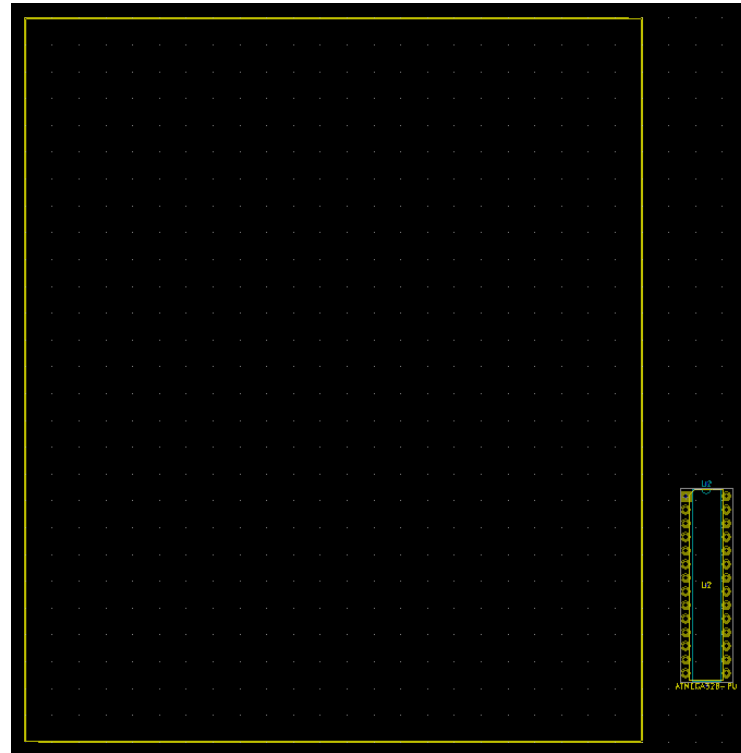
7. Drawing the borders of the board

- Let's start by drawing the borders
- Choose the Edge.Cuts layer:



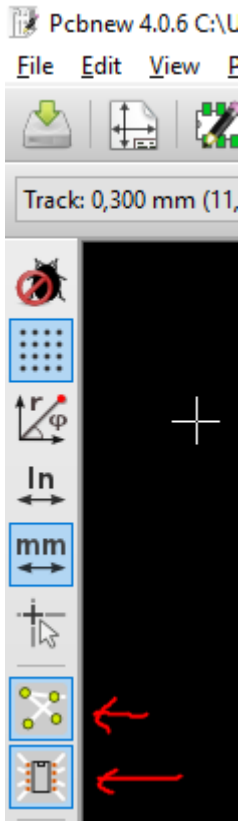
7. Drawing the borders of the board

- Then choose the tool to draw lines ->
- Now, draw some rectangular borders
 - Draw the box big enough so all the components and wirings will fit

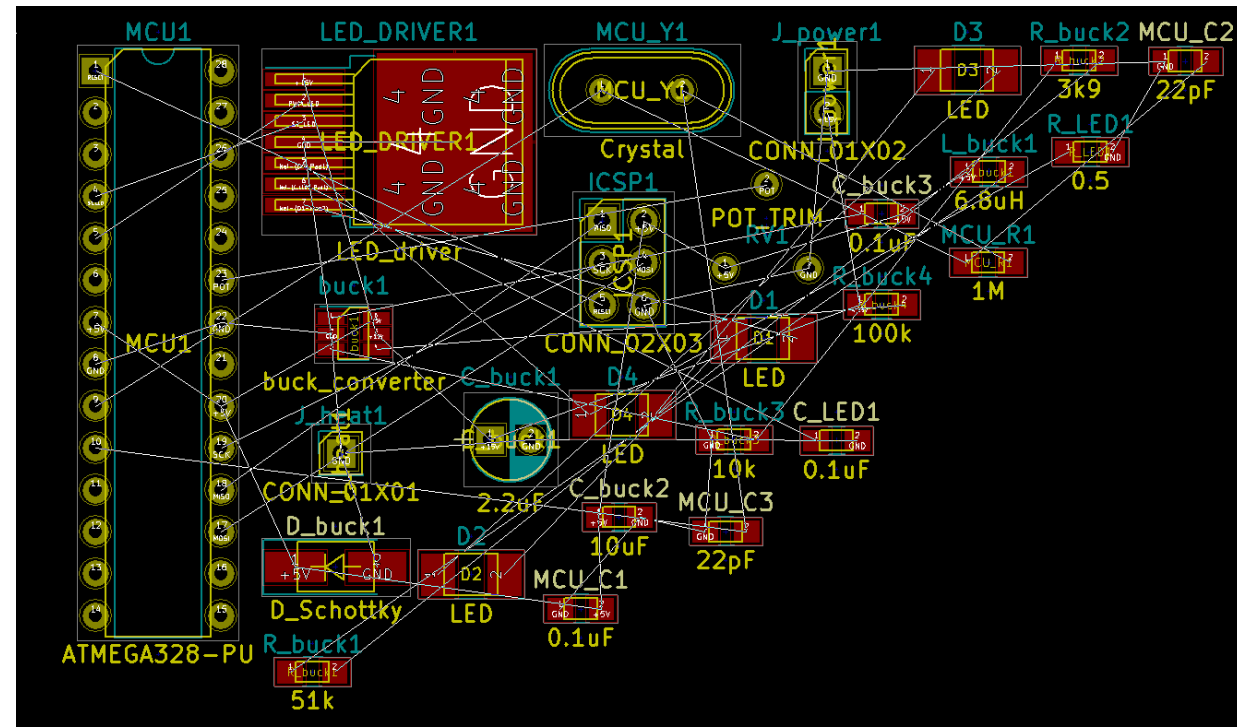


7. Moving the footprints

- First, enable "rats nests"

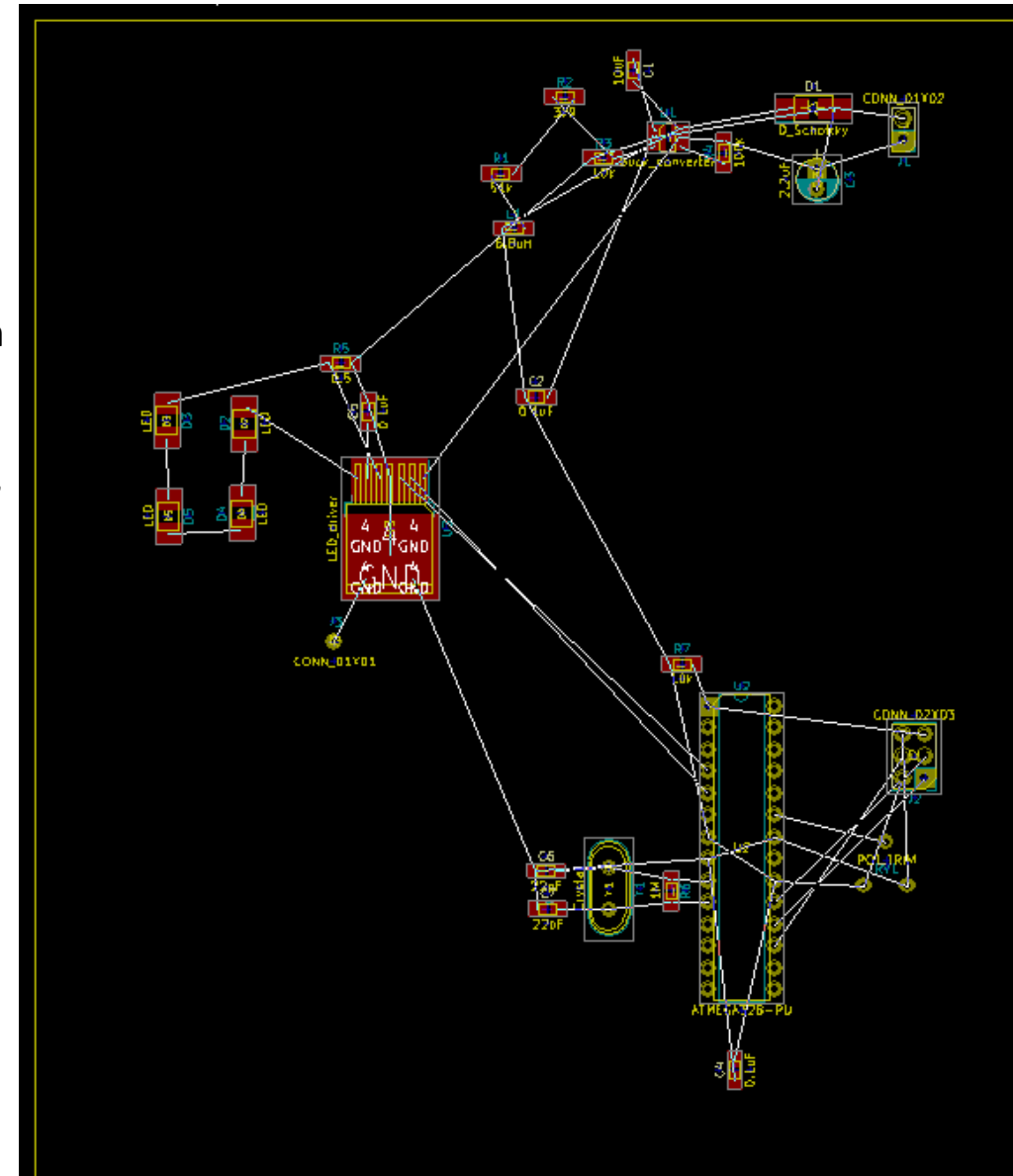


- Now when you are moving a component, you can see where it should be connected to



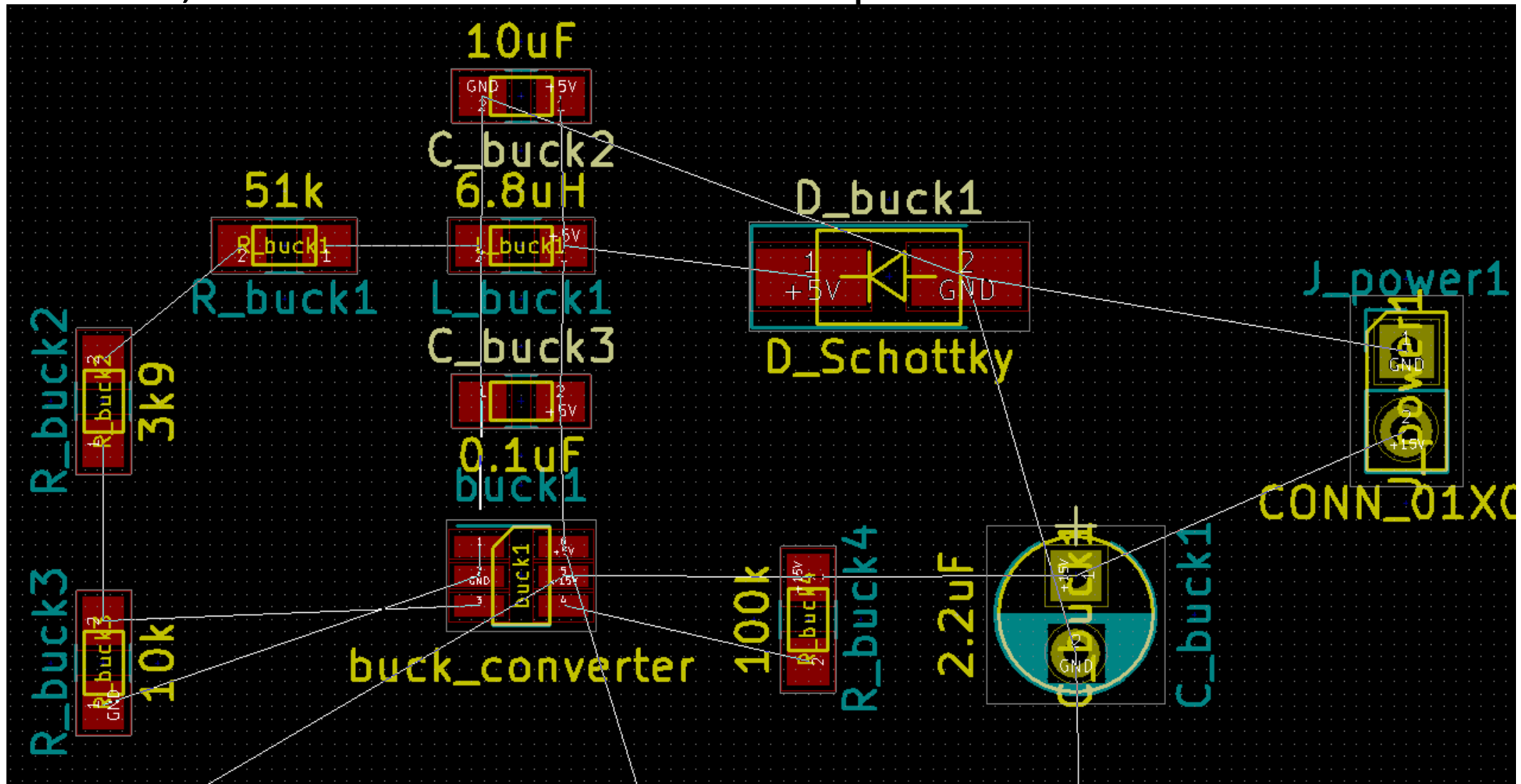
7. Moving the footprints, Spoiler

- Figuring out where components should be placed is a time-consuming task
- To save time during this demo, the board was drawn big, and a suggestion is provided
- Footprints are moved by hovering the mouse over them, and pressing M, and can be rotated in 90-degree increments by pressing R, and placed where desired by clicking
- Move the components as follows
 - 1: Move the microcontroller and the components related to it (MCU) to the lower right corner
 - 2: Move the buck converter and the components related to it (buck) to the upper right corner
 - 3: Move the rest of the parts (LED-driver and LEDs) to the left



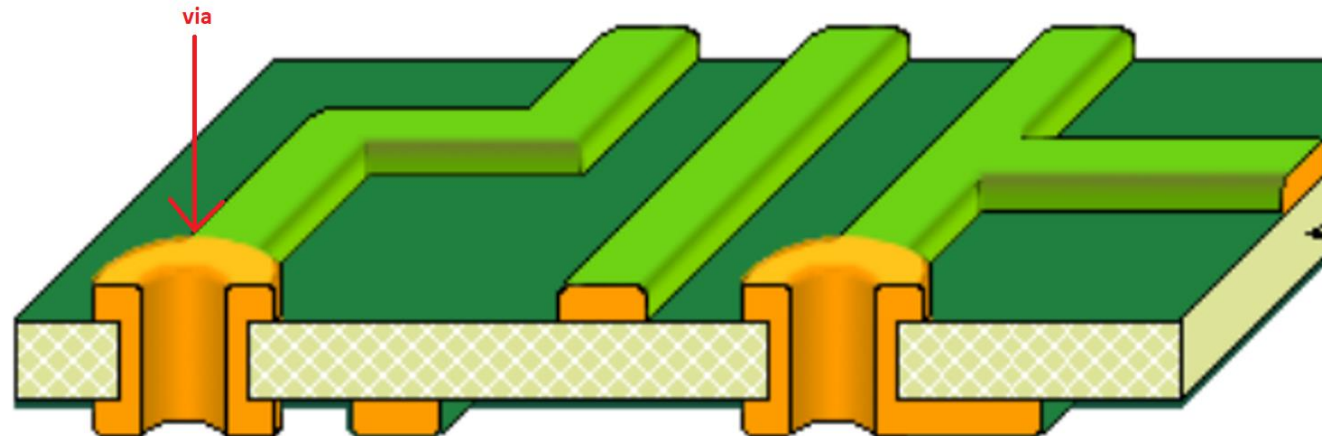
7. Moving the footprints, buck converter

- First, move the buck converter components like this



7. Drawing the wiring

- Now its time to draw the wires between components
- Try to avoid making sharp 90 degree turns if possible: they may pick up radio interference
- Try to have thick wires when possible: low resistance
- Wires can be placed on both sides of the board, different sides are connected via vias



7. Drawing the wiring

- Sometimes it can be difficult to connect wires without intersecting, sometimes a jumper is a possibility
 - In some cases a piece of thin wire may be used



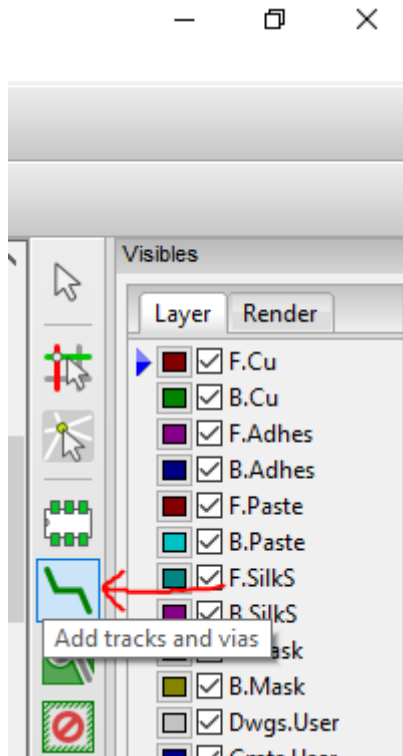
- A zero-Ohm smd resistor can be also used like a "bridge"



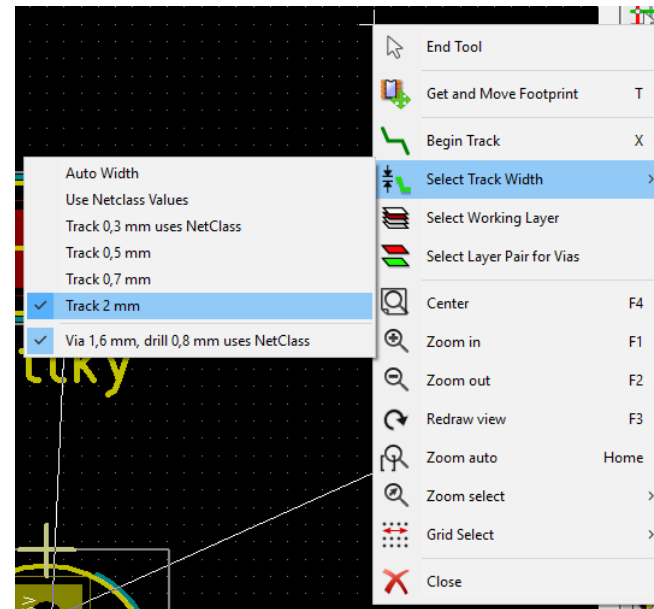
- Try to avoid these tricks if possible

7. Drawing the wiring

• 1:



2: Right click and choose 2mm track width:

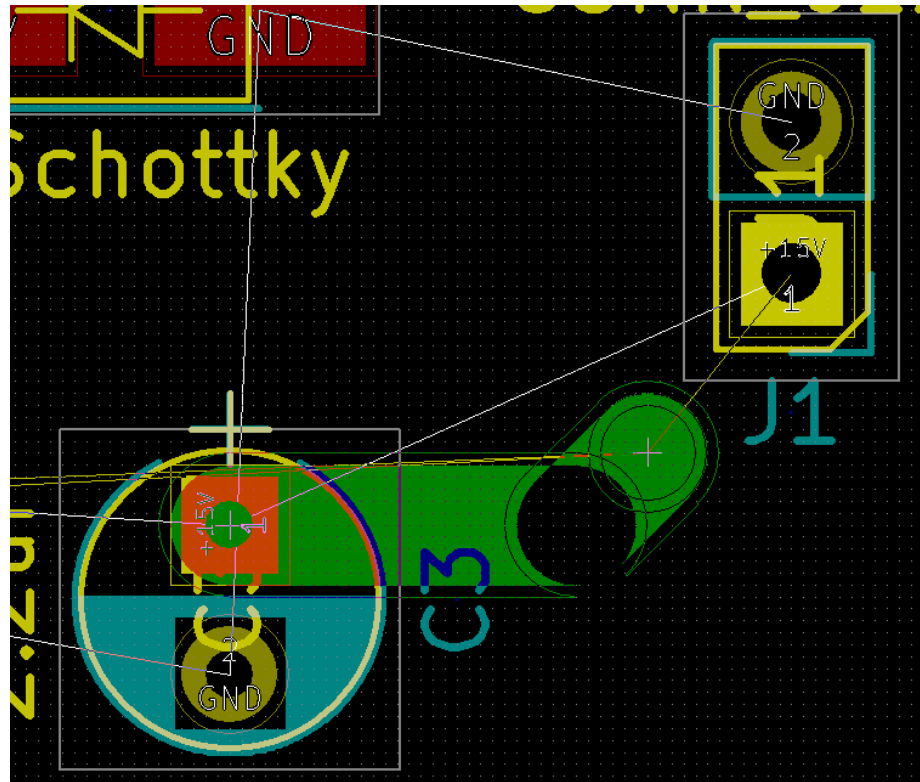


• 3: Choose to start drawing wires on the backside

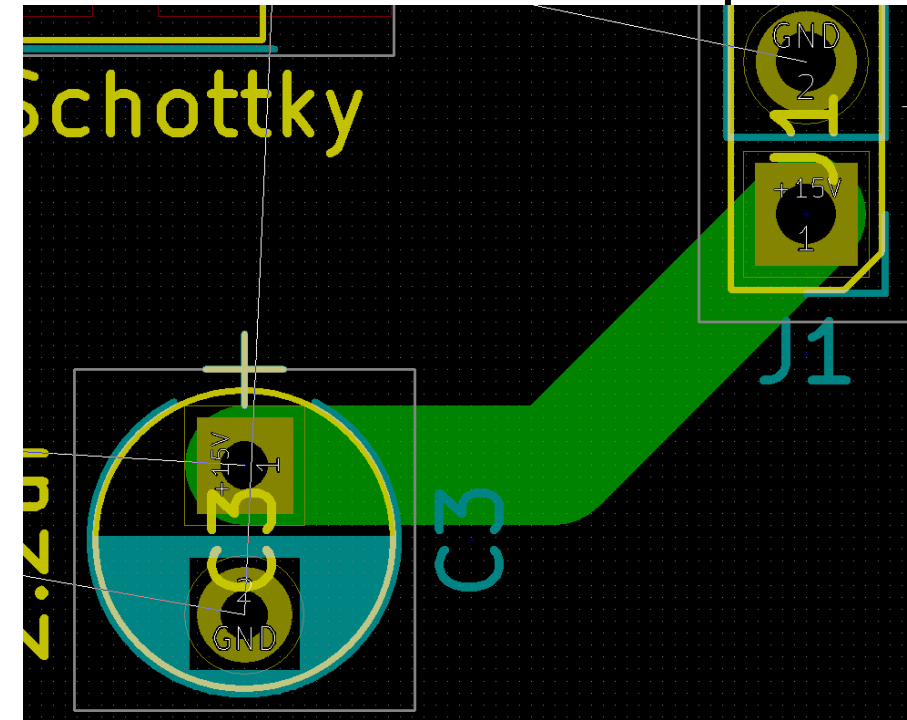


7. Drawing the wiring

- 1: Click the upper pin of the round capacitor 2: Double click the



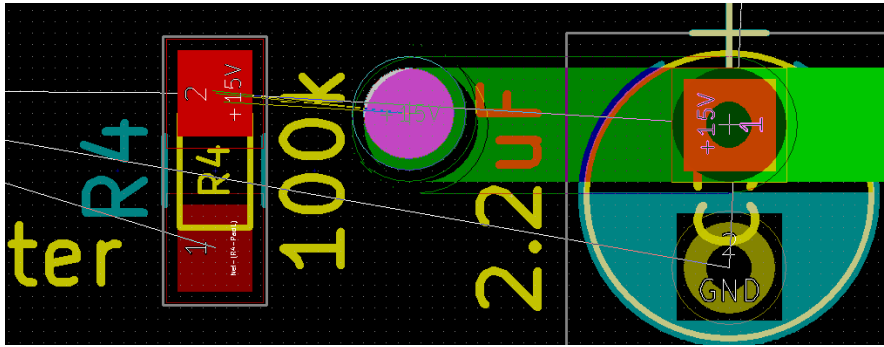
lower connector pin



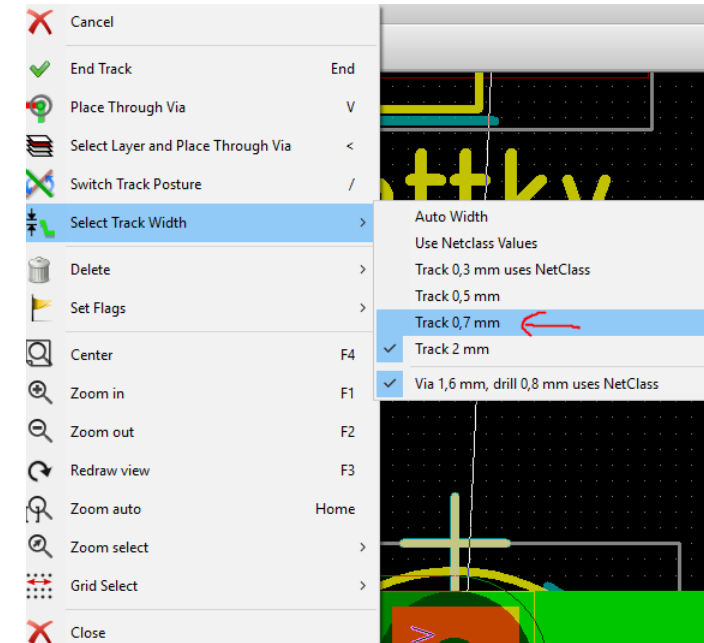
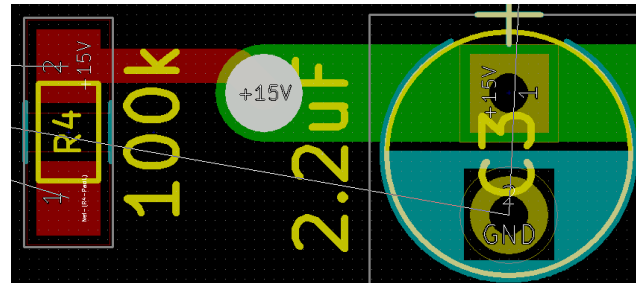
- These two pins are now connected

7. Drawing the wiring

- 1: Click on the same round capacitor pin, move the mouse a bit left and press V, this creates a via here

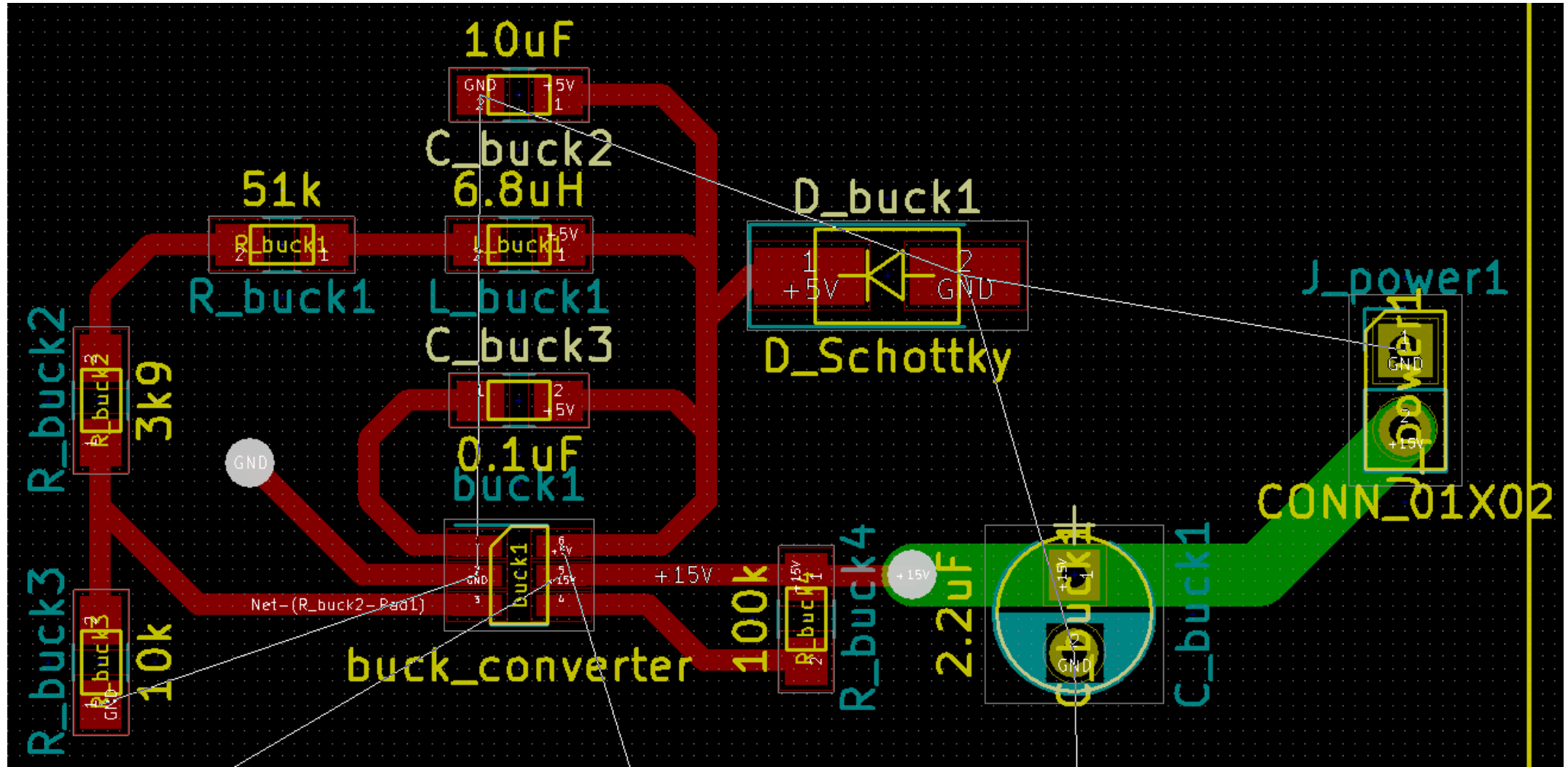


- 2: Then right click and change the track width to 0.7
- 3: Finally, double click on the resistor upper pin



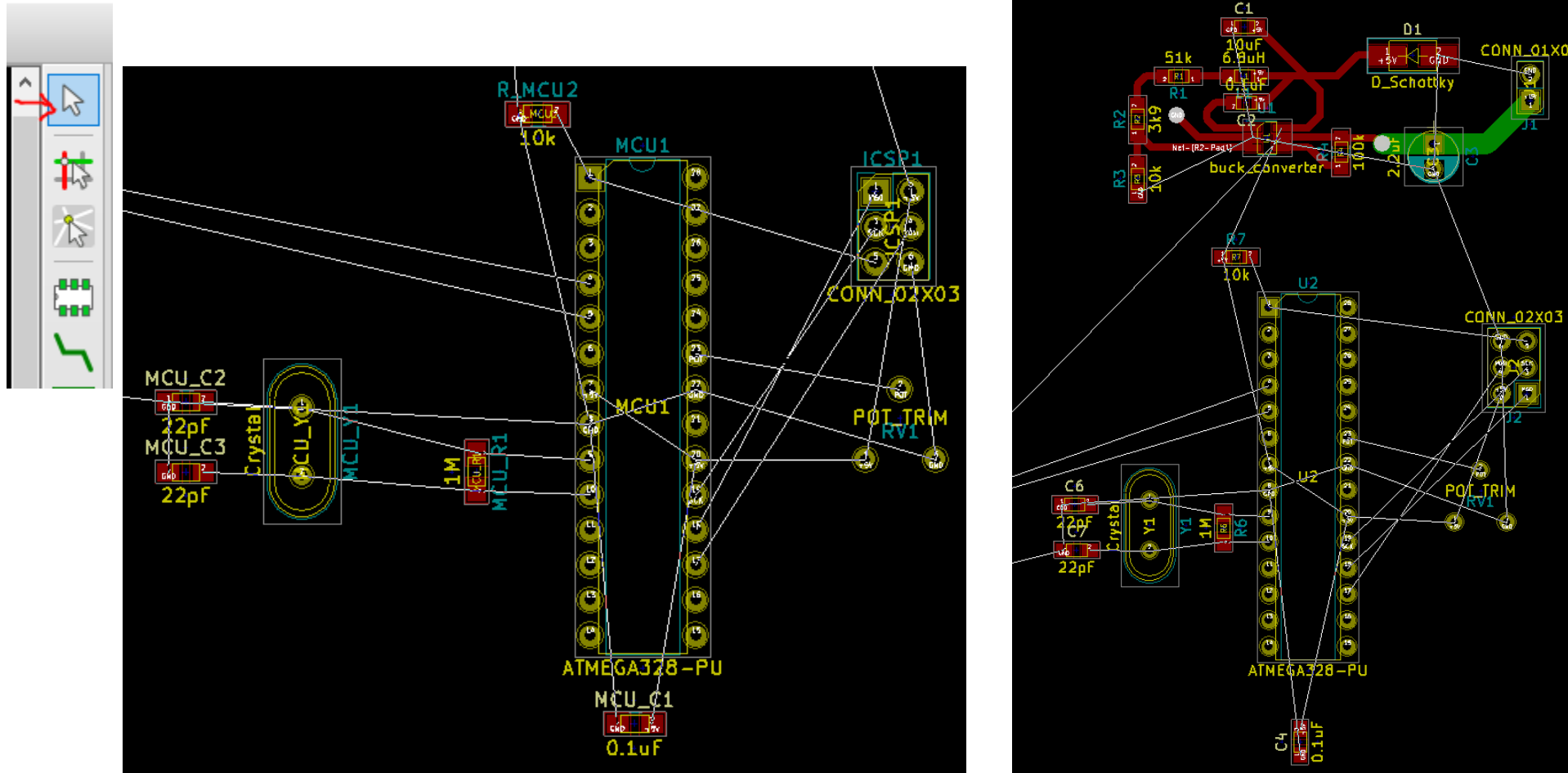
7. Drawing the wiring, buck converter

- Lets draw the other buck converter wires
- Hint: start the wires from the 6-footed converter



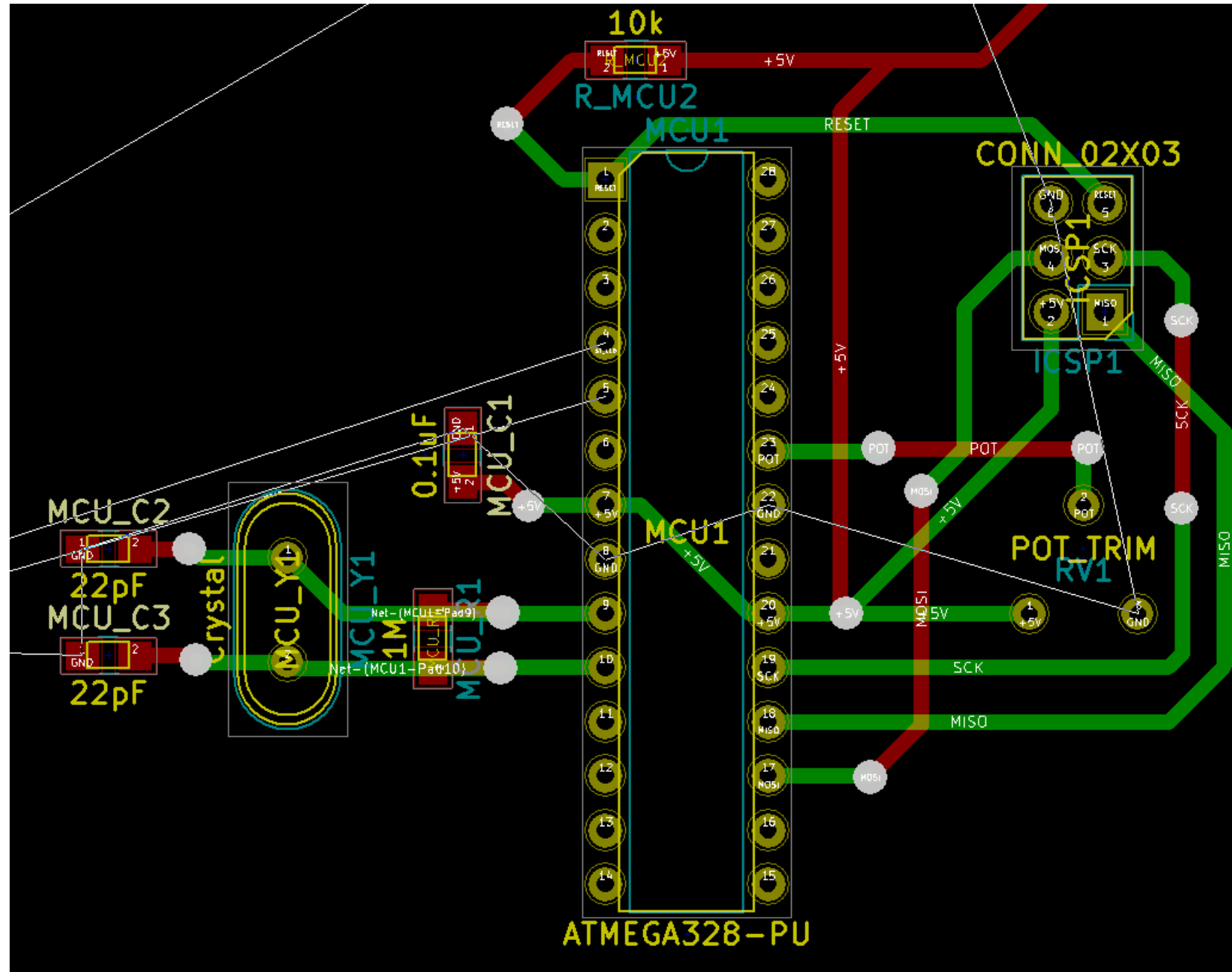
7. Drawing the wiring, the microcontroller

- Let's move the microcontroller et al. a bit closer to the buck converter, lasso them



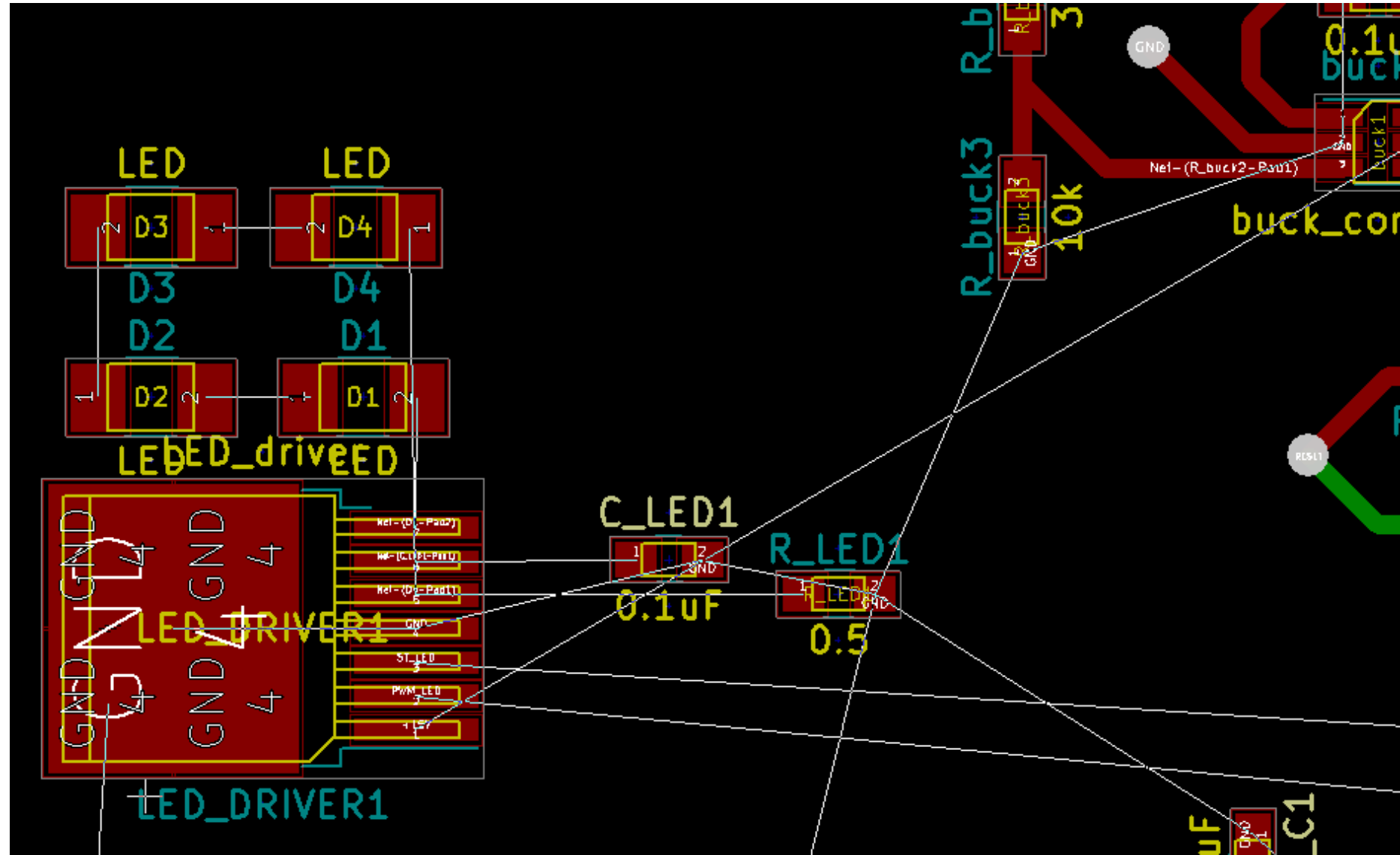
7. Drawing the wiring, the microcontroller, Spoiler,

- Principle:
 - Red wires are on the frontside, green on the backside
- Connect to holes (yellow) from the back
- Hint: you can move the components around for a bit



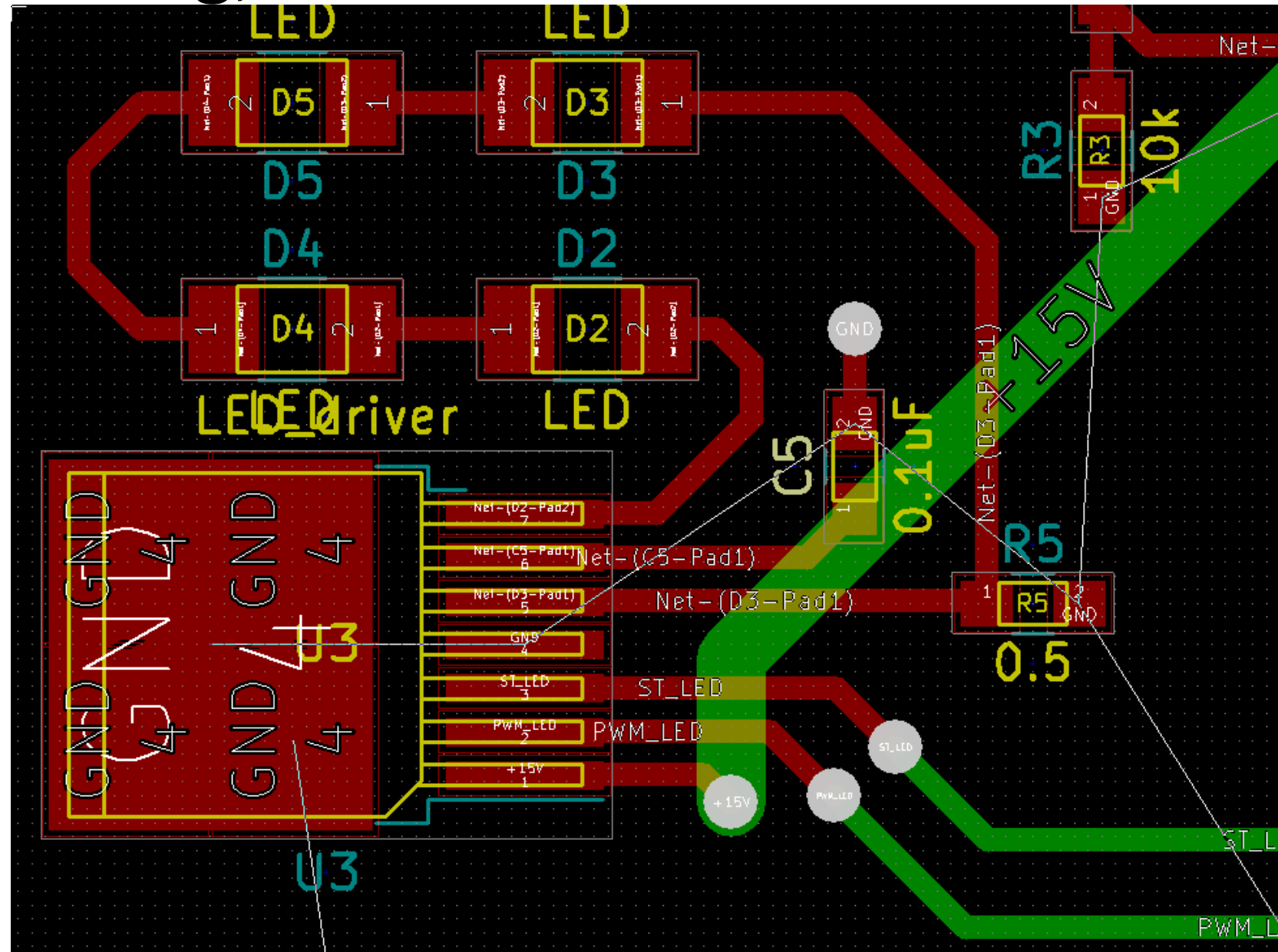
7. Drawing the wiring, the LED-driver and leds

- Let's move the LED-driver et al. a bit closer:



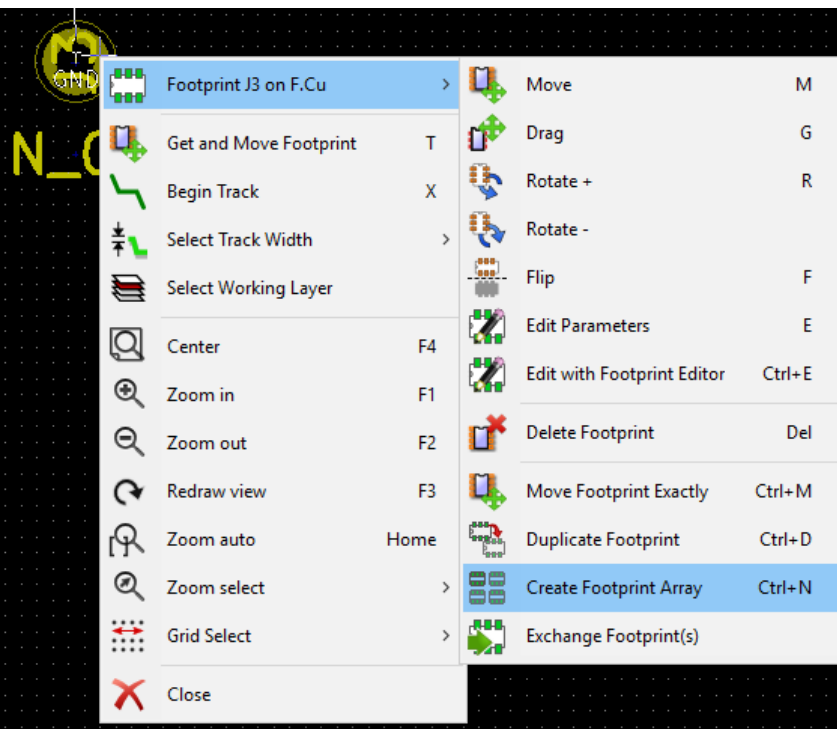
7. Drawing the wiring, the LED-driver and LEDs, Spoiler

- Lets wire them up



7. Drawing the wiring, heat vias

- The LED-driver might heat, so let's put heat up vias underneath it just in case
- Right click it, and create a duplicate
- Right click the duplicate, and create an array of them



Create Array

Grid Array

Circular Array

Horizontal count:

3

Vertical count:

3

Horizontal spacing:

3

mm

Vertical spacing:

3

mm

Horizontal offset:

0

mm

Vertical offset:

0

mm

Stagger:

1

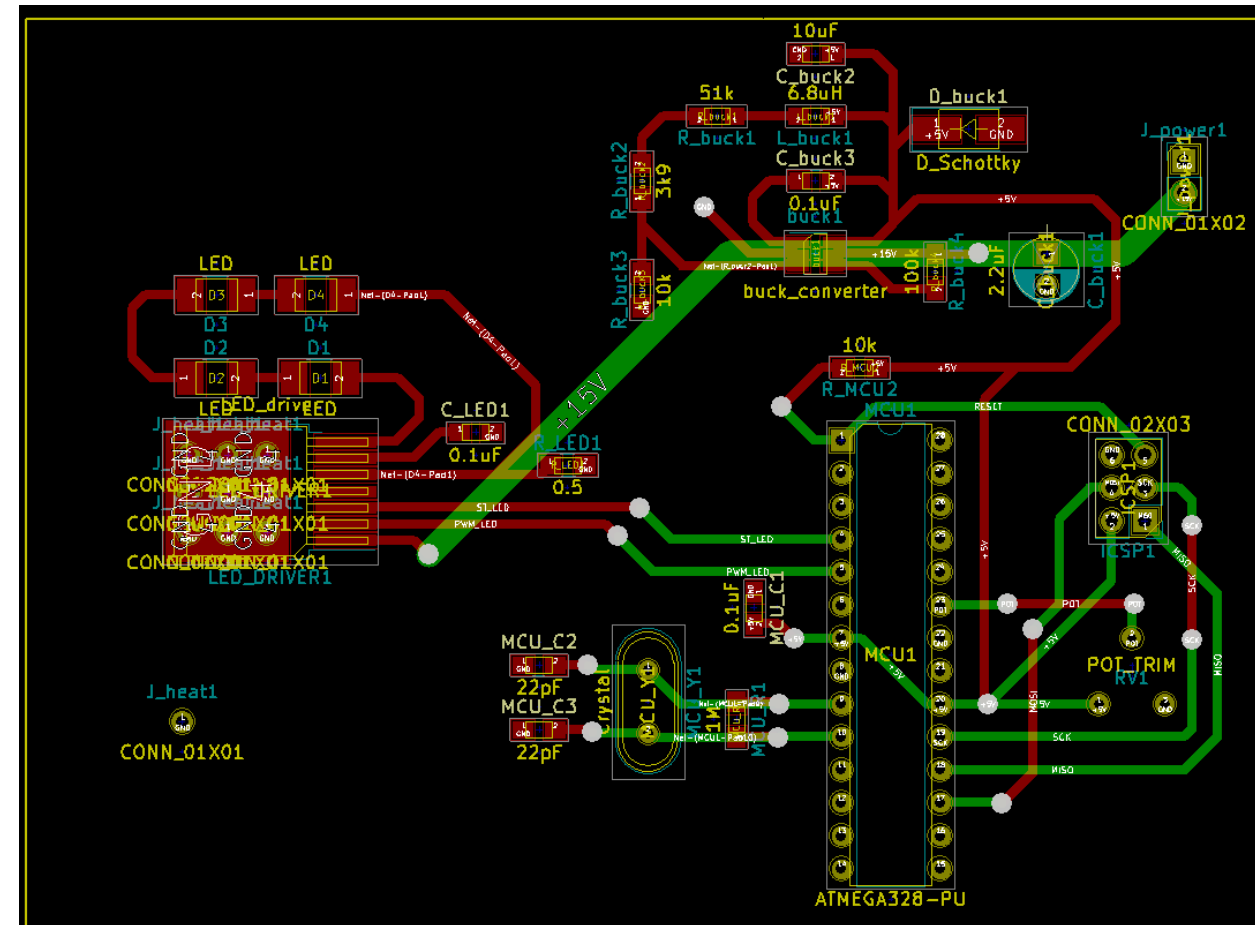
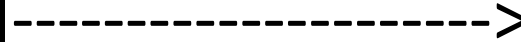
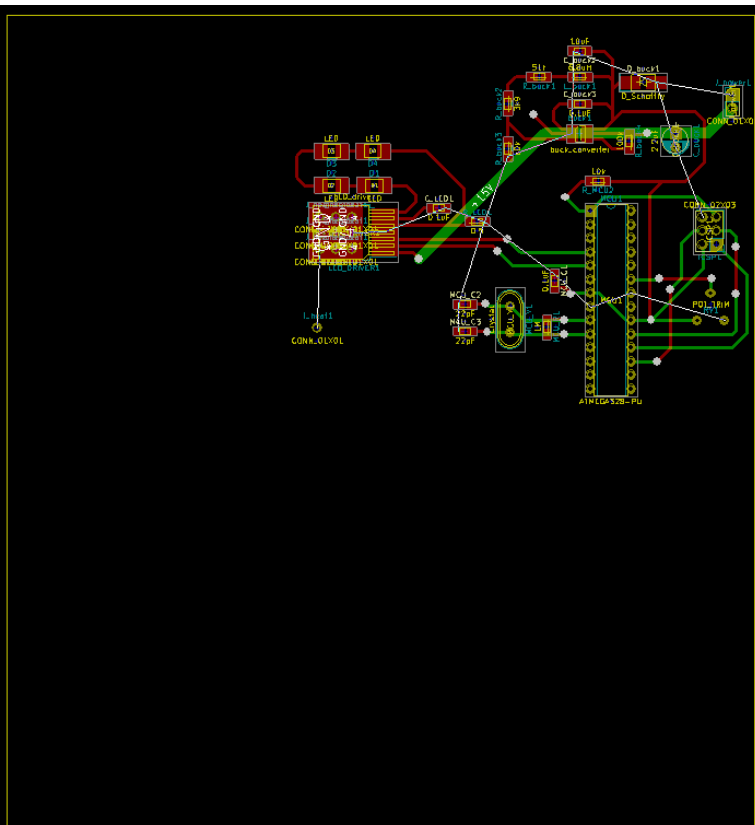
7. Drawing the wiring, heat vias

- Then, move the array underneath the LED-driver



7. Drawing the wiring, resizing the board

- As we can see, the area we chose at the beginning is too large
- Time to delete the old borders and draw new with Edge.Cuts...

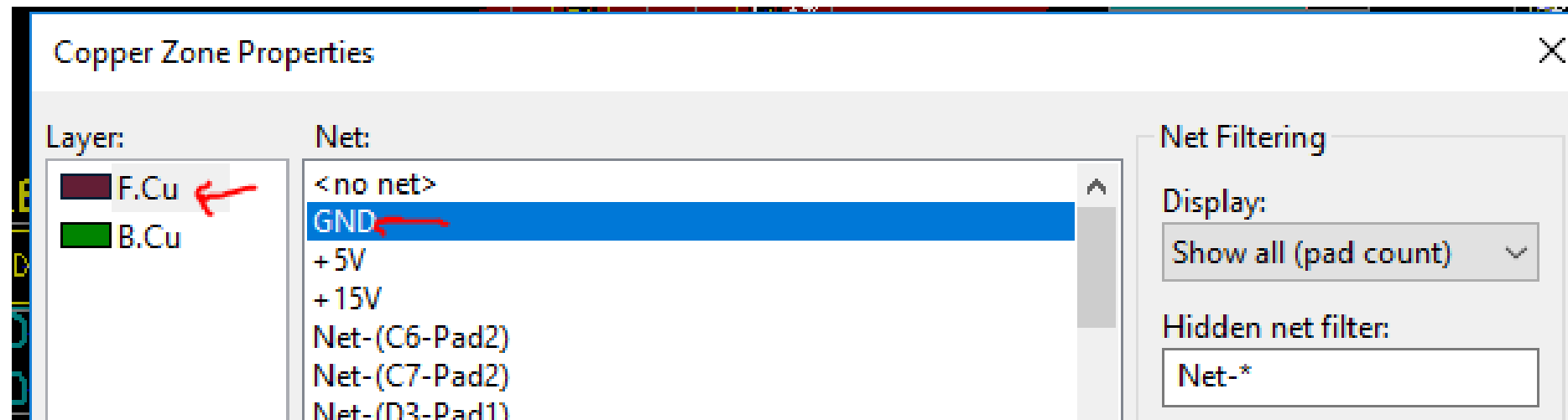


7. Drawing the wiring, adding fills

- We never connected the ground pins because we can simply fill the empty areas with ground plane

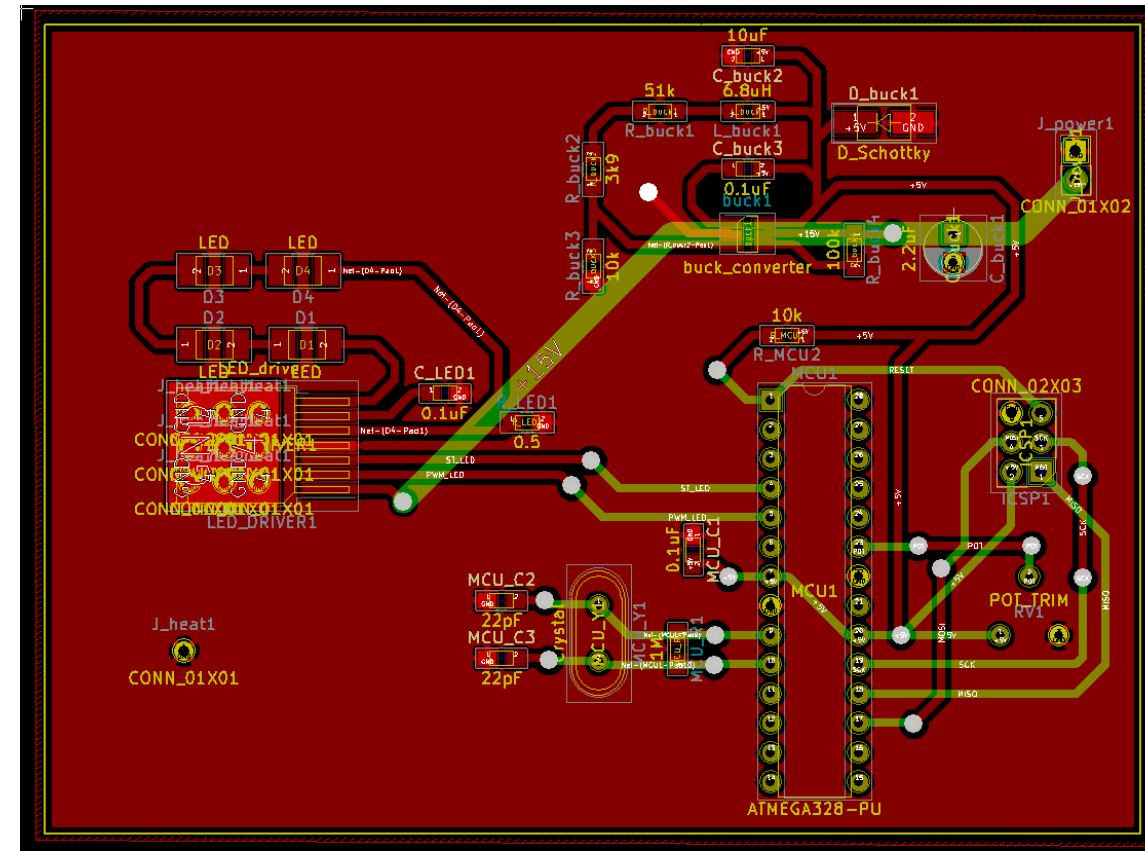


And then click on the corner of the board, and choose:



7. Drawing the wiring, adding fills

- Include the whole board in the area, and press B
- We can see that the empty areas are filled with copper
- Do the same thing, but for backside

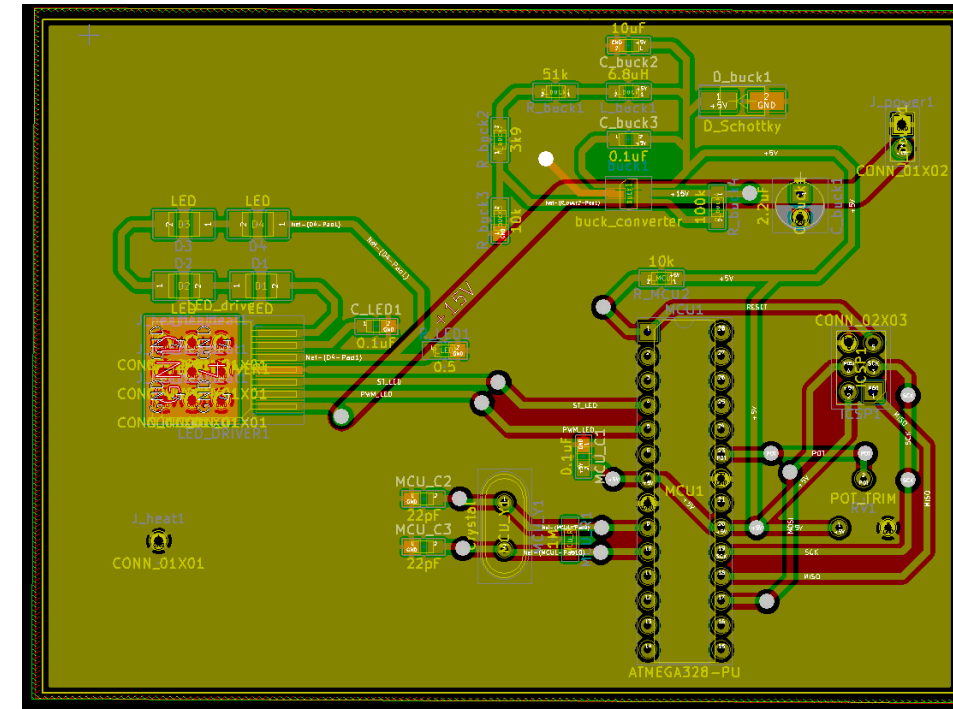
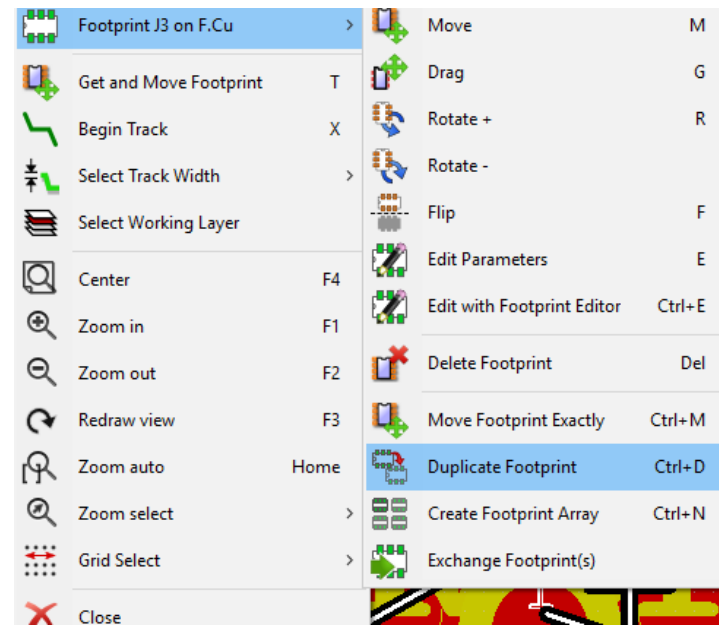


7. Drawing the wiring, adding fills

- Looks good, but now we cannot see things properly, press CTRL + B to remove fills temporarily
- Its usually not a good idea have large unbroken areas of filling, so let's add some additional heat vias in the "empty" areas
- Paste the duplicate many copies of the heat via in various places

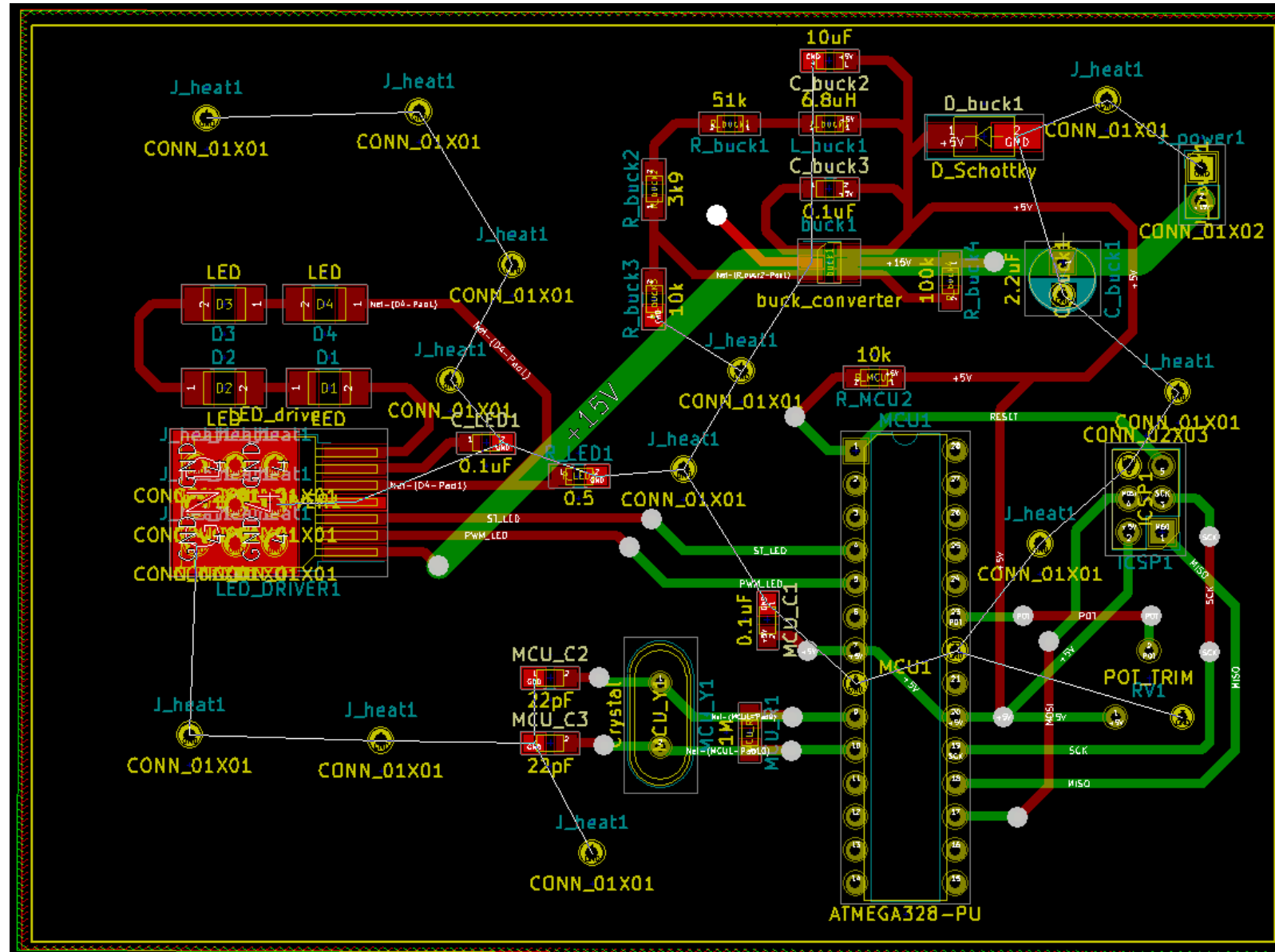
in various places

Hint: CTRL + D



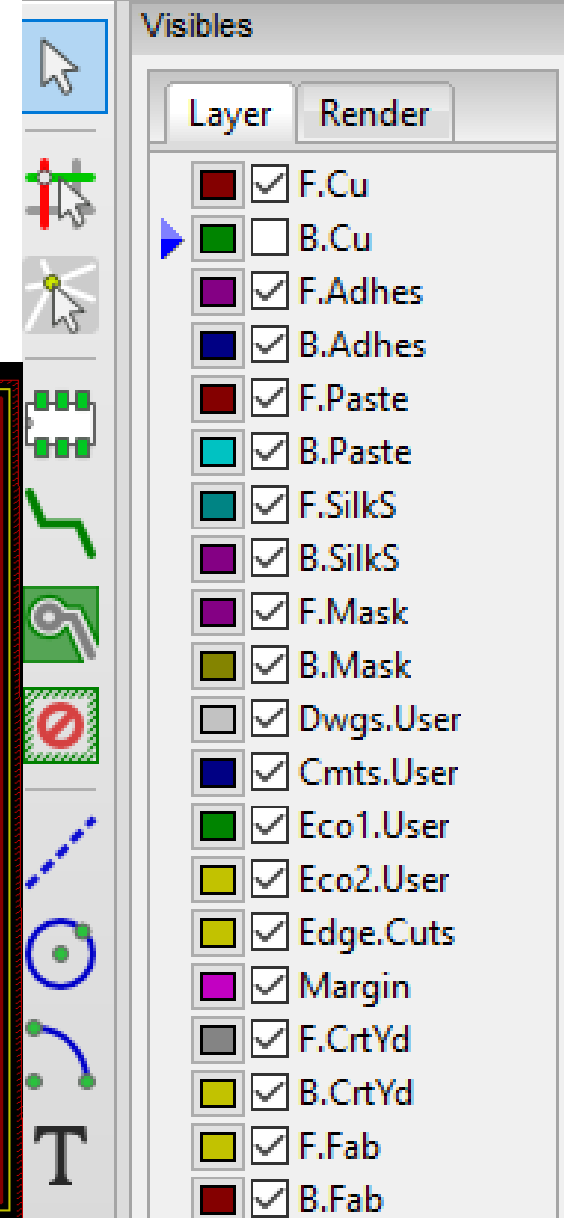
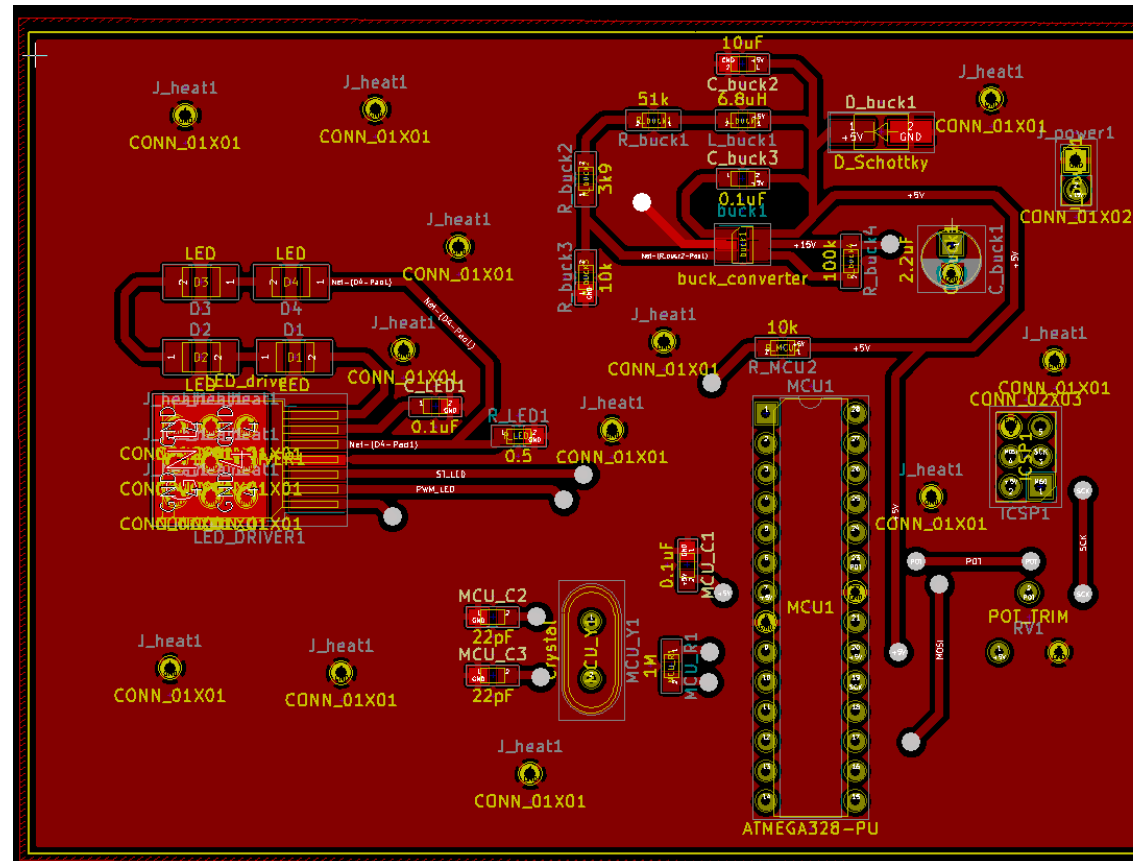
7. Drawing the wiring, adding fills

- Start adding duplicates in empty areas
- Press B to add fills



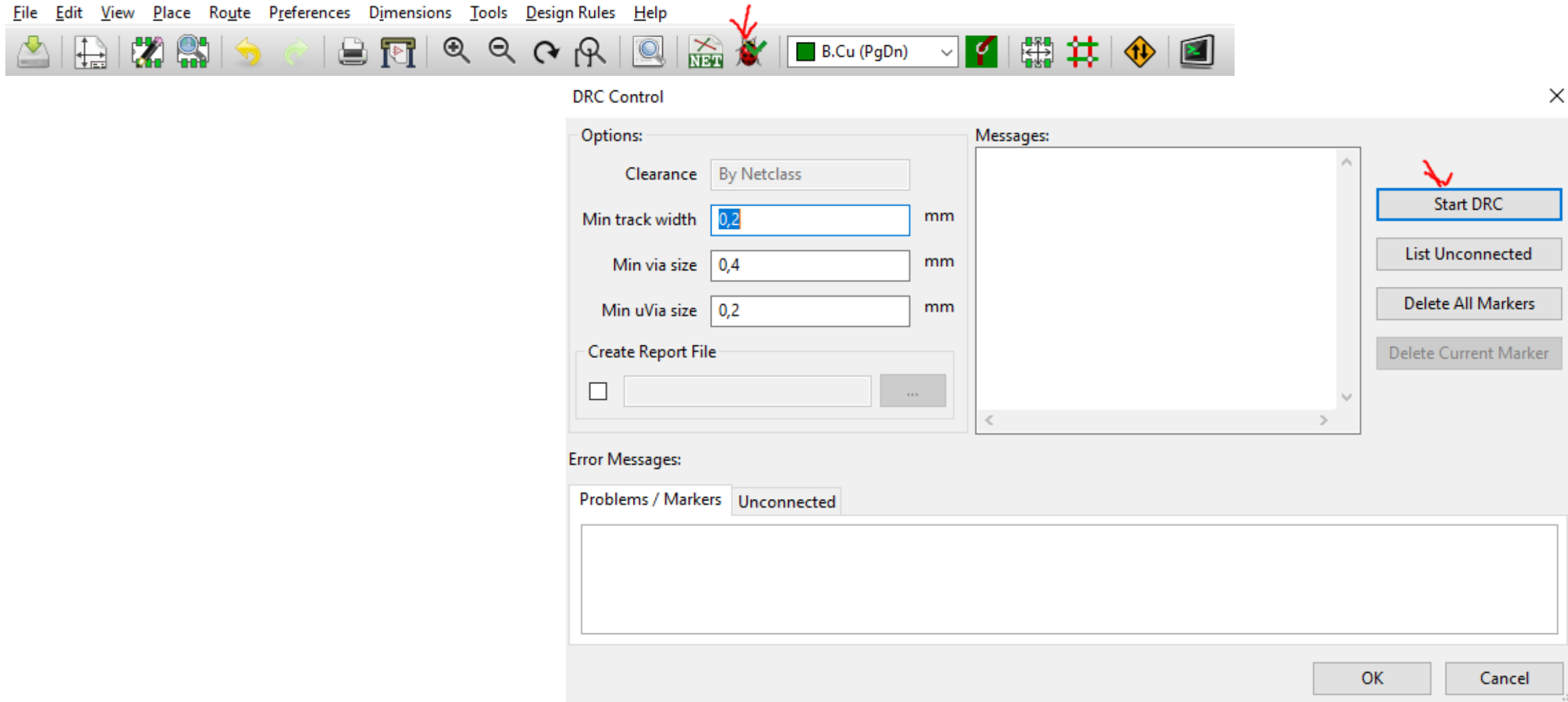
7. Drawing the wiring, adding fills

- To enable you to see clearer, you can choose which layers you want to see
- That's better ->
- You can also choose if you want to see fills on the left

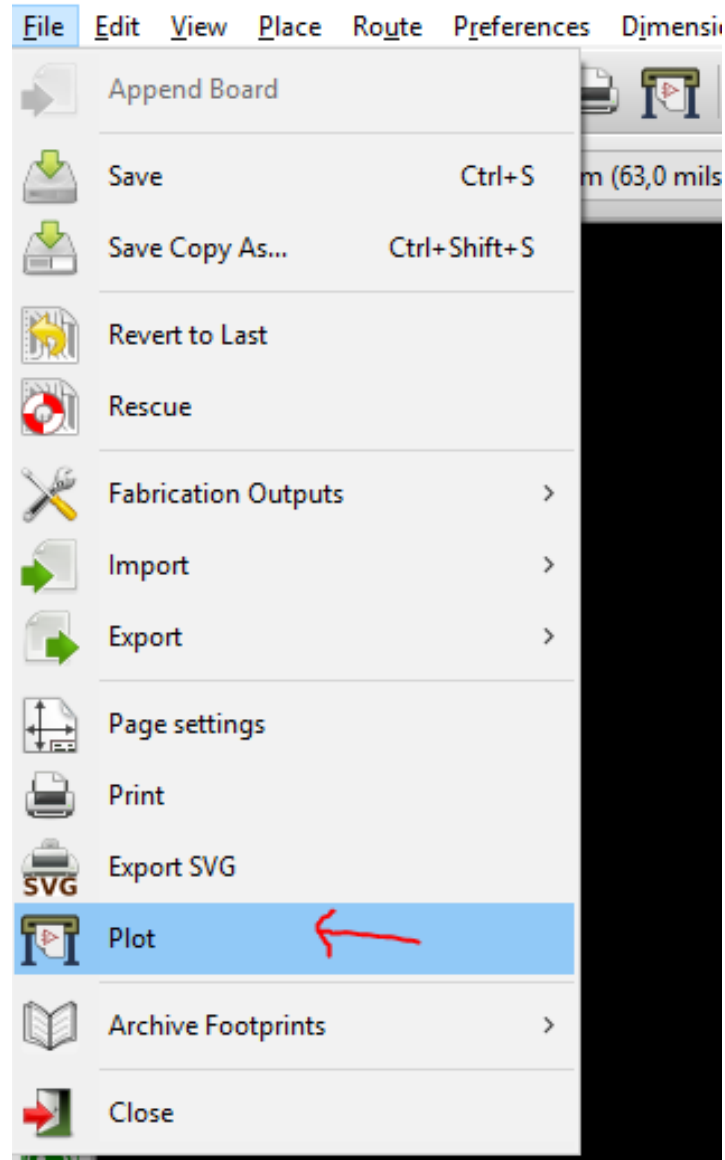


7. Drawing the wiring, checking errors

- Finally, let's see if we have made any errors

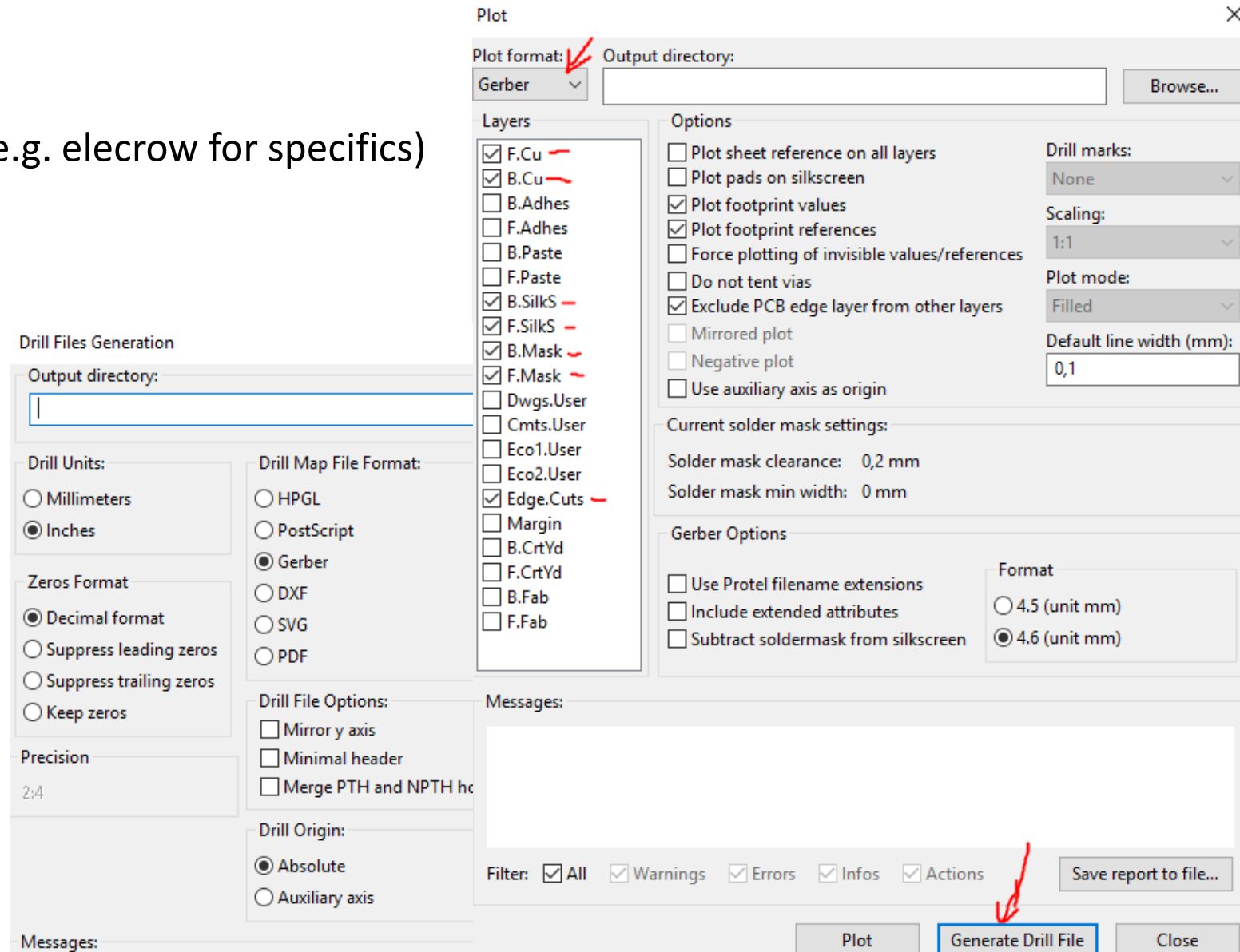


8. Create files needed to manufacture



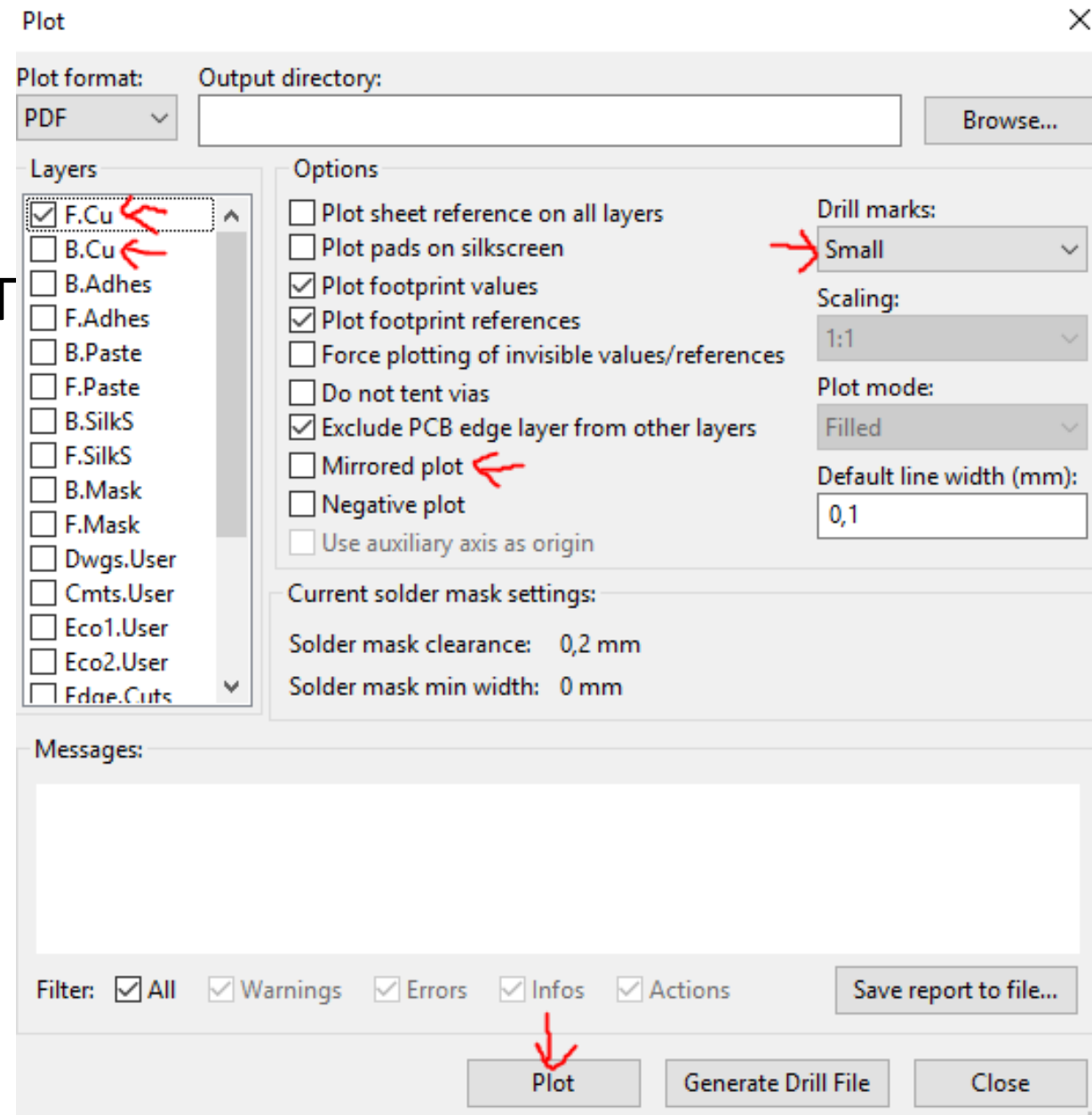
8. Create files needed to manufacture, order online

- Generate gerbers: (check website e.g. elecrow for specifics)
- Front and back copper layers
- Front and back silk screens
- Front and back masks
- Mechanical layer (Edge.Cuts)
- Also generate a drill file
- Some Places to order (non-exhaustive):
 - Elecrow
 - Seeed
 - Oshpark
 - Try googling and asking...
 - Great variance in price



8. Create files needed to manufacture, in the workshop

- Plot the copper layer pdfs
- REMEMBER TO MIRROR THE FRONT LAYER, BUT NOT THE BACK LAYER
- Press plot, use the printer in the workshop to print the layers on projector slides
- Use the slides as I'll explain now...
- If necessary, I'll explain again later and will help you make the first PCB in the workshop



The end

- Questions?
- Comments?
 - Other?