

---

## SMD PCB Workshop

### Rolling Dice Game

---

Learn how to create Printed Circuit Boards (PCBs) the right way using KiCad software. In this workshop you will create a PCB for a rolling dice game, using Surface Mount Parts as required in higher level UQ Subjects.

*Prepared by:*  
Tim Hadwen

# 1 Introduction

Printed circuit boards (PCBs) are the backbone for nearly all modern electronics. Simple PCBs are often used to help prototype a project, with more advanced PCBs used for iPhones and Televisions often requiring many more layers (in some cases over 100 layers).

## 1.1 PCB Concept Refresher

### 1.1.1 Layers

Most PCBs designed by students at University and in projects are 2 layers, top and bottom. However in some cases as size restrictions exist, more layers are required with some designs moving to 4 and 6 layers. Often if custom RF (Radio Frequency) circuitry is required, 4 layers will be required as layers on a 4 layer board are more accurate and thinner allowing the required  $50\Omega$  traces.

### 1.1.2 Traces

Traces are the wires that link between components on a PCB. These traces can be any width you require with standard widths being 0.2-0.3mm however high current power traces can be as thick as 1mm depending on current requirements. These traces should never have right angle bends, as this increases the resistance in the trace.

### 1.1.3 Planes

Planes make up large sections of copper over the PCB surface. This surface acts as an extremely low resistance trace allowing for better ground signals and in some cases high current power capacity. In 4 and 6 layer boards it is typical to have both a ground and power plane, however in 2 layer boards (like in this workshop) typically a ground plane is placed on the bottom layer. Traces should be avoided on power and ground planes to stop the breaking up of planes, although in most cases some traces will be required on the bottom layer of a 2 layer board.

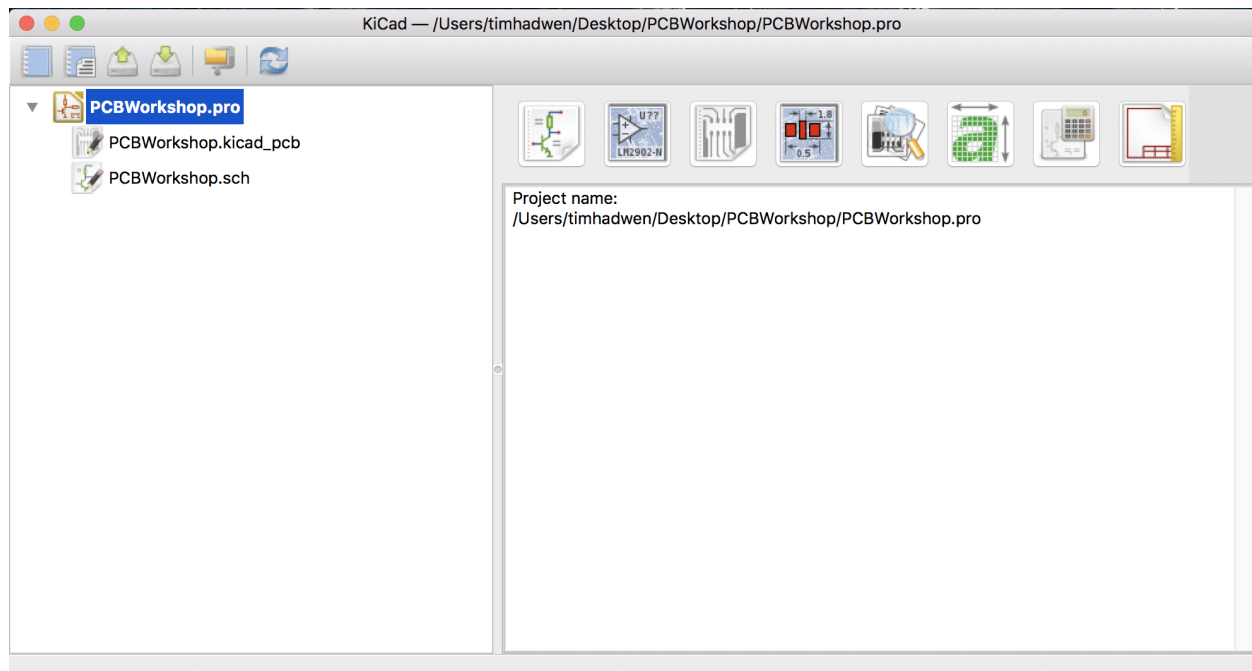
### 1.1.4 Footprints

Footprints are sections of the PCB that holds components, for example a footprint for a surface mount resistor is shown below, with 2 copper sections and a white line surrounding it. Often markings are added to denote what part number and what value is required in this position.

## 2 Workshop

### 2.1 Project Creation

In order to start designing a PCB in KiCad you will first need a project! To create a project click **File** → **New Project** → **New Project** and choose a name for the project.



Once the project is created we can see the project with both **kicad\_pcb** and **sch** files, these are the **Circuit Board** and **Schematic** respectively. These are all the files we need to create a PCB.

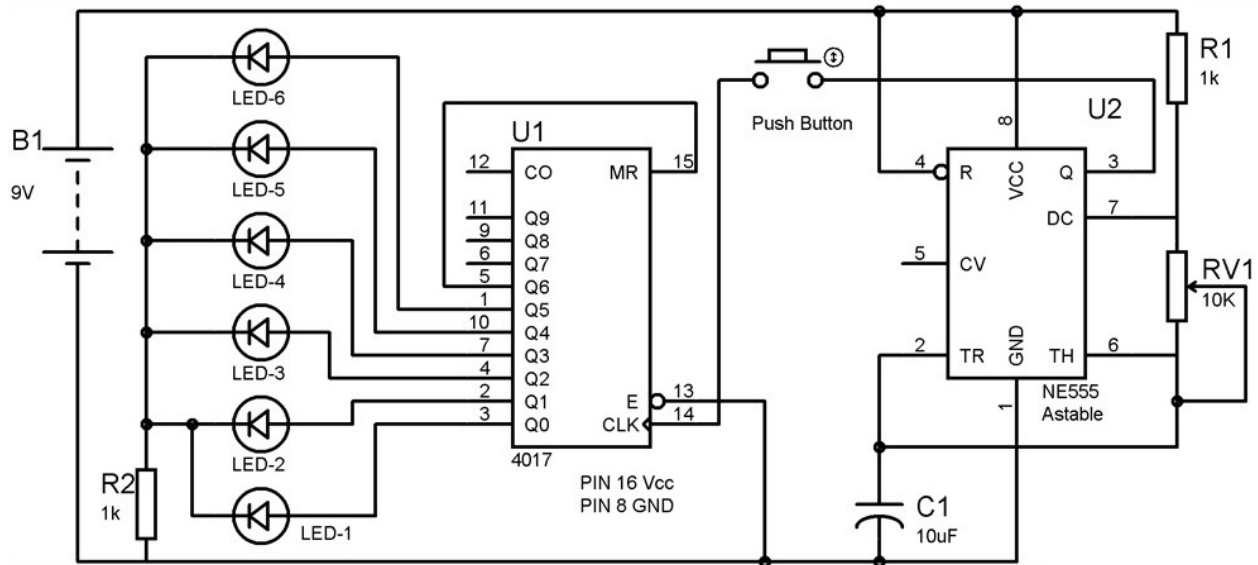
On this window we can also see a series of buttons on the right. These open up the programs associated with PCB Creation. The important ones (the first 4, used in this workshop) are listed below.

- EESchema - Schematic Creation
- Schematic Library Creator - Creates schematic symbols for use in EESchema
- PCBNew - Layout and routing of the Circuit Board once schematic is completed
- PCB footprint editor - Creates the footprints for placement on the PCB

### 2.2 Schematic Creation

In order to create a KiCad Schematic we must first have some schematic designed. In this case we have provided you with the one below! The circuit we are building is a Electronic Dice, that on a button press will generate a random number between 1 and 6 and show that number on a series of LEDs. It does this by using a 555 timer and a CMOS Decade Counter Chip.

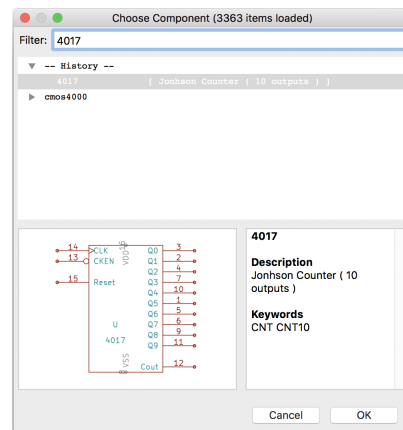
While the button is pressed the lights will cycle until the button is released, showing a the result. The speed can be adjusted by changing the resistance of RV1, a 10k $\Omega$  potentiometer.



First, you'll need to know a few keyboard shortcuts. They are listed below. Your mouse must be hovered over the part you want to Rotate, Delete or Move and also the position you want to start or end a wire.

- A - Add a schematic part
- Backspace - Delete a schematic part
- R - Rotate a part
- M - Move a part
- W - Start a wire
- K - End a wire

Now that you know that, we will add the CMOS Chip and 555 Timer. To do this press **A** on the keyboard. You can type in **4017** to get the CMOS chip to show up as a Johnson Counter as shown below.

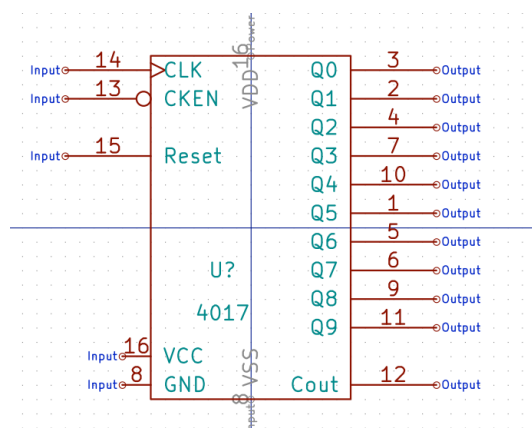


However, sadly this footprint isn't quite what we want as it has no VCC or Ground connected to it! So we should probably fix that!

### 2.2.1 Modifying a schematic part

Go back to the home page, and click on the Schematic Library button. You can then click **File → Select Current Library** and select one called **CMOS4000**, which is where our 4017 Decade Counter chip is from. Now we need to select which part we want to edit. To do this press the **Load component to edit** button (Op amp with down arrow) and select the 4017 part.

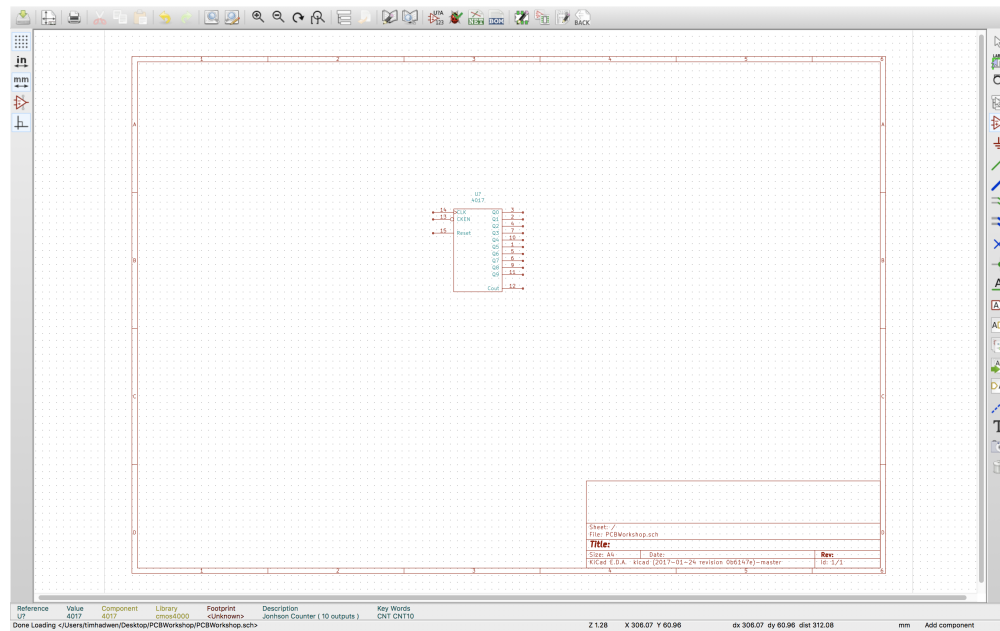
Now all we need to do is add 2 pins, one for Ground (Pin 8) and one for VCC (Pin 16). So click the pin button on the right side and place 2 pins. Name them GND and VCC and give them the right pin numbers. It should end up looking like this!



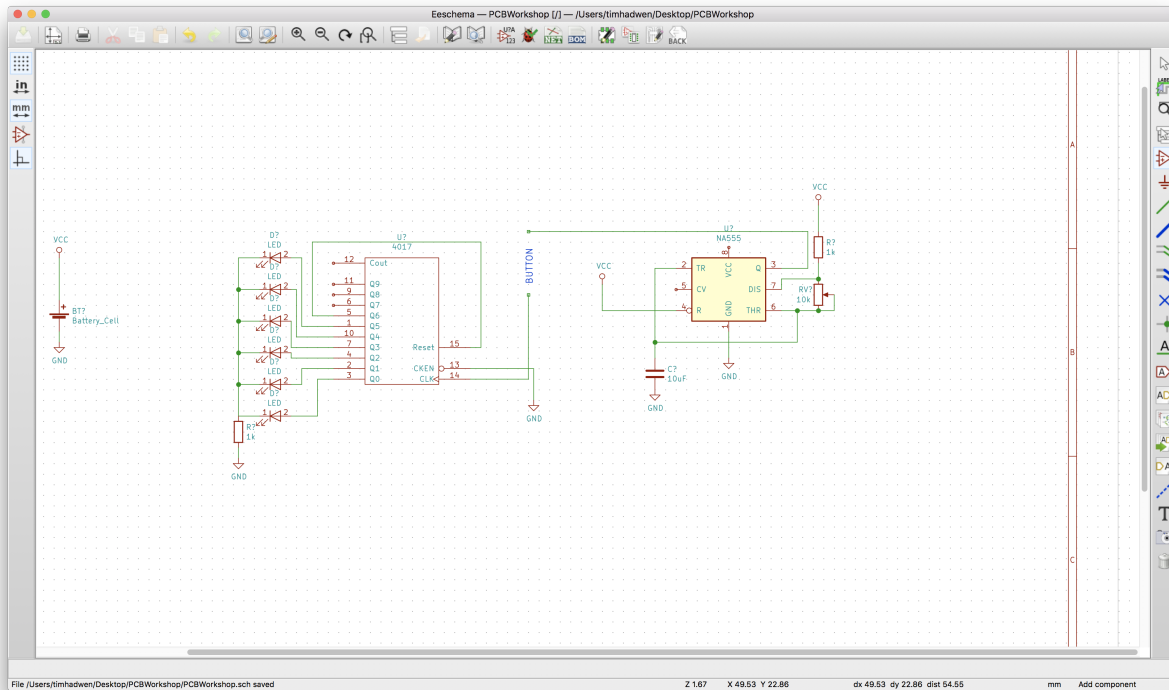
You can now click save and close this window!

### 2.2.2 Continuing on with schematic creation

Open up the schematic window again repeat the steps from before (Click A, Search for 4017) then click ok and add the part to the schematic by clicking anywhere on the sheet, probably somewhere in the centre is good for this part since we will need to connect things to both sides. Remember we can always move it later!



You can now go ahead and add the rest of the circuit, except the button. We will need to create a custom footprint for that in the next section. You should end up with something that looks like this.



## 2.3 Part Library Creation

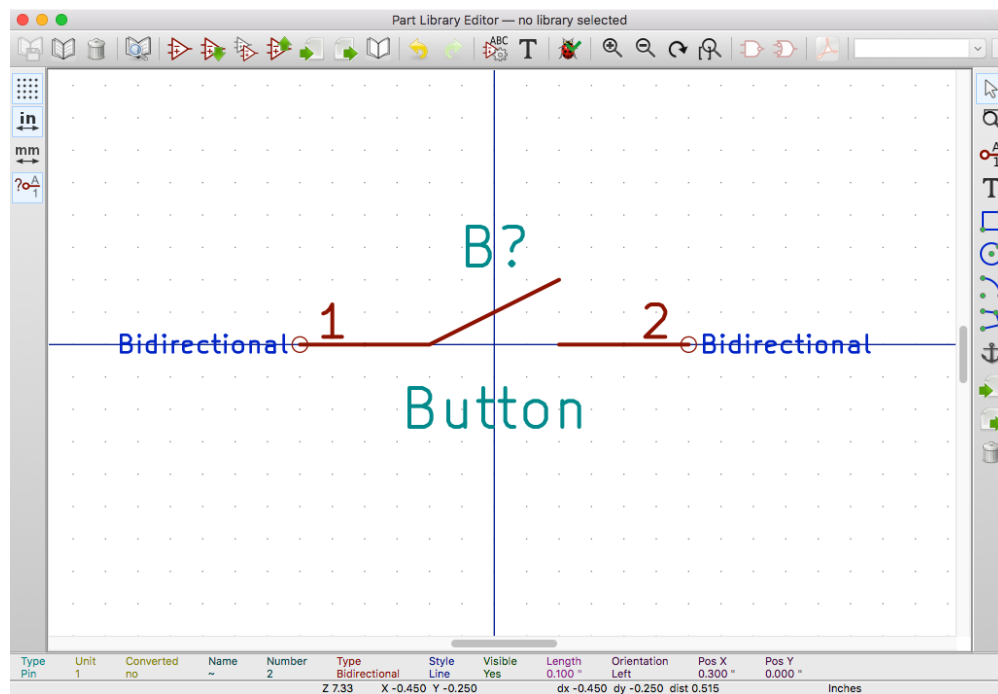
Since there isn't already a button part we will have to create one! That's done pretty easily in KiCad and is a very useful skill to have especially in more complex designs or when using obscure parts.

To do this we click the **Schematic Library Creation** button back on the home screen. Then click the new part button at the top and give the part a name. In this case we will just call it **Button** and give it a designator **B?**. The rest can stay as it is.

We are then presented with a screen with **Button** and **B?** in the centre. We move these out of the way and draw our schematic symbol. This will show up on the schematic we showed earlier.

You can create pins using the pin button on the right side, and lines using the polygon line button below that.

Once you're done you should end up with something like this.

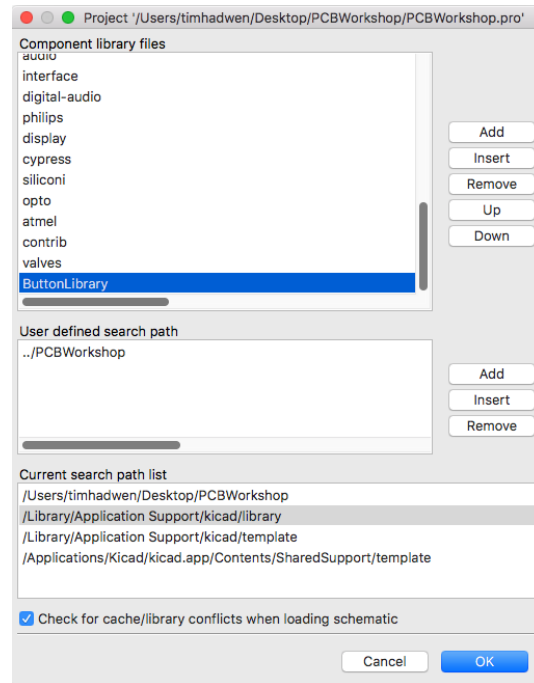


In order to save this to a library, we need to create a new one. Luckily KiCad can handle this for us using the **Save component into new library** (the book icon) along the top bar. Call it something like **ButtonLibrary** and click OK.



## 2.4 Using custom parts in EESchema

In order to use this newly created part we need to add it to EESchema's libraries. To do this, click **Preferences** → **Component Libraries**. Then click add and navigate to where you saved the library. It should then show up in the list as below.

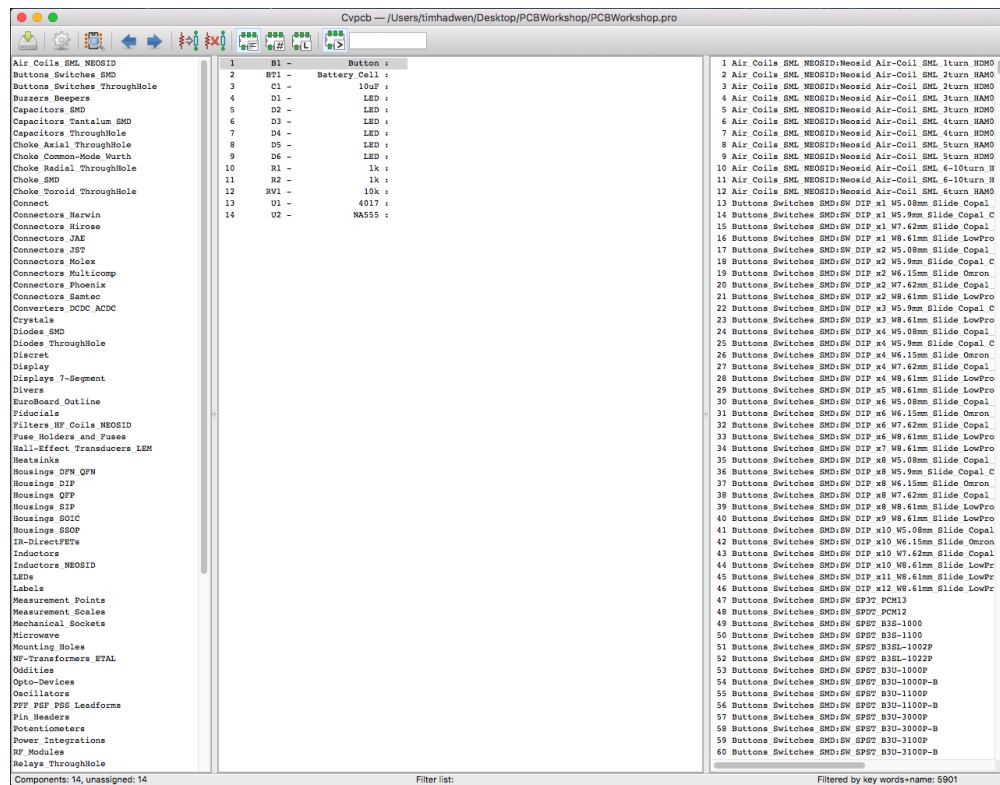


You can then add it to the schematic just as you did earlier (Press A, find ButtonLibrary, and place it where it needs to go). Remember to connect any wires to that part you need!

## 2.5 Preparing for PCB Design

That great! We now have a completed schematic, however we now need to do a couple of things to get ready to lay out the PCB itself. First we need to number all the parts, as they are currently things like **R?**, **C?**, **U?** and **B?** which isn't that useful. In order to do that click the **Annotate Schematic** button on the top bar (Picture of an Op Amp). Click ok a few times and now every part should be numbered!

Now we need to associate each part with a **Footprint** or the physical layout of that part! To do that click the Associate button (3rd from the right). You should be presented with something like this.



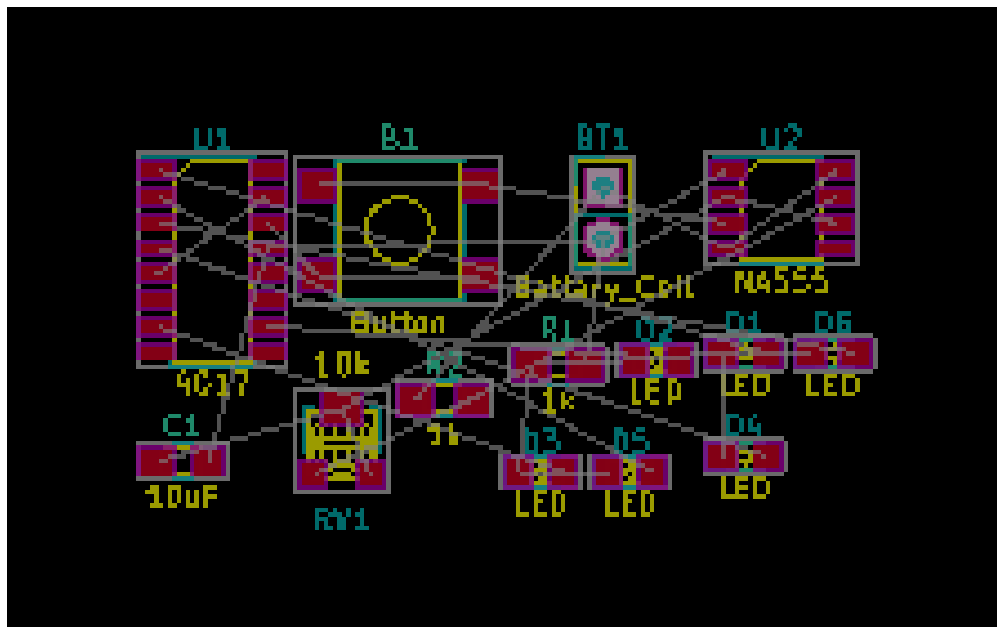
Here we have all the libraries on the left, our parts in the middle and what KiCad thinks is a good match on the right. Luckily its pretty good! For this PCB we are going to use all 0805 size SMD parts, so that we can solder it by hand if we want to later! Lets leave the button until last because its a bit more complex! So click on C1 and we are presented with a big list on the right. We want **C\_0805\_HandSoldering** so double click on that. For the LEDs we want **LED\_0805**, Resistors we will go with **R\_0805\_HandSoldering**. For the Potentiometer, KiCad isn't being so smart so we will use the **Filter Box** instead. At the top type in Potentiometer and then click the button to the left to enable that search. We are looking for **EVM3E** so you can type that in as well to make life easier! You can also click the third button from the left to see what the footprint looks like, which is helpful for working out which one you need. For the 4017 Logic Chip we want a SOIC16 package, for the 555 timer we want a SOIC8 package, for the Button we can use **SW\_SPST\_B3S1000**

(you may need to search for it) and finally for the battery we are just going to add a 2 pin header **Pin\_Header\_Straight\_1x02\_Pitch2.54mm**. In the end you should end up with this:

1	B1 -	Button : Buttons Switches SMD:SW SPST B3S-1000
2	BT1 -	Battery_Cell : Pin Headers:Pin Header Straight 1x02 Pitch2.54mm
3	C1 -	10uF : Capacitors_SMD:C_0805_HandSoldering
4	D1 -	LED : LEDs:LED_0805
5	D2 -	LED : LEDs:LED_0805
6	D3 -	LED : LEDs:LED_0805
7	D4 -	LED : LEDs:LED_0805
8	D5 -	LED : LEDs:LED_0805
9	D6 -	LED : LEDs:LED_0805
10	R1 -	1k : Resistors_SMD:R_0805_HandSoldering
11	R2 -	1k : Resistors_SMD:R_0805_HandSoldering
12	RV1 -	10k : Potentiometers:Potentiometer Trimmer-EVM3E
13	U1 -	4017 : Housings_SOIC:SOIC-16_3.9x9.9mm_Pitch1.27mm
14	U2 -	NA555 : Housings_SOIC:SOIC-8_3.9x4.9mm_Pitch1.27mm

Click save and then back to the schematic! Now we need to export a list of wires, parts and connections for the PCB Software to read! Click the Netlist button along the top, click Generate and then save it with everything else! Now close the schematic and open PCBNew by either double clicking on the **kiCad\_pcb** file or clicking the PCBNew button (3rd from the left).

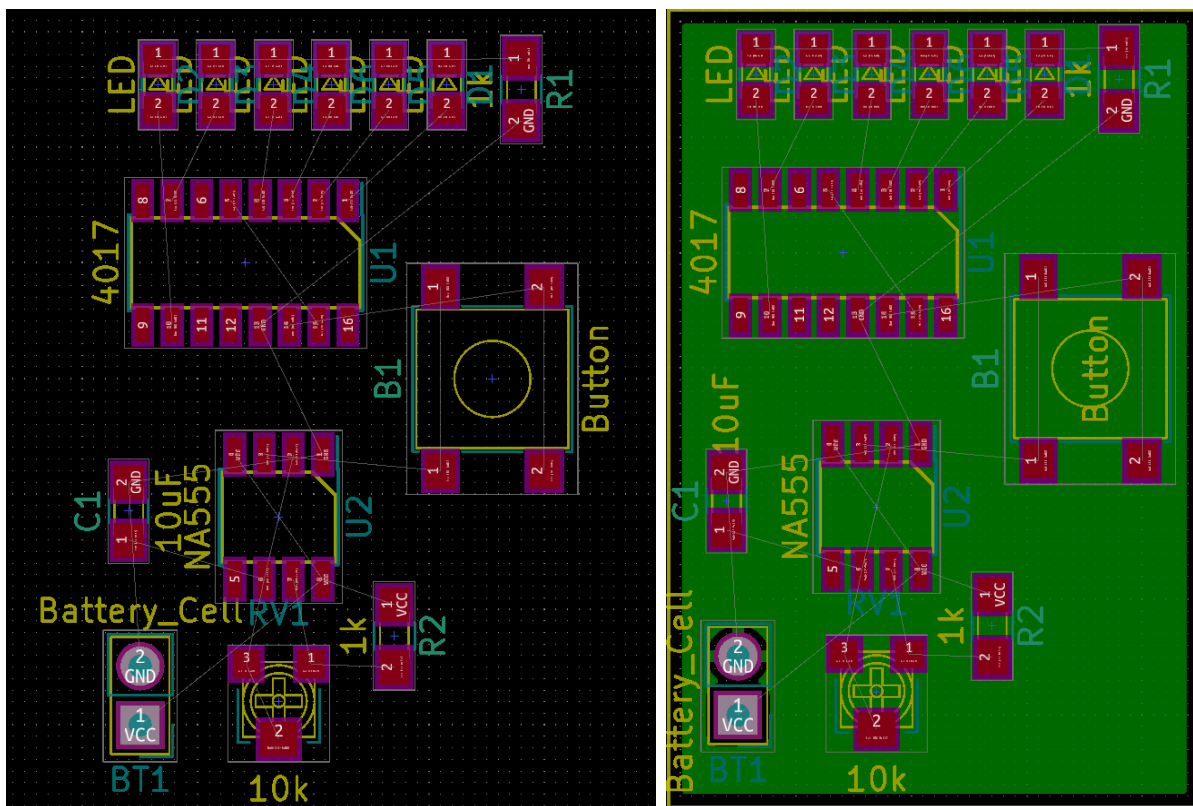
For now this will be blank, so lets import that netlist! Again click net and navigate to your file, as this is the first time we are importing we can just click **Read current netlist** and then close the window. We now have all the components on the PCBNew canvas. It should look something like this.



For now lets place them out in a way that sorta makes sense, so the 6 LEDs should be in a line somewhere, the power connector near the edge and everything else somewhere in the middle. It helps to think about how things wire up too, the small yellow lines mean these

pads need to be connected so take care to have these close to each other. Of course there isn't a right way to do this but the way we came up with is below:

Now we need to setup an outline and a ground plane. For the outline click on the edge cuts layer on the right side and then on the dotted line. Draw an outline of the board making sure all the components are within it! This should probably be square but that doesn't matter too much. Then to create a ground plane click the B.cu layer and then the plane button (its the small dot within a green square button on the right side) and again follow that outline with it. A window will pop up and you want to make sure you've selected GND as the net name, this connects ground to this plane. Once you've done that there should be a large green area under your parts! That's the ground plane!



Now we need to go about connecting everything together. We can use the route tool which will help us follow the yellow net lines and replace them with wires. Even for simple boards we are going to need to put some things on the bottom layer and to do this press **V** while routing to switch layer. To connect something to the ground plane click **V** and then **Escape**. Once you do anything with the ground plane or bottom layer make sure you press **Shift + B** to update the bottom layer. Once you're done you should end up with something like this:

