ELEN90053 Electronic System Design 2020 – Workshop 2 - Altium Designer Tutorial

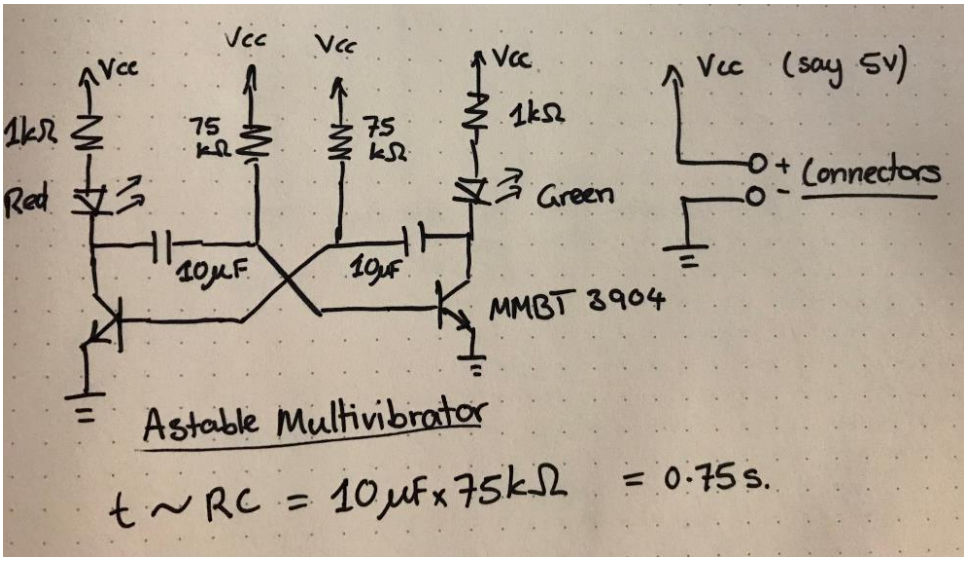
This isn't just a workshop about 'how do I use Altium Designer' - it's also an introduction to important general ideas around how we design an electronic system fabricated with real-world components on a real printed circuit board.

In this workshop we will consider the process of taking an electronic system from concept through to selection of components, schematic capture (drawing the schematic in software) and the layout of a printed circuit board (PCB) using CAD/CAM software, design rule checks and the generation of industry-standard photoplotter files which can be sent to a PCB foundry for fabrication.

The first step in designing an electronic device is to imagine what your finished device will look like and define what it will do. It's important to design your PCB layout with consideration of the industrial design of the plastic case that your PCB will need to be mounted inside, for example.

What kinds of inputs and outputs will it have? What major electronic subsystems will be needed, and what main kinds of components? What kinds of connectors will be used to interface to the outside world? Will user interface and UX issues need to be considered, with devices such as knobs or displays?

For purposes of this tutorial, here's a circuit I'm going to use.



This is an astable multivibrator, where the cross-coupled feedback between two transistors and two RC networks provides a constant square-wave oscillation determined by the two RC time-constants, making the two LEDs flash alternately.

I've sketched this circuit in my notebook to emphasize that many important steps early in your project have nothing to do with CAD/CAM software. Whiteboards, notepads, websites and datasheets are all valuable tools you will use to plan your design.

Computer aided design is exactly that – it is a tool that can *aid* your work, but ultimately you are the engineer designing the system. We will look at Altium Designer soon, but we don't need to worry about it straight away.

Picking real-world components for your circuit can seem daunting at first. Suppose you need an op-amp, for example, you might go to Digi-Key and search. You get over 40,000 op-amps.

In the Altium reference documentation, you might read about the **Manufacturer Part Search** in Altium Designer – but we won't be using that in this course so don't worry about it.

You will be picking your parts 'manually' from vendors, or from part choices specified by demonstrators. You will be using local Altium library files that you have drawn up yourself or those that may be supplied by demonstrators.

When picking real-world electronic components for your design, here are some basic principles.

Select components that have appropriate *electronic parameters* and performance for your circuit.

Select components that *you can buy* – check that they are stocked at a supplier that you can purchase them from, in the quantity required. Read the *datasheet*! Check the physical package of the part, check the specific part number variant you are ordering.

Check that you have appropriate CAD/CAM *libraries* that match the *pinout and package* of the parts you are ordering or *create* those libraries.

Pick the cheapest parts, subject to the above. After these other factors are considered, you're still going to have hundreds of op-amps to choose from. Just pick the cheapest one available that is appropriate for your engineering. You might use an expensive ultra-bandwidth Analog Devices opamp in a circuit that requires it, but you shouldn't use expensive parts for every circuit that doesn't require them.

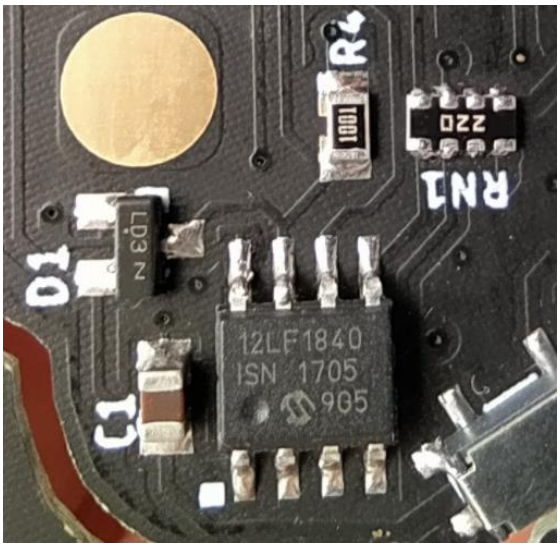
Look up parts for purchasing (or just to see what is available) using Digi-Key as your first distributor choice. They are simply the fastest, best distributor with the biggest range of stock. Their website also provides convenient searching and filtering by appropriate parameters, and a direct link from every product page to the datasheet of that component.

Surface-Mount Devices

In this subject, we are going to focus on PCB layout and assembly using surface-mount devices (SMD) for most components in our projects, where components are soldered onto surface pads on one side of the PCB without passing wire leads through holes in the board.

Surface-mount technology is ubiquitous in modern electronics manufacturing, and allows substantial miniaturization, higher density of components, and has benefits for automated assembly of components on PCBs.

This photo shows a few examples of SMD components in very common standard packages, including an 8-pin SOIC (small-outline integrated circuit), a diode in SOT-23 (small outline transistor) package, and a resistor and capacitor in '0805' package.



These are all examples of standard package shapes standardized by electronics standards organizations such as JEDEC.

(The *package* is the shape of the component, which corresponds to a particular *footprint* of space on the PCB layout and exposed metal pads arranged in a particular position in our CAD/CAM software.)

An '0805' package denotes a small rectangular device, usually with two pads, with dimensions of 80 thou x 50 thou.

A package like this is commonly used for ceramic capacitors, resistors and LEDs.

You can get smaller packages for more miniaturized designs, such as '0603' or '0402', but we will use 0805 package for such devices in this course, as they are a bit easier to handle and solder by hand with a soldering iron. The fact that most components use common, standard package shapes makes building your CAD package libraries much easier.

A 'thou' is one thousandth of an inch, and this unit is common in PCB layout engineering, even in countries where the metric system is established. For example, the spacing between pins on a standard DIP IC or breadboard is 100 thou, and the space between pins on a SOIC device is 50 thou.

In PCB layout, it is quite common to use a grid set to 100 thou, or 25 thou or so. (Some literature and software will use units of 'mil', including Altium. This is exactly the same thing as a thou, but I advise against saying mil as it can create confusion due to the casual use of spoken 'mill' to mean millimeters among engineers.)

Layers of a PCB

A typical printed circuit board consists of a number of different structures layered together.

Substrate: The substrate of the PCB, which mechanically holds everything together, is a dielectric material, commonly fibreglass-reinforced epoxy resin. In PCBs which have >2 copper layers, there are multiple dielectric layers, stacked up between the copper layers.

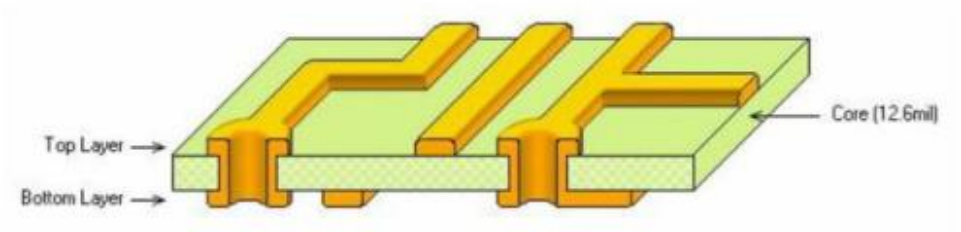
(Although fibreglass-epoxy is by far the most common, some special applications use specialised PCB dielectrics such as Teflon, providing low loss in microwave applications, for example.)

Copper layers: Every PCB contains one or more layers of conductive metal, patterned with photolithography to define the connections in the electronic system.

Two layers of copper (one top and one bottom) is very common, although some older devices commonly use single-layer PCBs, and more complex modern devices often use four or more layers – multiple layers "buried" inside the PCB between a stack of multiple dielectric layers.

Two-layer PCBs are most common, with a top and bottom copper layer. Although it is possible to manufacture PCBs with multiple internal layers of copper conductors, like 4-layer or 8-layer boards, we will only worry about two-layer

boards in this course.



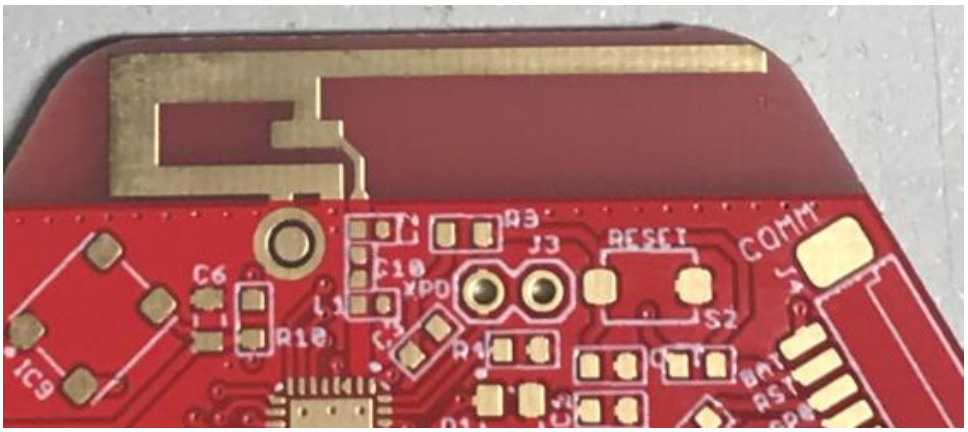
Solder mask:
The soldermask is a paint-like coating applied to the outer surfaces (top and bottom) of the PCB.

The soldermask coating on the PCB is patterned with photolithography, creating a hole or opening that corresponds to each pad where a device is soldered onto the PCB, leaving a metal pad exposed for solder to flow onto.

Soldermask has several functions – it gives the PCB a nice aesthetic color, it provides a substrate onto which the silkscreen markings are printed, and it provides a basic level of insulation and protection to the copper conductors.

There are two soldermask layers defined, when you send your PCB design files for fabrication.

Soldermask also keeps the solder contained during soldering, allowing the copper to flow only onto the pads where solder is intended to be, and not onto the other copper traces.



On this PCB design, I have excluded all the soldermask from the area surrounding this RF antenna, as well as excluding the copper ground pour from the area. (This does not need to be soldered, but it is done because the soldermask layer shifts the intended permittivity of the metal-air structure.) You can see the exposed FR-4 (fibreglass composite) substrate, which has almost no color. The red color of the soldermask on the bottom side is slightly visible, since the fibreglass is slightly transparent.

Silkscreen:

The silkscreen is the printed layer (usually white) on the surface of your PCB which contains documentation and markings, such as the numerical designations of components, markings of component positions, functions or pinouts of connectors, and other documentation for users to read on the PCB during assembly of parts on the PCB, and system integration. PCB version numbers, manufacturers or logos may also be printed on the silkscreen.

As with the soldermask, a typical PCB design has two silkscreen files designed for fabrication – one for the top and one for the bottom.

It is important to note that the silkscreen ink can only be patterned on top of the soldermask – it cannot be printed onto exposed metal. This means that your component markings, designators or other information printed on the silkscreen must be arranged so that it does not overlap onto exposed component pads.

Choosing Parts

Here, I have picked a suitable bill of materials for our astable LED oscillator project. You won't be manufacturing a real-world PCB for this example circuit, and the main ICs for your Digital Storage Oscilloscope will be specified for you, but these are the kinds of things you need to plan when building a real-world project.

In a real BOM, you would often include part designators (such as R1, C2) that correspond to the schematic, but I will omit these here.

Qty	Manuf. Part No. (MPN)	Description	Package	Manufacturer	Digi-Key SKU
2	MMBT3904	NPN BJT, Ic = 200mA	SOT-23	Micro Commercial	MMBT3904TPMSCT-ND
2	C0805C106K4PACTU	10uF ceramic, 16V, X5R	0805	Kemet	399-8012-1-ND
2	RMCF0805FT1K00	1kΩ resistor 1/8W, 1%	0805	Stackpole	RMCF0805FT1K00CT-ND
2	RMCF0805FT75K0	75kΩ resistor 1/8W 1%	0805	Stackpole	RMCF0805FT75K0CT-ND
1	APT2012ZGCK	Green LED, 350 mCd	0805	Kingbright	754-1795-1-ND
1	APT2012SECK/J3-PRV	Red LED, 350 mCd	0805	Kingbright	754-1791-1-ND
1	5000	Test point, red	1mm Hole	Keystone	36-5000-ND
1	5001	Test point, black	1mm Hole	Keystone	36-5001-ND

MMBT3904 transistors are used, which are cheap NPN BJTs in a SOT-23 package. These are pretty generic, and many different kinds of BJTs would work fine in this circuit. Red and green LEDs are chosen in 0805 packages, with 350 mCd brightness which is reasonably bright but not too expensive. (It can be tricky to identify polarity on SMD LEDs – the component footprint on the PCB needs to be marked with the orientation, and the cathode aligned with a tiny marking on the LED body.)

Resistors and capacitors are quite generic, and you can usually interchange them from many different suppliers. SMD resistors have a power dissipation constraint related to the package size, for example 125mW is typical for an 0805 resistor. If you need higher power dissipation, you need to move up to a larger package size on your PCB or even a through-hole resistor.

Capacitors can often be considered quite generic as well, but you need to consider that both capacitance and voltage constrain the practical size of the capacitor package. For example, if you need a 22uF 50V ceramic capacitor in an 0805 package, that might be unrealistic and you might need a PCB layout designed for a larger SMD package.

Here I've chosen a 16V ceramic capacitor, since we want to have a healthy margin on capacitor voltage ratings, we might want to use a 9V battery, and the capacitance of some ceramic capacitors drops under applied DC bias.

I've also chosen some test points, which mount in a 1mm hole and provide convenient points for power connections which are color-coded and can reliably hold a test probe or clip. These are completely optional, but directly soldered

through-hole wire-to-board joints are prone to failure, hard to disassemble and repair, and are generally not ideal for electronics manufacturing.

Now that we've got the basic circuit design, and picked the parts we want, we need a schematic library file and a PCB footprint library file in our Altium Designer project.

You don't need to worry about creating any libraries in this subject. The demonstrators will provide you with Altium schematic and PCB library files with the parts and footprints you need for all the components you are expected to use, both for this simple multivibrator example and your Digital Storage Oscilloscope project.

I will discuss the process of library creation at the end of this tutorial – this is not essential for your project, but I think it's worthwhile for you to know.

You can get started straight away drawing the schematic of your design in Altium Designer using our library files, followed by PCB layout.

(Altium has some more advanced capabilities to model the electrical characteristics of the components in libraries, such as signal integrity, transmission-line models and SPICE models. However, these can be ignored, we won't use them in this course.)

Here is a tutorial available on the Altium website.
https://www.altium.com/documentation/altium-designer/from-idea-to-manufacture-driving-a-pcb-design-through-altium-designer?version=19.1

The Altium website provides comprehensive documentation for Altium Designer, and it's worth checking the website for further advice on specific Altium Designer functions.

Creating New Project and Schematic

In Altium Designer, a PCB project consists of a set of files. There is a project file, for example `Tutorial.PrjPCB`, a schematic file such as Tutorial.SchDoc, a PCB layout file such as `Tutorial.PcbDoc`, and a PCB library file `Tutorial.PcbLib` and schematic library file `Tutorial.SchLib`. You might have multiple schematic files (`*.SchDoc`) within your project, where a more complex electronic design is split up into subsystems across multiple sheets for easier design and readability.

To create a new project in Altium Designer, we select `File -> New -> Project` to open the Create Project dialog.

Select Local Projects in the list of Locations and confirm the Project Type is `PCB <Default>`. Enter a suitable name for the Project Name and select an appropriate directory where you want to store the project. Another directory, named as per your project name, will be created in this location. Click Create to create your new project.

The project will appear in the `Projects` panel. If this panel is not displayed, click the Panels button at the far lower right of the screen, and select `Projects` to turn on that panel. Select `File -> Save Project` to save your project, saving it in an appropriate location with some desired file name, like `Multivibrator`, for example. (You don't need to add the file extension, it will be added automatically.)

The next step is to add a new schematic sheet to the project. Right-click on the project filename in the Projects panel and select `Add New to Project -> Schematic`. A blank schematic sheet named `Sheet1.SchDoc` will

open in the design window and this schematic file will appear in the Source Documents folder under this project tree in the Projects panel.

Right-click and select `Save As` (or from the File menu) to save the new schematic file. We will save this in the same location as the project file and give it an appropriate name such as `Multivibrator` and click Save. (No need to add the file extension.)

Since you have added a schematic to the project, save the project file again by right-clicking on the project file in the Projects panel and selecting Save.

We now right-click on the project, select `Add Existing to Project`, and add both a Schematic Library file and PCB Library file to our project (`SchLib` and `PcbLib` files). These library files will be provided for you to download by your demonstrators, just locate and select those two files to add them to the project.

Drawing your Schematic
Now we have a blank schematic sheet open, ready to draw a schematic.

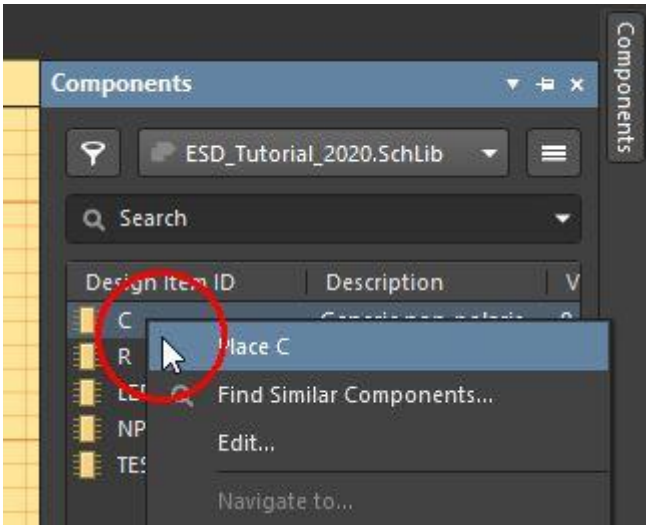
The properties of the schematic sheet (and properties of most other objects) are configured in the `Properties` panel. If the Properties panel is not visible, it can be opened by clicking `Panels` at lower-right and selecting `Properties`.

It is best to set both the `Snap Grid` and `Visible Grid` to 100 mil when drawing schematics and ensure that all symbol pins lie on the 100-mil grid. Under `Page Options` on the Properties panel, it is best to set the `Sheet Size` to A4, which allows easy printing of the schematic.

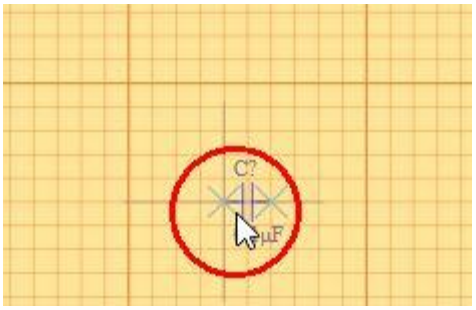
Certain preferences can be adjusted under the `Tools -> Preferences` dialog, such as color schemes, grids and the auto-pan behavior. I like to turn off the auto-pan, personally.

To place components on the schematic, we use the `Components` panel, and select components from within the library files in the current project.

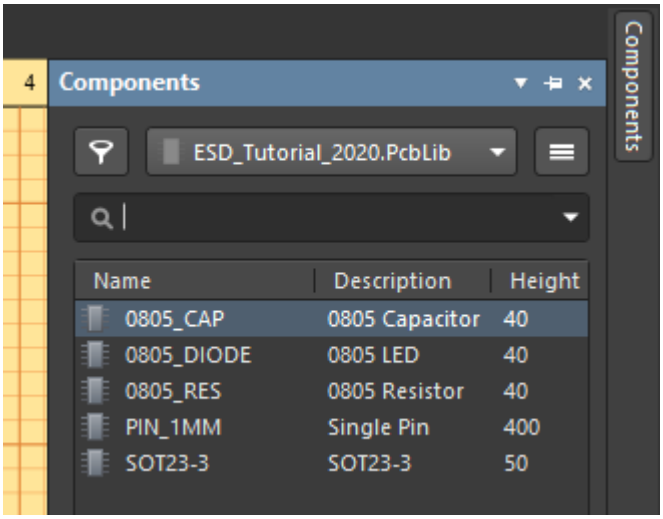
We right-click on a component and select Place from the context menu.



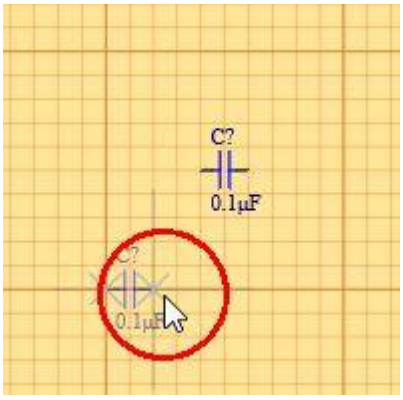
We then carry that component over to the schematic sheet and left-click to place.

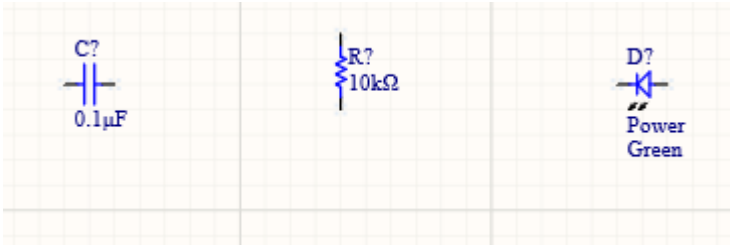


If you can't place the components when you right-click on them, a common mistake is that you have selected the PCB Library from the drop-down list. Here you can see I have selected the `PcbLib` file and each item has a black IC icon – not what you want. Select the `SchLib` file as shown in the screenshot above (the items in the library have a yellow schematic symbol icon).



We can click again to place another of the same component, or right-click to drop out of placement mode.

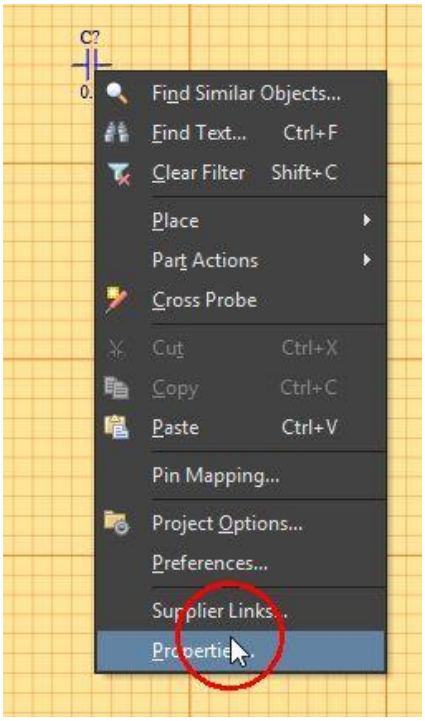




Here, I've started placing a few symbols on my schematic. We will then need to set the value or other properties of each component. If you need more components with the same value, it can be easier to just copy-and-paste an existing component with that value.

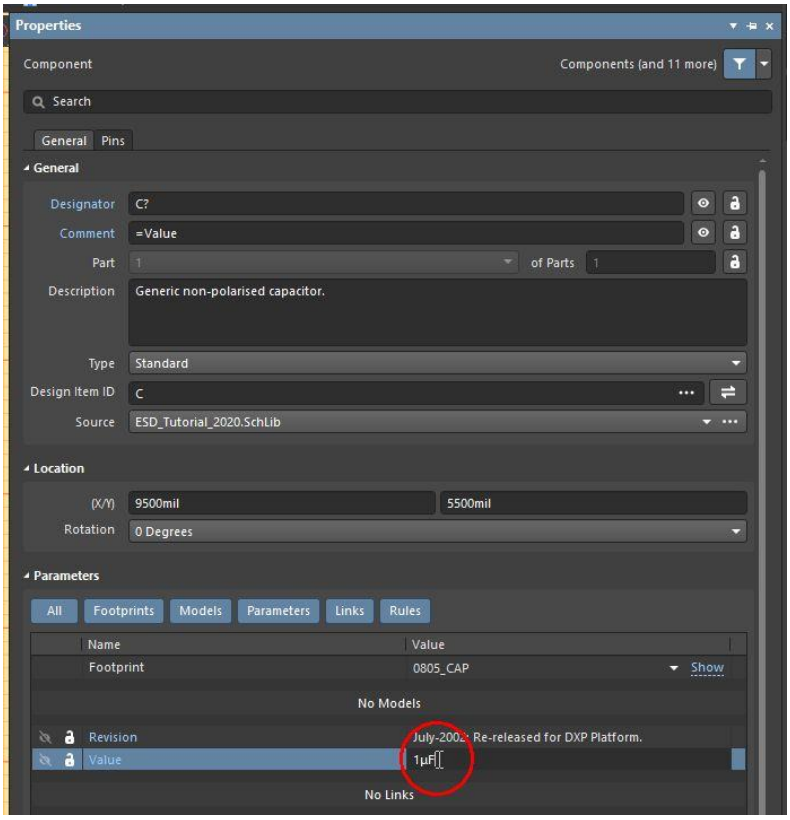
Press Space while placing a component, to rotate the component in 90-degree increments.

Right-click the component and select Properties, or press Tab, to display the properties panel and edit the properties of an object, such as the value of a resistor, or the net name of a Power Port.



Press X while placing a component to mirror the component along the horizontal axis, or press Y to mirror on the vertical axis. For example, on my multivibrator schematic below, one of the BJT symbols is drawn in a mirrored way. Control + Mouse Wheel will allow you to zoom in and out of the schematic. Or, use the PageUp and PageDown keys.

Right-click and drag to slide the schematic sheet around.



You will need to edit these parameters such as the values of resistors and capacitors.

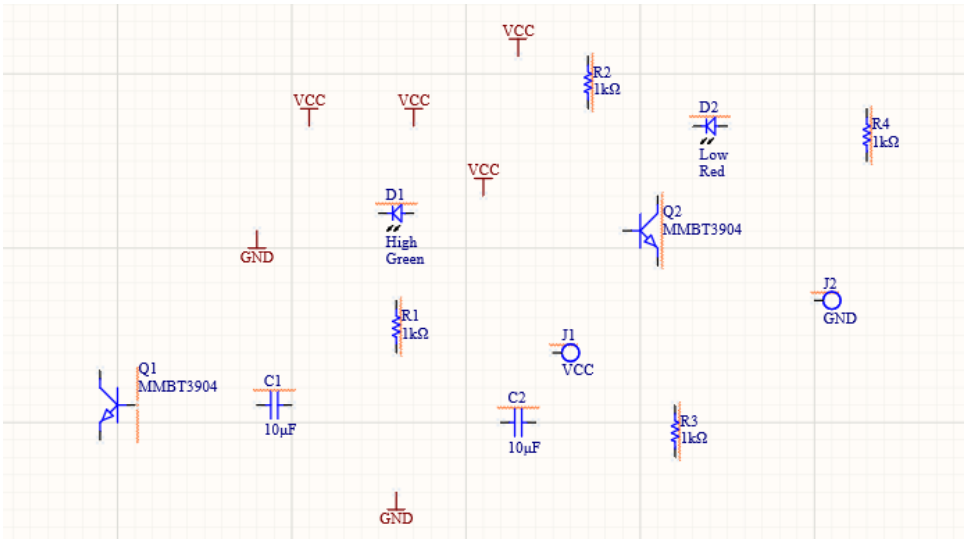
We also want to change the designators of components, so that they have numbered designators (such as R1, C1 etc) instead of “R?” which Altium displays by default.

This numbering is automatically managed for you so that it is easily updated, kept consistent and without gaps, and be managed in intelligent ways across large systems with multiple schematic sheets etc.

To assign the designators to all the parts on your schematic:

Select `Tools --> Annotation --> Annotate Schematics` from the menus.
Click `Update Changes List`, then click `OK`.
Click `Accept Changes (Create ECO)`. Then click `Execute Changes`.
Then click `Close`, and `Close` again.

Now all the numbering of components has been done for you.



Now I have placed all the components needed for my multivibrator design, I have started changing the component designators and values, copied-and-pasted some of the components, and I have also added some Power Port symbols with appropriate net names for VCC and Ground. (Do this using the Place -> Power Port menu.)

Power Ports

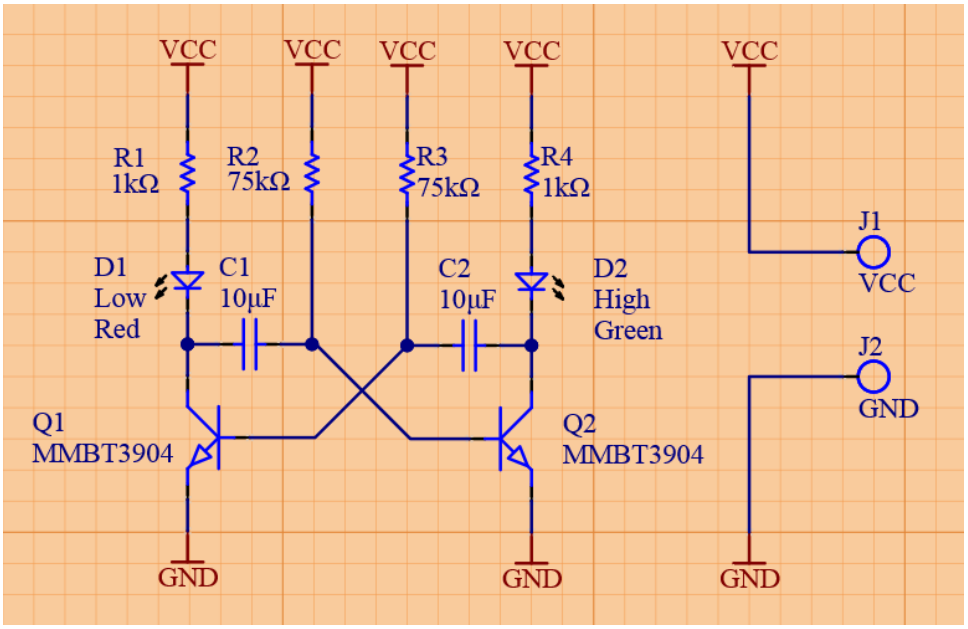
Power Ports are a very common important tool that makes your schematic much neater and easier to read. Power Ports are a shorthand symbol that represents a net such as Ground or VCC, where there are lots and lots of connections on that net from most parts of the circuit.

Each Power Port has a net name, and every one with the same net name is implicitly wired up together, even without explicit lines drawn on the schematic. (You can have multiple different Power Ports with different net names, for example +5V and +3.3V rails, isolated analog and digital VCC, or multiple isolated Ground nets.)

Drawing an explicit wire for every connection to Ground on the schematic would create a very messy and cluttered schematic that is hard to read, so this system of implicit nets connected together is much better.

Drawing Nets

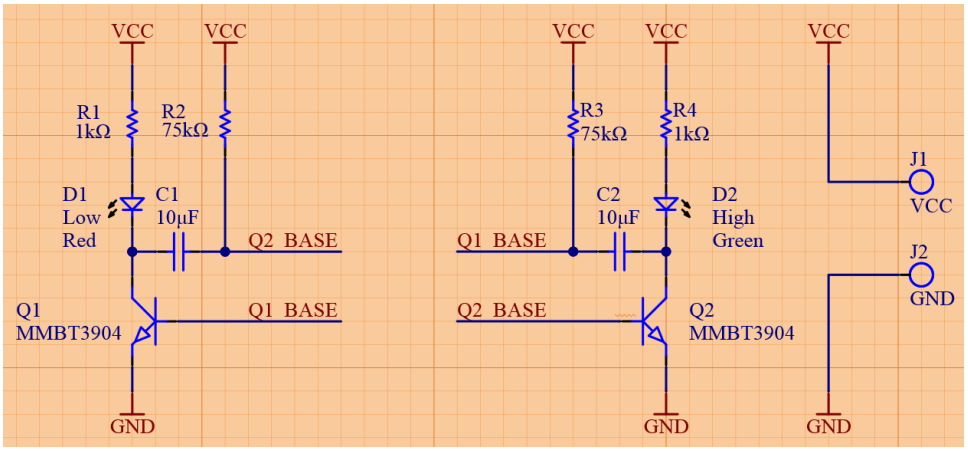
Now, we use the Place -> Wire tool to draw the schematic wires that connect together the pins on our schematic parts. Basically, click on a pin, draw a wire, route the wire and click on another pin. Right-click or press Esc to exit wire placement mode. (Don’t forget to save your files as you work!)



Here, I have completely drawn and wired up my multivibrator schematic with all the parts and connections needed, illustrating the use of Power Ports to make the VCC and Ground connections less cluttered and easier to read. I have also set the values and designators of the parts. (I also changed the background color, which is up to you.)

Net Labels

We can also place Net Labels on schematic nets. To do this, select Place -> Net Label, then press Tab to open the properties of this Net Label, and enter the name you want for this net. Then click the Pause icon to go back to the schematic editor. When placing a Net Label, the bottom-left marker on the Net Label should go on the net you want to attach it to. Right-click or press Esc to exit net label placement mode.



Here, I have deleted the two wires that make up the cross-coupled feedback connections to the base of each transistor in our circuit and replaced them with Net Labels, calling the two nets `Q1_BASE` and `Q2_BASE`.

These two diagrams are exactly the same circuit, they have exactly the same netlist and translate to exactly the same PCB – but this demonstrates how the use of Net Labels can make a complex schematic much easier to read.

Net Labels are a very similar concept to Power Ports, except Net Labels can be used for any net in the circuit and don't get a special symbol like power supply and ground nets.

Under the `Class Generation` tab in the `Project Options` dialog, uncheck the `Component Classes` checkbox, since we don't want component rooms or component classes in this simple, single-sheet design.

Now, you can compile the project by selecting `Project -> Compile PCB Project` from the main menu. If there are any electrical rule error messages during compilation, you can open the Messages panel to check them.

Now we are ready to start creating a PCB layout!

PCB Layout

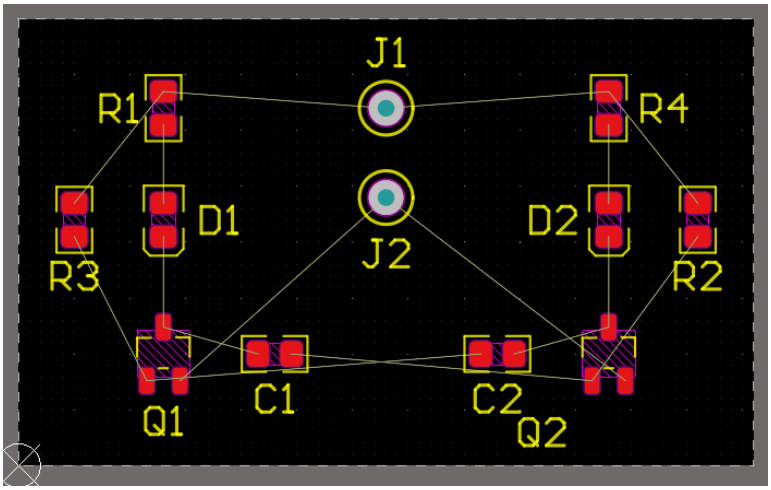
Right-clicking on the project tree, using the `Add New to Project` item and selecting `PCB`, we add a new PCB layout file to our project. We can then select Save As and save this file, again saving it as Multivibrator with the extension automatically added.

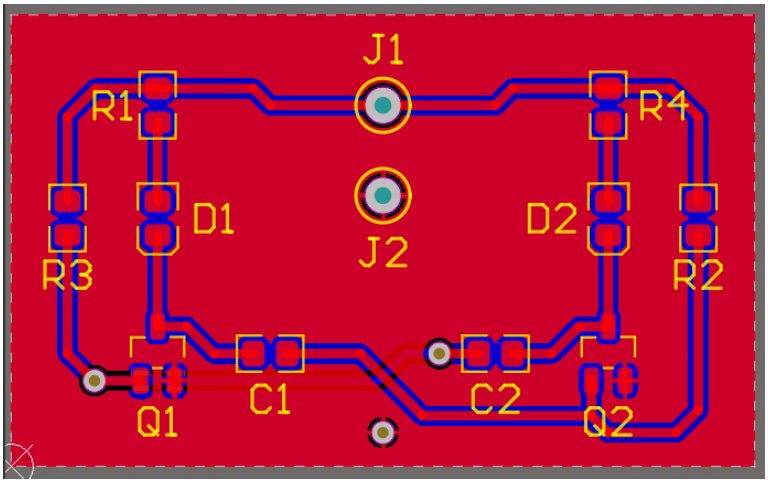
Use the `Edit -> Origin -> Set` command to set the coordinate system origin to the lower-left corner of the board.

Press `Ctrl-G` to display the Grid Properties and change the grid size. I recommend working in imperial units and trying a grid size of 50 mil or 25 mil.

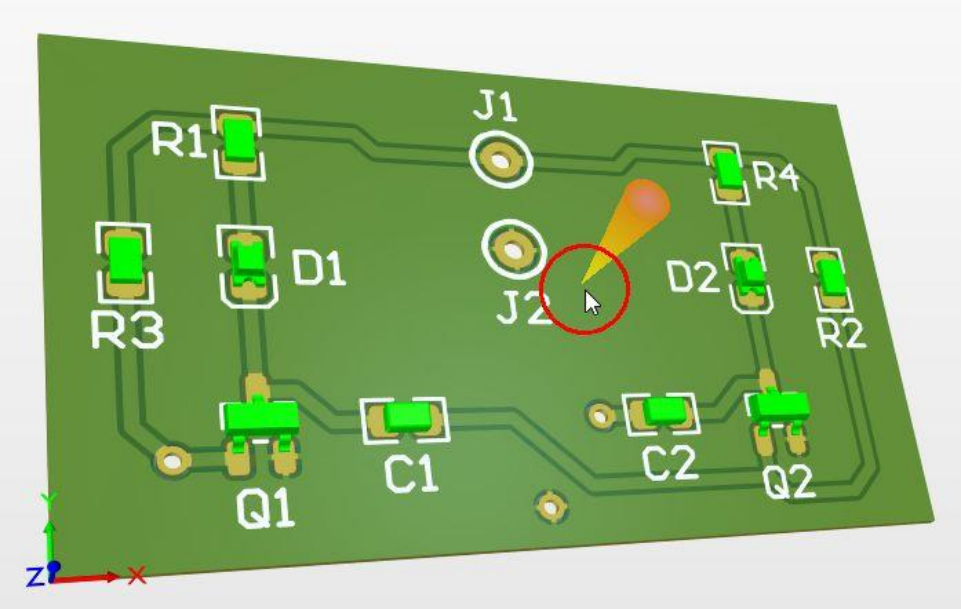
From the PCB editor, select `Design -> Import Changes From` to update the PCB layout with the component packages and netlist generated from the schematic you have drawn.

Now it is time to place your components in appropriate places, and route copper PCB traces between them. Here I have made a PCB of about the right size, and started moving the components into rational, organized locations based on the topology of the circuit.





Here's my mostly finished PCB layout – I've routed copper tracks for all the nets, including a couple of vias (through-board plated copper structures that connect a trace on a top layer to the bottom layer) which connect to the base of Q1. I have also added polygon pours, connected to ground, on both the top and bottom layers.



Using the 3D preview is a useful tool to check the appearance of your PCB – to get a feel for what the finished PCB will look like with components soldered onto it. (This assumes that Altium has 3D model libraries for each of the components used in your design.) For example, this can help you check the height of components (assuming the models are correct).

The 3D view can also be useful to help ensure that your component designator silkscreen markings are in appropriate locations with soldermask under them, and ensure **silkscreen markings are not overlapping an area where exposed metal is present**, since the silkscreen can't be printed onto exposed pads.

As a quick shortcut to access the 3D rendering view, simply press **3**, and press **2** to switch back to the 2D layout view.

Design Rules

Design Rules are essentially a set of specifications provided by a PCB foundry, describing the most precise tolerances that can reliably be manufactured using their lithography process.

Basically, when you order your PCB fabrication, your PCB artwork needs to comply with the design rules supplied by the fabricator.

Meeting those design rules means that both you and the factory have confidence the boards can be fabricated, without open-circuits or short-circuits accidentally introduced during photolithography and etching.

A PCB foundry may allow orders with smaller design rules, but this may require different materials and processes, at higher cost. As with all engineering and manufacturing processes, you should work with the coarsest tolerances that are acceptable for your system, as this means manufacturing will be easiest and cheapest.

For example, some typical PCB design rules for this course will look like this. Although you can fabricate smaller, high-density PCBs, we will work with relatively coarse design rules in this course to ensure that everybody's PCBs can be fabricated quickly, with confidence of full fabrication yield with no defects, and low cost.

Width of a copper trace or structure shall be at least **10 thou**.
Space between two adjacent copper traces (different nets on the same layer) shall be at least **10 thou**.

Vias (where copper is plated into a hole to interconnect two layers) will have a diameter of at least **40 thou**.
Drill diameter shall be **20 thou** or larger.

Traces and copper structures shall be excluded at least **40 thou** in from the board edge.

Silkscreen text will have a height of at least **36 thou** and a width of at least **8 thou**. (It is essentially printed, with a low-DPI printer, and you want the details to be readable.)

The Design Rule Check (`Tools -> Design Rule Check`) in Altium Designer is a software tool that lets you specify the design rules required, then Altium will automatically detect any places on the PCB where these are breached, allowing you to fix them.

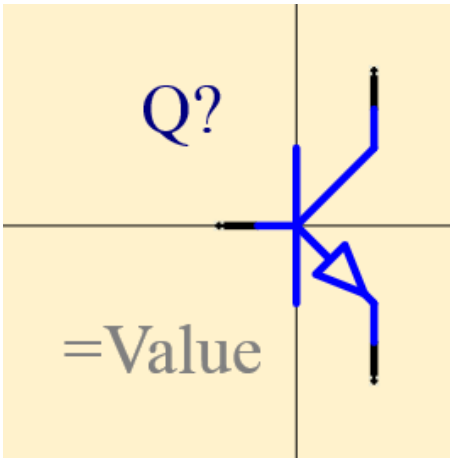
This includes things like short circuits (overlapping different nets on the same layer) and any nets on the netlist which are not routed on the board.

https://www.altium.com/documentation/altium-designer/design-rule-checking-ad?version=19.1

Optional – Creating Library Items

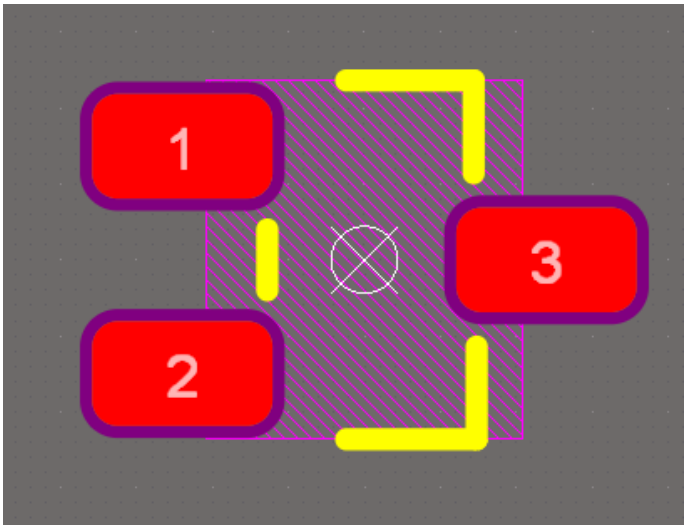
We draw schematic symbols for a resistor, non-polarized capacitor, LED, NPN BJT, and test point. An important point here is to draw the pins of each device, where wires will link up to the symbol on the schematic, on a 100-thou grid.

A schematic is typically laid out on a 100-thou grid, and it is much easier and neater if all nets are connected to pins which lie on the grid.



Here I'm drawing a BJT schematic symbol in an Altium library, for example. A device such as a transistor, resistor, op-amp or logic gate has a standard familiar symbol you should use on your schematics, but many ICs are often simply drawn in schematic libraries as a generic rectangular box with some pins attached and these pins given names corresponding to the names on the device datasheet.

When you do this, it is best to group the pins in a logical arrangement, with inputs and outputs in sensible groupings, grounds at the bottom and power at the top etc. This helps to give you the neatest possible drawing when you need put this IC in a schematic.



This is a PCB library footprint drawn for a SOT-23 package.

When drawing a PCB package in a library, you need to carefully check the drawings and dimensions supplied in the manufacturer's datasheet and replicate this layout. The yellow line is an outline that represents the body of the component – this will be printed on the PCB silkscreen.

You need to correctly record how the pins are numbered and include a marking on the PCB that denotes pin 1 on the IC package.

(For packages with an obvious asymmetry, like SOT-23 or SOT-25 packages, this isn't needed as you can't get the orientation wrong.)

You will find that you only need to draw common packages such as SOT-23 and 8-SOIC once and you will keep using it for many devices you add to your library, so this becomes less of a chore over time.

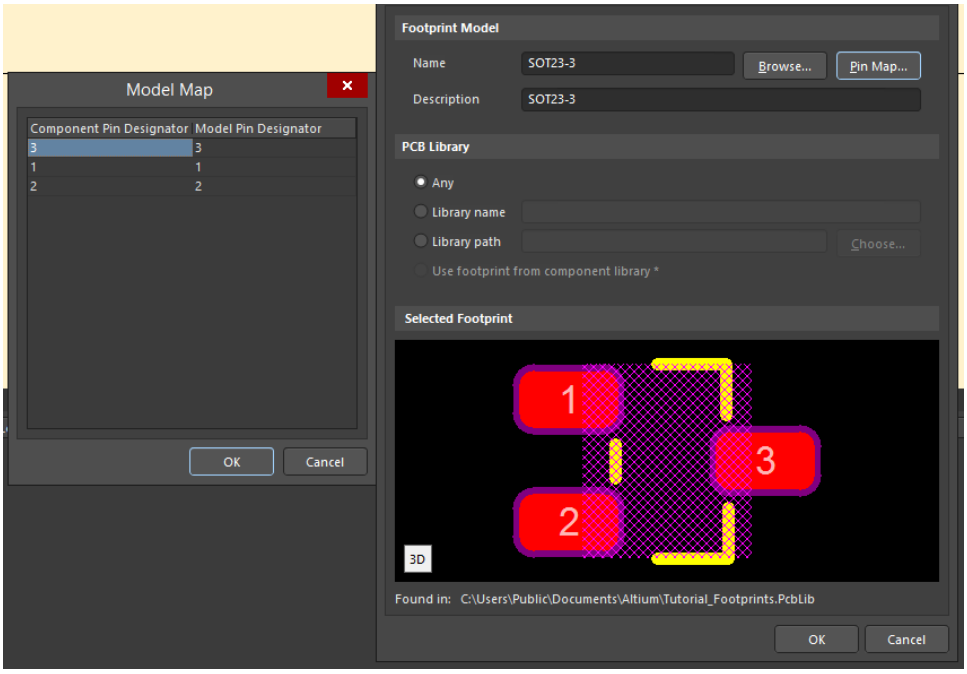
Creating and checking symbol and footprint libraries for the parts you need can be a lot of boring and tedious work. However, it saves a lot of trouble if you carefully ensure these libraries accurately represent the real-world components, and these libraries become an accumulated resource of your intellectual property that saves you time and makes schematic capture and PCB layout projects faster and easier in the long run.

As you work on more projects, your personal libraries of symbols and footprints will accumulate and you will re-use existing devices most of the time, and your library parts will be validated as you assemble real PCBs and test working circuits derived from these libraries, which will flush out any errors.

We will draw PCB footprint symbols for a SOT-23 package, an 0805 resistor package, an 0805 capacitor package, an 0805 LED package and the through-hole test point parts I'm using. The 0805 resistor and capacitor footprints are identical, except for their 3D rendering model.

Altium has the ability to have a 3D visualization of each library component and to create a visualization of the finished PCB assembly, but this isn't important for this tutorial.

You could use exactly the same 0805 PCB footprint design for both a resistor and capacitor, it doesn't matter. The 0805 LED footprint I have drawn is almost identical except for an 'arrow' symbol on the PCB silkscreen to indicate the cathode orientation when LEDs are assembled on the PCB.



You then add a PCB footprint entry to each symbol in the schematic symbol library, and carefully check the datasheet to specify the mapping between pin numbers on the PCB footprint and pins you have named and numbered on the schematic symbol.

In this case I have named the base pin on the schematic symbol pin 1, the emitter pin 2 and the collector pin 3. This doesn’t really matter – it's arbitrary, as long as the map is set up correctly. We consult the MMBT3904 datasheet and see that Pin 1 is base, Pin 2 is emitter and Pin 3 is collector, meaning that we need a 1-1, 2-2, 3-3 pin map.