

Circuit Design with EAGLE

Version 0.2

Table of Contents

Introduction to EDA

Part 0: What are we making, anyway?

Part 1: Create a new project

Part 2: Design the schematic

Part 3: Lay out the PCB

Part 4: Generate Gerbers

Appendix A: Extra Resources

Introduction to EDA

Firmly situated in the Information Age, we find ourselves surrounded by technology. Whether it be your cell phone in your pocket or the thermostat in your house, electronic devices will almost universally contain a **printed circuit board (PCB)** of some kind. Some are fairly simple (e.g. solar-powered lights in a garden), and some very complex (e.g. the motherboard of your computer). But they all allow for the interconnection of individual components contained on them.

How are these boards designed? Typically, an electrical engineer will model them in a piece of software called an **Electronic Design Automation program (EDA)**. They typically allow you to perform two actions:

- create, modify, and test circuits (CAD)
- model those circuits for fabrication (CAM)

The relative affordability and power of personal computers - coupled with the open-source software movement - has created a space where non-electrical engineers are able to utilize these same tools to create their own PCBs for small- and medium-scale projects!

EDAs typically separate the two aforementioned steps into:

1. Schematic design (CAD)
2. PCB design (CAM)

We'll be focusing today on both of these, with the goal of having a finished PCB design that you then fabricate - either through a 3rd party fabricator, or in-house (if you have a fitting CNC machine)! You'll be learning **Cadsoft's EAGLE**, which is the most ubiquitous EDA on the market. It also has a few interesting quirks, which you'll see soon enough. But due to its widespread adoption (as well as its pricetag of free for a simplified version), it's the best choice for entering the EDA world.

Let's get started!

Part 0: What are we making, anyway?

For your first EDA-designed project, it is vital that we keep the scope concise and understandable, while maintaining reliability. For these reasons, the circuit that you will be designing is a simple **digital-to-analog (DAC)** breakout board.

You will have had experience at this point with Arduino: a wonderful and accessible microcontroller environment which allows you to rapidly design and prototype systems that combine software and hardware. One of the finer points you may have stumbled across by now is that while these boards have plenty of Analog Inputs, *no Atmel board has a true Analog Output*. “Analog outputs” on Arduino are faked using a technique called **Pulse Width Modulation (PWM)**. While this is a fantastic technique for emulating analog signals on digital devices, at some point you’ll likely need a true analog signal (such as a function generator or an audio signal); Arduino boards cannot produce this type of signal.

There are many digital to analog converters on the market; they’re fairly inexpensive and easy to set up. We’ll be creating a DAC breakout today using the [MCP4822 from Microchip](#). (Note: the DAC we’re using communicates via SPI. If you’re unfamiliar with the SPI communications protocol, please ask your instructor.)

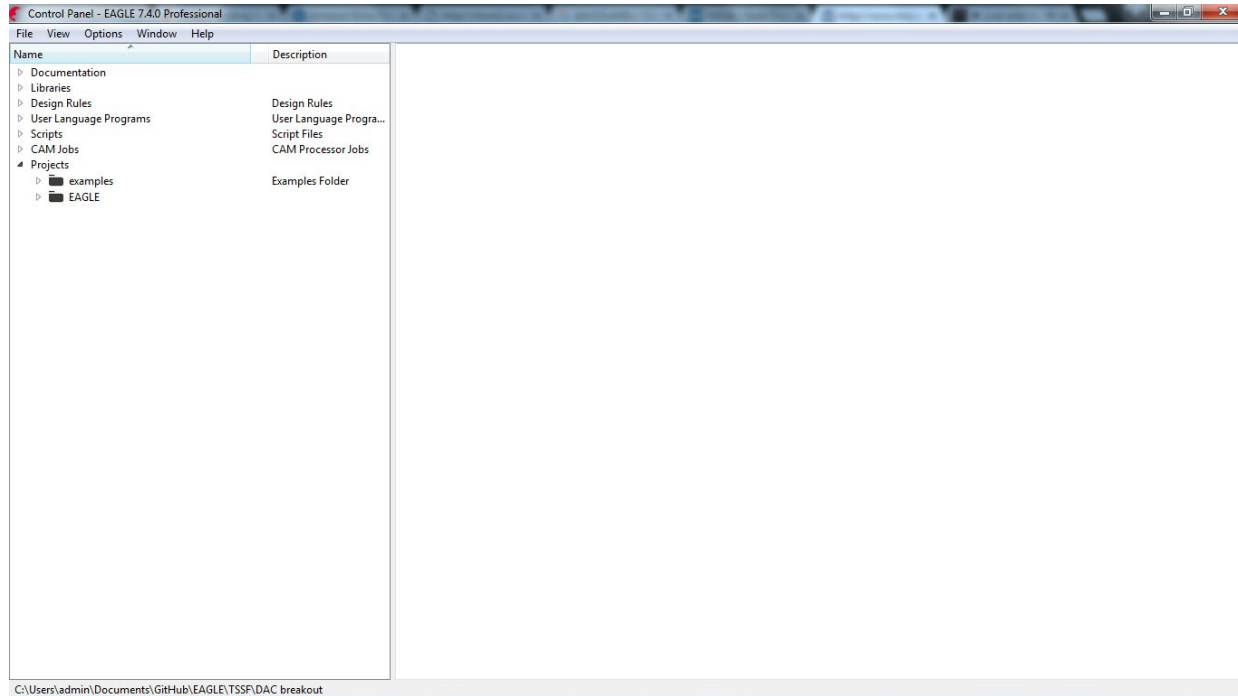
As an additional step, we’ll need a few extra **libraries** in order to create this board. EAGLE libraries give you access to more parts with which to design. They’re currently (as of 07 Dec) located in the “lbr” folder. Move these into the following local folder:

`C:\Program Files\EAGLE 7.5\lbr`

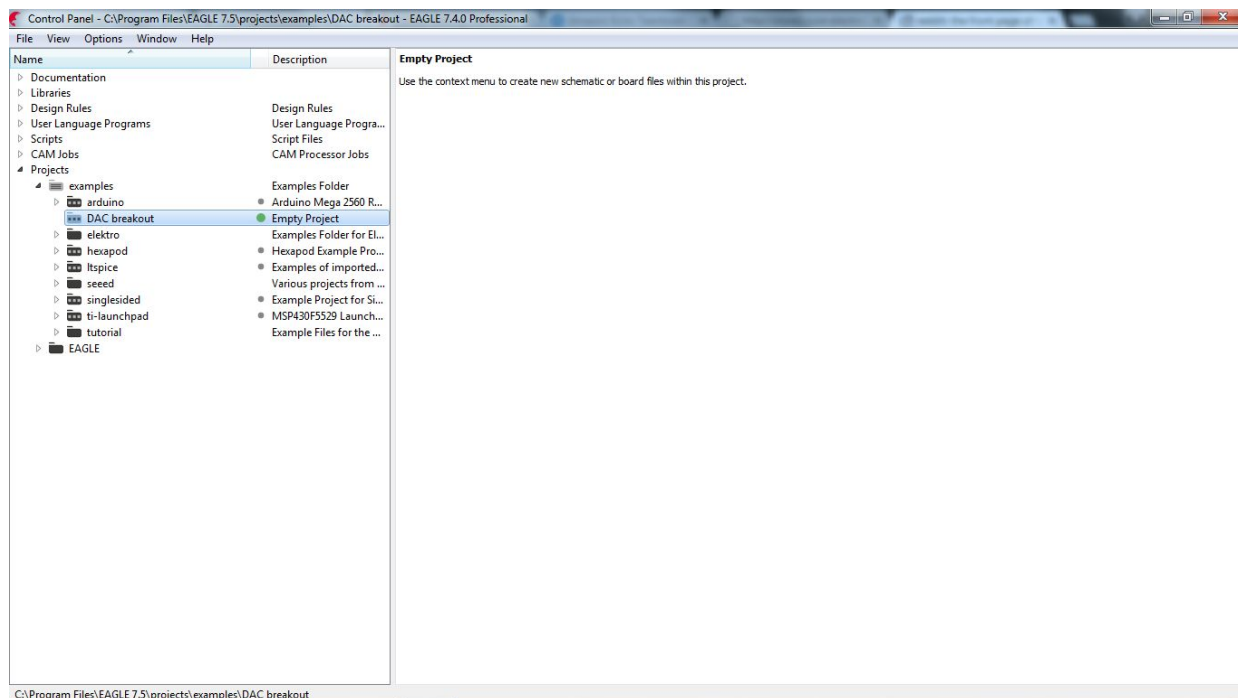
Part 1: Create a new project

Opening up EAGLE, you'll be presented with the window below.

This is the **Project Window**, and it is effectively just a browser for all things EAGLE. You will spend very little time here; it's mostly just for navigation and project selection.

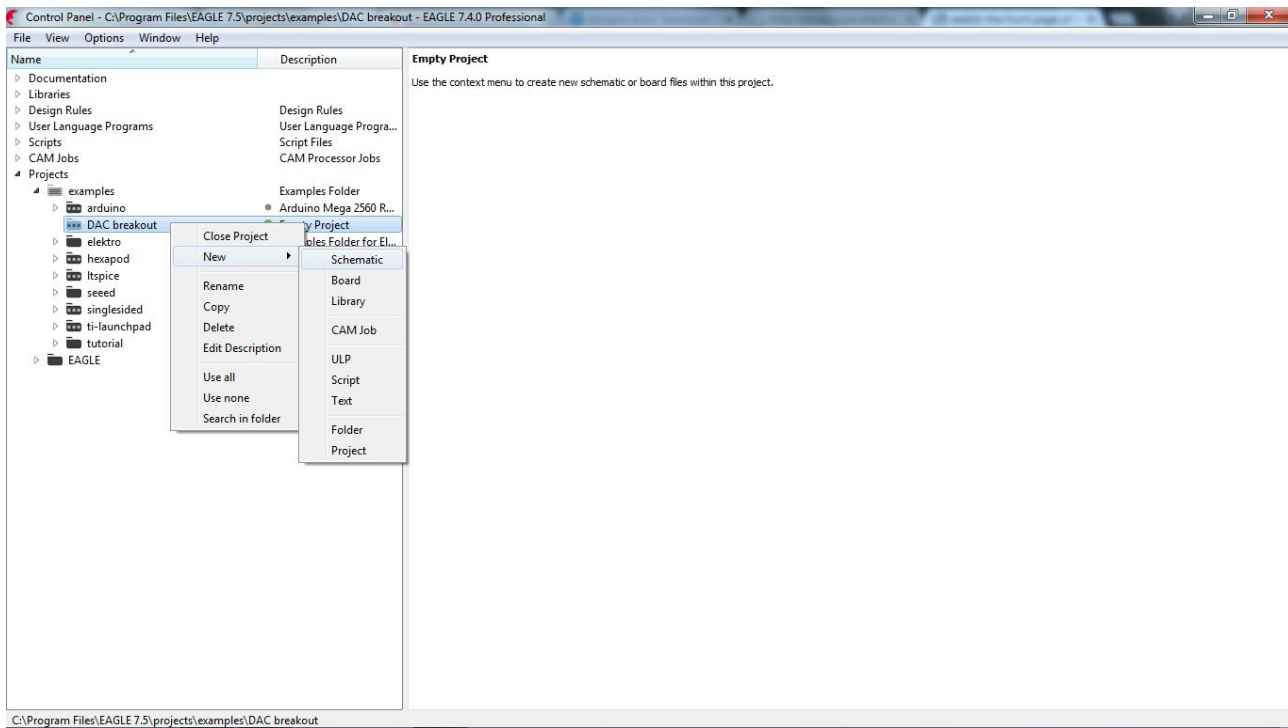


The first thing you'll want to do is create a new project! You can do this by right-clicking on one of the folders inside "Projects" and selecting "New Project". Rename it so that it is called "DAC breakout":

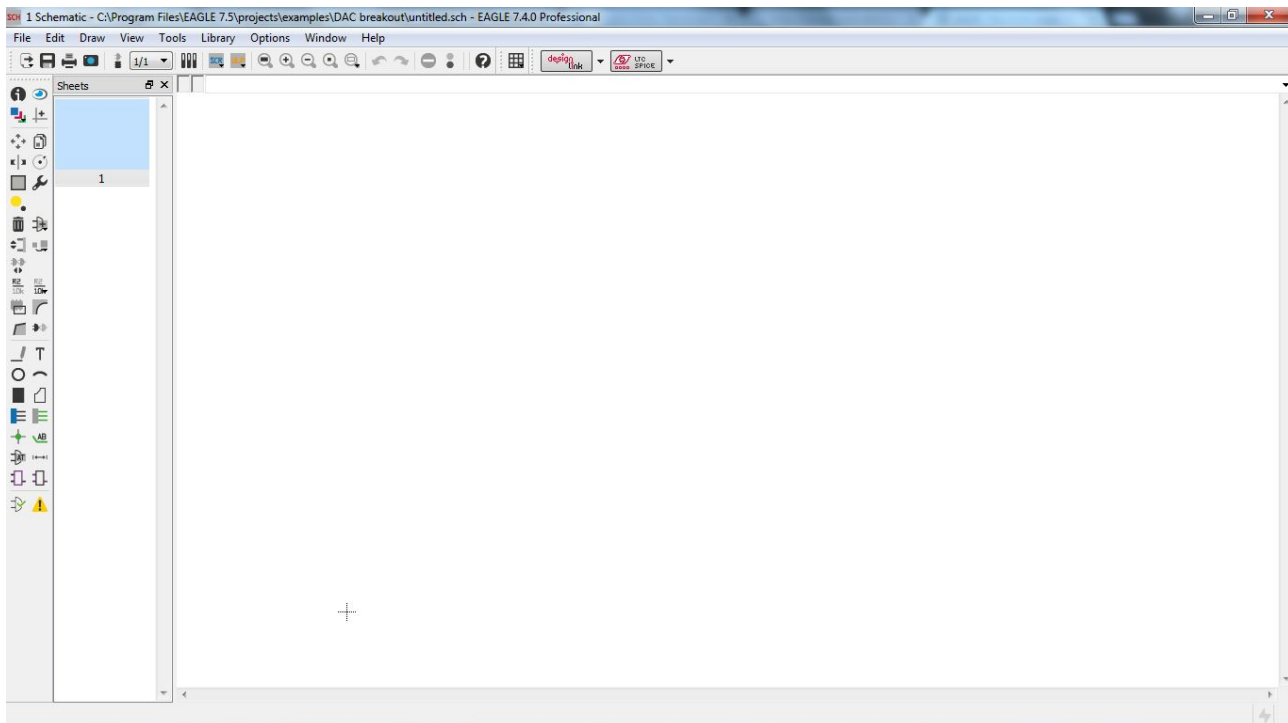


Next, you'll want to add a new schematic to your project! Right-click on your project, and select:

New → Schematic



The window below should appear.



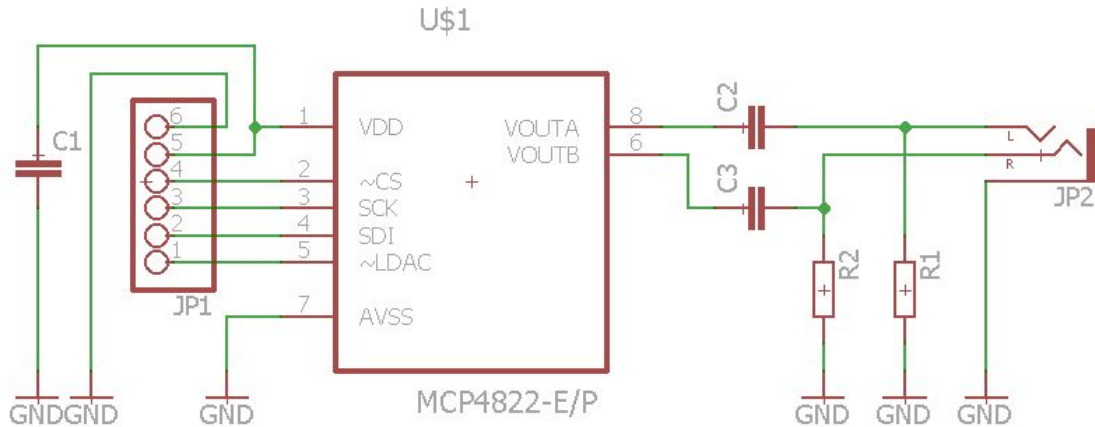
Save this so that it's got a real name! Name it "DAC Breakout.sch":

File → Save as

Part 2: Design the schematic

You have now entered the realm of the **Schematic Capture window**. This is where you will design your circuit! What we'll be doing, in essence, is defining the components and the electrical connections among them.

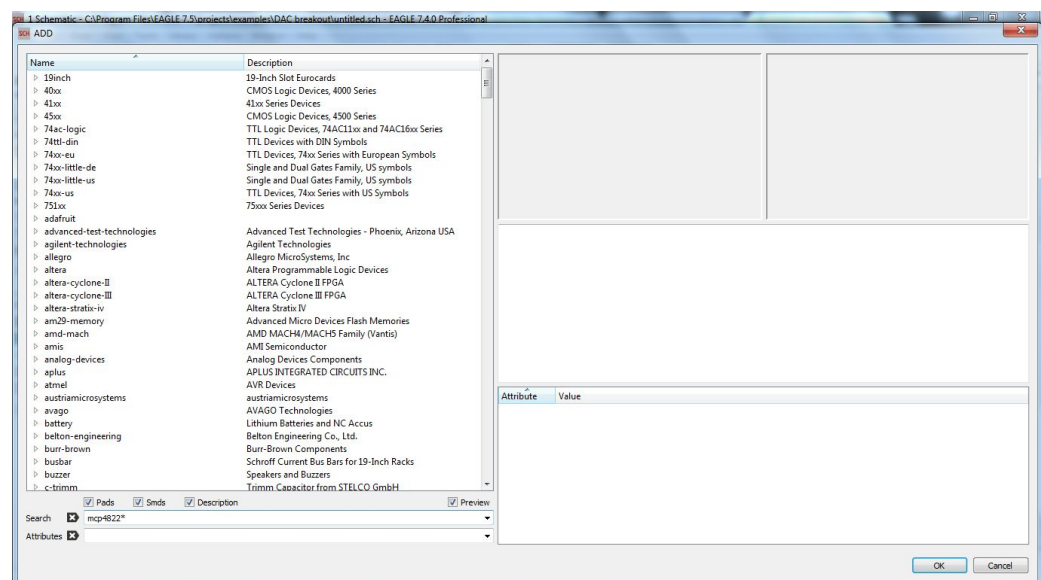
For reference, this is the circuit you'll be making:



You're going to spend quite a bit of time in the Schematic Capture window, so let's quickly take a look at the **tool palette**, which is on the far left side of the window (and the far left of this page). Because EAGLE was built on top of command-line scripts, its toolset functions slightly differently to most other **User Interfaces (UIs)**: each type of action requires a different tool. So, for example, to delete something you must select the Delete tool (trash can). Using the Delete key on the keyboard does not work. (As a side note, hovering over any of these tools should present you with a tooltip of their name!)

Add the components

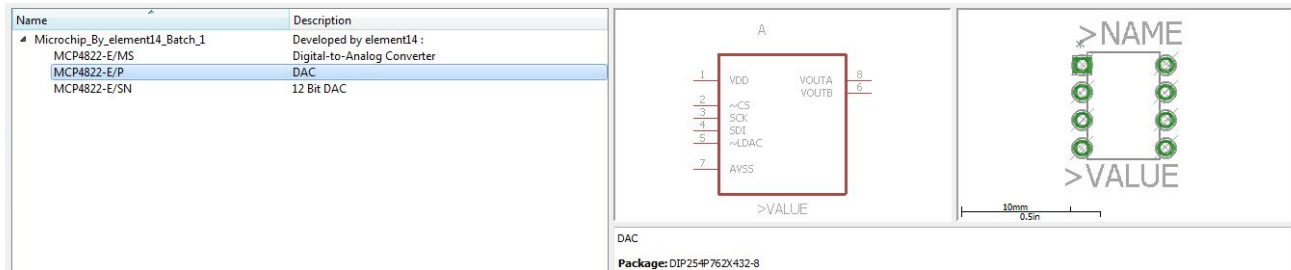
There are many many tools here, and we most assuredly will not be using every single one of them today. The first (and arguably the most important) tool we're going to use is called the **Add Component tool**. Click on it, and a new window should pop up that looks like below:



Let's start by adding the heart of the breakout board - the MCP4822 - to the project. In the Search bar at the bottom, type:

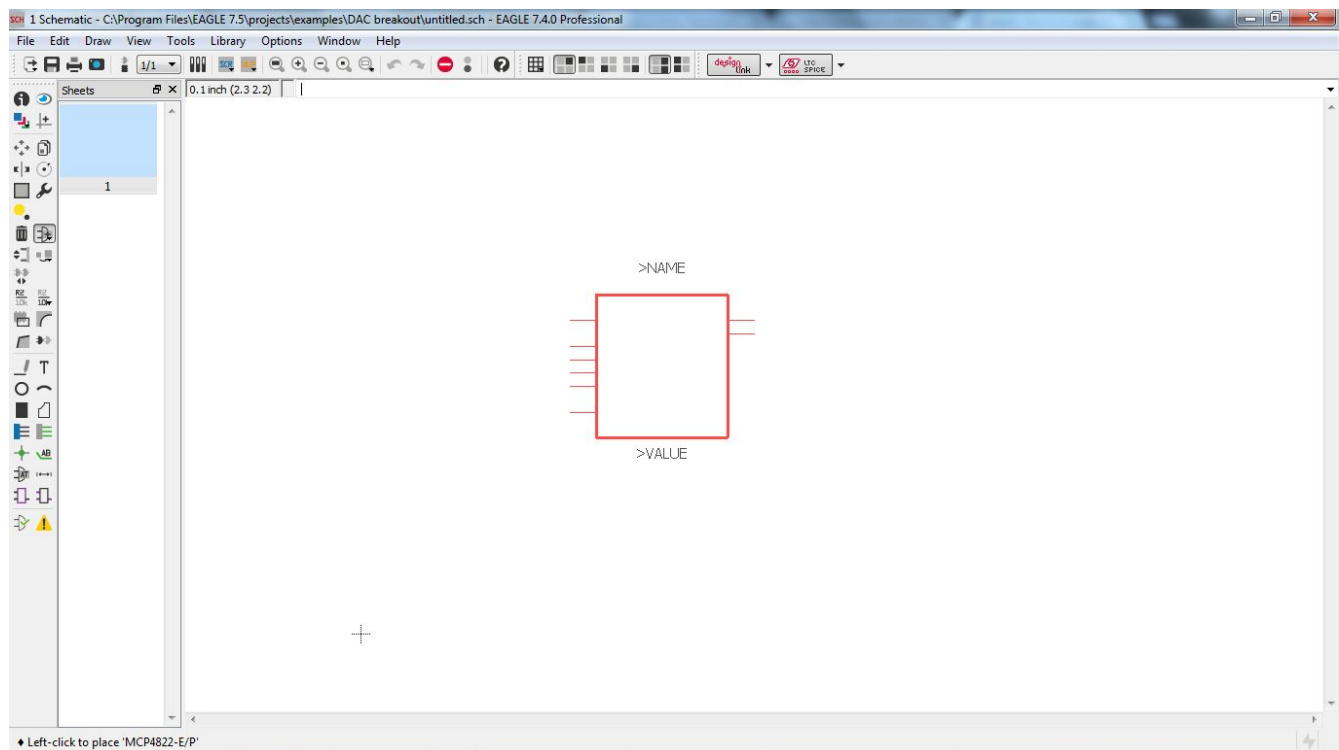


You'll likely return three results, as below:

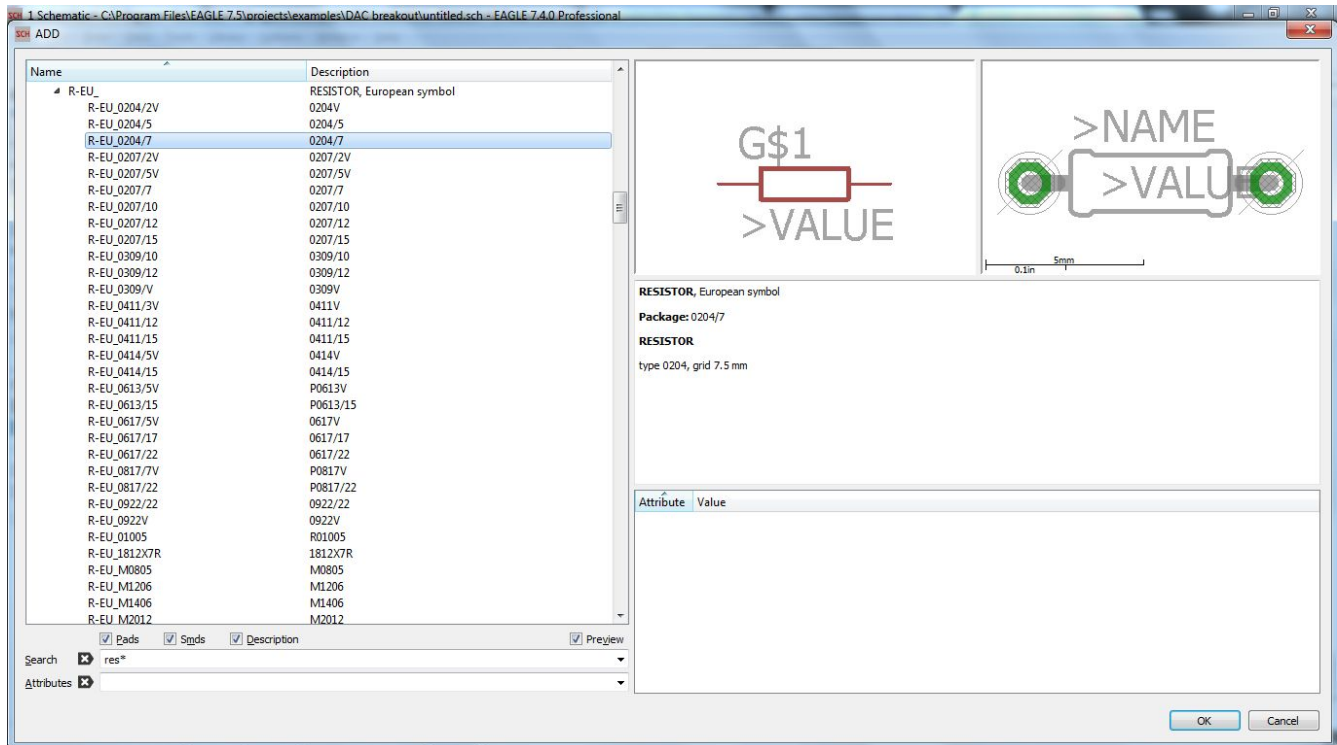


BIG OL' NOTE: because you're both modelling the *schematic* and the *PCB*, components will usually have associated with them both a **schematic symbol** and a **footprint**. For the schematic it's not a big deal which component you choose here; in fact, they should all probably look the same. But your footprint will be determined by which **package** you have selected. Take a look at the [datasheet for the MCP4822](#) - it comes in *three* different packages. The easiest one to work with is likely one you've encountered before: the PDIP. (It's the same as the Arduino Educato's processor chip.)

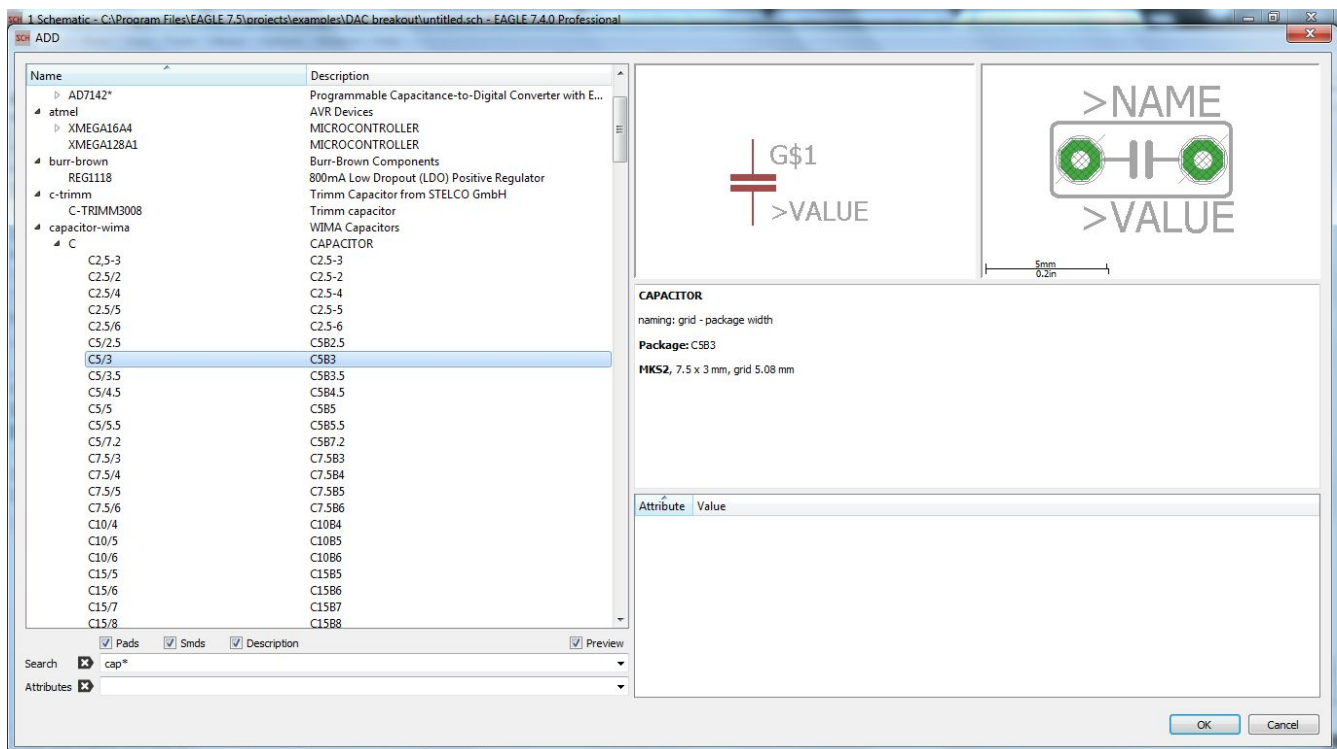
Find the PDIP package and add it. You'll be kicked back to the Schematic Capture window, where you can place it wherever you'd like. Once you've placed it, hit escape to be sent back to the Add Component window.



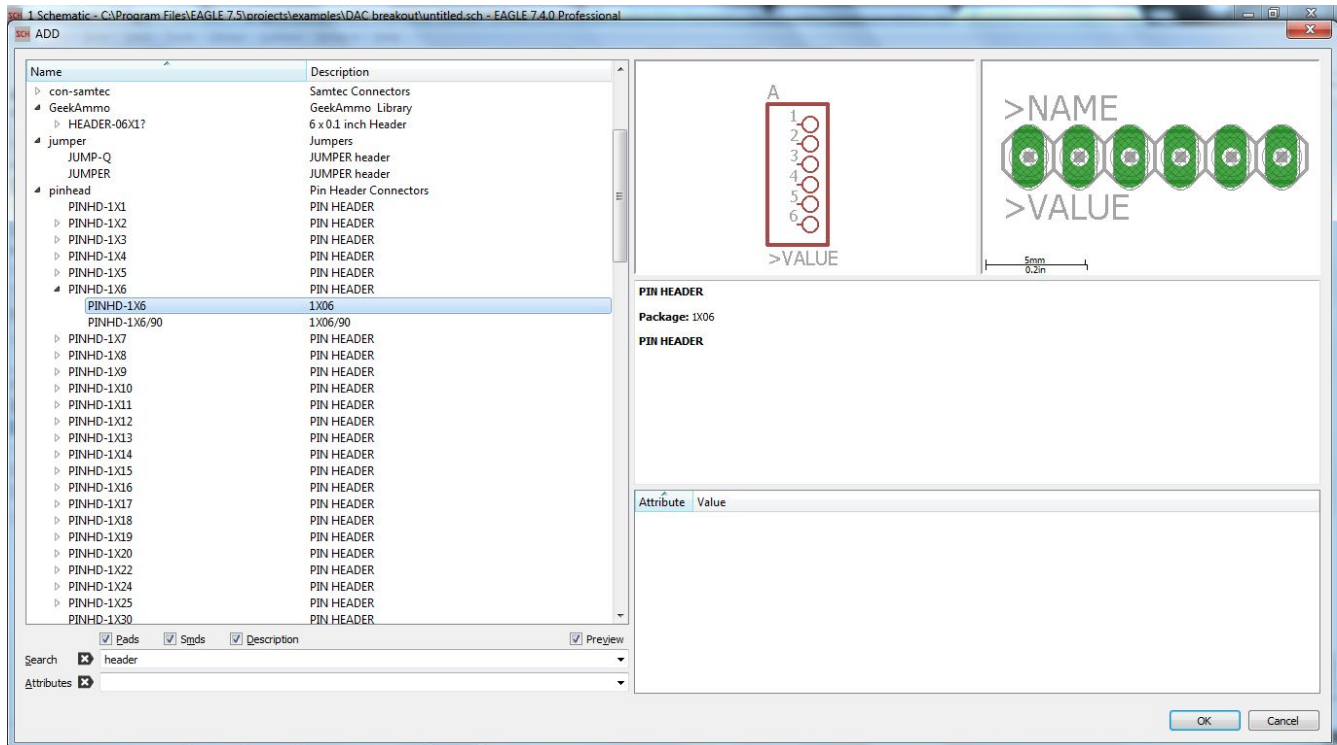
Now add two resistors (note that the actual resistance value doesn't matter right now; we'll add that later):



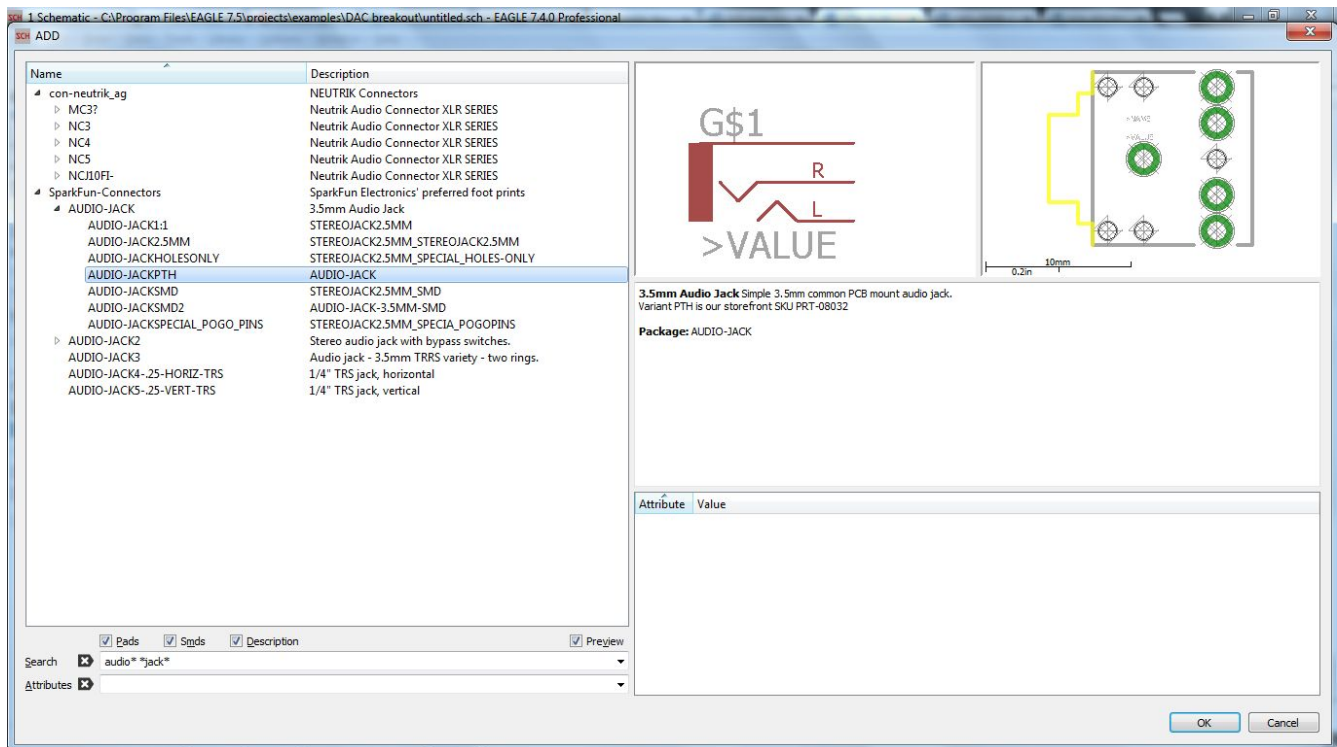
Then three capacitors:



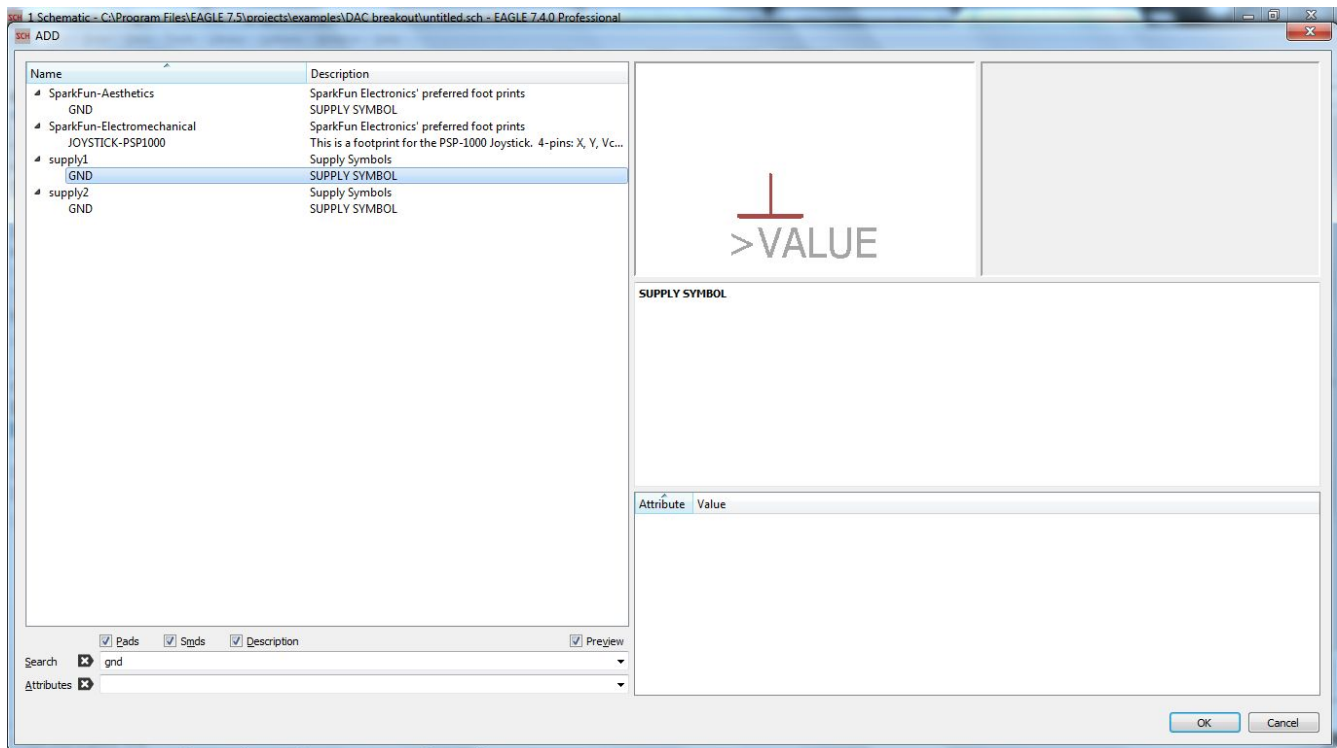
To connect to your microcontroller, we'll need a header strip:



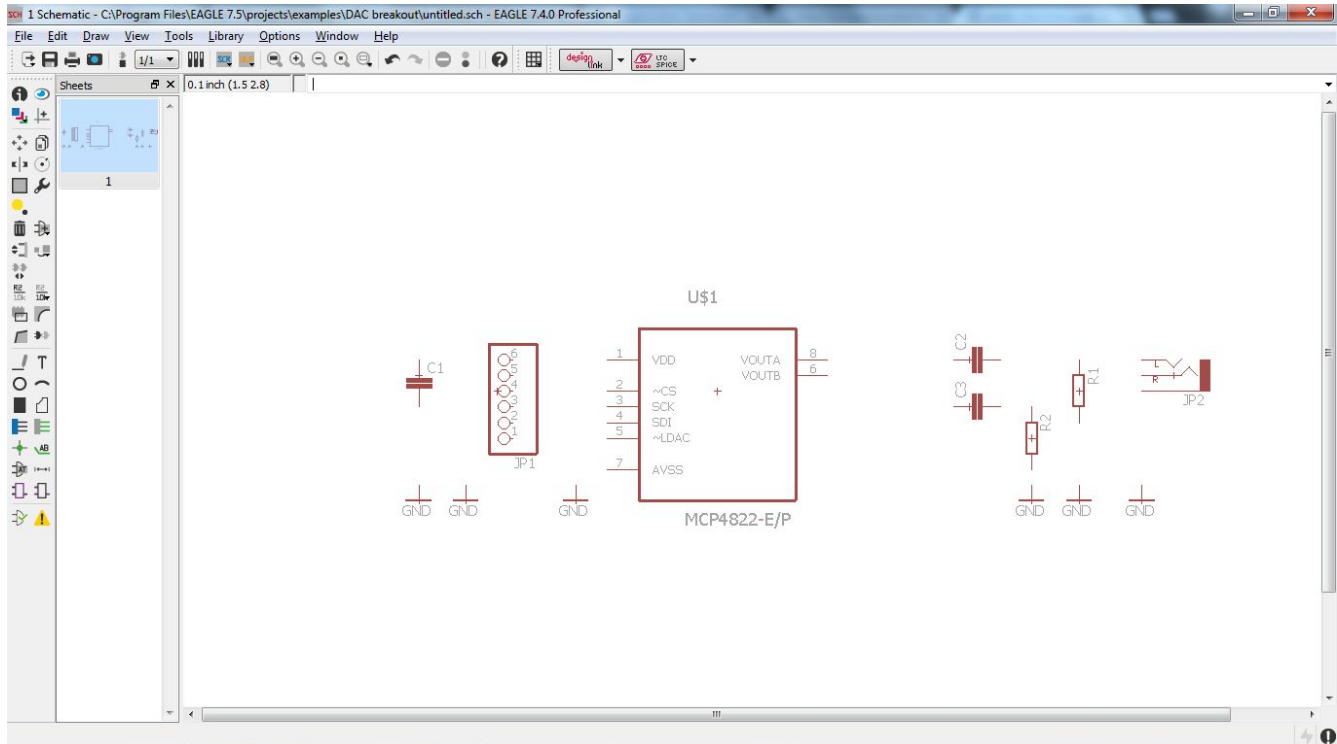
To be able to hear your analog signal (assuming you're going to be outputting audio), add an audio jack:



And one last (but very crucial) part of the circuit: ground. In order to not clutter the ensuing wiring, you can add several of them; the software knows to connect them all:

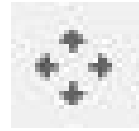


Your schematic should now look something like this:



Connect the components

If the position of the components isn't optimal, use the **Move tool** to position them to your liking.



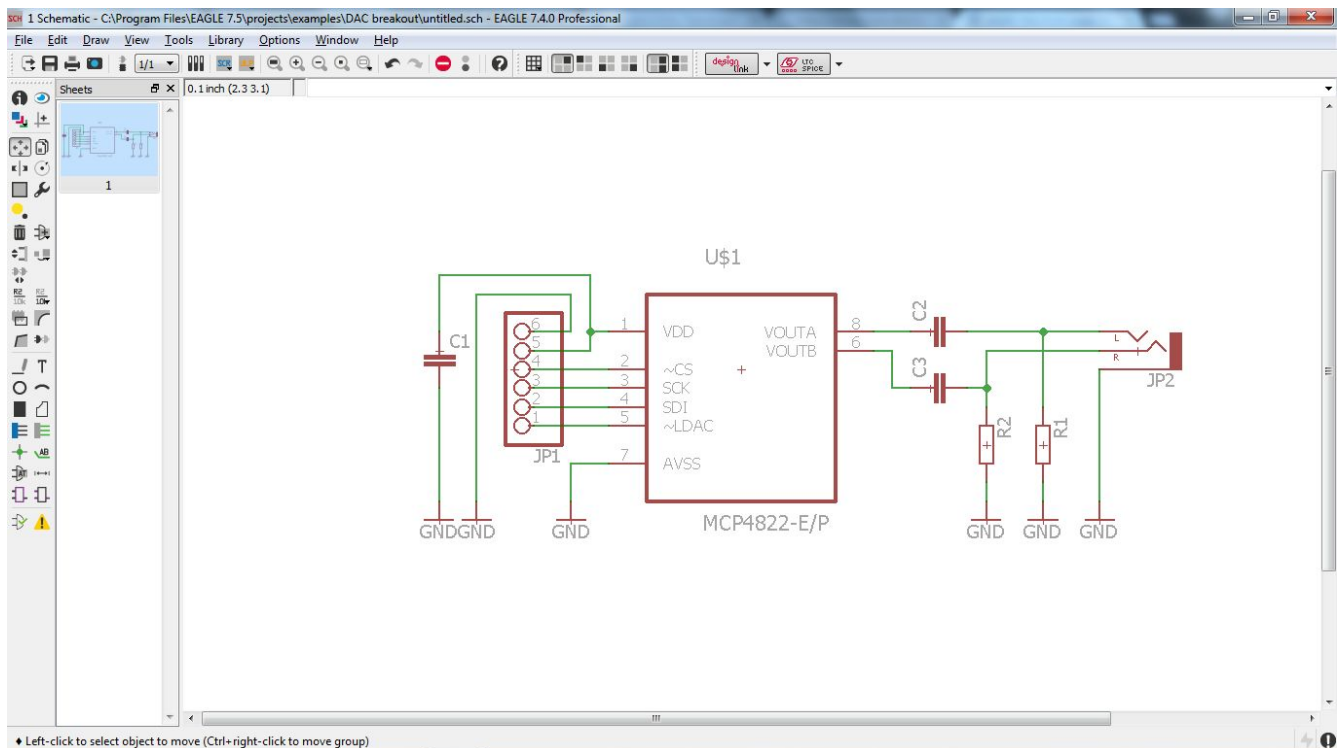
If components need to be rotated, use the **Rotate tool** (or right-click while moving using the Move tool).

Then use the **Net tool** to



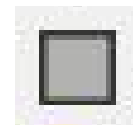
connect the pins!

Your schematic should ultimately look like the following:



BIG OL' NOTE: if you want to perform an action to more than one object at once, you'll need to first use the **Group tool**. For example, if I want to move three close components together:

- Choose the Group tool
- Drag a box around those components
- Select the Move tool
- Right click on the group and choose "Move: Group"



Give values to components

If you're going to fabricate this circuit, you'll need to give values to all the components! Some of the components already have values (e.g. MCP4822); some don't need values (e.g. header strip). But there are several which do. They are:

- C1: 1uF (the capacitor sitting by itself between the header strip and ground)
- C2: 0.22uF
- C3: 0.22uF
- R1: 100kohm
- R2: 100kohm

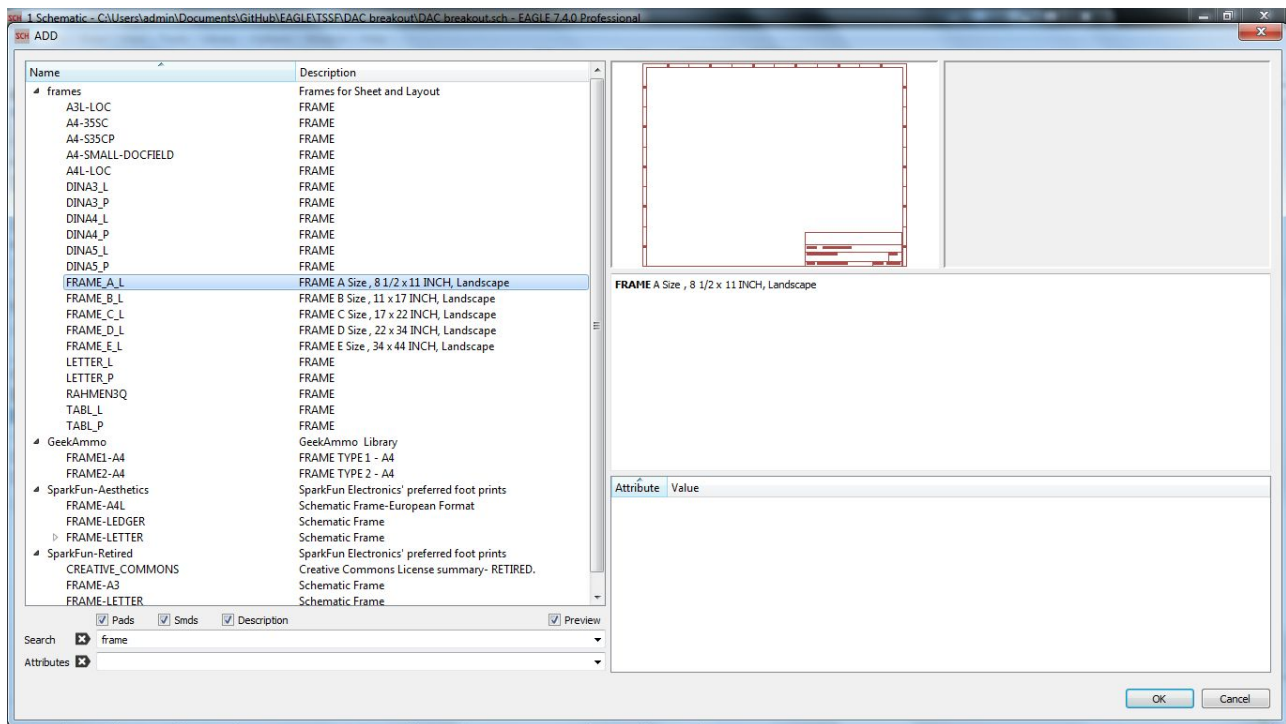


Use the Value tool to change their values.

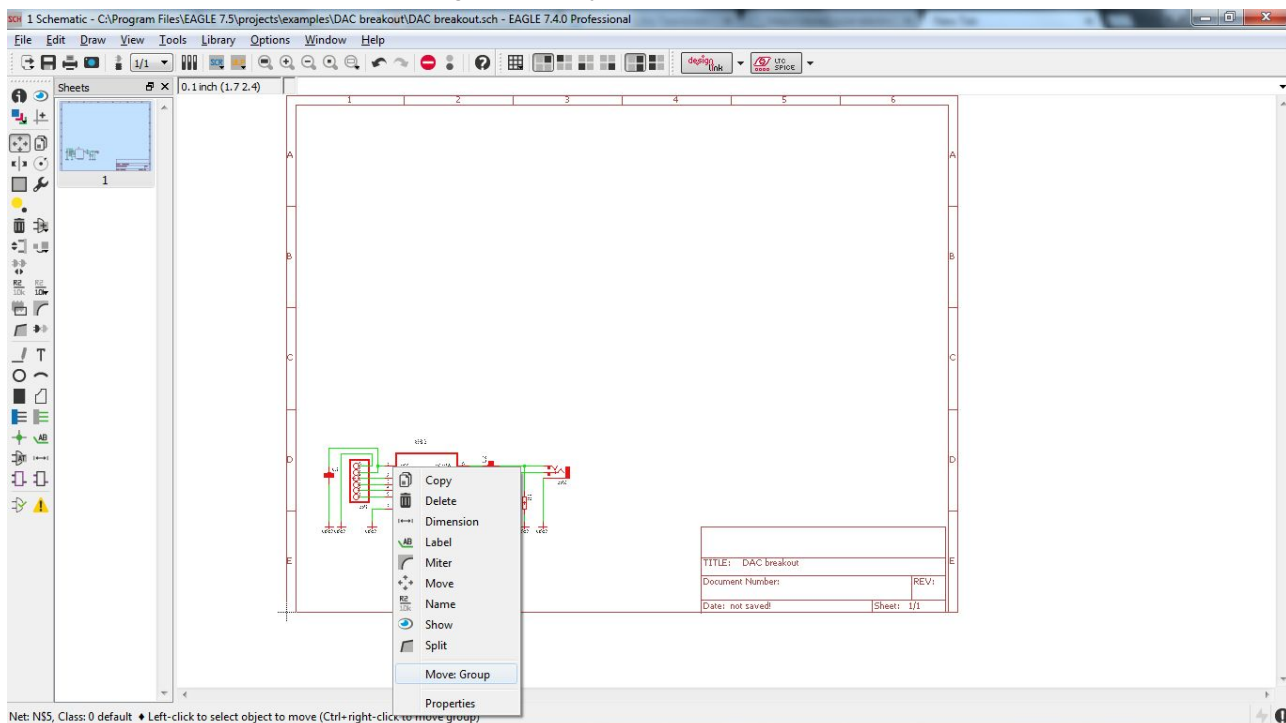
Don't worry if the numbering system for your capacitors and resistors is different; as long as the values are correct, the software has no problem lining them up.

Cleanup

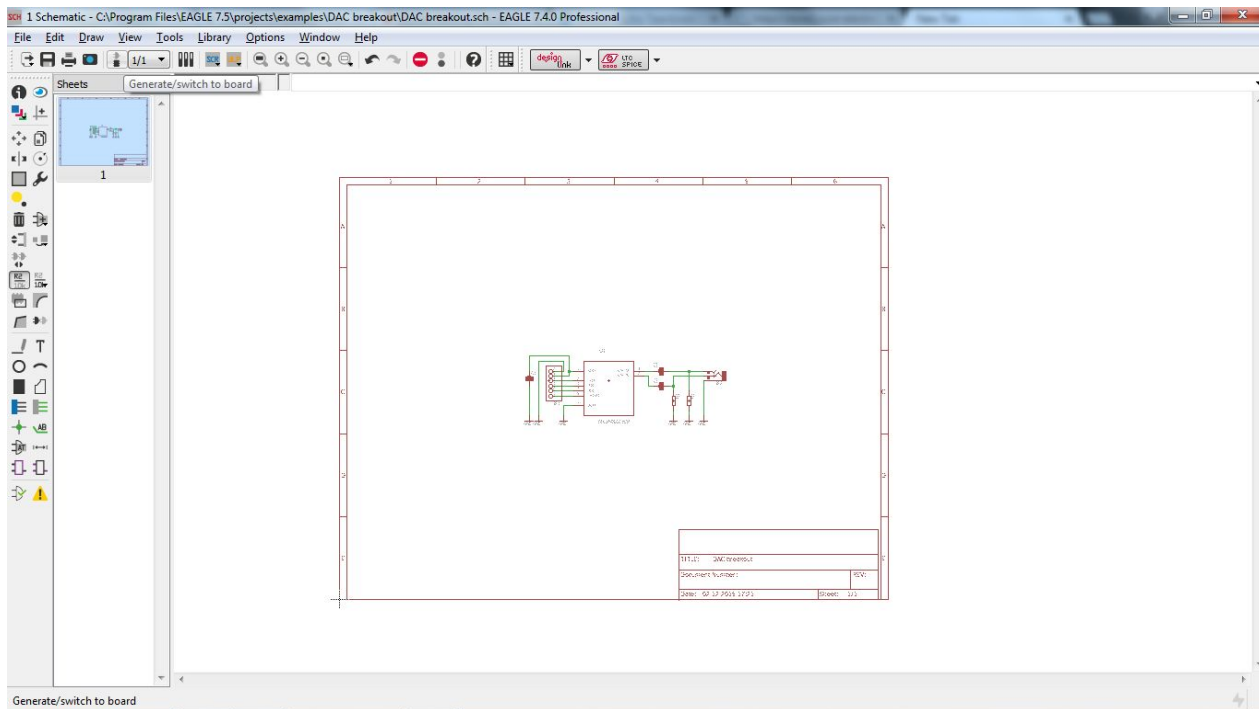
One last step before we design the PCB: in order to make this schematic as presentable as possible, you should clean up any stray connections, and try to minimize overlaps. Additionally, if this ever gets printed out, you'll want a frame around it which will help other engineers identify and understand your design. Add a frame:



Place the bottom left corner at the same spot as the “+” overlay. Then select every part of your circuit using the Selector tool, switch to the Move tool, right-click on your circuit and select “Group: Move”:



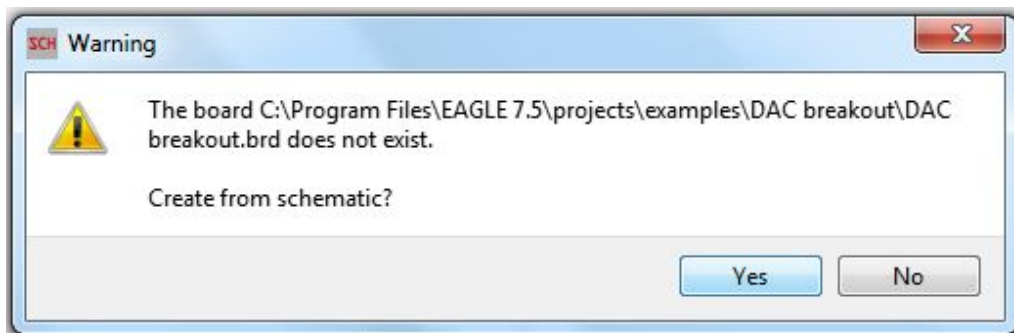
Move it all into the center, and then save. Congrats: you've designed your first circuit using an EDA!



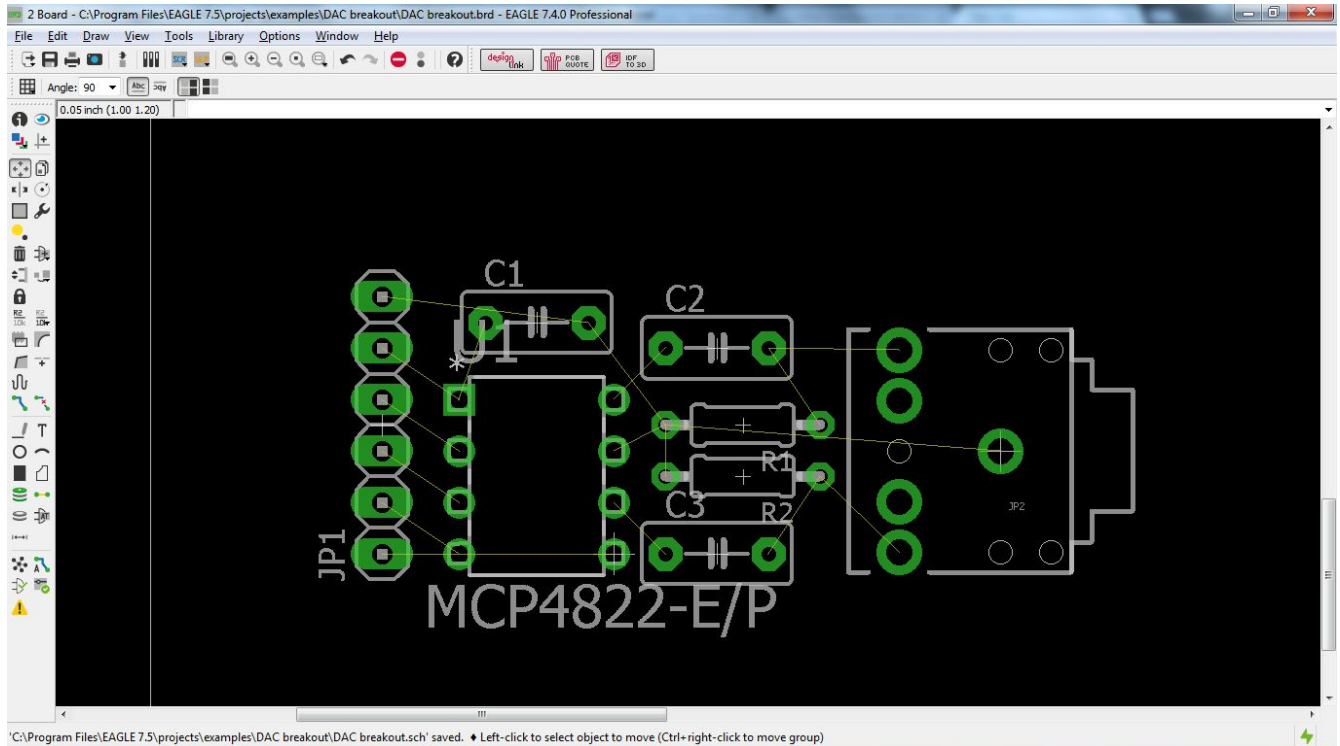
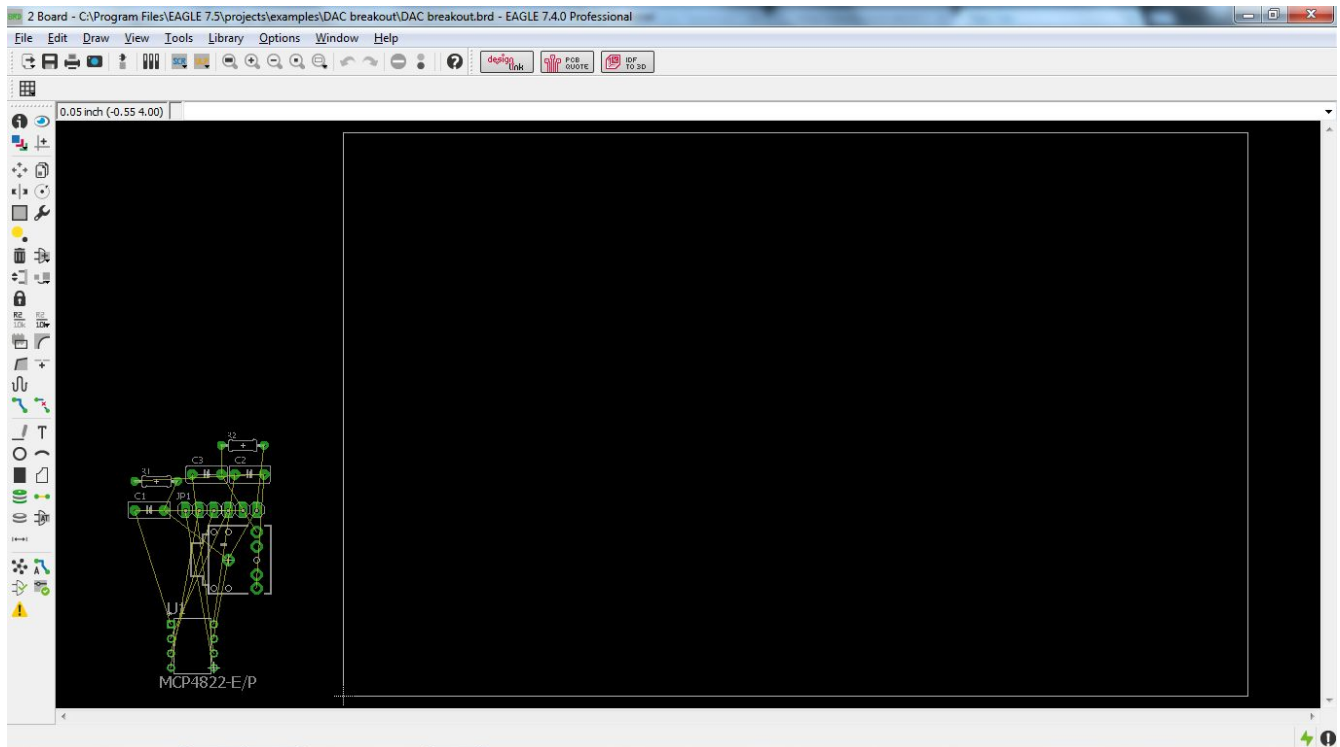
Guess what our next step is? Click the “Generate/switch to board” button on the top toolbar:

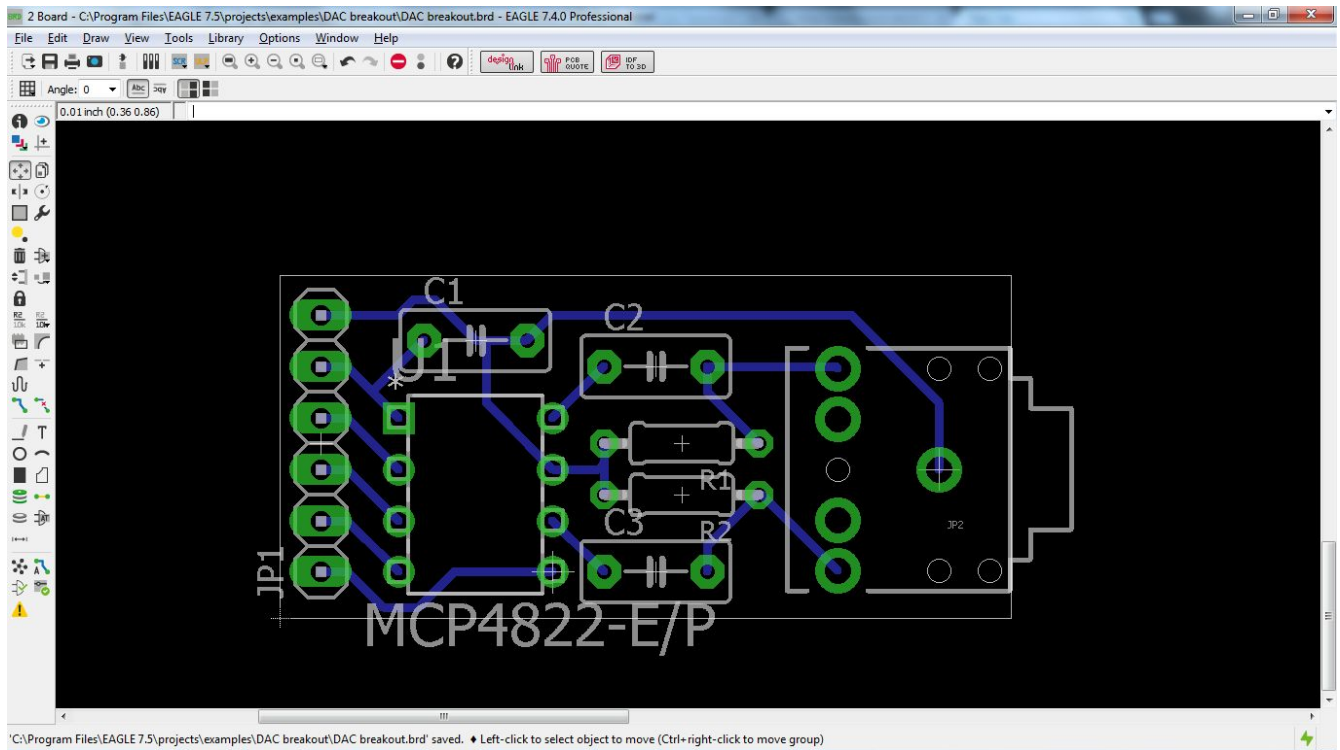
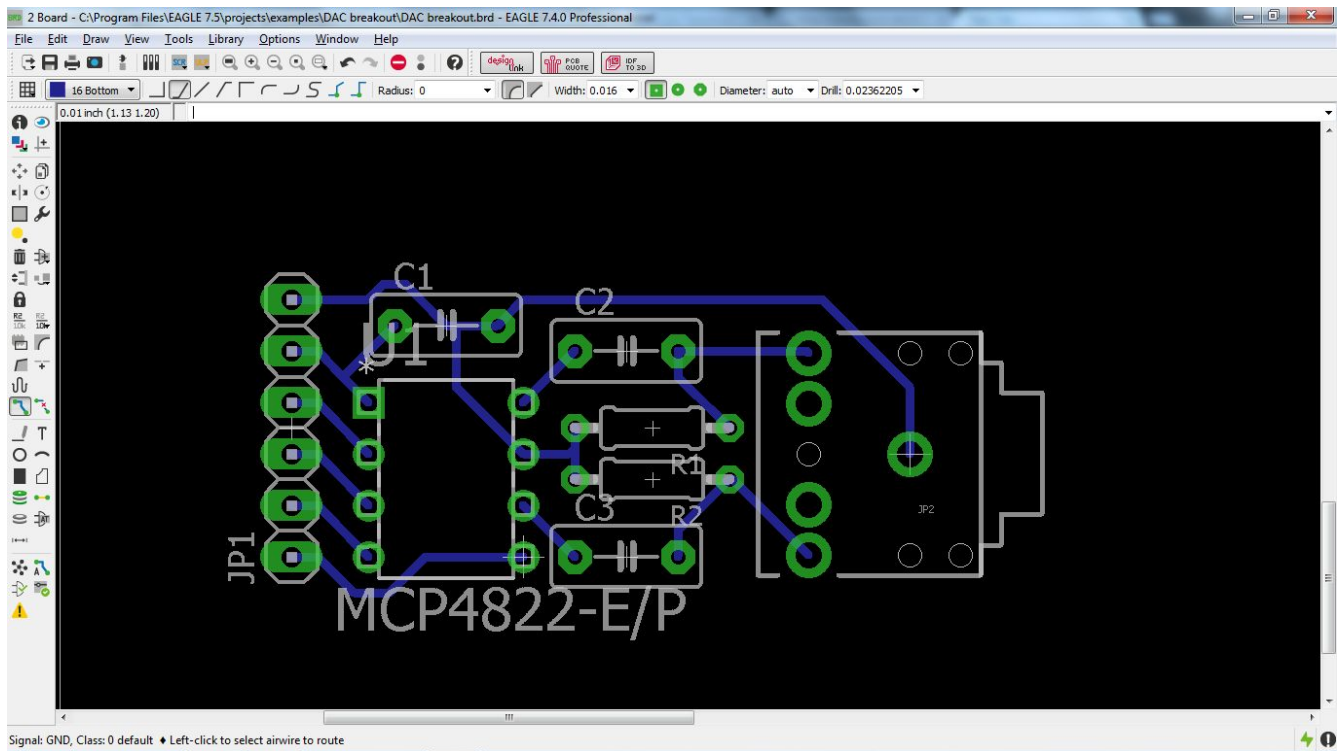


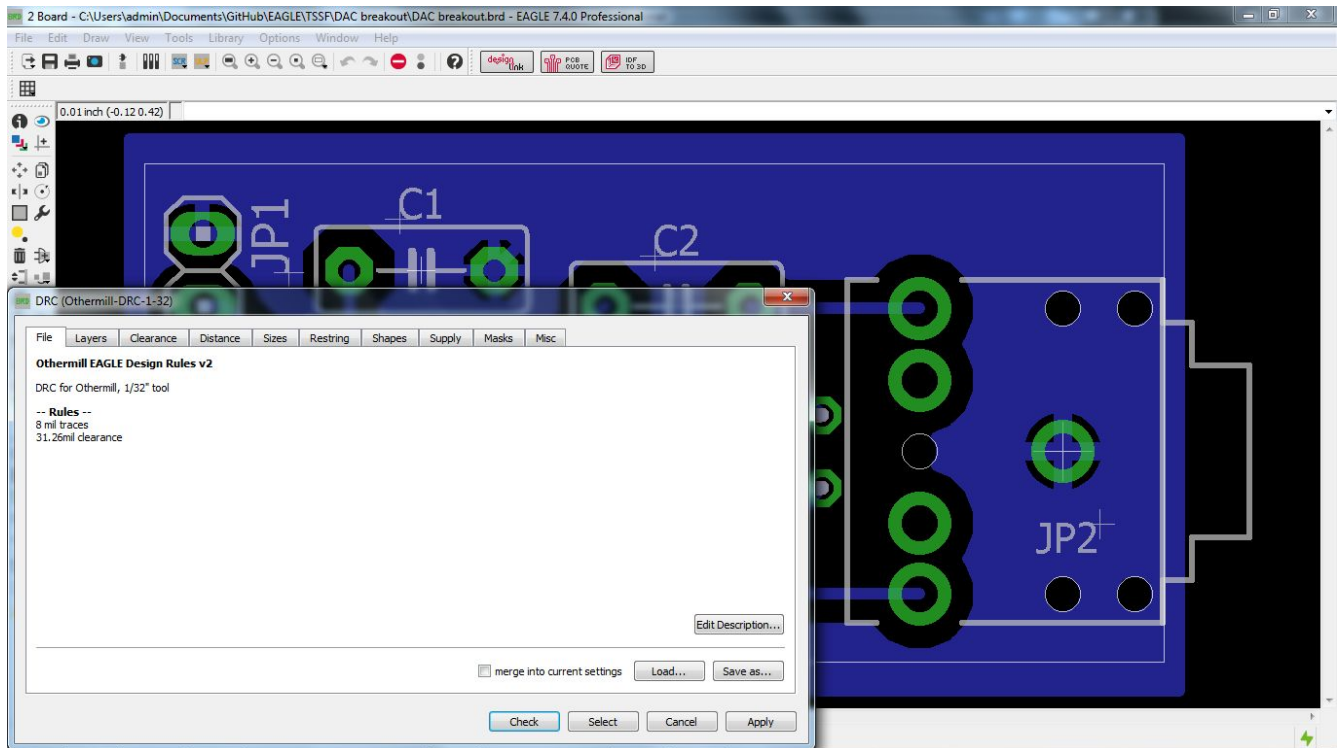
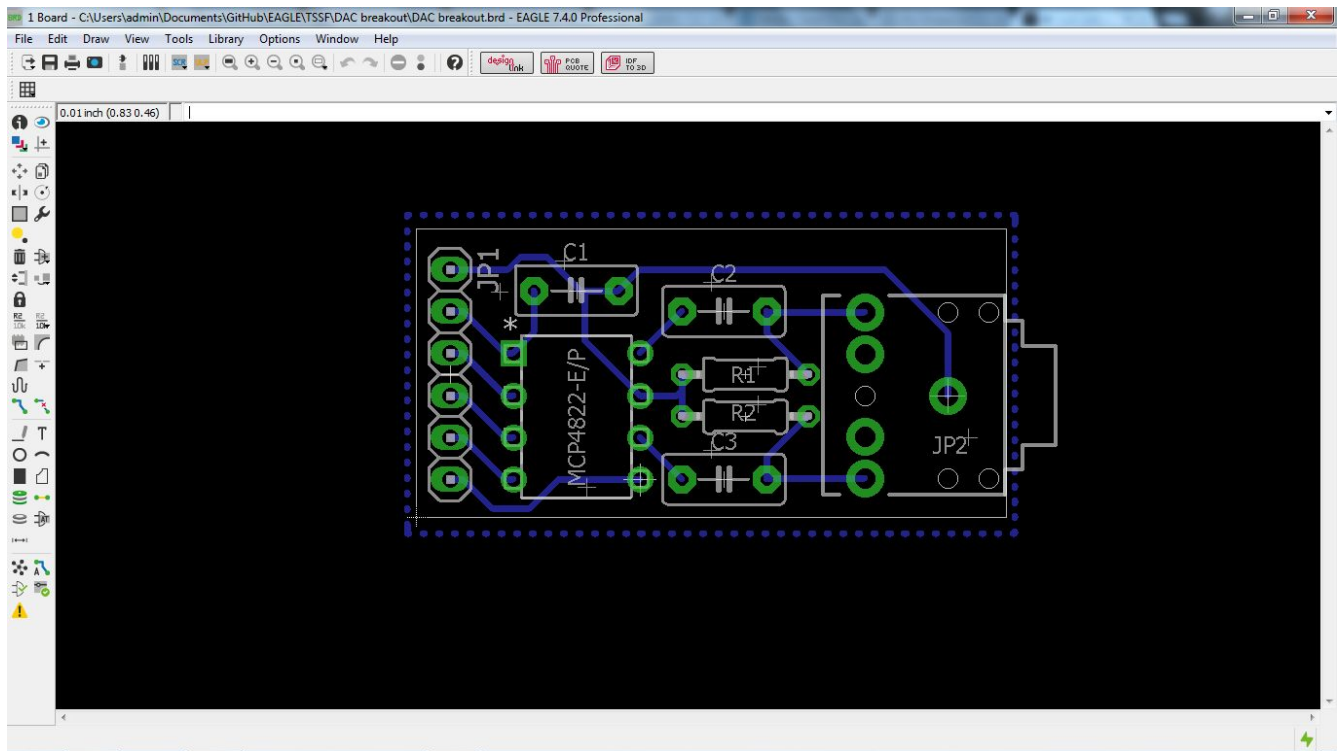
A warning will pop up. This is to be expected: yes we *do* want to create a board from the schematic!:

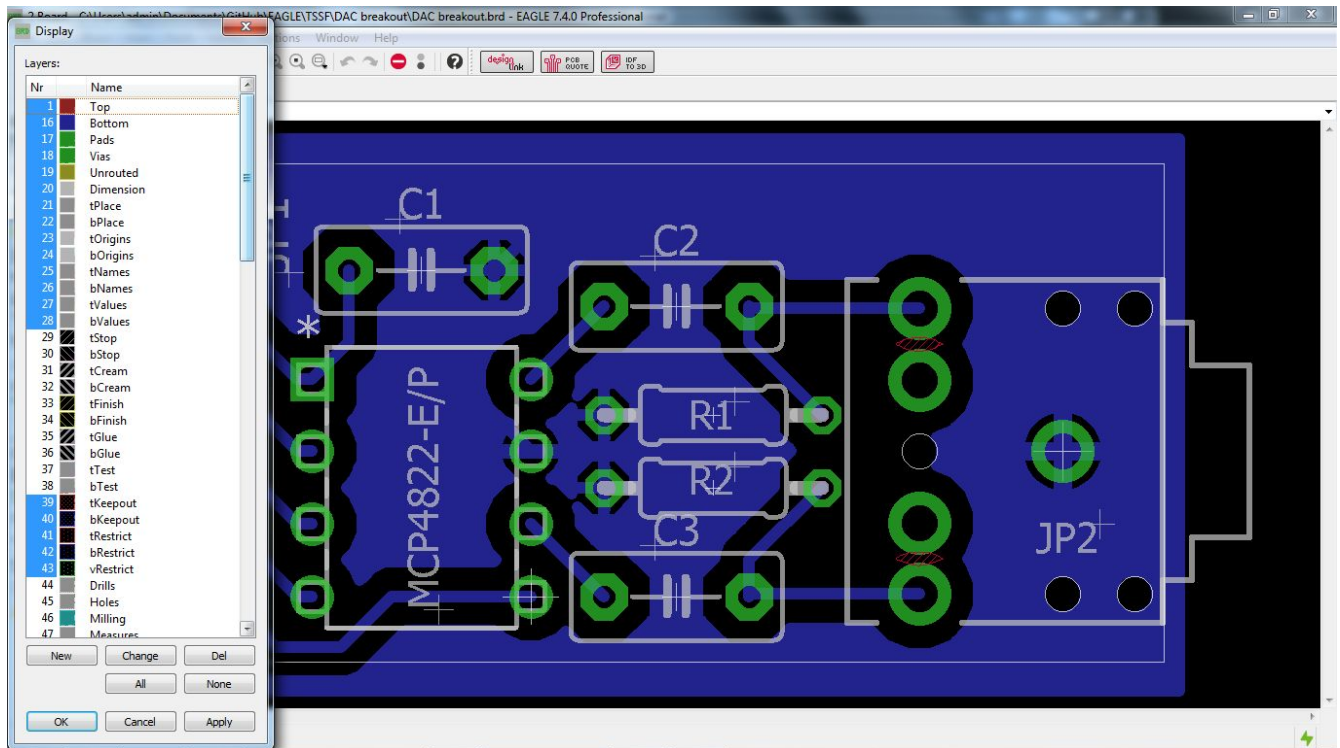
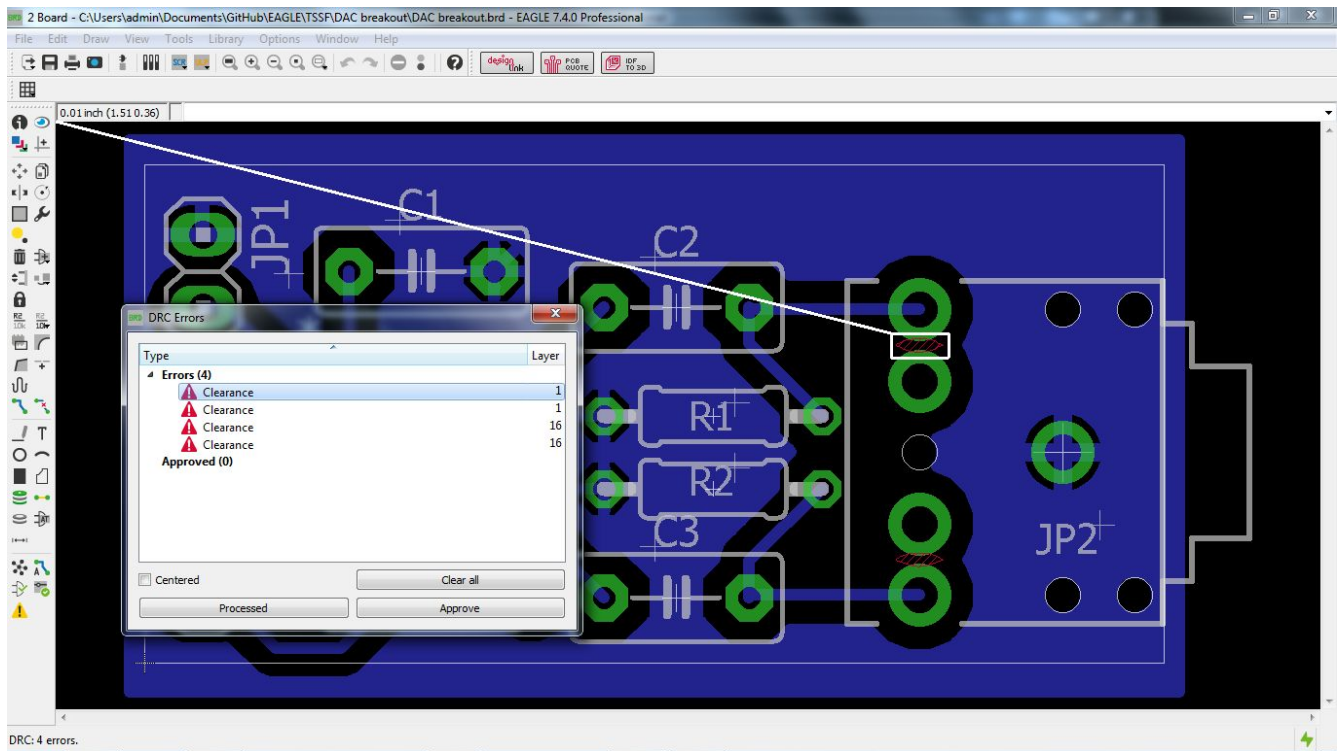


Part 3: Lay out the PCB









Part 4: Generate Gerbers

Appendix A: Extra Resources

[Element14 libraries](#) for Eagle

[MCP4822 library](#)

[make a new part!](#)

Sparkfun's [tutorial on EAGLE](#)

Eagle [layer descriptions](#)

[how to make the perfect PCB](#)

[BOM EX](#) and [BOM EX](#)

[How to design the perfect PCB part 2](#)

[Cutting mylar stencils](#) (probably broken)

[code for MCP4822](#)