Lab 3: PCB design with EAGLE

In this lab you will design a PCB board that will replace all the wires and boards you've used in the first two labs.

1. Pre-Lab

On the website are two EAGLE tutorials. Do them both.

Q1. For the first tutorial, get a screen capture of your board design. It should look similar to the figure below.



Q2. Now do the second (component design) tutorial and get a screen capture of the footprint and symbol. They should look a lot like the two figures below.

1/0 0	P2P1	TYO 0	



Now take a look at the EAGLE design file at

<u>http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove</u>. A modified copy of the schematic can be found on the following page.

Q3. Identify all the parts circled on the following page. For each part (A-H) find a part number and provide a brief description (a sentence or so) of what the part is.

Go ahead and do a design-rules check at this point from the board view (tools \rightarrow drc...) just to see what you get. You'll notice there are a fair number of errors. Most of those are quite acceptable (for example they use the metal layer for art on occasion and so too-line wires are fine) others aren't (wires too close to holes).



2. In-Lab



TECHINSIGHTS

Pictured clockwise from top left: Nexus Q, Roomba 5xx series, and iPhone 4 logic board.

(The in-lab is based heavily on the Eagle lessons from the Michigan Embedded Systems Hub by Sam DeBruin).

Part 1: Basic Eagle Usage

In the pre-lab, you gained experience performing schematic capture and board layout with a simple board in Eagle and looked at a much larger design. In this section, we will be exploring a mid-sized board featuring a simple microcontroller-based system, including a power supply and supporting components such as LEDs.

Here are a few reminders before you get started:

- Movement within a schematic, board, or library is achieved by clicking and dragging with the middle mouse button.
- Zooming in and out is achieved by scrolling the mouse wheel.
- Tools can be used either by pushing the button or typing the corresponding command in the command line.
- When using a command, return to the default by pushing the ¹¹ button or by typing a semicolon followed by a carriage return.
- Nets named with the 'name' command in schematic will automatically be connected to nets with the same name elsewhere in the schematic.



- To select an object in a tight space filled with other objects, click near the object's origin. The object will highlight but not yet capture. Left click again to confirm or right click to cycle parts in the area. Left click anywhere in the schematic whenever the correct part is highlighted to capture it.
- When searching for a part in the 'add' dialog, the '*' symbol represents a wild card.
- For example, the part 'MA03-2' will successfully be found by the search '*03-2' but not by the search '03-2'.

If you are confused by any of the above items, ask your GSI.

Part 2: Create a New Schematic and Add Core Components

From the EAGLE Control Panel, select File -> New -> Schematic Save this schematic somewhere you can find it.

The core components of a design are those that will affect the functionality. In our simple example board, the core components are the voltage regulator and the processor.

2.1 Power Supply

We will start our design by adding our voltage regulator and the support components that it requires. For simplicity we will be using a simple LDO (low dropout regulator).

Some of our components will come from the SPLAB Eagle library; you can download the library from the Lab 3 section on the course website: <u>http://eecs.umich.edu/courses/eecs498-brehob/labs.html</u>

To add the library to Eagle, click the ¹ button or go to Library -> Use, then select the SPLAB.lbr file you just downloaded.

In the command line, type 'add' and press enter or press the the button. From the SPLAB library select 'NX1117CE'. Place this part in the center of the page in your schematic.



All voltage regulators will require some capacitors to function. Information on what capacitors and of what size can usually be found in the voltage regulator's datasheet. In our case, this datasheet can be found by searching for NX1117CE50Z on www.digikey.com or at http://www.nxp.com/documents/data_sheet/NX117C_SER.pdf

Find section 12, "Application Information" in the above document. As we can see, we need two 10uF capacitors. Because this is a lower bound, we will use 100uF aluminum capacitors with additional low ESR ceramic capacitors for increased noise immunity.

We'll add the aluminum caps first. Find the library called 'rcl' and under CPOL-EU select the part CPOL-EUD. Place this part twice in your schematic, one to either side of the voltage regulator.



In the command line type 'value 100uF' and click on both capacitors. This can also be accomplished by pressing the Button and entering the values manually.

In the command line type 'net' or select the button. Using this tool, connect the positive terminal of each capacitor to either the input or output pins on the regulator. Note that there are two output pins, both should be connected.

Add GND symbols by finding the library called 'supply1' and adding the symbol GND. Once this part is placed, use the command 'copy' or press the $\frac{\$\$}{1000}$ button to duplicate it. Connect the negative pins of the capacitors and the GND pin of the regulator to GND.



Add two unpolarized 0805 ceramic capacitors - one for the input and one for the output. These parts can be found in 'rcl' under C-EU -> C-EUC0805. Connect them across PWR and GND as with the others. The value for these is 100nF.

The last steps in creating our power supply are to add a power source and to label the +5V net. To add a DC jack, find the library 'con-jack' and add either of the two DC jacks listed there. This is sufficient for our purposes but more care should be taken with an actual board. Connect pin 1 to the regulator's input and pins 2 and 3 to GND. Add the +5V label from the library 'supply1'. Connect it to the output. The program will prompt you to connect nets, click 'Yes'.



This is a finished power supply that will take a DC voltage up to 20V and regulate it to +5V.

2.2 Processor

Place Part and Decouple

Use the 'add' command to place the Atmega328 from the SPLAB library.

In order for the processor to function correctly, it needs a very consistent power supply voltage. This involves placing capacitors between power and ground nearby to the processor. These capacitors will then buffer the voltage and prevent transients. For more information on decoupling in processors, FPGAs, and other devices ask your GSI. We'll also have a fair bit of lecture material on the topic later in the semester.

We will be using 100nF 0603 capacitors for our decoupling network. In the schematic, use the 'add' command and find the part 'C-EUC0603' either by searching for it or by selecting the library 'rcl' -> 'C-EU'.

Place two 100nF capacitors near to the 'VCC' pins and use the 'net' command to connect them according to the figure below. Place the 'GND' and '+5V' parts by searching in the 'add' screen or by selecting the library 'supply1'.



Add two more capacitors, one between AVCC and AGND and one between AREF and GND.

Connect AGND to GND and AVCC to +5V.



Place Crystal with Capacitors

The processor needs a crystal input in order to operate. This crystal should be connected between XTAL1 and XTAL2. Additionally, each terminal of the crystal needs a 22pF (10e-12) capacitor connected to GND.

Using the skills developed earlier in this lesson, place the component 'CRYSTAL 16MHZ' from the 'SPLAB' library and two 0603 capacitors from the 'rcl' library. Use the value command to label these capacitors as 22pF. Use the image below to check your work.



Reset Switch and Pull-Up Resistor

The reset pin on a microcontroller is used to power cycle the device and restart. This is especially useful in embedded design if the processor freezes or if the user wants to observe startup behavior. A low going pulse will reset and this pin must be high (+5V) in order for the device to operate. In our schematic we are going to have a button push force that pin low and a resistor 'pull' it high at all other times.

Place an 0603 resistor (rcl -> R-US -> R-US R0603) between the reset pin of the processor and +5V. Give this resistor a value of 10k.



Add a pushbutton switch from the library switch-omron -> 10-XX. Connect one side to the reset line and the other to GND.

When the switch is unpressed, the resistor will cause the reset line to be pulled to +5V. When the switch is pressed, current still flows through the resistor, but it is much lower than the current through the switch and reset is forced low.



Programming Interface

The last mandatory component to add before this is a functional board is the programming interface. This interface allows the user to download code to the processor. Information regarding the correct setup for the programming interface is usually not found in the processor datasheet but instead in an application note or reference document. This particular processor programs via the Serial Peripheral Interface (SPI).

The processor's manufacturer provides the programming hardware, usually a USB to serial converter, and a termination header to connect to your board. In this case, the header is 3x2 0.1". For future reference, this is the programming interface for most, if not all, ATmega chips.

Add the component MA03-2 from the library 'con-lstb'.



To keep the schematic clean and easy to read, we will not be connecting the new header directly to the processor. Instead, we will use the 'net' and 'name' commands to make symbolic connections. EAGLE will connect these for us in layout. For more information on EAGLE layout see Lesson 2: EAGLE Layout and Gerber File Generation or ask your GSI.

Using the 'net' and 'name' commands make the symbolic connections seen in the image below. Type 'label' or press the sutton and click on a net to display the net name.



The programming interface can also supply power to the board. This is useful for debugging - the battery doesn't need to be connected while the debugging interface is connected provided power requirements for the board are low.

Place power and ground symbols and connect them according to the diagram below.



The last step is to make the other half of the symbolic connection. As we can see, RESET is one of the programming lines, so we need to label our reset line as 'RESET'. We also need to make connections to the SPI lines MISO, MOSI, and SCK.

NOTE: When you complete a symbolic connection, EAGLE will prompt you to connect nets. This is a good sanity check to confirm that you made no spelling mistakes. If you think you completed a symbolic connection but you are not prompted to connect nets, check your work.

, <u>k</u>	Connect N\$2 and RESET?	
	No Yes	

Use the 'name' and 'label' commands to complete the RESET net.

Using the 'net', 'name', and 'label' commands, complete the other three symbolic connections according to the figure below.

(SCK)PB5 (MISO)PB4 (MISO)PB3 (MOSI)PB3 (SS)PB2 (SS)PB2 (OC1)PB1 (ICP)PB0 P\$1	Z SCK 6 MISO 5 MOSI 4 3 2

Let's briefly look at the layout to make sure those symbolic connections worked. Because this is the first time we open the board, the program will prompt you to create from schematic. Press 'Yes'.



From the command line type 'board' or press the 📱 button to switch to board view.

The board view will be pretty messy right now but by moving things around you should be able to see that your processor, programming header, and reset switch are marked as being connected.



We have now created a board that supplies power to a processor. The processor gets a clean power supply with the capacitors we added and we can program that processor using the interface that we just made.

Part 3: Support Components

The last step in this board design will be to add support components. Support components can be anything from LEDs, level converters, serial adapters, or even just pin-out headers. For our board we will just be adding two LEDs and a few headers.

3.1 LEDs

The first LED is a power LED to indicate to the user that the board has power. Although seemingly trivial, this LED is a really important debugging tool and should be included on any board that has the power budget to accommodate it.

Add an 0603 resistor, give it a value of '240', and connect one terminal to +5V.



Add an LED from the library 'led' -> 'LED' -> LEDCHIP-LED0603. Connect the positive terminal to the resistor and the negative terminal to GND. Give the LED a value of 'GREEN'.



The next LED that we place will be a debugging LED connected to one of the digital IO pins of the processor. The user will then be able to toggle this LED through software. We are going to place our LED on the SCK line. This line is used during programming and then becomes user IO.

An LED on SCK, therefore, will indicate to the user that programming is happening and then become available for other use.

Add the same resistor/LED pair that we added for the power LED, but use SCK as the high voltage.



Note that the LED will be illuminated when SCK is high and off when SCK is low.

3.2 Headers

Because this board is essentially a platform for numerous different applications, it is likely that various processor IO pins will be needed in the future. The pins on the processor itself are far too tiny to solder directly to so we will provide a header to make this easier.

Add an 8-pin header from the library con-lstb -> MA08-1 and place it near Port D of the processor.

Connect the pins of the processor and the pins of the header using the 'net' command.

	P\$11	1	_
(HINL)PD7	P\$10	2	
	P\$9	3	
(TE)PD5	P\$2	4	
(TE)PD4 (TET4)DD2	P\$1	5	
(INTE/PD3	P\$32	6	
(INTO/PDZ	P\$31	7	
CIXD/PDI	P\$30	8	
(RAD/PD0			_
		-	9112

Naming and labeling these nets is optional and may be useful during layout.

Part 4: Error Checking

The last step in the schematic process is to run the error checker. This can be run at any time and is a good way to check your work. Note that it will not catch all errors, just some.

In the command line, type 'erc' or go to Tools -> ERC...

If you followed the lesson correctly, you should have a few warnings about value - these should largely be ignored. Make sure you fix all errors and at least understand all the warnings.

G1. Have your GSI look over your schematic.

Part 5: Initialize and Outline Board

Within the lesson 2 folder, open part1.sch and type 'board' or press the button. If this is the first time opening the board, the program will prompt you to create it.

Using either the 'move' command or the 'info' command, change the dimensions of the board outline (layer 20 - Dimension) to 2.6" x 1" centered at the origin. Make sure the width of the board outline wires is '0'.

Part 6: Disperse Components

With the outline placed, the next step is to arrange the components within the board area. This step, and subsequent rearranging, is pivotal to the success of the board.

Start with the regulator, IC1, and the processor, IC2. Group these items separately and with enough space for support components. Make sure that the decoupling capacitors for the microprocessor and voltage regulator are as close to their ICs as possible. Place the connectors, headers, and switch last.

When you understand how to place components, open part2.brd to get the arrangement presented below.



This view is called the rat's nest and the yellow lines are called airwires. These lines represent connections that you need to make by routing.

Part 7: Connections and Routing

Before you start connecting components, place polygons on both the top (layer 1) and bottom (layer 16) of the board and connect them to the GND net. Use the commands 'poly' and 'name' to place and name the polygon.

In the figure below, a polygon has been placed on the bottom layer. When placing a polygon, it appears first as a dotted outline. Notice that the polygon overlaps the board outline; this will be corrected automatically by Eagle.



Use the command 'rats' to fill the polygon.



Three things are important to note from this view. First, notice that the polygon does not go to the edge of the board. This is because we have already input the minimum distance to the edge that the board house allows. Eagle follows this guideline.

The second thing to note is that two of the large pads in the bottom right no longer have an airwire connecting them. These pads are both part of the GND net and the polygon connects them. The command 'rats' not only fills polygons, it also redraws airwires (including removing those not still required).



The final thing to note from this view is the thermals. If you look closely at the aforementioned GND pins or the GND pin of the programming header (enlarged below) you will notice that the GND plane doesn't connect to the hole all the way around.



Instead, four small copper areas connect to the rest of the GND plane. This allows the pin to be soldered without the iron's heat being wicked away to other parts of the board, potentially causing damage.

Practice routing a few of the airwires. Once you get the hang of routing, move on to the next section. You will be given an opportunity to complete the board at the end of the lesson.

Part 8: Gerber Files

Once you have made the board in Eagle, you need to specify to the board house what the PCB looks like. To do this, use Eagle's CAM processor.

To open the CAM processor, go to File -> CAM Processor or press the 📟 button.



With the CAM window open, go to File -> Open -> Job...

Sunstone provides CAM jobs to produce Gerber files that will be compatible with their process. The files can be found on the Sunstone website under 'Sunstone Downloads' in the 'PCB Resources' section at the bottom of the page or at:

http://www.sunstone.com/pcb-resources/Downloads.aspx

The file is listed at the bottom: **Sunstone-EAGLE 5.0 Cam NEW.zip**. The default path includes the Eagle directory. Go to cam -> Sunstone-Eagle 5 and select 2LPlus-Sunstone.cam.

	Top Side	Bottom Side	Silkscreen Top	Soldermask Top	Soldermask Bott	om Silkscreen	Bottom	Top Stencil	Bottom Stencil
Section C Prompt Dutput Device File Street X Oinch Y Oinch	Dutline GERBEI	R_R5274X				vle Mirror Rotate Upside down gos. Coord Quickplot Optimize Fill pads	Nr A 1 1 16 16 17 18 18 20 202 23 24 22 23 24 24 22 23 24 24 23 25 26 27 28 290 33 344 33 366 36 367 38 390 90 41 43	Layer Top Bottom Pads Vias Unrouted Dimension tPlace tOrigins tNames tValues tValues tStop bValues tStop bValues tStop bStop tCream bFinish bFinish bFinish bGlue tCream tFinish bFinish bGlue tTest tKeepout tKeepout tKeepot tKestrict vRestrict	
								Delle	

Each of the tabs at the top represents a layer or collection of layers. Notice that the 'file' parameter for each tab includes both the extension and the placeholder '%N'. This placeholder will be replaced with the board name when the job is processed.

Create a folder on the desktop named 'output'. For each tab at the top change the 'file' parameter to include the path to the new folder 'output'. Once done, press the button to create the CAM files.

Open another job, in the same folder as before, entitled excellon.cam. Save the file to the same place and process the job. This will generate placement information for the holes in the board.

The final step in the CAM process is to generate the tool list file. This file provides information to the board house about what size drills your job will need.

In the command line, type 'run' and navigate to ulp -> drillcfg.ulp.

Eagle: Drill Config	juration
Select unit for output file	
💿 mm	OK
◯ inch	Quit

In the window that opens, specify the units as mm. Unless you specifically did otherwise, the Eagle drill sizes are in mm. Press OK.

T01 0.60m	ım	
T02 0.80m	nm Nm	
T04 1.02m	nm	
	Ok	

Press OK again to accept the sizes listed and save the .drd file with the rest of your Gerbers in 'output'.

To produce your board, compress the folder with your gerbers and upload the file.

Part 9: Self Check

For this self-check assignment, finish routing the board started during the lesson. The following images with their captions detail a few areas of the board and the key items to consider when working in these areas.



9.1 Power Supply



When working on the power supply, it is important to keep traces wide. All the system current will flow through these traces so it is best to make them polygons. As seen in the images above, there is a polygon for each of VIN and VOUT in the power supply. GND, as mentioned above, exists in a polygon that covers all uncovered areas of the board.

A special thing to note when working with overlapping polygons is the rank. To view or modify a polygons rank, use the 'info' command and click on the polygon's outer edge.

Polygons with higher rank (lower number) will be placed over lower ranked polygons. In this case, make sure the power supply polygons are filled instead of the GND plane polygon.



9.2 ATmega



As seen in these figures, the organization of components is critical. Place decoupling capacitors as close to their IC as possible. The crystal is of secondary importance to those capacitors and should be placed very near the microcontroller as well.

9.3 Button





9.4 Connectors



- **G2.** Show your board layout to your GSI. Be able to justify the decisions you've made.
- **G3.** Create a zipfile of your Gerber files you'd need for the manufacture of your board. Show your GSI.

3. Post-lab

Q4. In the lab we used both aluminum and low-ESR ceramic capacitors.

- a. What is ESR in this context?
- b. Why does the type of capacitor matter? You might find it useful (if mildly painful) to watch http://www.youtube.com/watch?v=VTh6hT2h9kg.