Microsensors, Implantable Devices and Rodent Surgeries for Biomedical Applications
TA: Kaushik Lakshmiramanan, Course Instructor: Dr. Hardik J. Pandya
Department of Electronic Systems Engineering
Indian Institute of Science, Bangalore
Week - 10
Lecture – 48

Greetings, and welcome back to the NPTEL course on Microsensors, Implantable Devices, and Rodent Surgeries for Biomedical Applications. We're continuing our journey in the third part of this lecture series, where we're focusing on crafting an electronic device for signal acquisition and stimulation of the brain, particularly in rodent models, with the ultimate goal of translating these advancements to human applications.

In our previous session, we delved into interfacing the Intan Tech RHS 2116 chip with microcontrollers and FPGAs to construct a comprehensive circuit. Today, we'll embark on a new chapter, exploring how to transform this schematic design into a tangible printed circuit board (PCB) using CAD software, often referred to as Electronic Development Automation (EDA) software.

We'll start with a brief recap of where we left off. Recall the schematic we developed based on the RHS 2116 chip. We successfully interfaced it with an FPGA and microcontroller to acquire signals at the desired sampling rate. Furthermore, we enabled the microcontroller to program the FPGA, which in turn programs the chip for electrode stimulation. We also examined the necessary external peripherals, such as the USB-to-UART interface and protection ICs. Additionally, we considered the batteries and other power supplies required for biasing the microcontroller, FPGA, and the biopotential acquisition chip.

Our last lecture concluded with a discussion on translating this circuit design into a physical PCB. We touched upon several software options, including open-source choices like Easy EDA and KiCad, as well as licensed software such as Altium, OrCAD, and Eagle. For this demonstration, we'll be using Altium to provide you with an overview of PCB design.

On your screen, you should see two windows. The top one displays the schematic, which mirrors our circuit design but is arranged for enhanced visualization of each component. The bottom window showcases the PCB layout of the schematic, translated from bare symbols into a physical representation on the PCB. Notice how the lines from the schematic are printed using copper planes and wires. Additionally, observe the presence of pads, which are crucial for mounting the ICs. This provides a general idea of how a schematic and PCB relate to each other.

Now, let's dive into the process of creating our own simple PCB. I encourage you to utilize Easy EDA or KiCad, install it on your preferred device, and follow along with me.

I'll be conducting a short demonstration on building a simple PCB, a skill you can leverage to later develop more complex PCBs as needed.

Let's begin our journey from a bare schematic to a PCB layout and ultimately generate a Gerber file. When you launch your PCB software (in this case, Altium Designer), you'll be greeted with a welcome screen, as shown here. At the top, you'll find panels for various windows, such as the file view and project window. We'll navigate these as we progress.

In our next step, we'll initiate a new project and then add a schematic sheet to it. This schematic sheet will serve as the canvas for our circuit design. Once the sheet is ready, we can populate it with the components required for our PCB. These components will include resistors, capacitors, integrated circuits, and any other necessary elements. We'll connect these components using wires, ensuring a faithful representation of our intended circuit.

Once the schematic is complete and meticulously checked, we'll transition to the PCB layout phase. Here, we'll arrange the components on the board, paying close attention to their placement and the routing of the connecting wires. This process demands careful consideration of factors such as signal integrity, power distribution, and thermal management.

With the layout finalized, we'll generate the Gerber files. These files contain the precise instructions for manufacturing the PCB. We can then submit these Gerber files to a PCB manufacturer who will fabricate the physical board based on our design.

By the end of this demonstration, you'll have a foundational understanding of the PCB design process. Armed with this knowledge, you'll be empowered to translate your circuit ideas into tangible reality, opening up a world of possibilities for your electronic projects. So let's roll up our sleeves and embark on this exciting journey of PCB design.

Now, I'll provide explanations on a need-to-know basis to avoid overwhelming you with functionalities. Let's explore the interface. On the sides, you'll notice the project and navigator windows. The project window conveniently displays your currently open projects and those ready to be opened. Meanwhile, the navigator window enables you to navigate through all the components within your project.

On the right side, you have the components window, your gateway to importing component libraries from the internet. These readily available libraries, often well-documented, streamline the process of adding components to your design. In Altium, specifically, you have a handy "manufacturer part search" feature located in the bottom corner. This search function allows you to directly access components from libraries uploaded by manufacturers themselves, further simplifying your workflow.

For our current purpose, let's create a new project. Navigate to "File," then "New," and select "Project." We'll opt for a default PCB project. You'll see several options, such as "multi board." This is useful for creating multiple PCBs, particularly for panelized or flexible PCBs. In our case, we'll stick with a standard PCB project. Let's name it "demo" and click "Create." Keep in mind that Altium stores your generated PCBs in the cloud, though you can also save them locally if you prefer.

You'll observe that Altium has already created a schematic sheet and a PCB layout within our project window. However, if your project window is empty, perhaps due to no existing project files in the source documents, you can easily add your own. Simply right-click on the project name, select "Add New to Project," and choose to add a schematic. The schematic will be added with the default name "sheet1.schdoc." This is the standard extension for schematics in Altium, though it might vary slightly in other PCB design software. Don't worry about this minor detail; the concept remains consistent across different tools.

Next, right-click again and add a PCB layout to your project. You'll immediately notice the visual distinction between the schematic, resembling a sheet of paper, and the PCB, which appears like a blackboard.

Now, let's focus on the schematic view. I'll guide you through placing various symbols that represent the components we need to incorporate into our design. Subsequently, we'll transfer these components from the schematic onto the PCB and proceed with routing to finalize our PCB design.

For our schematic, let's take the simple example of a USB-powered LED. This straightforward circuit will serve as an excellent starting point for understanding the PCB design process. We'll add the necessary components, connect them appropriately, and then translate this schematic into a physical PCB layout.

Remember, the key is to start with a clear understanding of the circuit's functionality. Once you have that, the process of creating the schematic and PCB layout becomes much more manageable. So, let's begin adding the components for our USB-powered LED circuit and bring this design to life.

Let's keep things simple. As you can see, there are numerous options available on the toolbar within the schematic. It's important to note that the toolbar for the schematic document differs from the one for the PCB document, though some options might overlap. I'll do my best to explain them as we encounter them.

In Altium, you can go beyond schematic and PCB design; it also offers simulation capabilities. For instance, you could create a sine wave generator and simulate the sine wave with varying frequencies and amplitudes for your simulation projects. However, since we're focusing on a simple circuit design without simulation, we won't be utilizing those components for now. Those simulation sources and probes are readily available if needed in the future.

Within the schematic view, you can place shapes to help organize and visually enhance your schematic document. Feel free to decorate it to your liking. Text frames and notes are also available for naming components and keeping track of changes, respectively. The text frame will neatly enclose the component along with its text, while the note can be used for documenting any modifications or observations.

We'll explain each component in more detail as we progress and try to utilize as many as possible. However, the most crucial components we'll need are ports, which represent the power supply connections (positive, negative, and ground), and wires to connect the various components together.

Now, let's get back to our simple USB-powered LED circuit. First, let's search for the most important component: the LED itself. On the right panel, you'll find the "manufacturer part search." You can open it directly or access it through the panels. Once it's open, type in the component you need, in our case, "LED." You'll be presented with numerous options, which you can filter using the provided filters.

We're looking for an LED with models suitable for the PCB layout. So, we'll choose one that has a model associated with it. If you were planning on simulations, you'd also want to select an LED with a simulation model. Let's see what options we have.

We have a plethora of LEDs, each from a different vendor. You'll notice they each have a brief description, category, and supply information. If you click on a specific LED, you'll see that it has a symbol, a 3D model, and a 2D model. The symbol will be used in the schematic, and when we import it into the PCB document, this symbol will be translated into the 2D PCB symbol and the 3D model as well.

For now, let's select one of these LEDs. You can either right-click on it and choose "Place" or simply drag it onto your schematic document.

Now that we have our LED placed, let's continue adding the other necessary components for our USB-powered LED circuit. We'll need a resistor to limit the current flowing through the LED, a USB connector to provide power, and some wires to connect everything together.

As we add each component, we'll ensure that it's correctly oriented and connected according to the circuit diagram. Once all the components are in place and the connections are made, we'll be ready to move on to the PCB layout phase.

Please remember that I'm designing a generic circuit here. You'll need to consult the datasheets of each component you select to ensure they're compatible with your circuit. It's essential to verify that the power supply you're using is suitable for the LED and that the supplied current matches the LED's requirements.

To check the LED's specifications, click on its properties. If you don't see the properties panel, go to "Panels" and select "Properties." In the properties panel, you'll find a "parameter" column listing the LED's requirements. In this case, the reverse voltage is around 5 volts, and there's no forward current mentioned. The required voltage is 2 volts, and the forward current is 30 milliamps.

Typically, LEDs are powered with a constant current. So, we'll use a constant current driver IC that supplies 30 milliamps. Alternatively, you could provide a constant voltage of 2 volts for a compliant circuit. Let's search for a 30 milliamp constant current driver. You'll find several LED drivers available.

If you see one from a specific semiconductor manufacturer, keep in mind that it might take a minute or two to load the model. Now that we have the driver, you can check its datasheet for detailed information. As you can see, this is a 30 milliamp current driver, suitable for powering our LED. Feel free to explore the specifics of each driver, but for the sake of time, I'll proceed with this one and place it in our circuit.

In prototypes, it's common to use resistors to control the current flowing through LEDs. However, since heat dissipation can be an issue, let's opt for a more professional approach by using the constant current driver.

I don't have a physical USB port available, and I don't want to add one externally. Instead, I'll place a power port that I can customize into a USB port later. To complete the circuit, I'll also add a ground port.

With all the components placed, let's connect them using wires. It's quite straightforward. Just ensure the polarities are correct for the anode and cathode of the driver. You can use "control" and scroll to zoom in and out for better visibility.

You might notice squiggly lines near the components. This indicates that each component needs a unique name, which we haven't assigned yet. One way to do this is to double-click on a component, which opens the properties tab where you can name the designators.

Another option is to go to "Tools," then "Annotation," and choose "Annotate Schematics." You'll be presented with options for automatically annotating the schematic. You can see the annotation order, typically from top left to right, then bottom. You can also customize this order.

For a quick solution, go to "Tools," then "Annotation," and select "Annotate Schematics Quietly." This will automatically identify designators that need changes and update them accordingly.

Now that we've addressed the component naming, let's proceed with connecting the remaining components and finalizing our schematic. Remember to double-check all connections and polarities before moving on to the PCB layout phase.

Now that we've connected everything, let's save the schematic within our PCB project. You'll notice that this creates a PCB folder for your entire project, and inside that folder, it will automatically suggest a filename for saving. I'm going to rename it to "Demo_PCB_Project," matching my project name for easy tracking later on.

With the schematic saved, you can further customize its properties by simply clicking on the sheet and accessing the properties panel on the right. You'll find a variety of properties you can adjust. When working with numerous components, you can use the filters to select specific components, wires, or any other symbols individually.

I'm satisfied with the current setup, so let's proceed. Once the schematic is complete, it's crucial to validate the project before moving on to the PCB layout. Go to "Project" and select "Validate PCB Project." In this case, it highlights an error, indicating that we're using net labels for ground and VCC but haven't connected them to any power source.

To resolve this, let's add a USB connector. You can choose any type you prefer: micro USB, standard USB, or USB Type-A. I'll select a micro USB connector. Once you've chosen the component, you can rotate it by clicking on "Place" or use the "Y" and "X" keys to flip it around the respective axes.

To copy symbols, hold "Shift" and drag, or use the familiar "Ctrl+C" and "Ctrl+V" shortcuts. I'm arranging the components neatly. Don't worry about the additional pins on the USB connector (SHIELD 1 to 6); we won't be using the data lines for this simple circuit, so we'll leave them as is.

Now, let's name this new component. Go to "Tools," then "Annotation," and choose "Annotate Schematics." The component is now labeled. Let's save our progress.

Next, go to "Project" and "Validate PCB Project" once again. The validation messages will appear in the messages toolbar, which you might need to open and position at the bottom if it's not visible. In my case, I had to open it using "Panels" and selecting "Messages."

Now, we can see the compilation was successful, and no errors were found. This means we're ready to translate our schematic into the PCB layout.

Before we proceed with the transfer, let's prepare our PCB layout to receive the schematic components. This involves defining the board outline, setting up design rules, and configuring any necessary layers.

Once the PCB layout is primed, we can initiate the transfer process. Altium provides a seamless way to import the schematic components into the PCB layout, maintaining their connectivity and ensuring a smooth transition.

After the transfer, we'll meticulously arrange the components on the board, route the connecting traces, and conduct thorough design rule checks to guarantee the PCB's manufacturability and functionality.

By the end of this process, we'll have successfully transformed our schematic into a tangible PCB design, ready for fabrication and testing. So let's continue our journey and prepare the PCB layout for the next exciting phase.

Now that we have our PCB layout open, let's save the file using the same name as our project for consistency. With the file saved, I strongly recommend setting the origin of the PCB document. This will make it easier to track the dimensions of your PCB as you design it.

In the bottom corner of the window, you'll see X and Y coordinates displayed in mils (thousandths of an inch), reflecting the imperial system. There's also a grid that indicates how much the cursor moves with each step. You can switch these X and Y coordinates from mils to millimeters (the metric system) by pressing the "Q" key. You'll also notice the grid adjusts accordingly.

You can further modify the grid to a larger size using the "G" key. Observe how the grid becomes wider, providing more spacing between grid lines. Typically, a PCB document starts with a vast workspace, but we'll adjust it later to fit our PCB dimensions.

Let's set the origin now. Go to "Edit," then "Origin," and select "Set." Since we have a large grid, I'll zoom in using "Ctrl" and the scroll wheel. Remember, holding "Ctrl" allows you to override the grid snapping, giving you finer control over the cursor placement.

I'll position the origin at what I consider the bottom-left corner of my PCB. If it's not perfectly aligned, I can always readjust it using "Edit," "Origin," and "Set" again. Once the origin is set, you can use "Ctrl + Page Down" to view the entire PCB or simply zoom out using the scroll wheel. Utilizing these shortcuts significantly boosts your productivity, allowing you to navigate and complete tasks more efficiently.

With the origin established, let's bring the schematic symbols onto the PCB. Switch back to the schematic document, click on "Design," and select "Update PCB Document." This action updates the PCB document with your schematic symbols.

A "Change Order" dialog box will appear, listing the components to be added to the PCB. Notice how "D1," the symbol name, is translated to the PCB document when you click "Add." This highlights the importance of unique component names, as it allows you to selectively add components. You can even add nets, which are the names assigned to the wires connecting components. Common wires will share a common net name or net label.

Since I want to add all components, I'll validate the changes and execute them once the check is complete. You can then close this panel, and you'll see all three components from the schematic document now present in your PCB layout. You'll likely notice a change in scale compared to the schematic view.

To adjust the PCB dimensions to our requirements, let's switch to the "Board Planning Mode." In the view panel, you'll find three different views: "Board Planning Mode," "2D Layout," and "3D Layout." We're currently in the 2D Layout view. By switching to Board Planning Mode (or pressing the "1" key), we can modify the PCB dimensions.

With Board Planning Mode active, go to "Design" and select "Edit Board Shape." Since our grid size is relatively large, each cursor movement will correspond to 2.5 mm increments. I'll adjust the height and width of the board by dragging the edges. You can see the board dimensions displayed in the top-right corner relative to the origin. In this case, it's 25 mm x 20 mm, roughly a 2 cm x 2 cm board, which should suit our needs.

Now, press "2" to return to the 2D Layout view or navigate to "View" and select "2D Layout." This is where the critical aspects of PCB design come into play. The primary tasks here are arranging the components and routing the connections between them.

Observe the invisible lines associated with the pins. If you recall the schematic, these lines represent the wires we connected, and they're automatically linked based on their net labels. This visual aid helps maintain connectivity as we arrange the components on the board.

In the next steps, we'll strategically place the components, considering factors such as signal integrity, power distribution, and overall board layout aesthetics. We'll then route the traces, ensuring proper clearances and minimizing signal interference.

Once the layout is complete, we'll conduct design rule checks to verify compliance with manufacturing standards and identify any potential issues. Finally, we'll generate the Gerber files, which contain the manufacturing instructions for our PCB.

By following these steps, we'll have successfully transformed our schematic into a physical PCB design, ready for fabrication and testing. So let's continue our journey and arrange the components on the board.

So, the net labels are crucial here. You can see that a net was created by default, and it's associated with specific components. By pressing "Ctrl" and clicking on any net with your mouse, you can highlight all the labels connected to that net. For instance, if I "Ctrl + click" on the ground net, it highlights all the labels marked as ground. Similarly, clicking on another net will highlight all its associated labels. To clear the selection and return to the normal view, use "Shift + C" or right-click and choose "Clear Filter."

Now, let's arrange the components. Remember, you can rotate them by clicking on a component, holding the mouse button, and pressing the spacebar. My general approach is to place the power supplies near the origin and then work outward from there.

To cross-verify which component is which, especially when it's not immediately obvious, right-click on the component and select "Cross Probe." You'll see a flash on the corresponding component in the schematic, indicating the association. To utilize this effectively, I recommend splitting the screens. Now, you can easily cross-probe each component to confirm its identity. You can clear the filter by right-clicking and selecting "Clear Filter" to return to the normal view.

While cross-probing is helpful, it might become cumbersome for large-scale PCBs with numerous components. In such cases, the 3D layout view can be beneficial. Go to "View" and select "3D Layout" or simply press the "3" key. You'll see a top-down view of the PCB as translated from the 2D layout, making it easier to identify components like the USB connector, the constant current diode (LED driver), and the LED itself.

Keep in mind that there are smaller LED drivers available in the market. For this demonstration, I didn't explore all the options, but you can easily find components with specific parameters using vendors like Mouser or DigiKey, which offer extensive filtering options. We'll delve into that a bit later.

For now, let's focus on the layout. You might notice the grid appears a bit jittery because the grid size is quite large. Let's adjust it by pressing "G" and reducing the grid size. This makes the movement smoother, allowing for more precise component placement.

Feel free to arrange the components according to your preferences and the specific requirements of your design. Consider factors like signal integrity, power distribution, and the overall aesthetics of the board layout.

Once you're satisfied with the component placement, we'll move on to the crucial step of routing the traces, connecting the components and establishing the electrical pathways on the PCB. We'll ensure proper clearances, minimize signal interference, and adhere to design rules for a successful fabrication.

Remember, PCB design is an iterative process. You might need to rearrange components or reroute traces as you progress. Don't hesitate to experiment and refine your layout until you achieve the desired outcome.

I want to ensure the components are placed precisely 1 mm apart. I'll position the first one at the origin and then drag the others, maintaining a 1 mm spacing between them.

Observe the yellow lines or indicators. These will be translated to white in the 3D layout. This is known as the silkscreen layer or the top overlay layer in Altium Designer. It's printed on top of the PCB and serves as a helpful reference during assembly, indicating where each component should be placed.

Let me show you another useful mode in the 3D layout. Switch to 3D Layout using the "3" key and then press "Shift + Z." You'll now see only the pads on the board. This represents the bare PCB you'll receive from the manufacturer, and if you're assembling the components yourself (rather than outsourcing it), this view is invaluable for identifying the correct placement for each component.

Let's return to the 2D layout by pressing the "2" key and start routing the components. Before we begin routing, I want to optimize the component placement to minimize crisscrossing connections. Notice the intersection in this area. I'll rotate a component to improve the layout and create a cleaner flow.

Now, since I want a linear arrangement, I'll adjust the components accordingly. You can also move the silkscreen elements as needed. I'll reposition them for better clarity.

At this point, the board size is still larger than necessary. So, I'll go back to "Board Planning Mode," select "Design," then "Edit Board Shape," and further reduce the dimensions.

Back in the 2D layout, you can see the PCB edges are marked using a mechanical layer. Let's extend the USB connector to align with the PCB edge.

Speaking of mechanical layers, let's explore the various layers involved in a PCB and their functions. A PCB consists of a dielectric material in the middle, acting as an insulator. On either side of the dielectric, you have copper layers for conducting signals and power. These copper layers are separated by additional insulating layers. This layered structure repeats, allowing for multiple layers in a PCB.

To visualize the layers, go to "Design" and select "Layer Stack Manager." It might take a moment to load, but you'll see the layer properties. As mentioned, there's a dielectric in the middle, followed by a top layer, a top solder layer, and then the silkscreen layer. This configuration represents a two-layer PCB, which is the minimum. You can have PCBs with four, eight, or even more layers, but the cost increases with the number of layers.

Each layer serves a specific purpose. The top and bottom layers are primarily used for routing signals and power. The solder layers provide a surface for soldering components. The silkscreen layer, as we saw earlier, is for printing component outlines and labels. Other layers, such as the mechanical layers, define the board outline and other physical features.

Understanding the layer stack is crucial for successful PCB design. It allows you to visualize how the different layers interact and ensure proper signal integrity and power distribution.

Now that we have a basic understanding of the layers, let's continue with the routing process. We'll connect the components using traces on the appropriate layers, following design rules and best practices for optimal performance.

In Altium Designer, you have the flexibility to choose the weight (thickness) of the copper used in the top and bottom layers of your PCB. These are the layers where the wires or copper traces will be printed, forming the electrical connections. You can also customize the thickness of these copper layers according to your design requirements.

The dielectric material used in the PCB is typically FR4 substrate, although you can explore other materials available within the software. For now, we'll stick with the default settings. Feel free to experiment with different materials and layer configurations as you gain more experience. However, for this project, we'll be using a two-layer PCB since we primarily need a single layer for our circuit.

Let's return to the layer view. You'll notice that each layer is represented visually, and there are additional layers beyond the top and bottom copper layers. These additional layers serve various purposes.

Mechanical layers, for instance, are used to define the board outline and other physical dimensions. However, they won't be translated onto the final PCB. Courtyard layers, on the other hand, help prevent component interference by defining a keep-out zone around each component. If you try to place a component within another component's courtyard, you'll encounter an error.

You can toggle between viewing all layers simultaneously or focusing on a single layer using "Shift + S." This allows you to isolate specific layers for closer inspection.

Let's go back to single layer mode. If you attempt to place a component within another component's courtyard, you'll see an error highlighted. Besides these layers, you'll find assembly layers, 3D layers for visualizing 3D models, overlay layers for silkscreen prints, and paste layers for defining solder paste areas. Each of these layers has corresponding top and bottom versions.

In our current design, we're primarily using the top layer. You'll notice the bottom layer is mostly empty except for one connection. This is because we're using a through-hole component.

Let's switch to the 3D view to illustrate this. You can see that the USB connector is a through-hole component, meaning it passes through the entire board, from the top layer to the bottom layer, through the dielectric in between. That's why it appears on both the top and bottom layers.

You can change the viewing angle in the 3D model by holding "Shift" and using the right arrow key. You can also rotate the model freely or use specific number keys for predefined views: "1" for front view, "Ctrl + 1" for back view, "2" for side rotation, "3" for isometric view, and "0" for another isometric view. Feel free to explore these options.

Back in 2D mode, let's complete the routing process. Make sure you're in "all layer mode" by pressing "Shift + S." You can find all the available shortcuts in the "View" and "Project" menus.

Now, you might see some green crosses on the board. These indicate unrouted nets, meaning there are connections in the schematic that haven't been translated into physical traces on the PCB yet. Our next task is to route these connections, creating the necessary pathways for signals and power to flow between the components.

We'll utilize the routing tools provided by Altium Designer to carefully route the traces, ensuring they adhere to design rules and avoid any conflicts or shorts. We'll also consider factors like signal integrity and electromagnetic compatibility to ensure the PCB functions as intended.

Once the routing is complete, we'll conduct a final design rule check to catch any potential errors or violations. Then, we'll generate the Gerber files, which will be sent to the PCB manufacturer for fabrication.

With these steps, we'll have successfully completed the PCB design process, transforming our schematic into a physical board ready for assembly and testing. So let's continue and tackle the routing phase, bringing our USB-powered LED circuit to life.

Each PCB design has its own set of PCB rules, which dictate the allowable widths for pads and tracks and the minimum distances between components. You can't exceed these specified widths or place components too close together. However, these rules are customizable.

Let's address the violation we encountered. Right-click on the pad with the green cross and go to the violation. You'll see a "clearance constraint," indicating that the pads are not sufficiently spaced apart. Click on the clearance constraint to access the design rules, specifically the "PCB Rules and Constraints Editor."

Alternatively, you can go to "Design" and then "Rules" to reach the same editor. You'll find various design rules here, but we're interested in the "Clearance" rule. It defines the minimum clearance between different nets. However, the net in question is associated with the USB connector's footprint, and we can't modify it directly.

To resolve this, go back to the clearance rule and select "Ignore Pad to Pad Clearance within the Same Footprint." Click "Apply," and the errors should disappear. There might also be a "width constraint" violation, indicating a track is too narrow. For our purposes, this isn't a major concern, so we'll leave it for now.

In a real-world scenario, you'd need to check your PCB designer's capabilities and adjust the rules accordingly to ensure the widths and clearances are feasible for manufacturing. Some PCB vendors have limitations, and you might need to make the board slightly larger to accommodate their requirements.

Now that we've addressed the violations, let's connect the components using one of the routing options. Before we do that, let me explain the various options available in the toolbar. You'll find tools for drawing lines, arcs, circles, and other shapes. I'll also demonstrate how to change the board shape after routing.

For routing, we'll use the "Interactive Routing" option, but for a quick connection, we can simply click on "Quick Routing." Since our grid size is large, let's switch to a smaller grid of 1 mil or 10 mils.

Notice that once you start routing, the connection can only be made to the same net. This is enforced by the net labels, preventing accidental connections to different nets.

While routing, you can press the "Tab" key to modify the width and other parameters of the tracks. This is useful for high-current applications requiring wider tracks or for applications like antenna design or SPI communication where specific track lengths are critical. For our current design, we'll stick with the default options, but feel free to experiment with these settings as you gain more experience.

Now, let's complete the routing using these tools. We'll carefully connect all the components, ensuring clean and efficient traces. Once the routing is finished, we'll proceed to the final design rule check and then generate the Gerber files for manufacturing.

Remember, practice makes perfect. The more you experiment with routing and explore the different options available in your PCB design software, the more proficient you'll become. Don't hesitate to try different approaches and refine your techniques as you progress.

Excellent! Now that we've established the connections, let's take a look at the 3D view. You'll notice a change: copper tracks have been laid out beneath the top overlay, automatically connecting the components. This is what will be printed on the PCB, completing our circuit.

To get a better look at the 3D representation, go to "Panels," then "View Configuration." Here, under "View Options," you can enable "3D Bodies" to visualize the entire circuit in three dimensions. This is how our circuit will appear once assembled. When we plug it into a 5-volt supply, it should ideally power the LED.

This is a functional circuit, albeit a simple one. I wanted to demonstrate this example so you can apply these principles to more complex schematics. Before we conclude, let me show you how to modify the shape of the PCB.

I'll start by drawing a shape on the board. Before placing it, I'll press "Tab" to access its properties and change the layer to a mechanical layer. This ensures that the shape won't be translated into copper on the final PCB; it's merely for defining the board outline. I'll also add some fillets to make the shape more visually appealing.

With the shape in place, I'll define the board shape based on this object. Go to "Design," then "Board Shape," and select "Define Board Shape from Selected Objects." Since I've already selected the shape, I'll simply click on it.

Now, switch to the 3D view. You'll see that the board shape now incorporates the fillets we added. You can use "Shift" and the right mouse button to rotate the view and examine it from different angles.

Before sending the design for fabrication, you can also adjust the PCB thickness. While you can leave it as is and specify the desired thickness to the PCB manufacturer, it's good practice to set it beforehand. This way, you can visualize the changes and ensure the thickness is suitable for your application.

The standard thickness is usually around 0.5 mm to 0.6 mm. Let's make it thinner, say 0.15 mm. Once you save the change, you'll see the board thickness decrease in the 3D view. You can always revert it back if needed.

Controlling the PCB thickness is especially important for compact designs where weight is a concern. Thicker boards naturally weigh more.

Now that we've finalized the board shape and thickness, we're ready to prepare the design for fabrication. This involves generating the Gerber files, which contain the manufacturing instructions for the PCB. These files will be sent to the PCB manufacturer, who will use them to produce the physical board.

We'll explore the process of generating Gerber files in Altium Designer, ensuring all the necessary layers and information are included. Once the Gerber files are ready, we can submit them to the manufacturer and eagerly await the arrival of our custom-designed PCB.

Remember, PCB design is a continuous learning process. There are always new techniques and tools to discover. I encourage you to explore further, experiment with different designs, and continue expanding your skills in this exciting field.

Excellent! Now, to send this design for fabrication, we need to generate Gerber files. Go to "File" and then "Fabrication Outputs." You'll find options for Gerber files and Gerber X2 files. Gerber files are widely accepted by most PCB manufacturers, so we'll choose that option.

In the Gerber setup, select the layers you want to include in the Gerber files. Each layer is exported as a separate file. For now, I'll select all layers and enable "Software Arcs" to ensure smooth curves are represented accurately. Click "Apply" to generate the Gerber files.

Altium will process the design and create the Gerber files, which will be stored in your project's output folder, within the same directory where you created your PCB project. You can then archive these files and send them to your chosen PCB manufacturer.

Within Altium, you can also use the "CAMtastic" menu on the left to preview each layer individually and make any necessary adjustments. This allows you to verify the design and ensure it meets your expectations before sending it for fabrication.

In our case, we've already validated the PCB project and confirmed there are no errors. However, for more complex designs, it's highly recommended to perform a "Design Rule Check" (DRC) before generating Gerber files. The DRC allows you to specify clearance parameters, track widths, and other design rules to ensure your PCB adheres to manufacturing standards and avoids potential issues.

To run the DRC, go to "Tools" and select "Design Rule Check." Configure the desired parameters and click "Run DRC." A report will be generated, highlighting any violations or areas that need attention.

In our simple design, the DRC might show a few minor violations, such as the width constraint we encountered earlier or some solder mask constraints. These are generally acceptable, but for a professional PCB, it's best to minimize or eliminate all errors. You can click on each error to see the specific constraint violation and take corrective action if necessary.

Now that we've covered the PCB design process from schematic to Gerber generation, I encourage you to apply this knowledge to more complex projects. Explore the capabilities of your PCB design software, experiment with different designs, and challenge yourself to create sophisticated circuits.

Remember, the forum is available for any questions or doubts you might have. Feel free to post your queries, and we'll do our best to provide assistance and guidance.

Once the PCB is fabricated, the next step is to package it appropriately for placement on the rat. We'll explore the possibilities of additive manufacturing to create a custom enclosure that allows for free movement while securely housing the PCB. We'll also discuss how to interface the acquired signals with a graphical user interface (GUI) using software like MATLAB or Python, enabling real-time visualization and analysis of the data. My colleagues will cover these topics in more detail in subsequent lectures.

To give you a glimpse of the final system, here's a simple representation of how the packaged device might look. Once mounted on the rat's back, it will facilitate experiments using sensors implanted in the brain. The acquired data will be transmitted to the GUI, providing a comprehensive platform for measuring brain potentials and even stimulating the brain.

Thank you once again for your attention. If you have any further questions, please don't hesitate to reach out to us on the forum. I look forward to seeing you in the next lecture.