Lesson 1: EAGLE Part Creation and Schematic Capture

Sam DeBruin

January 16, 2012

1 Introduction

The purpose of the first two lessons is to provide a usable knowledge of the PCB (printed circuit board) creation software EAGLE. Lesson 1 covers the part creation process as well as schematic capture. Lesson 2 will cover the schematic to board process, layout, and Gerber file generation. For both of these tutorials, make sure you have an up-to-date copy of the SPLAB EAGLE library. Machines in the SPLAB will already have these libraries installed. To obtain the SPLAB library, email cse-splab@umich.edu.

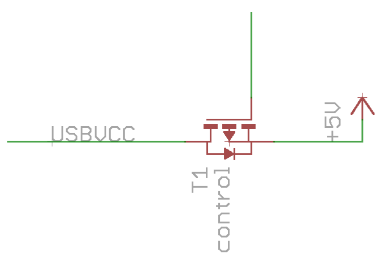
This lesson details designing a simple microcontroller based system in order to learn the important concepts of PCB design. This involves making a power supply, adding the processor and other core components, and adding support components like LEDs.

At the end of this lesson there is a self check portion wherein students reproduce the Arudino Duemilanove and add a sensor. This self check is designed to reinforce the material of lessons 1 and 1a (part creation). Lesson 2 builds on material from Lesson 1 but not from the open ended self check.

2 Basic EAGLE Use

If you are new to EAGLE, there are a few things that you should know 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  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 SPLAB instructor.

3 Create a New Schematic

From the EAGLE Control Panel, select File → New → Schematic

Save this schematic somewhere you can find it. If you are on a SPLAB computer, save your schematic to the Desktop in a folder named Lesson1_YOURNAME where YOURNAME is replaced by your uniquename.

4 Add Core Components

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.

4.1 Power Supply

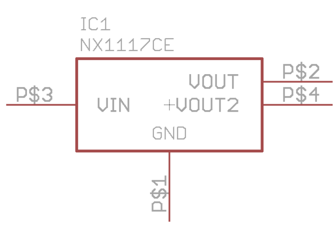
4.1.1 Place Part

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).

In the command line, type 'add' and press enter or press the  button

From the SPLAB library select 'NX1117CE'.

Place this part in the center of page 1 in your schematic



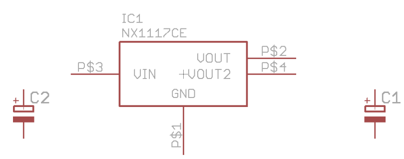
4.1.2 Add Capacitors

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/NX1117C_NX1117CE_SER.pdf`

Find section 12 - Application Information. As we can see, we require 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. For more information on using capacitors in power supplies including capacitor type and ESR, ask your SPLAB instructor.

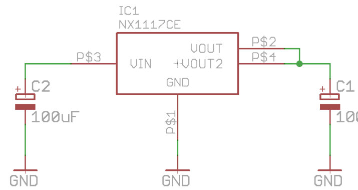
We'll add the aluminum caps first. Find the library called 'rcl' and under CPOL-EU select the part CPOL-EUD. Place this part twice in your schematic, one to either side of the voltage regulator.



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  button to duplicate it. Connect the negative pins of the capacitors and the GND pin of the regulator to GND.

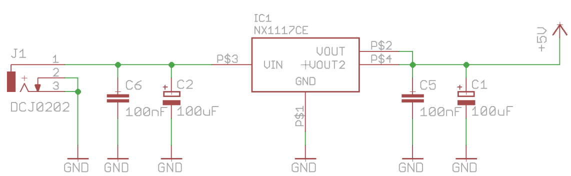


Add two unpolarized 0805 ceramic capacitors - one for the input and one for the output. These parts can be found in 'rcl' under C-EU → C-EUC0805. Connect them across PWR and GND as with the others. The value for these is 100nF.

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. Ask your SPLAB instructor if you have any questions.

4.2 Processor

With the power supply placed, the most important item remaining is the processor. For this, we are going to have to create a new part.

Refer to Lesson 1a: Eagle Part Creation for more information on creating a part.

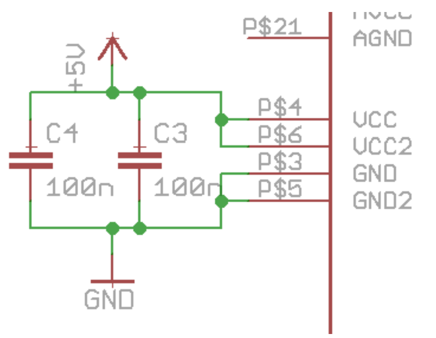
4.2.1 Place Part and Decouple

Once you have made your component, use the ’add’ command to place the new processor on page 2.

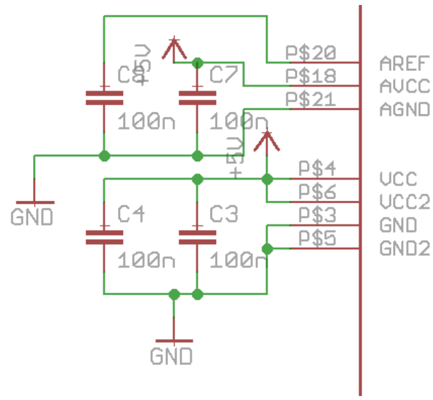
In order for the processor to function correctly, it needs a very consistent power supply voltage. This involves placing capacitors between power and ground nearby to the processor. These capacitors will then buffer the voltage and prevent transients. For more information on decoupling in processors, FPGAs, and other devices ask your SPLAB instructor.

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’.



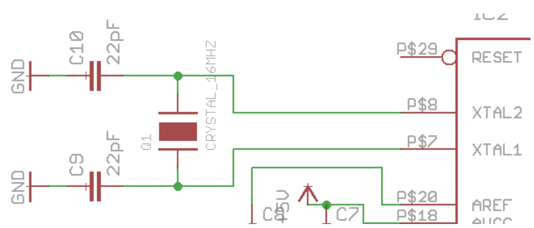
Add two more capacitors, one between AVCC and AGND and one between AREF and GND. Connect AGND to GND and AVCC to +5V. Ask you SPLAB instructor if you are unsure of what these pins are or why these capacitors are needed.



4.2.2 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 (10^{-12}) capacitor connected to GND.

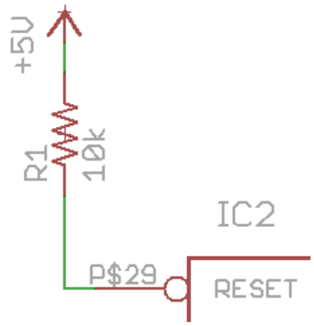
Using the skills developed earlier in this lesson, place the component 'CRYSTAL_16MHZ' from the 'SPLAB' library and two 0603 capacitors from the 'rcl' library. Use the value command to label these capacitors as 22pF. Use the image below to check your work.



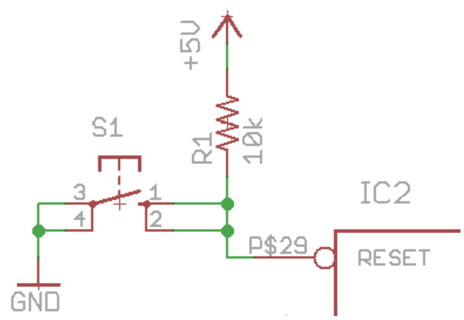
4.2.3 Reset Switch and Pull-Up Resistor

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. For a more thorough explanation of reset conditions or pull-up and pull-down resistors, ask your SPLAB instructor.

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.



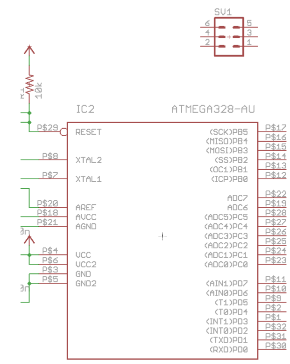
When the switch is unpressed, the resistor will cause the reset line to be pulled to +5V. When the switch is pressed, current still flows through the resistor, but it is much lower than the current through the switch and reset is forced low.

4.2.4 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). For more information on SPI and other serial communication, ask your SPLAB instructor.

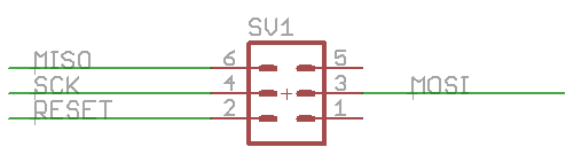
The processor's manufacturer provides the programming hardware, usually a USB to serial converter, and a termination header to connect to your board. In this case, the header is 3x2 0.1". For future reference, this is the programming interface for most, if not all, ATmega chips.

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



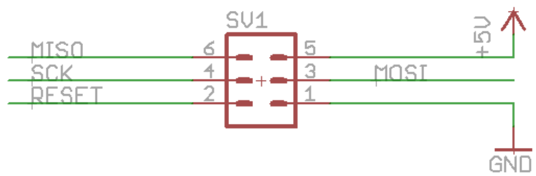
To keep the schematic clean and easy to read, we will not be connecting the new header directly to the processor. Instead, we will use the 'net' and 'name' commands to make symbolic connections. EAGLE will connect these for us in layout. For more information on EAGLE layout see Lesson 2: EAGLE Layout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

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



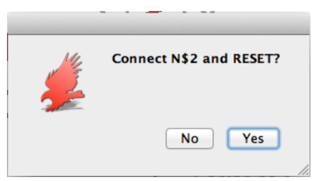
The programming interface can also supply power to the board. This is useful for debugging - the battery doesn't need to be connected while the debugging interface is connected provided power requirements for the board are low.

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.

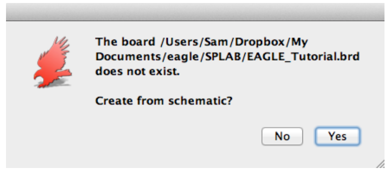
NOTE: When you complete a symbolic connection, EAGLE will prompt you to connect nets. This is a good sanity check to confirm that you made no spelling mistakes. If you think you completed a symbolic connection but you are not prompted to connect nets, check your work.



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

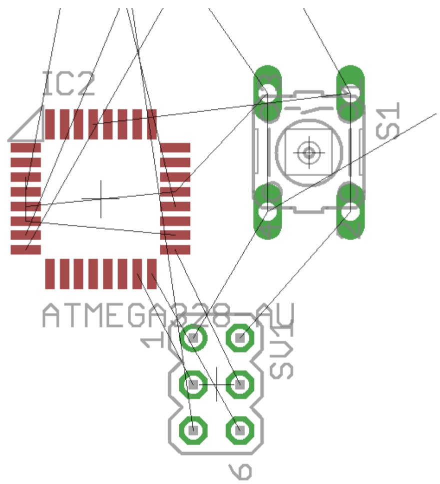
Using the 'net', 'name', and 'label' commands, complete the other three symbolic connections according to the figure below.

Lets 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.

5 Support Components

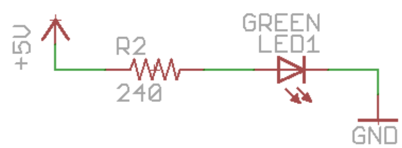
The last step in this board design will be to add support components. Support components can be anything from LEDs, level converters, serial adapters, or even just pinout headers. For our board we will just be adding two LEDs and a few headers. For more information on level converters, serial adapters, etc ask your SPLAB instructor.

5.1 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.

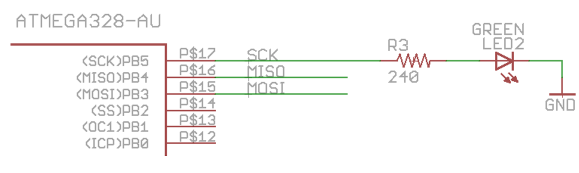
Back on page 1 of the schematic add an 0603 resistor, give it a value of '240', and connect one terminal to +5V.

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.

On page 2 of the schematic, add the same resistor/LED pair that we added for the power LED, but use SCK as the high voltage.



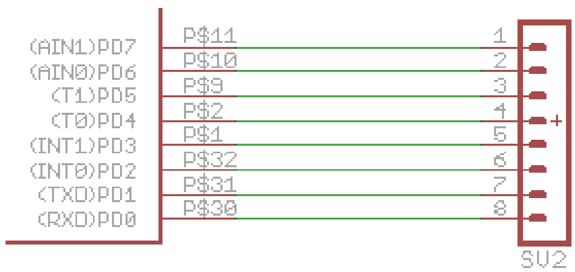
Note that the LED will be illuminated when SCK is high and off when SCK is low.

5.2 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. If you are unsure of what a header is, ask your SPLAB instructor.

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.



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

6 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. If you have questions, ask your SPLAB instructor.

7 Summary

This concludes Lesson 1. Through this lesson we created a new schematic, added a power supply and DC input jack, implemented a microcontroller with its support components, and added some debugging and power LEDs. This setup, although simple, is the first step in the world of embedded design. We can blink an LED via software and we have many IO ports for extended applications.

For information on how to use EAGLE Layout to turn this schematic into a board and then use EAGLE CAM to generate the Gerber files, see Lesson 2: EAGLE Layout and Gerber File Generation.

8 Self Check

The last aspect of this lesson is a self check assignment. This assignment will test your understanding of the material that you just covered.

The self check for lessons 1 and 1a is to reproduce the schematic of the Arduino Duemilanove and add a sensor. The schematic for the Arduino can be found at

`http://arduino.cc/en/uploads/Main/arduino-duemilanove-schematic.pdf`

The SPLAB library contains parts for IC2, IC5, T1, X4, and F1. The name of the part in the SPLAB library can be found below:

Arduino Part Label	Part Name	SPLAB Library Name
IC2	FT232RL USB to Serial Converter	FT232RL
IC5	LM358D Dual Op Amp	LM358D
T1	NDT2955 PMOS MOSFET	PMOSSOT223
X4	Mini USB Connector	USBSMD
F1	Inductor	INDUCTOR0603

The second part of the self check is to add a sensor of your choice. Go to www.digikey.com → Product Index → Sensors, Transducers and select a sensor to add. Temperature sensors are always a good choice. Using the sensor's datasheet and the skills developed in this lesson make a new part for the sensor, add it to your schematic and connect it to the processor. Ask your SPLAB instructor if you have questions.

Once you are finished with the self check assignment, ask your SPLAB instructor to check your work.