



Physics 398DLP
Design Like a Physicist
Fall 2018

George Gollin
University of Illinois at Urbana-Champaign

Unit 2: Breadboarding and Schematic Capture

Unit 2: Breadboarding and schematic capture

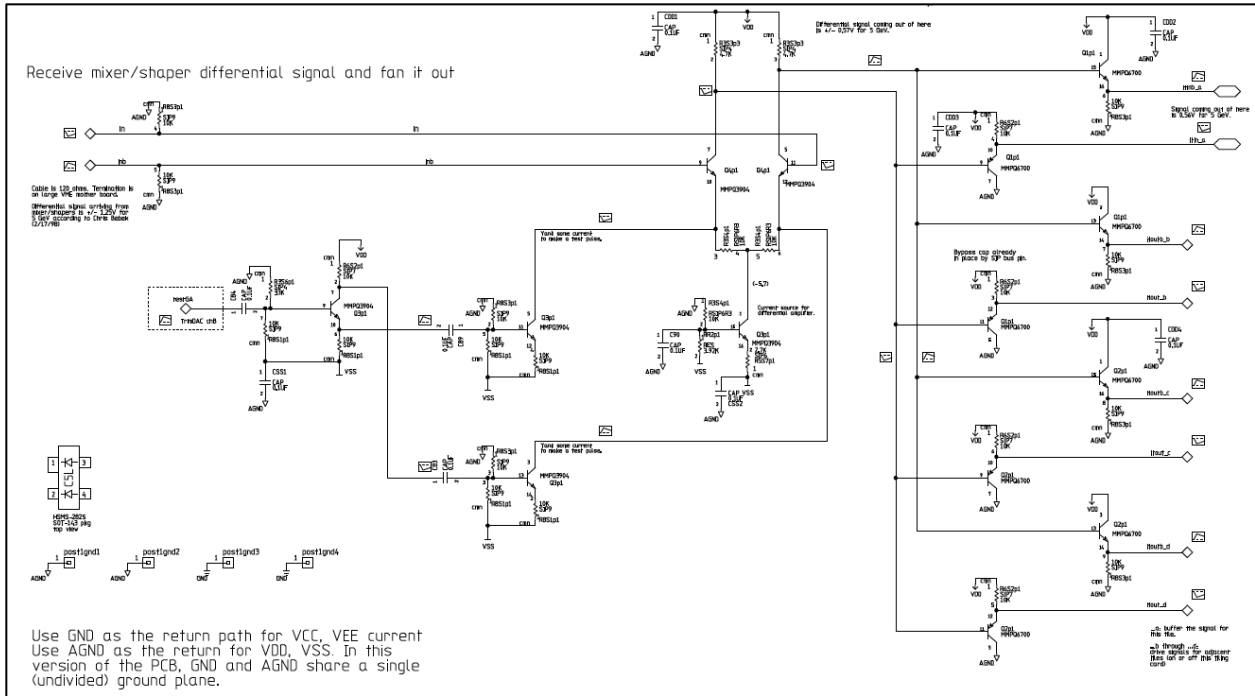
Goals this unit	3
Introduction: circuit schematics and PCB layouts	3
Autodesk EAGLE schematic capture	5
Capturing your circuit's schematic	10
Installing microSD storage	19
Installing the Real Time Clock	19
DAQ time.....	19
Volunteers needed.....	19
Post-class assignment.....	20

Goals this unit

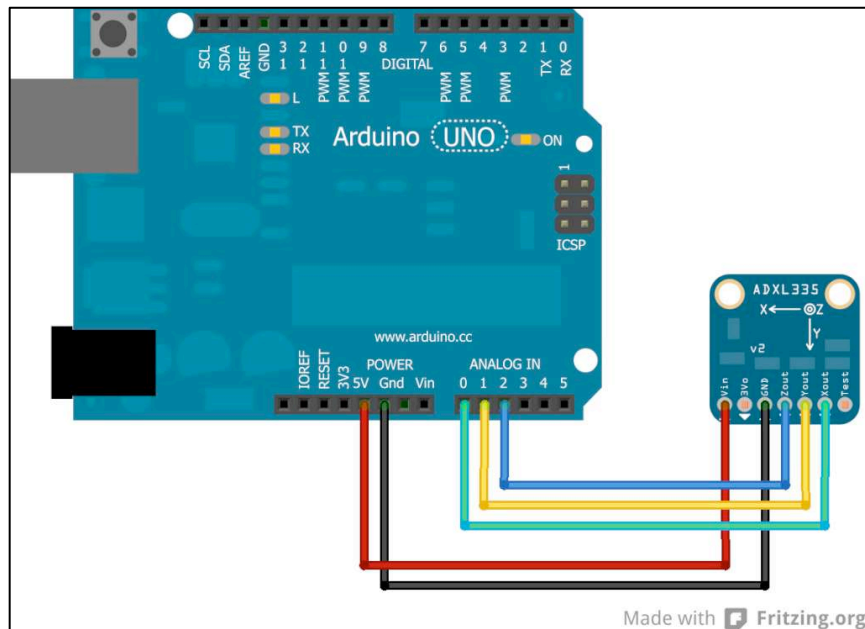
- Reports (1), (2), (3) that were assigned last week. (We'll scatter these throughout the four-hour class meeting, rather than hearing them sequentially.)
- Select volunteers for the following reports: (1) how the ADXL326 accelerometer works; (2) how an electret microphone works; (3) how the MLX90614 IR sensor works.
- Open a project in Eagle schematic capture/PCB and create a schematic that represents your breadboarded prototype as it currently exists, and reflects what you intend to do with it today. I will help with this, and show you how to work with the libraries of parts that you'll need. (Hands-on demo by GG, for sure: placing parts, connecting stuff, naming nets.)
- Install the microSD card reader and more sensors on your breadboard, making sure to leave room for the AA battery pack.
- Download software to communicate with your new sensors and test that they respond properly.
- Soldering clinic with Todd Moore.
- Class visitor: Professor Eric Benson (Art + Design). *Thinking about design.*

Introduction: circuit schematics and PCB layouts

One of the very first steps in designing a circuit is the creation of a schematic diagram that indicates which parts will be used in the circuit, and how they are wired to each other. Here's an example of (a small) part of a schematic for a mixed-signal board I designed for the CLEO III experiment at Cornell some years ago. (A mixed signal board holds both analog and digital signal lines.)

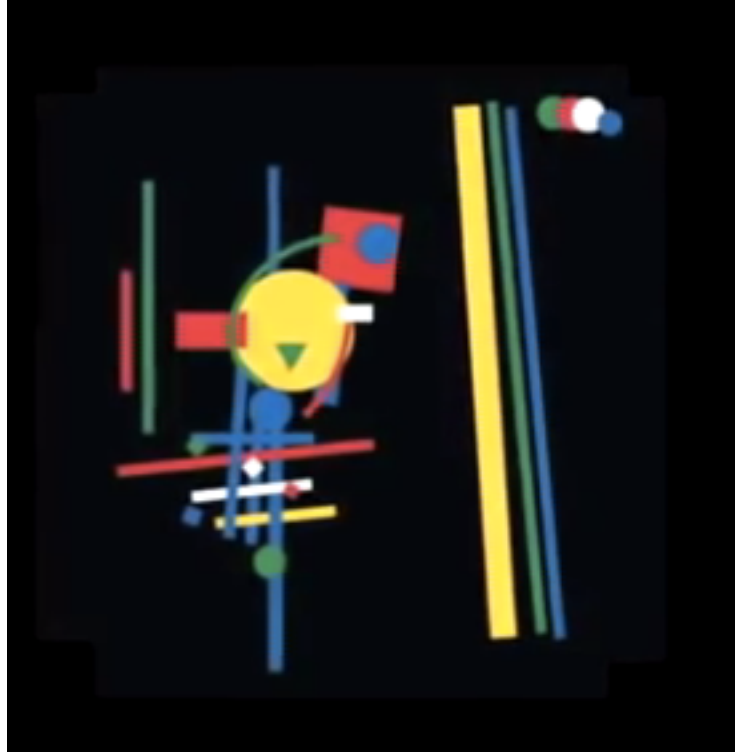


If you went to the Adafruit site to see how to wire up an ADXL326 accelerometer, you probably saw a representation of the circuit that looked like the following. That's not a schematic: though the illustration clearly represents the components and interconnections it is inefficient in its use of space.



You should have installed EAGLE's schematic capture and PCB layout tools last week. We'll work with the schematic tool this afternoon.

Did any of you play with Colorforms when you were little? It's a creative toy which includes a variety of thin vinyl shapes—circles, rectangles, and the like—that will adhere to the smooth, black surface that a child uses as a drafting environment.

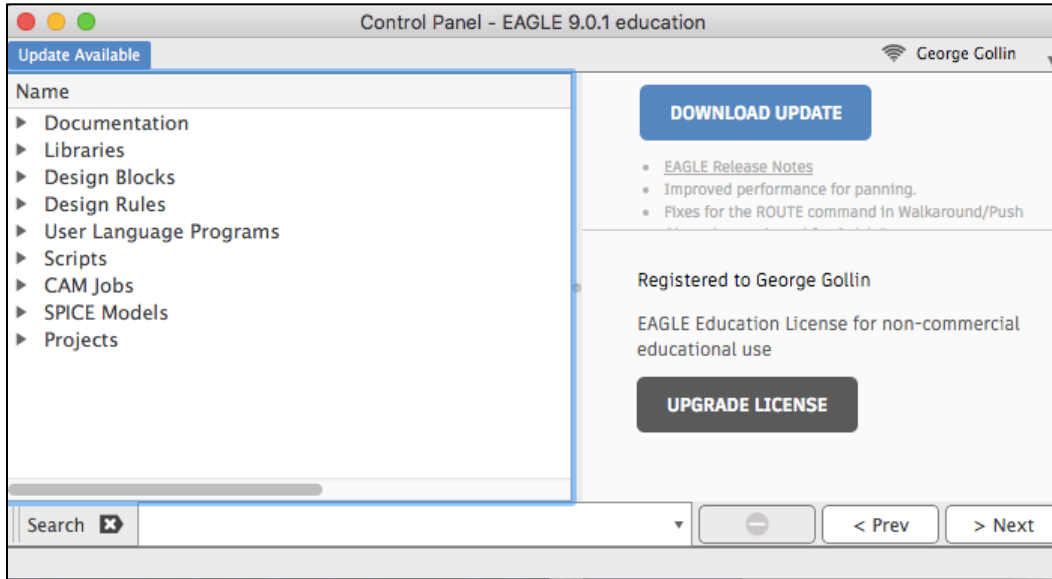


A schematic capture tool works in similar fashion: the designer selects parts from a library of components, places them on the design surface, and draws the interconnections between component pins.

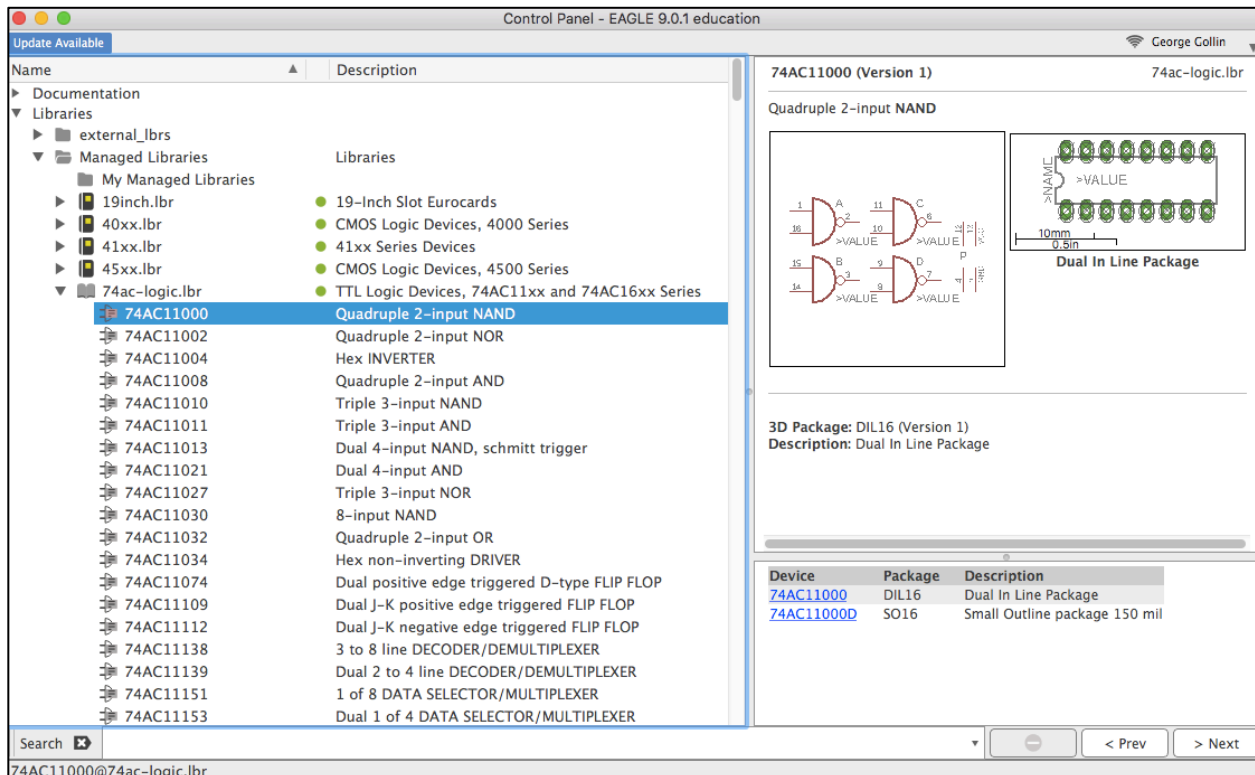
The data characterizing each part in the library include two distinct representations: one specifying just the pins to which connections are made, and another that includes geometric information. We use the first to define the topology of the circuit under construction, while we employ the other to decide where we can actually place components on a circuit board.

Autodesk EAGLE schematic capture

Let's create the schematic for your breadboard circuit now. Please open EAGLE. You should find that it starts by displaying the control panel.



Click on the arrowheads next to “Libraries,” “Managed Libraries,” and “74ac-logic.lbr,” then click once on “74AC11000.” The device displayed is a quad 2-input NAND.



“TTL” stands for “transistor-transistor logic.” It’s a logic family of integrated circuit chips in which 0 (or false) is represented by a voltage between ground and 0.8 V, while 1 (or true) is represented by a voltage between about 1.8 V and 5 V. The truth table for a NAND gate

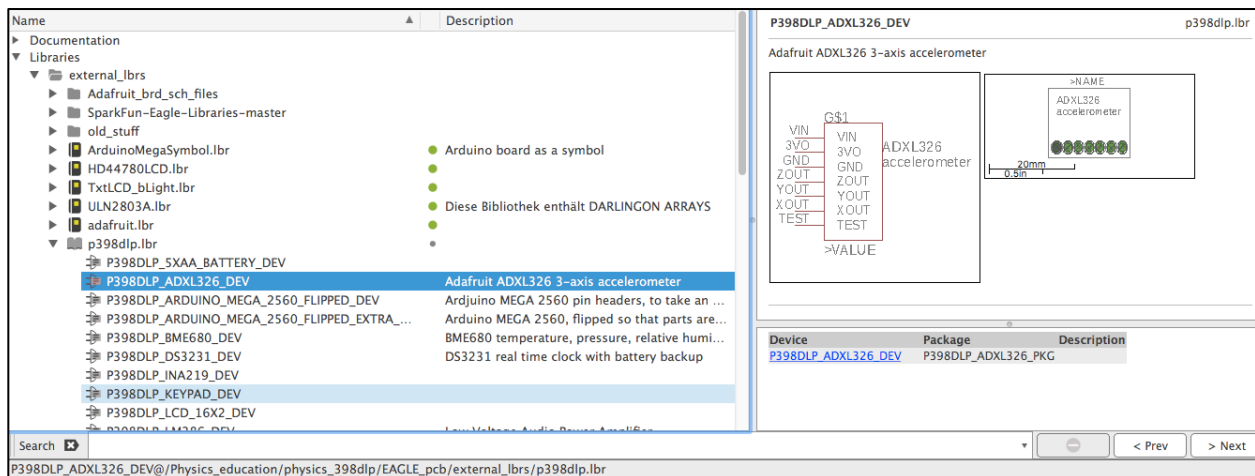
is simple: the gate's output is the logical complement of an AND of its inputs. The NAND's output is 1 if either or both of its inputs are 0. If both inputs are 1 the output is 0.

Take note of the two diagrams for the device. The one on the left is what EAGLE will offer to you to place on a schematic. The one on the right represents the actual footprint, including conducting pads to which you'll solder the chip's pins, for installation on a printed circuit board. Since this chip is available in two different packages, there are two choices listed in the Device/Package/Window box at the bottom. DIL16 is a 16-pin "dual inline package" while SO16 is a 16-pin "small outline" package.

We're not actually going to use this part, but I thought it would make a good example.

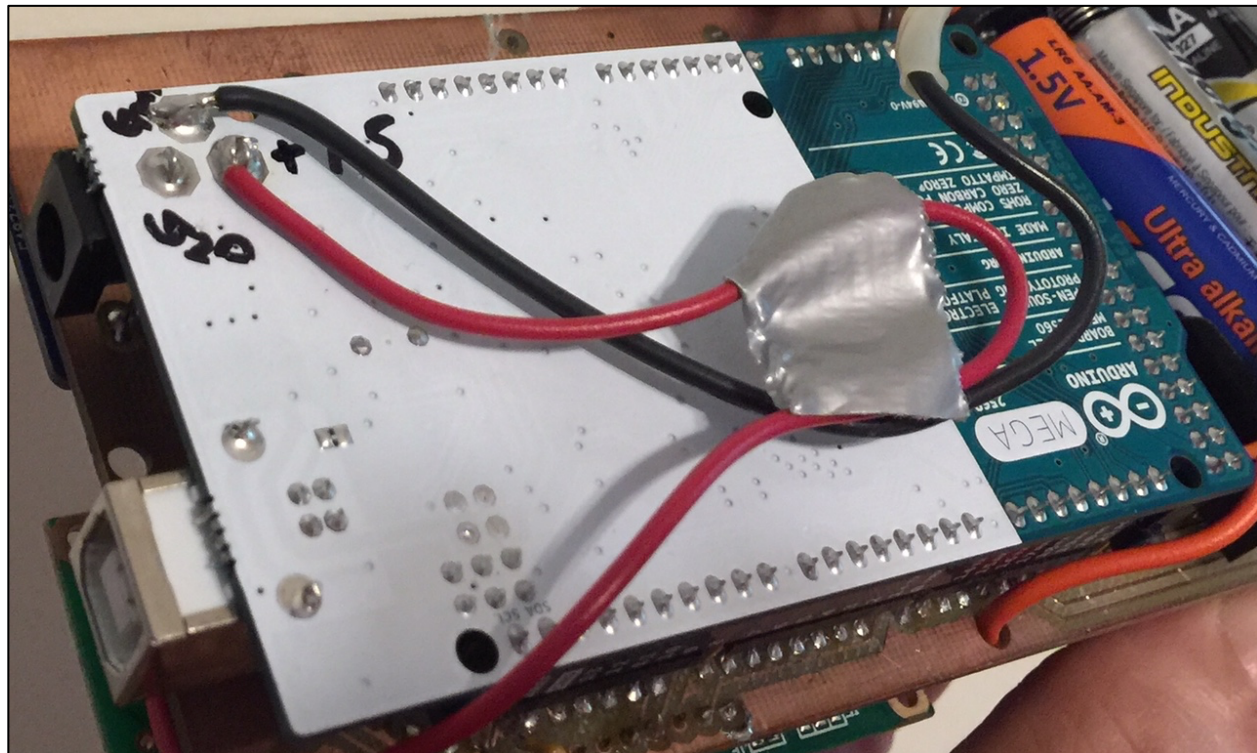
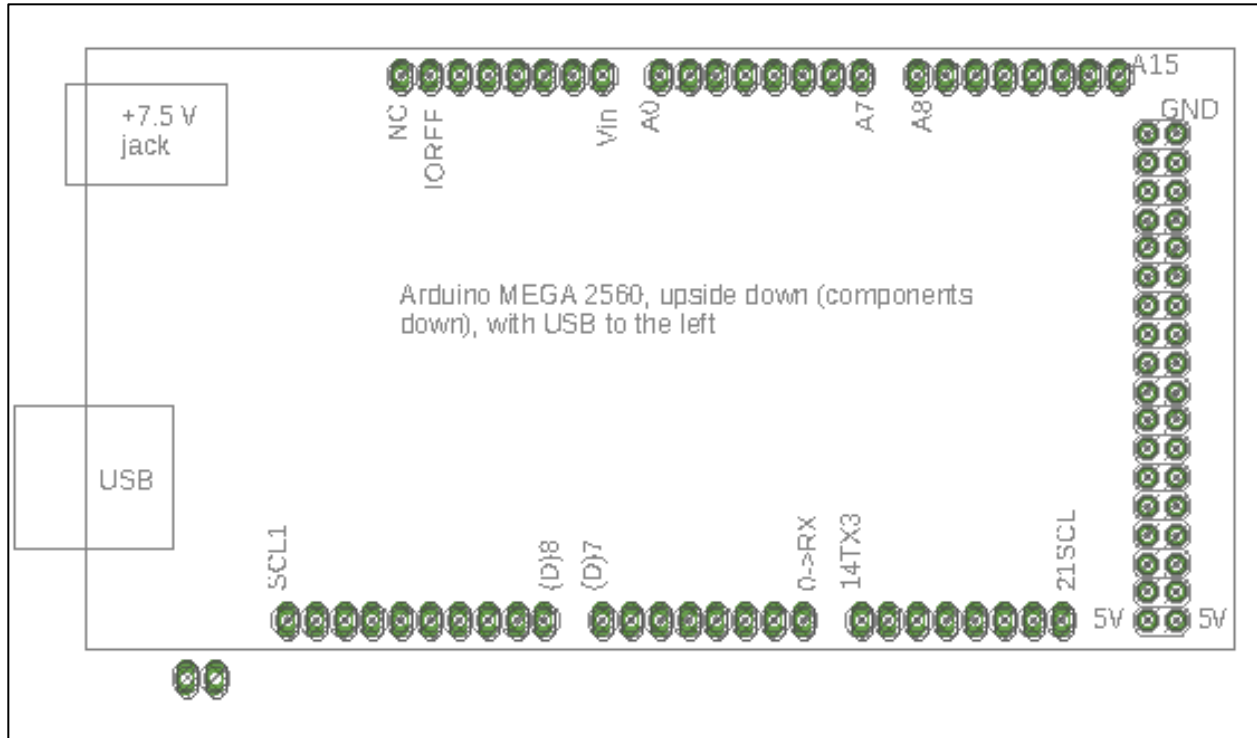
If you haven't already done this, please create a sensibly-named folder into which you will store all your EAGLE stuff. (Mine is named Eagle_pcb.) Inside this folder create another folder named external_lbrs. Go to the course web site and download https://courses.physics.illinois.edu/phys398dlp/fa2018/Eagle_pcb/external_lbrs/p398dlp_eagle_libraries.zip, then unpack it and move the files it contains into your external_lbrs folder. You may need to go to View → Refresh to update EAGLE's awareness of the libraries it has at its disposal.

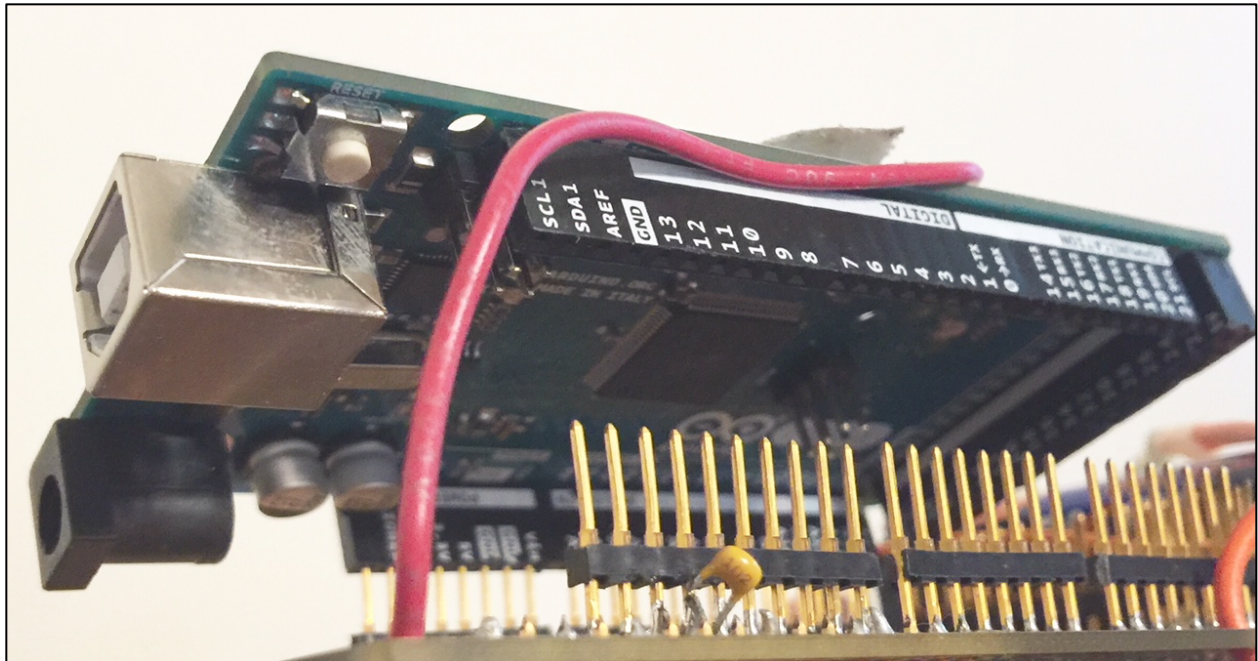
In the EAGLE control panel window look inside external_lbrs/p398dlp.lbr. You'll see a number of parts that I have created for your convenience. Here, for example, is how the ADXL326 accelerometer presents.



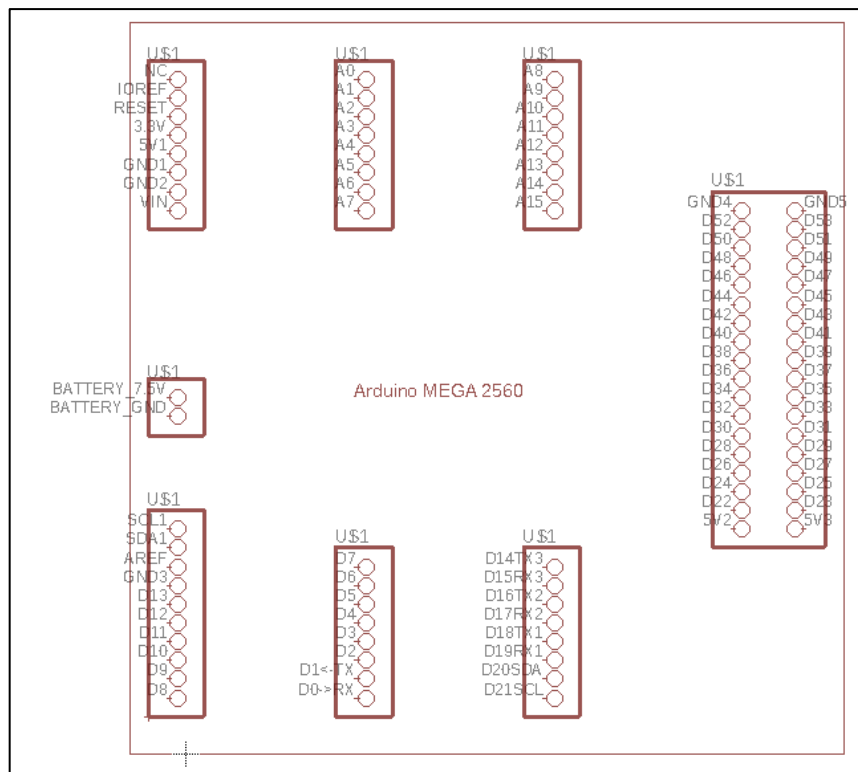
I've also created a part called "P398DLP_ARDUINO_MEGA_2560_FLIPPED_DEV." It's an Arduino Mega 2560, but upside down so that it can be plugged into properly positioned pin headers on a circuit board.

To help you visualize what I mean, here's a view of the EAGLE symbol that is used when designing a printed circuit board, as well as a pair of photographs of an Arduino installed on a prototype data logger. To orient yourselves, take note of the positions of the power jack and USB connector.





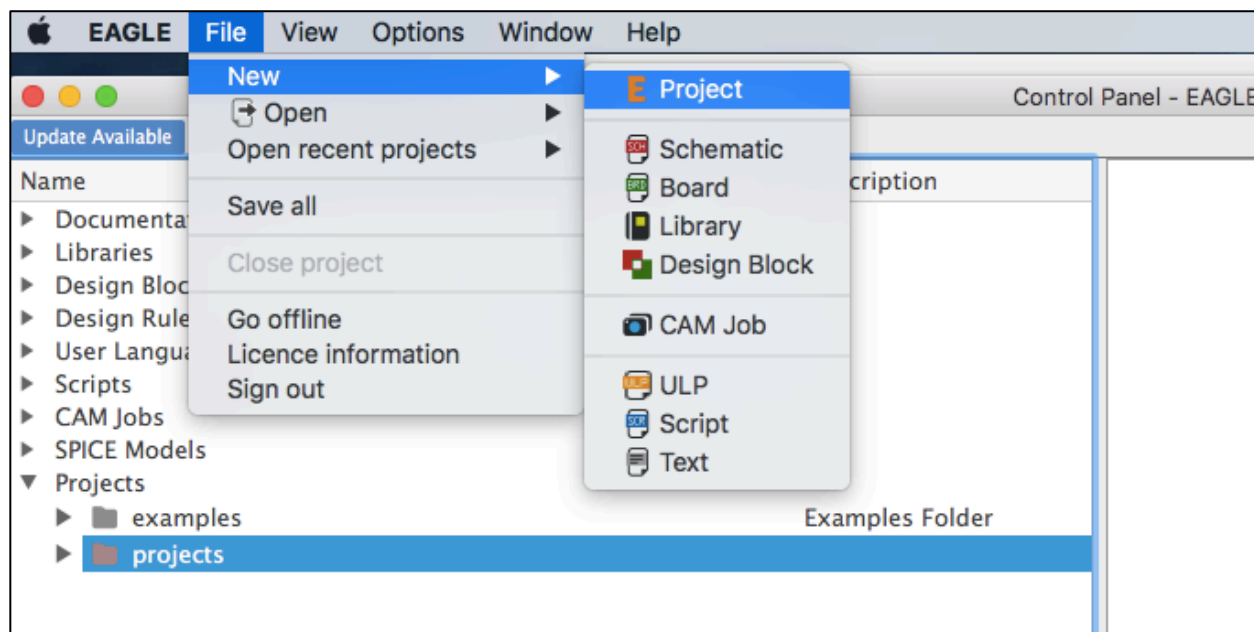
Note that this isn't the same as the representation of the Arduino that you'll use when defining the schematic if your circuit: there's no need for the placement of pins on the schematic symbol to bear any resemblance to the positions of the pins on the actual device. Here's what the symbol looks like after it's placed on a schematic.



Capturing your circuit's schematic

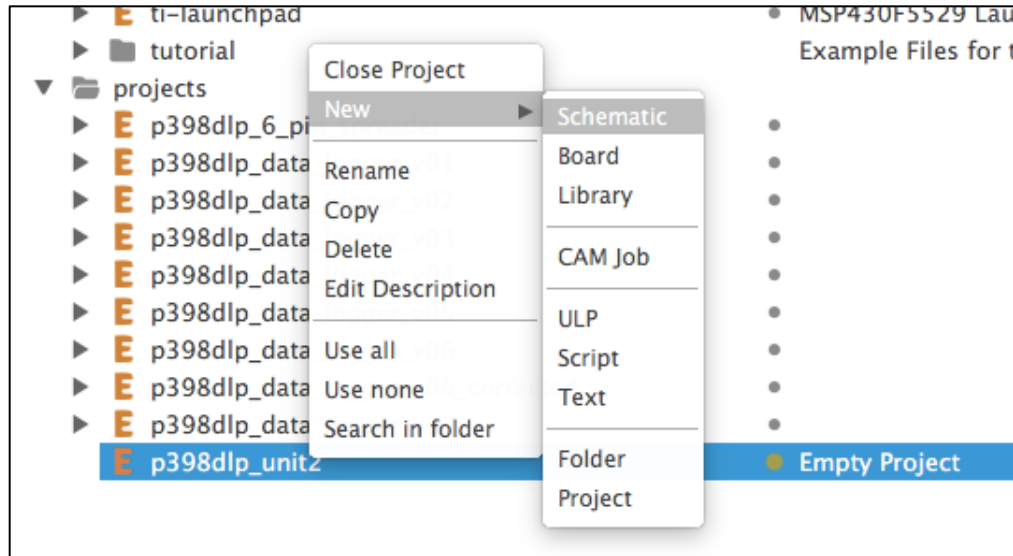
By now you should have an Arduino, an ADXL326, and a BME680 on your breadboard. Please make sure that you've wired your device as described in the unit 1 material.

In the same folder that you've used as the home for your external_lbrs folder, create a new folder named "projects." In EAGLE, go to File → New → Project:

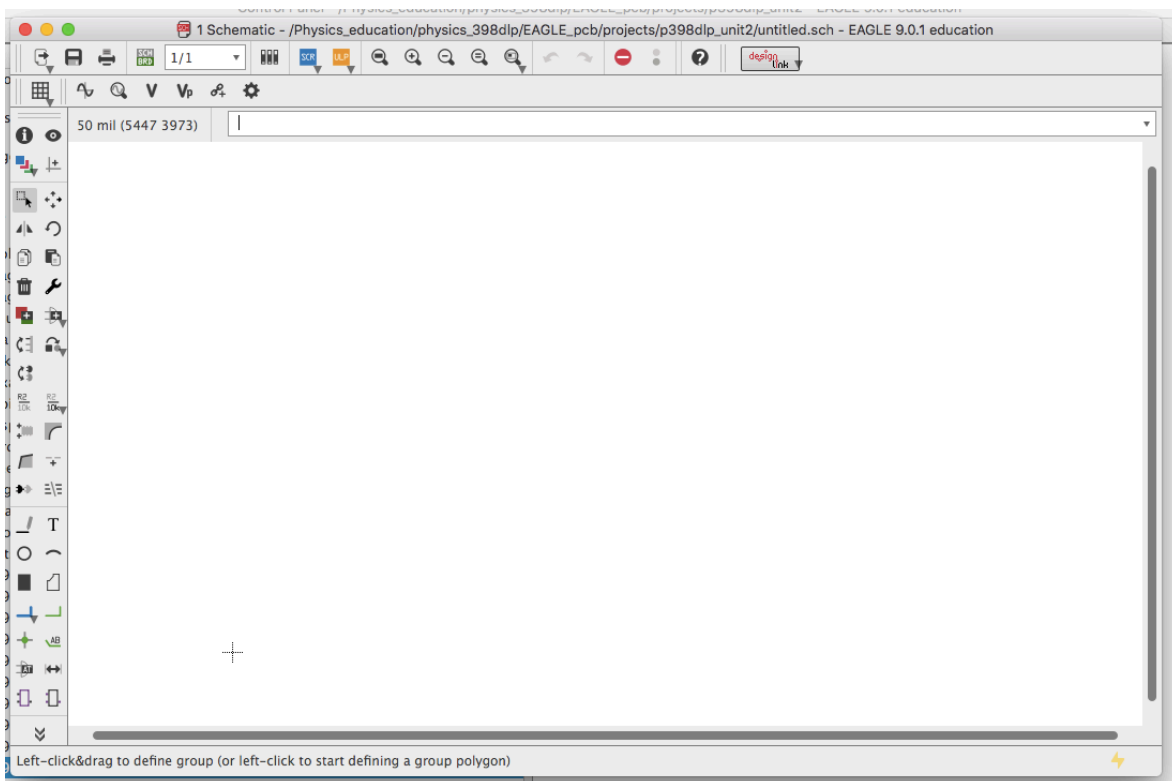


Give your project a sensible name, of course. If the project doesn't land in your projects folder you can drag it over to that folder from wherever EAGLE creates it. I am calling my project "p398dlp_unit2."

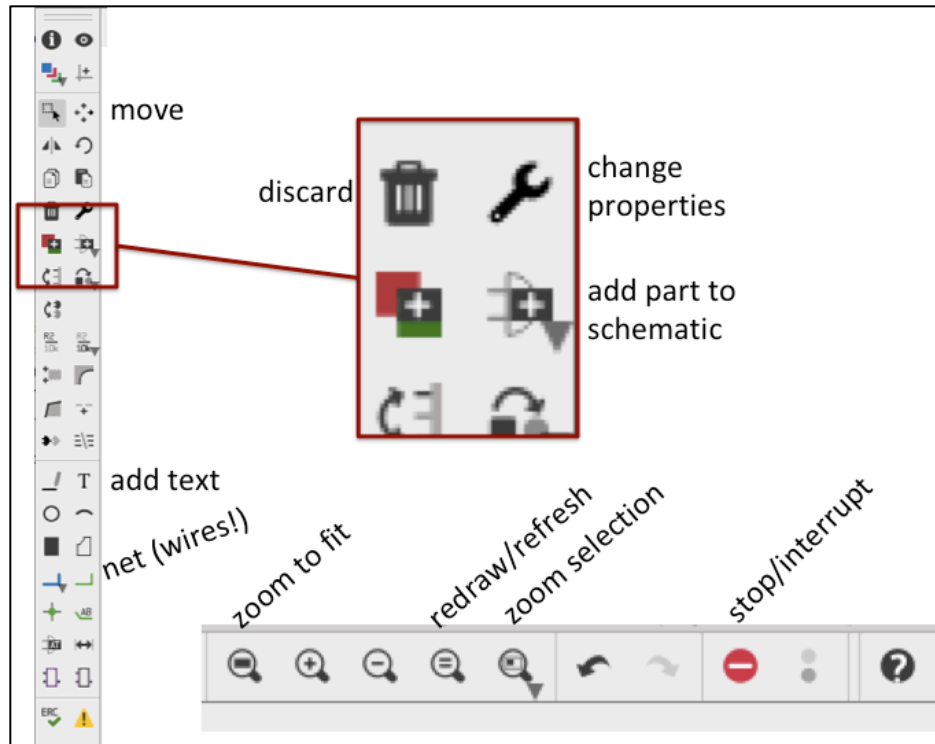
Right click on the project and go to New → Schematic.



The schematic capture window will open. It should look something like this:

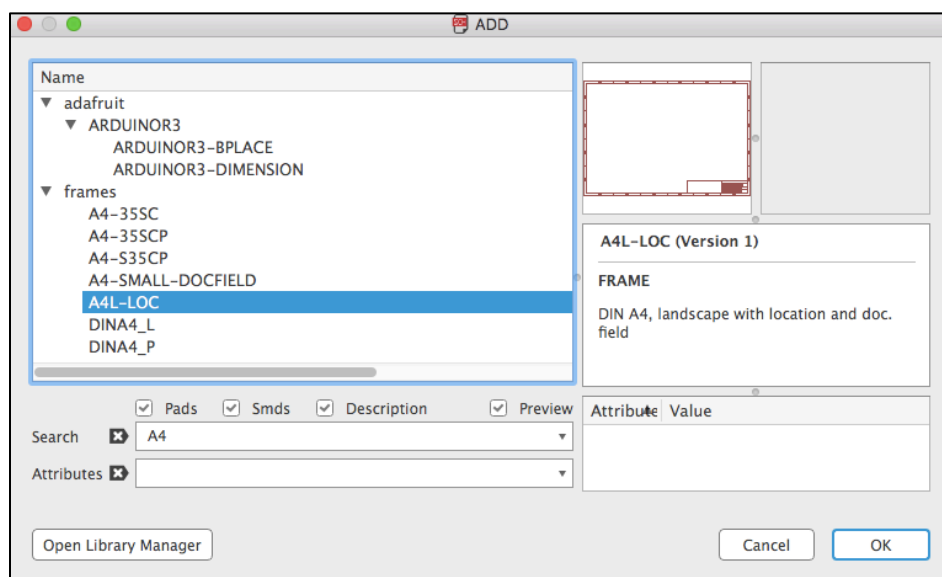


Looks intimidating, right? Don't worry—you'll get the hang of it pretty quickly. Here's the meaning of some of the tiles on the toolbars at the left and top of the frame.

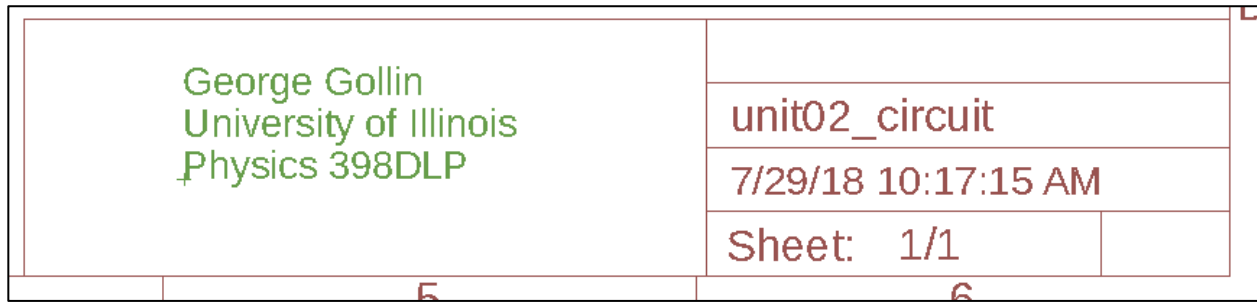


The stop/interrupt button is especially useful: if EAGLE decides to ignore your commands and keeps beeping at you, click on it.

Let's put a frame on the schematic and fill in some of its fields. Click the "add part to schematic" button and search for "A4." Select A4L-LOC and click on the schematic to drop it into place. You'll need to click on the stop/interrupt button to avoid placing a second frame on your schematic.

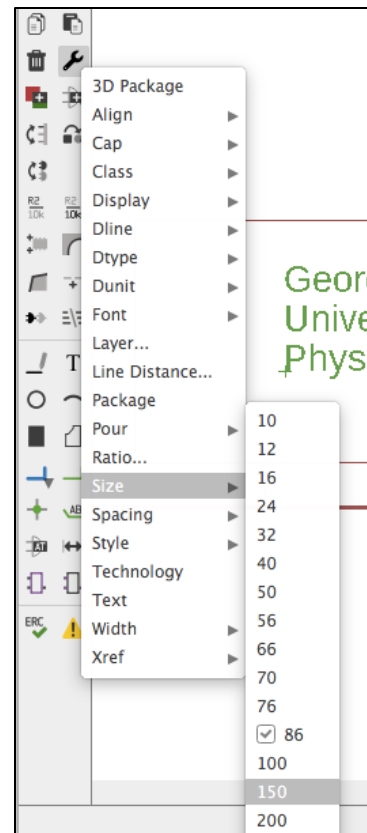


Click on the “zoom selection” button, then drag a rectangle over the information box at the lower right. Now save your schematic, then go to View → Redraw. You’ll see that the filename and date appear in the box. If you’d like to add more text (your name, for example), click the “T” tile in the left toolbar. Enter what you want, then click once to drop the text into place. Take note of the small + sign in the lower left corner of the text field. That indicates the location of the handle you can use to reposition or modify the text field.



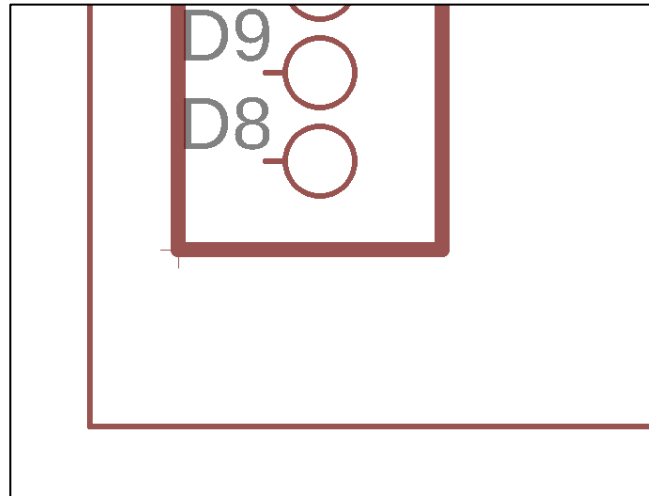
Here’s what you would do to make the just-added text larger. Do the following, in sequence:

1. click the change tile (the open-end wrench);
2. select size, then the new font size;
3. left click once on the text’s + sign;
4. click the “move” tile (the four-outwards-arrow symbol on the left toolbar)
5. left click once on the text’s + sign;
6. move the text to a new position;
7. left click once to drop the text in place;
8. click the stop/interrupt button.



Now put an Arduino Mega 2560 onto your schematic. Please use the P398DLP_ARDUINO_MEGA_2560_FLIPED_DEV version.

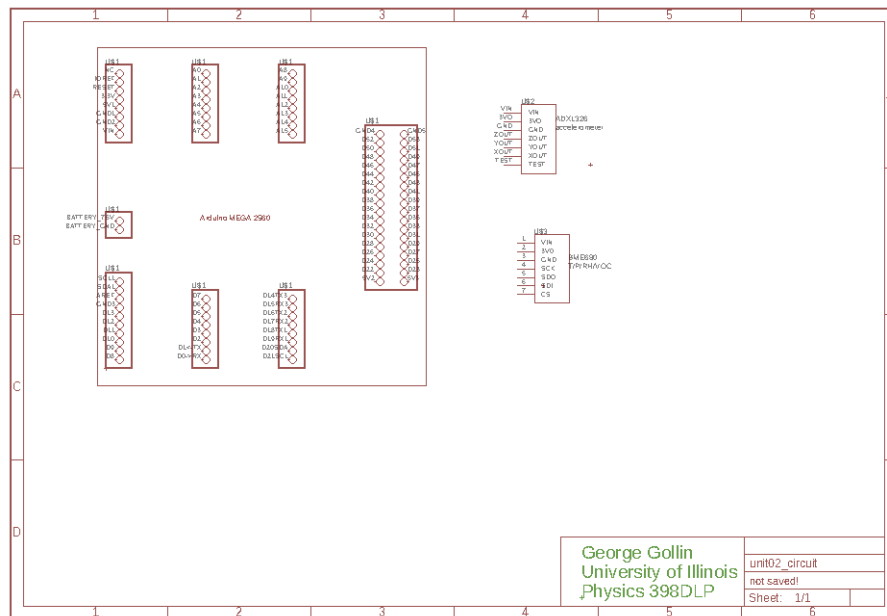
Here's what to do if you put two of them on the schematic by mistake. Locate the small (and difficult to see!) + sign on the part you'd like to remove. It's at the bottom left corner of one of the blocks of pins, as shown below.



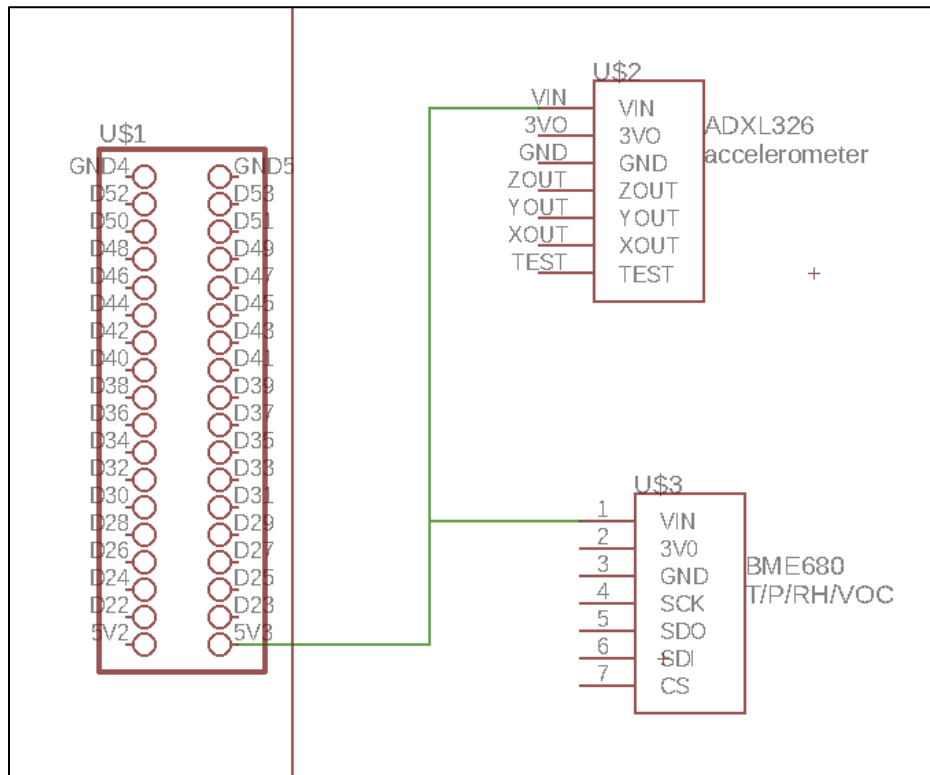
Left click once on the trash can tile on the left toolbar, click once on the + sign, then click once on stop/interrupt.

Now add an ADXL326 and a BME680 to your schematic. Sometimes the search tool for parts is balky, so you may need to help it out: in the Add Parts window, select the p398dlp library. You'll find the devices there.

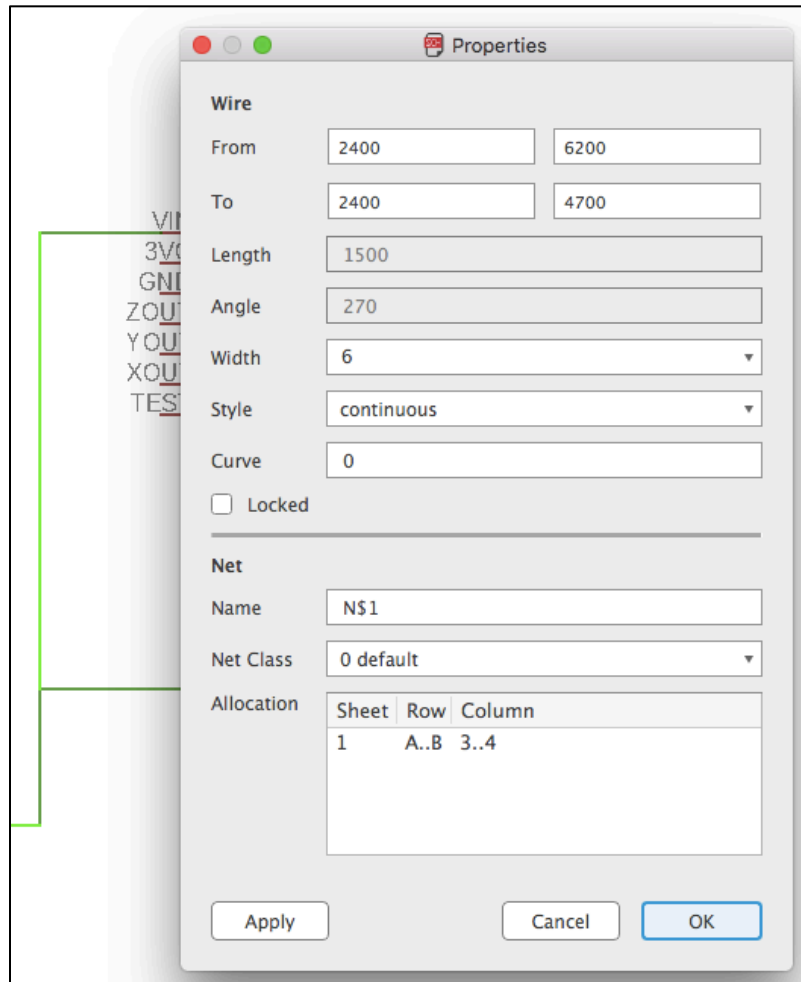
Your schematic ought to look something like this now:



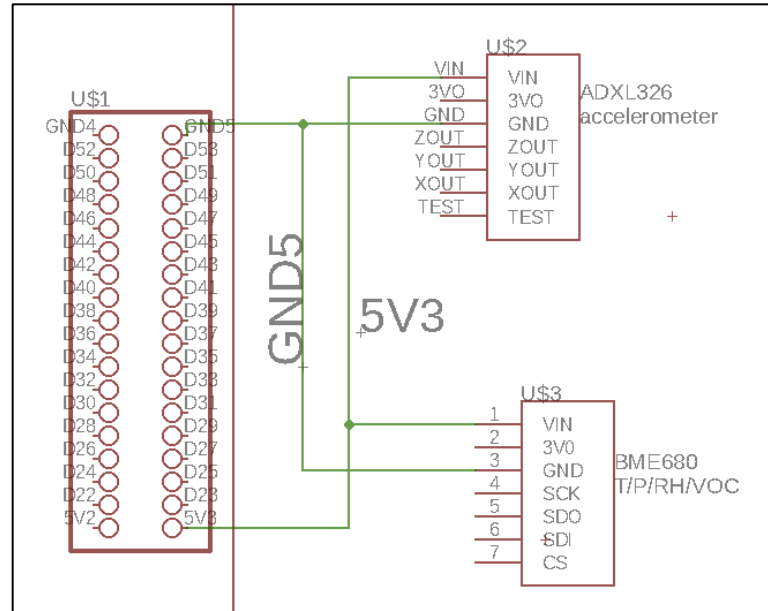
Let's connect the VIN pins on the two sensors to each other, and then to one of the Arduino's 5V lines. Click once on the "net" button on the left toolbar, then click on one of the pins. You can turn corners and connect to other pins by clicking. If you decide you don't like the placement of nets that you've established, click once on the "move" tile, then click once on the part of the net you'd like to move.



Now right click on the net you've just drawn and select "properties." You'll see something like the following screen shot. Many of the fields can be edited. I'd suggest you change the net name from the uninformative "N\$1" to something more descriptive like "5V3." Right click on the net a second time and select Name, then click in the "Place label" box.



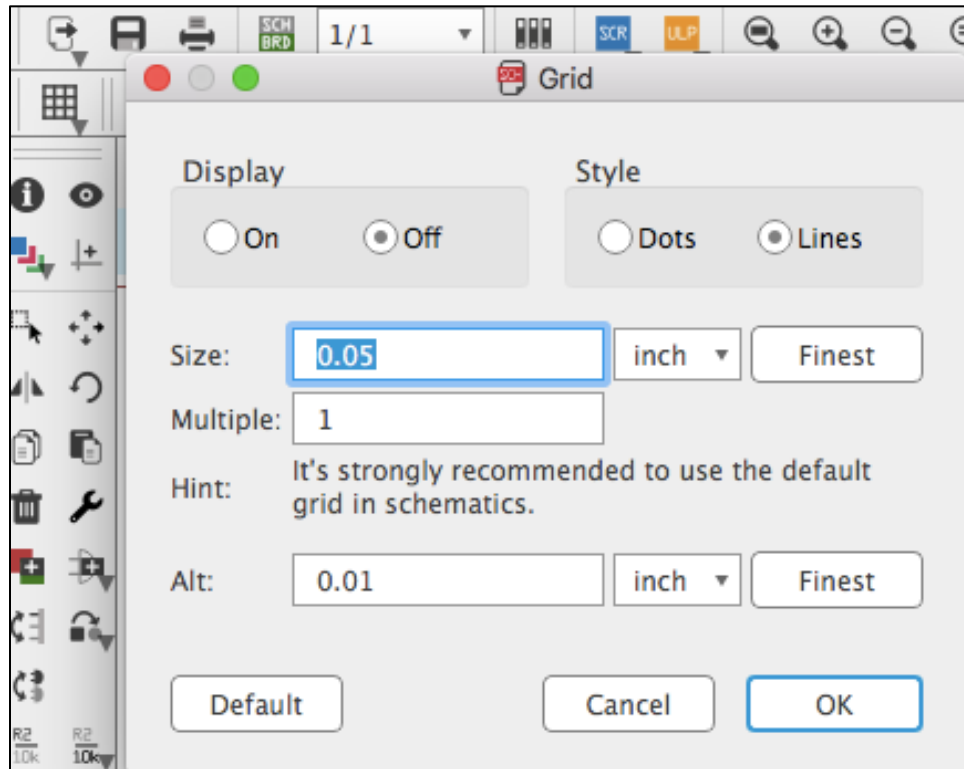
Now connect the GND pins to an Arduino ground. Add a visible label to the net. You may find that it's too crowded to allow a clean positioning of the net name on your schematic. If that happens you can rotate the text this way: click the rotate tile in the left toolbar, then click on the net name's + sign. Click the stop/interrupt button, then click the "move" button to reposition the text.



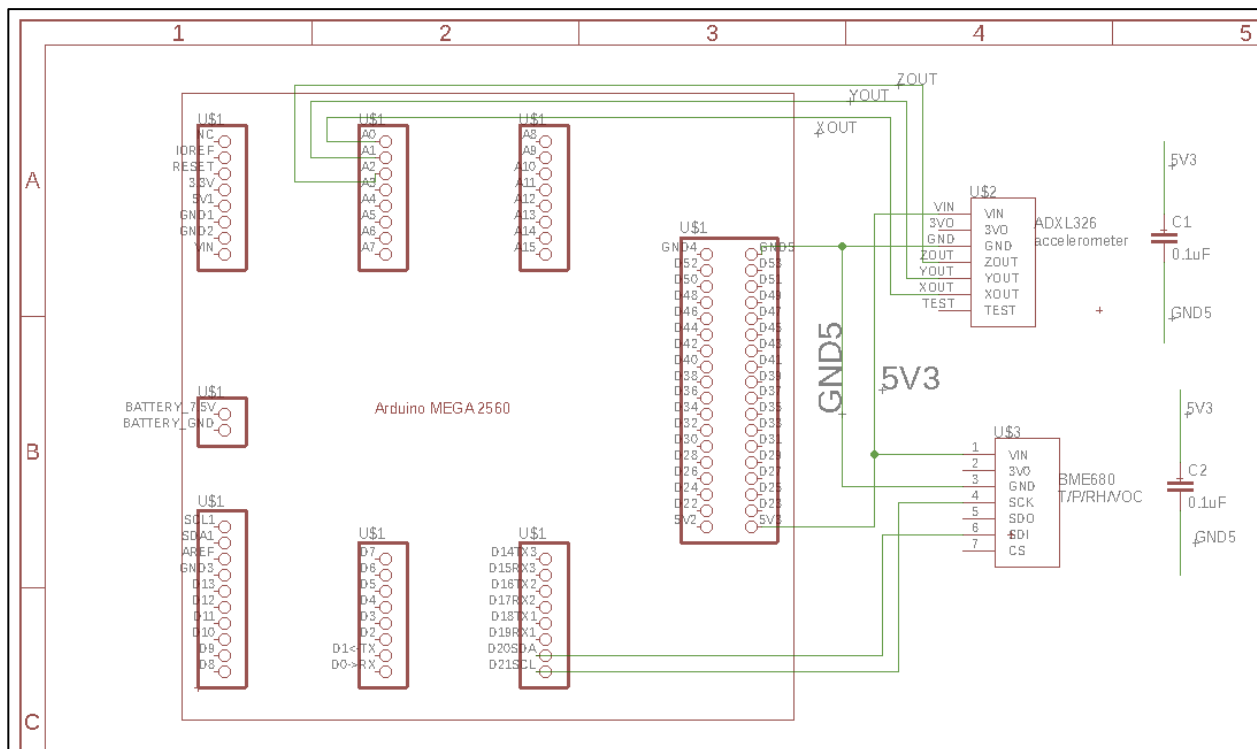
Now connect the rest of the pins as you've done on your breadboard. Naturally, the pins that you've left unconnected on your breadboard circuit will not be attached to any nets on your schematic. Label the nets.

It's generally a very good idea to put a capacitor between power and ground close to the inputs of each integrated circuit or, in our case, breakout board. So please put a pair of $0.1\mu\text{F}$ capacitors onto your schematic. You can find them in the "capacitor-wima" library among other places; consider using the C5/2.5 part, but you'll need to set the capacitance values by hand. You can connect them by attaching nets to their pins, then labeling the nets with the same names you had used for your 5V and ground nets. The nets don't actually need to be drawn so that they are attached to the VIN and GND pins of the sensors, as long as the net names are the same. To set the capacitances, right click on the devices, select "Value," and enter " $0.1\mu\text{F}$."

You've probably noticed that EAGLE forces objects to be placed on a grid. You can control the grid size using the control button that's immediately above the left side toolbar. I've reduced the grid size from 0.1 inches to 0.05 inches in my schematic.



Here's what my schematic looks like:



Installing microSD storage

Please install the microSD card breakout board now. You'll be using the SPI (Serial Peripheral) interface communications protocol, rather than the slower I2C—Inter Integrated Circuit—protocol used by the BME680. There's demonstration software at the Adafruit site that you can use to test it.

Add the microSD breakout to your schematic. I've included the part for this breakout board in the p398dlp library.

Installing the Real Time Clock

Please assemble the DS3231 real time clock module, then install it on your breadboarded circuit. You'll need to solder a pin header, then the backup battery carrier (careful when doing that!) before sliding in a CR1220 battery.

Oh—CR1220 means this: CR = LiMnO₂; 1220 = 12 mm (actually 12.5) diameter, 2.0 mm thick.

Find some demo software and use it to set the clock's time.

DAQ time.

“DAQ” is professional shorthand for “Data Acquisition Software.” Working with the other members of your team, please write a program that opens a microSD file, then executes a loop 100 times in which you record the time (from the DS3231, of course), the temperature, pressure, humidity, and acceleration ADC values for the *x*, *y*, and *z* sensors. Have your loop execute approximately once per second; after 100 passes have it close the microSD file. It'll be easiest to inspect the file if you write it as human-readable plain text rather than binary.

Write (or find) another program that reads your data file from the microSD card and writes its contents to the serial monitor window.

Remove the microSD card from the breakout board and try to open it with the SD card input on your laptop, if you have one, or else on someone else's laptop.

Volunteers needed

Here are the topics for presentation net week:

- How the ADXL326 accelerometer works
- How an electret microphone works
- How the MLX90614 IR sensor works

As I said last week, I'd like a report to last at most ten minutes, be carried in at most ten PowerPoint slides, be presented to the class by all the team members, and be suitable for upload to the course web site. (That means proper attribution of sources, and so forth.) Keep in mind that your audience is other students in the class, rather than a group of professional engineers.

Post-class assignment

1. Using TinkerCad, create a 3D-printable design for a small box with a lid in which you'll keep some of your breakout boards. It's fine to borrow a design you find on the web, but figure out how to engrave your name onto the lid of the box. Please don't make your box larger than about $2.5\text{ cm} \times 2.5\text{ cm} \times 2.5\text{ cm}$. I suggest you make the walls and lid 2 mm thick. I'll print it for you (you can pick the color!) once you've used Cura to create a gcode file.

2. Finish adding sensors to your breadboard prototype, downloading code from the web. Keep your schematic diagram up to date.

