DesignSpark Tips and Tricks
Day #4: A simple project

By Neil Gruending
(Canada)

Last time I talked about how to set up and use libraries in DesignSpark. Today, we’ll make a simple bi-color LED driver to learn how to use the schematic and circuit board editors. There are several ways to drive a bi-color LED, and today we will use an H-bridge variation which is hardwired to turn one of the LEDs on.

Figure 1.
Schematic of the bi-color LED driver.

Drawing the schematic
The first thing we need to do is to create a project file to link the schematics and PCB by using the “File->New” command. You then add a schematic to the project by using the “File->New” command again, but make sure that you check the “Add To Open Project” button. At this point you can also choose the technology file to use for the schematic, as we talked about in an earlier article. Figure 1 shows the schematic we will be using.

DesignSpark has a schematic entry tutorial at [1], which covers how to add components and edit the schematic.

Moving visible component fields such as reference designators around in a DesignSpark schematic is different than in some other packages because all of the fields are moved in one block. For example, in the example schematic, there’s a part number and reference designator visible for each transistor. Clicking on either one will highlight both fields and they can be dragged to a new position as a group. This can cause problems when mirroring components, which DesignSpark calls flipping, because the text can be incorrectly lined up. Fortunately, the text alignment can be easily changed by right clicking the text and selecting the “Properties” menu. There in the “Text” tab you will find an alignment field that will let you choose between “Left,” “Center,” and “Right” text alignment.

Also, don’t forget that power and ground symbols are components in DesignSpark. The default ones are in the DesignSpark scheme library, but you can also create a library of symbols to your own liking. Note that if you connect a power symbol to an existing net, DesignSpark will warn you that it will rename the net even though that’s what you want.

The LED component
For the transistors and resistors I used some existing schematic symbols and PCB footprints from the DesignSpark libraries. However, for the LED I modified an existing DesignSpark LED symbol and then made a custom PCB footprint for it. Making a custom PCB footprint is a lot easier when you use the footprint wizard. You can access the wizard by opening the PCB library where you want to save the footprint with the library manager and then clicking on the “Wizard...” button.
The PCB footprint wizard will then ask a series of questions to make a footprint. Since they are a generic as possible, it’s important to pick the closest type possible to minimize later editing. In the case of my LED, I used an axial component with 2.54-mm (0.1") lead spacing, so that all I needed to do was to edit the silkscreen and mark the polarity on pin 1. DesignSpark also includes similar schematic symbol and component wizards.

Getting ready for layout
Now we’re ready to lay out our circuit board by creating a new PCB file using the “Tools->Translate to PCB” menu, which will start the New PCB Wizard. We are going to create a two-layer metric design that’s a 20 mm square. If you tell the wizard to place the components outside of the board, you will get something similar to Figure 2. I like to place components on a 0.25-mm grid, so I changed the working grid to 0.25 mm before placing the components on the circuit board to get an arrangement like Figure 3.

Before routing the board I want to talk about the routing grid used when placing the copper traces on the circuit board. DesignSpark doesn’t include an interactive autorouter, which means that you have to set the routing grid to the width of the trace you’re routing. This way when two traces touch, the spacing is 0 mm and there’s a gap between them. The spacing is at least the trace width. This works because the routing grid is applied to the center of the trace instead of the edges. Therefore, if you want to route a 0.2-mm trace, then you would set the routing grid to 0.2-mm spacing to get 0.2-mm trace spacing. The downside of this technique is that all the trace widths should be multiples of the smallest size. For example, 0.2 mm and 0.6 mm would work but 0.2 mm and 0.35 mm would not. Also, be sure to create a style for each trace width that you want to use in the design technology settings (Settings->Design Technology... and then select the “Track Styles”). That makes the different trace widths much easier to manage in more complicated designs, because you can change the current trace width by just changing the style. You can change the current track style by pressing “s” while routing and then choosing the new style you want.

The same is also true for vias and in DesignSpark—you configure the via styles using the “Pads Styles” tab in the “Design Technology” window. I recommend making a “SignalVia” style and any other styles that you may need. I chose to make the signal vias in this design with a 0.45-mm hole and a 0.95-mm pad. You can change the current style used for vias by right clicking while routing and going into the “Change Via Style” menu to choose the style you want. DesignSpark uses the “Settings->Defaults” menu to set the track and via defaults, although I haven’t been able to get DesignSpark to recognize the new settings even with a restart.

More information is available from the DesignSpark website about PCB setup and
placement is available at [2].

**Layout**

Once everything is configured it’s time to lay out the circuit board. The result is pictured in Figure 4.

I routed all of the signal traces on the top layer and used a polygon pour to make a bottom ground plane. When routing the board it’s important to make sure that you double click on the “rat’s nest” interconnect line when starting to route a trace. As you run the trace, you can change how DesignSpark deals with corners by right clicking and choosing a different segment mode. If you look closely at the layout, you can see a “rat’s nest” line between Q4 and Q5 for the ground connection, which means that DesignSpark thinks that those two transistors aren’t connected to ground. Fortunately, DesignSpark includes a design rule check (DRC) in the tools menu that can verify all of the board connections—see Figure 5.

Once you click the “Check” button, DesignSpark will then verify that the layout meets all of the selected design criteria and it will generate a report summarizing any errors. Errors are also marked in the layout. If you mouse over them, the error message will be displayed. All of the clearance rules are set in the design technology settings “Spacing” tab, where there’s a table listing all of the clearances between different object types.

For more information about routing a board with DesignSpark, a tutorial is available at [3].

**Conclusion**

Today we created a simple PCB from a schematic and then verified the design using DesignSpark’s verification tools. Next time we’ll generate a BOM and the Gerber files so that we can build the design.

(130230)

**Internet References**