



PCB workshop

March 21, 2022

Bram den Ouden

1 Introduction

Hey there and welcome to the klushok workshop! This workshop is intended for absolute beginners and will guide you through making your very first PCB. The software we'll be using is:

- Kicad v6: <https://www.kicad.org/>
- Inkscape: <https://inkscape.org/>
- falstad circuit simulator: <https://www.falstad.com/circuit/circuitjs.html>

KiCad is a free and open source PCB design tool with similar functions to paid competitors like eagle and altium. It is actively maintained and there is a large community which helps new users or issues. Since Kicad is open source its users will always be able to open/edit older projects.

Inkscape is another open source firmware. It is used to design vector graphics and can be compared to adobe illustrator.

Falstad is a javascript tool which can accurately and visually simulate the operation of basic circuits.

We highly suggest you download and install Kicad and Inkscape whilst reading through this manual.



1.1 Design choices

During the design process of a PCB countless large and small changes will have to be made, these are left to the participant of the workshop. Some of the choices you should keep in mind whilst reading through this manual are:

- How many LEDs will you add to the design
- How will you controll these LEDs
- What is the power source and expected input voltage
- Will you use surface mount (SMD) or through hole (THT) components
- What is the maximum/minimum size of the PCBs

1.2 Workflow

Designing a PCB is not a linear progression. Each next step might result in having to make small adjustments in earlier steps. In general a step which simplifies a task in the next step is worth the iteration since the steps are built to be progressive in time consumption: A falstad simulation consumes time in the order of minutes to 1 hour due to simplifications like replacing all LEDs with a single current source or resistor. The next step would be creating the schematics which can take up to multiple hours depending on complexity of the circuit. The final step in the process is placing components and routing the traces. This step can take anywhere between hours and days depending on complexity and size of the PCB since each simplification made in earlier steps has to be routed manually whilst keeping all copper and components surrounding your module in mind.

The steps presented in this manual will follow a time increasing order:

1. First you will simulate your circuit to filter out any small mistakes and obtain values for your components. Simulations can also help to find critical points in your circuit. An example: which component becomes critical with a change in supply voltage?
2. A schematic must implement many components which could be left out in simulations. A certain load could be replaced with a resistor in simulations but will have to be replaces by those, for example, 20 leds in the schematics. There is still some simplification possible by using hierarchical sheets for repeated modules and copy-pasting components. This stage can often be sped up by using the recommended schematics provided by most datasheets.



3. The last step is creating the actual PCB. This step requires the user to layout all copper wires according to the connections described by the schematics. This step leaves very little room for copy-paste or repeated modules and is thus often the most time demanding step.

2 Schematics

Schematics are block diagrams of an electrical circuit. **Components** are represented by blocks with one or more **pins**. Each pin represents a metal legs of a component to which something can be soldered. The connection between multiple pins is called a **net**. Each pin connected to the same net is considered to be electrically connected.

Net labels can be used to connect two nets without actually drawing the wires. Using these in just the right amount is an art on itself but greatly improves readability of schematics. Besides connecting pins, net labels can also be used to name a net. This can be usefull when defining the copper wires on the PCB. Schematics also allow thicker wires called **busses** to be used. These contain many wires which don't have to be drawn individually and can be used to greatly improve readability.

2.1 Circuit

The circuit we will design will blink leds at a configurable speed and duty cycle. The basis of this circuit is an astable multivibrator. Astable indicates that the circuit does not have a stable position, multivibrator suggests it will oscillate multiple outputs. These suggestions are exactly what the circuit does: It has no stable position and will therefore keep oscillating. Figure 1 shows a basic configuration of such a circuit. Output 1 and 2 contain a square wave which can be used to drive single LEDs or LED drivers. Details of how this circuit works can be found on <https://www.electronics-tutorials.ws/waveforms/astable.html> and you're strongly advised to read this before continuing.

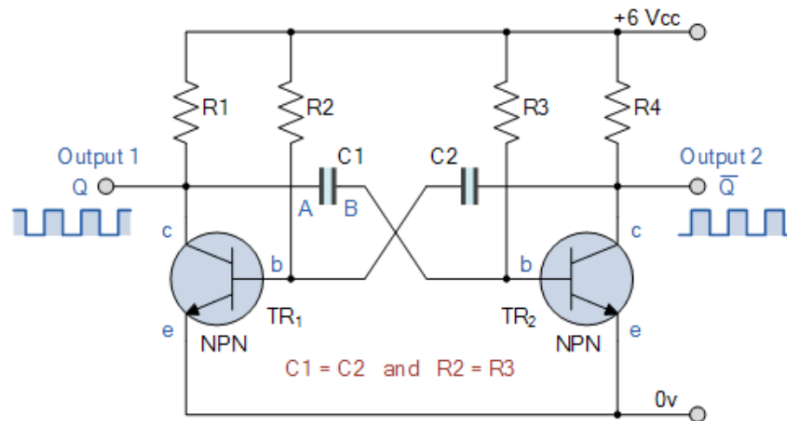


Figure 1: Astable multivibrator. Source: <https://www.electronics-tutorials.ws/waveforms/astable.html>

Besides the oscillator implemented by the astable multivibrator some LED drivers are required. These LED drivers are added to ensure that the behaviour of the LEDs does not influence the oscillator. Depending on the outcome of the earlier mentioned design choices there are different ways to implement LED drivers: A higher input voltage may allow driving multiple LEDs in series whilst lower input voltages (like when using a coin cell battery) will require driving the LEDs in parallel with individual resistors.

When deciding on the configuration of the LEDs, the switching element itself must also be considered: A BJT requires no extra components and since it operates on current, its behaviour is less dependent on supply voltage. A MOSfet can switch large currents but requires a gate-source voltage higher than its threshold value. A quick way to validate your design is to use the Falstad Circuit Simulator mentioned earlier. It is highly advised to have all critical parts simulated in falstad before moving on to Kicad.

2.2 Creating a Kicad project

To create a new KiCad project first open KiCad 6. A screen similar to figure 2 will appear. Go to the main screen and click on 'file'->'New Project' to create a project. An explorer window pops up asking for a location and a name. Keep in mind that when making a project, KiCad will create a folder with the same name as a project directory. Click 'save' and you will now see a file tree appear with your project name, a kicad_sch file and a kicad_pcb file. The sch file will contain your schematics and the pcb file will contain the PCB. On the right side of the window there are many more options which are not needed for this very first PCB we will create.

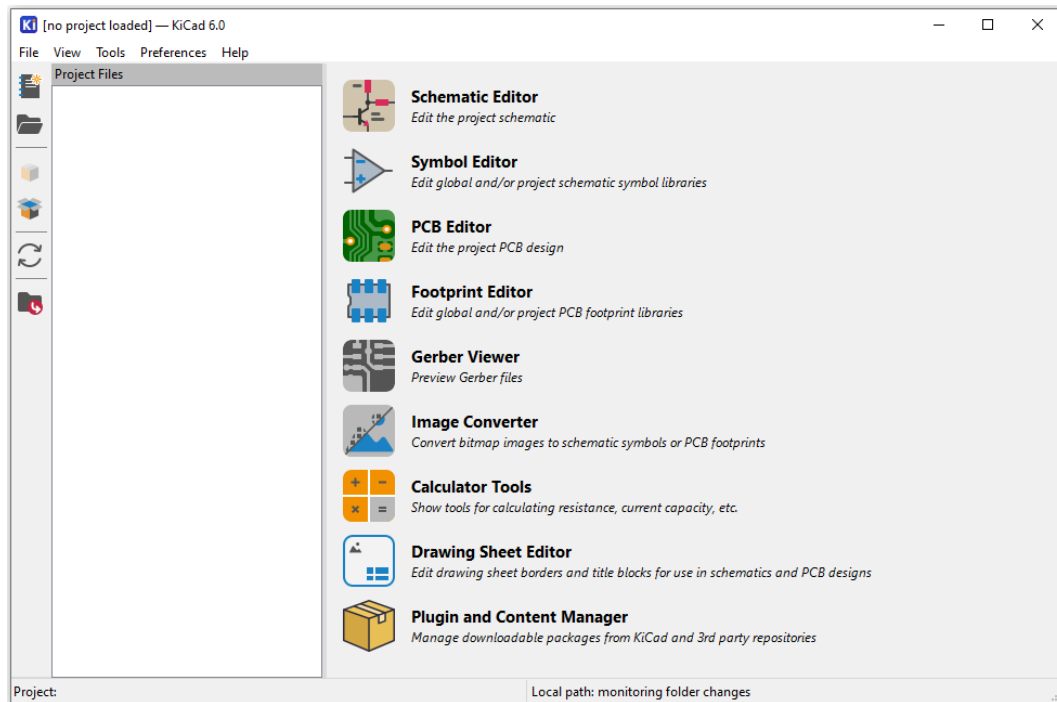


Figure 2

2.3 Creating schematics

Double click on the kicad_sch file in the file tree to open the schematic editor. An empty layout will appear in addition to a side- and top bar as shown in figure 3. Click on the op-amp icon in the right side bar to add components (shortcut 'a'), the first time you click this it might take some time to open.

Search for 'R' to add a generic resistor, click once on a result to see the schematic or double click a result to place it. The resistor will be stuck to your cursor until you click, which will place it. Repeat this process until all required components are placed.

Note: For the transistor you will need to select a specific model, check availability on <https://klushok.etv.tudelft.nl/inventory/components> or use the 2N3904 as a default NPN transistor.

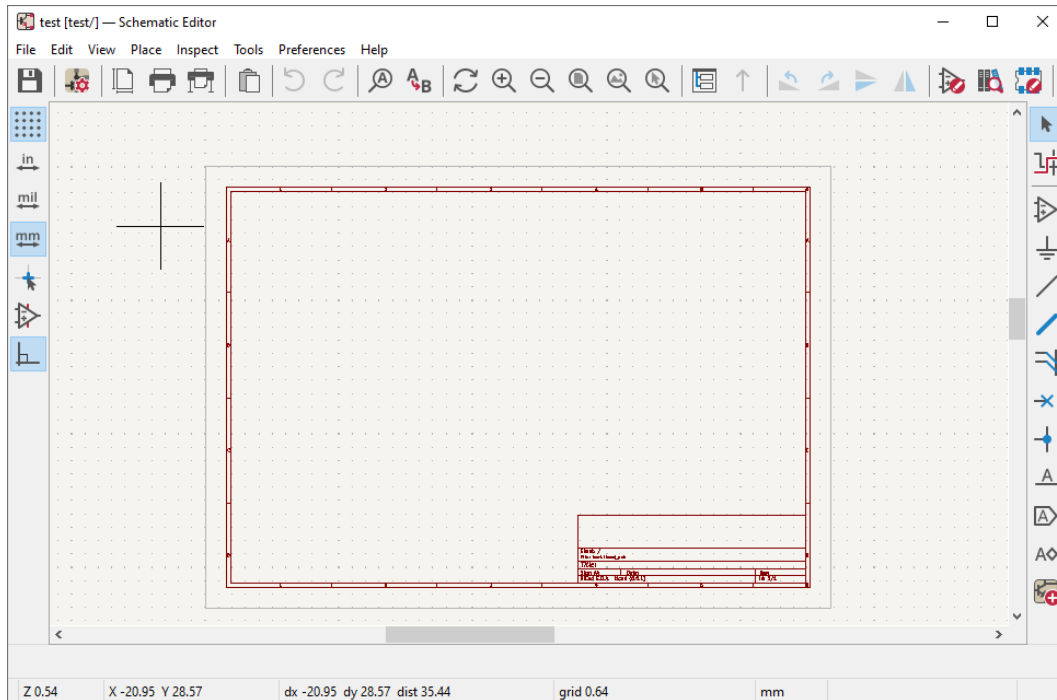


Figure 3: KiCad schematic editor

2.3.1 Basic commands

Left click a component to select it, click again to drag the component. The value and identifier of a component can be moved by clicking the identifier or value and dragging them. Right click and hold to move your view across the schematic. Scroll to zoom in or out and click left and hold to select multiple components. By selecting a component and rightclicking them you can open an action menu. Go to properties in this action menu to adjust multiple parameters such as the associated footprint, value or datasheet link. All options have their keyboard shortcut listed next to them. Adding wires can be done by clicking on the thin stripe in the sidebar of the schematic editor, pressing 'w' on the keyboard or by clicking the pins which have circle around them. Use these wires to connect all placed components. Take your time to make sure the components are positioned in a logical order, use net labels where needed.

2.3.2 Validation

Once you've made the schematic, verify that it matches the simulated circuit and carefully double check no incorrect connections were made by for example crossing wires. Ask a committee member to double check if you're in doubt.



3 PCB

3.1 Keyboard shortcuts

list of shortcuts or link to where these are listed

3.2 Footprints

not all devices have default Footprints ensure the footprint matches the version of your symbol (2n3904 THT has a different pinout from the smd version for example). select component, press e to open properties, click book icon in the footprint field to open footprint editor, select correct footprint. Do this for 1 of each type to make the next step easier.

now click the 'run footprint assignment tool' button in the top bar of the schematic editor. If you have not already done so, a popup will ask you to annotate the schematics. Annotation means replacing the questionmarks in the device identifiers by numbers.

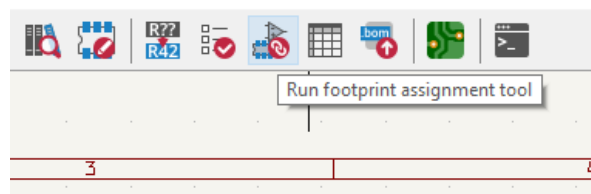


Figure 4

3.3 PCB edge

Outline from svg:

1. Open PCB editor
2. File->import->graphics
3. Select 'Edge.cuts' for option 'graphic layer' as shown in figure 5

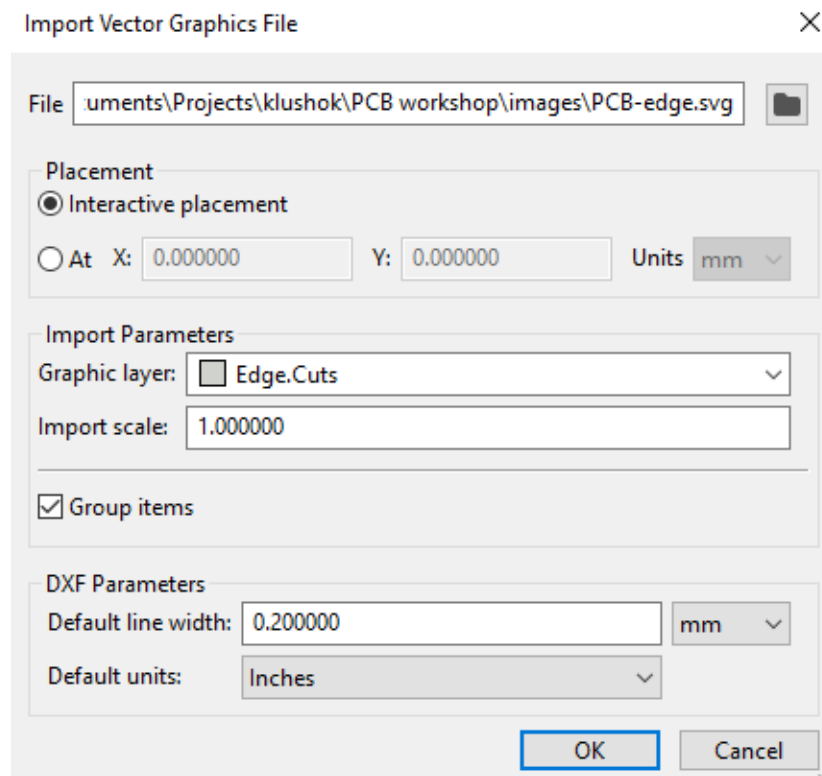


Figure 5

4. Select the vector file containing the desired outline
5. Drag the outline to the desired location and click to place
6. Open the 3D viewer (alt + 3) to verify that the outline has been placed correctly

3.4 Placing components

Import the design made in the schematic editor by clicking the "update pcb" button shown in figure 6. This will import all components with their assigned footprints to the PCB editor. Place them somewhere next to your previously defined board edge. The components are connected using thin wires which are called the ratsnest. These lines indicate the shortest path between two pins of a single node and must eventually be replaced by copper traces.

Drag individual components to the location you'd like on the PCB. Start with large or very important components, end with smaller stuff like resistors. Through hole components have a copper pad on both sides of the PCB which is something you should keep in mind when placing them next to surface mount components.

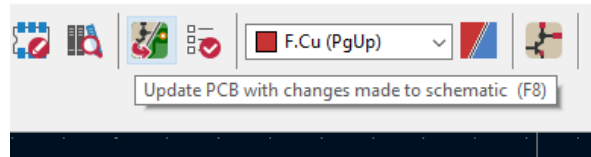


Figure 6

Although components can be placed on both sides of the PCB, soldering becomes a lot easier if you keep them to a single side.

Once the components are generally in the location you'd like, check if the ratsnest shows feasible routes, try to rotate components to have a minimally tangled ratsnest.

4 Conclusion

This workshop has taught you how to create a PCB but the most important part of making the actual PCB is experience. Each step can take anywhere from minutes to hours depending on the complexity of the circuit and how well the designer can separate the complete design in smaller blocks with nicely placed ports.

We hope you had a great first experience with designing PCBs and many more will follow. Feel free to ask members of the klushok committee for advice at any time, we all like to help!

Thank you for participating and we hope to see you on future workshops!

The Klushok committee