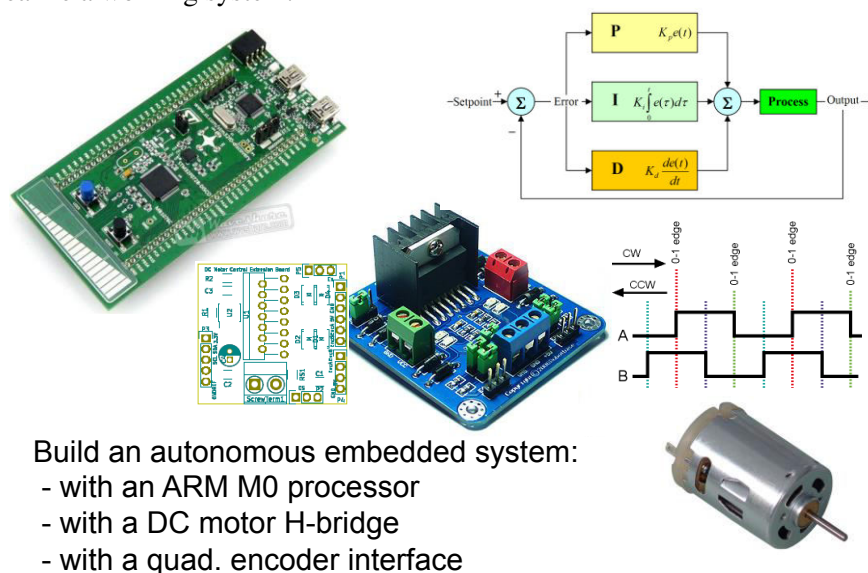


## Homework 1: Schematic PCB Design

Homework is due Jan. 26<sup>th</sup> 11:59pm on Canvas. Don't put this off to the last minute! It will be highly appreciated if you could submit files earlier. This way, we'll be able to give your early feedbacks that will improve the quality of your PCB. NO LATE SUBMISSIONS WILL BE ACCEPTED. We work with pretty tight constraints (manufacturing and fabrication) that we cannot postpone.

### 1. Objective

The class consists of a directed mini-project that aims at building an autonomous embedded system (Fig. 1). This system aims at controlling the speed of a DC motor using feedback loop. The system will consist of (1) an ARM Cortex-M0 based microcontroller (here physically integrated on the STM32F072 DISCOVERY development board along with an hardware programmer/debugger STLink2), (2) a custom-made extension board that contains all the required power interface between the DC motor and the microcontroller (the content of this homework); and (3) the DC motor itself with a quadrature encoder. The different labs and homeworks are designed in such a way that you'll acquire all the necessary knowledge and skills to realize a working system.



Build an autonomous embedded system:

- with an ARM M0 processor
- with a DC motor H-bridge
- with a quad. encoder interface

Fig. 1. Mini-project objective

Our goal is to get hands-on experience with PCB design. You will need to read and understand datasheets, and to create a 1" x 2" printed circuit board (PCB) that will serve as interface between the DC motor and the microcontroller board. The PCB design is split into two homeworks. In the first part, you will create the schematic of the board, and make an initial plane PCB. In the second part, you will work on the PCB layout. The split is there so that you can get feedback on your first part before you start with the second part.

To realize your PCB design, we will use a common software suite for Electronic Design Automation (EDA) - PCB schematic and layout -, named EAGLE. EAGLE is available on many platforms (Windows, Linux and Mac OS X) as freeware – so feel free to install it on your personal machines. It is also available on the lab images.

## **2. Readings and Tutorial Videos**

EAGLE is very popular with the hobbyist and educational communities and has a lot of online tutorials. These come in a variety of formats and depending on your personal preference you may chose to read or watch example projects.

Most of the following sections are taken from a series of tutorials developed and generously shared under a creative commons license by Sparkfun Electronics.

Unless you are already familiar with PCB design, it is strongly recommended that you read the following sections and visit the full tutorials online.

### **EAGLE User Manual:**

- Available from the EAGLE control panel (main window) after expanding the “Documentation” tab in the left hand list.
- Manuals can be found under the “doc” subfolder of the EAGLE install directory.

### **EAGLE Tutorial Videos:**

- CadSoft EAGLE guided tour
  - <https://cadsoft.io/tour/>
- Jeremy Blum & Element14 EAGLE Tutorials
  - <http://www.jeremyblum.com/category/eagle-tutorials/>
- Tangentsoft EAGLE and Soldering Tutorials
  - <https://tangentsoft.net/elec/movies/>

### **Sparkfun EAGLE Tutorial List:**

- Basic theory of PCB structure
  - <https://learn.sparkfun.com/tutorials/pcb-basics>
- How to read schematics, EAGLE schematic tips
  - <https://learn.sparkfun.com/tutorials/how-to-read-a-schematic>
- Install and configure EAGLE
  - <https://learn.sparkfun.com/tutorials/how-to-install-and-setup-eagle>
- Designing a schematic
  - <https://learn.sparkfun.com/tutorials/using-eagle-schematic>
- Converting a schematic to board layout
  - <https://learn.sparkfun.com/tutorials/using-eagle-board-layout>
- PCB layout with surface-mount parts
  - <https://learn.sparkfun.com/tutorials/designing-pcbs-advanced-smd>
- Creating custom part symbols and footprints
  - <https://learn.sparkfun.com/tutorials/designing-pcbs-smd-footprints>

## **3. Instructions**

The overall system block diagram, shown in Fig. 2, illustrates the different elements that will (1) be interface to the board and (2) the elements that will compose the board:

- **STM32F072 DISCOVERY board:** This board contains an ARM Cortex-M0 based microcontroller whose pins are breakout on pinheaders. Review carefully the datasheet **HERE** and in particular, understand what are the logic levels of the different pins. Connections between the discovery board and the extension board will be made through jumper wires.

[www.st.com/resource/en/user\\_manual/dm00099401.pdf](http://www.st.com/resource/en/user_manual/dm00099401.pdf)

- **DC Motor:** The DC motor is a **Pololu item #2824**. The motor will be operated with 6V, and

the stall current is 2500 mA. It is recommended to keep this value in mind while wiring the H-bridge and while sizing this current sense resistor to reach the full window of the ADC. Look also at the voltage range acceptable by the quadrature encoder – relate that to the logic levels of the digital pins of the microcontroller. Finally look at the connection of the motor and how it will be connected to the board (pins arrangement on the pin header).

<https://www.pololu.com/product/2824>

- **H-bridge:** The H-bridge (please read about the principle of an H-bridge) will serve as power interface between the motor coil and the control system (ARM processor). We will use a ST L298N chip (in multiwatt package). Carefully study its datasheet available **HERE** and understand how this chip is controlled and wired. Note that the “power” DC supply will be made with a screw terminal. Please also be careful of the different decoupling capacitances required with this device to operate properly.

<http://www.st.com/web/en/resource/technical/document/datasheet/CD00000240.pdf>

- **Sense resistor:** The sense resistor will be used to measure the current in the motor coil. The resistant must be sized according to the ADC range and the maximum current of the motor. The analog value will be brought back to the microcontroller board with a 3-pin pin header; the analog signal being sandwiched by 2 shielding ground lines.

- **Temperature sensor:** In order to monitor the temperature around the H-bridge, we will add close to the H-bridge (or its optional heat sink) a TI LM75A. It is strongly recommended to read its datasheet **HERE**. This device will communicate through I<sup>2</sup>C (a serial communication bus) with the microcontroller. The I<sup>2</sup>C signals along with the power supply and digital control of the H-bridge will be centralized on a common digital pinheader.

<http://www.ti.com/lit/ds/symlink/lm75a.pdf>

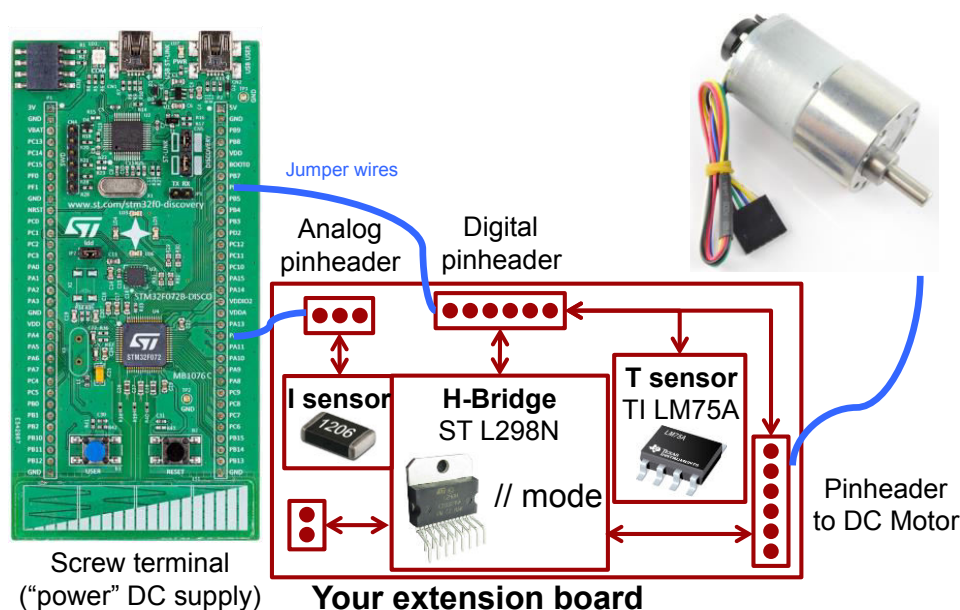


Fig. 2. System Block Diagram

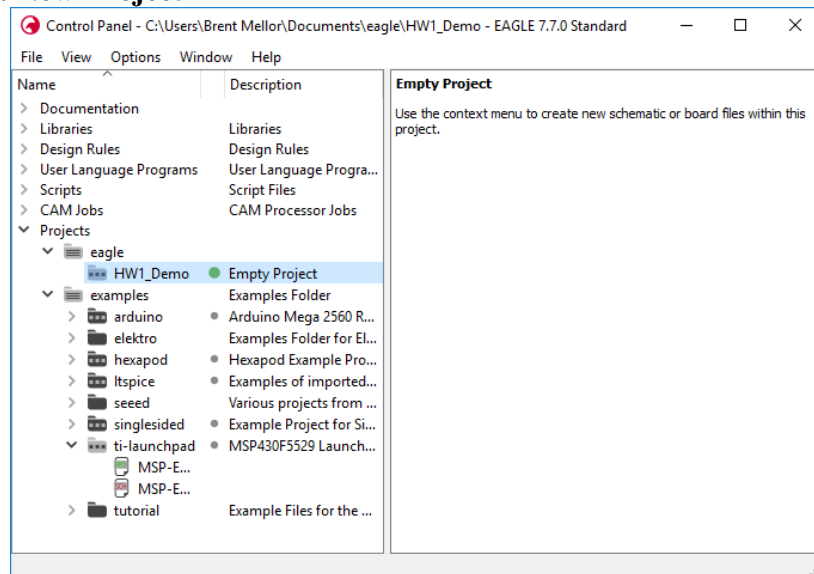
#### 4. Creating the Schematic

This section covers the initial steps required to start designing a schematic within EAGLE. It's a condensed version of the SparkFun tutorials “How to read a schematic” and “Using EAGLE: Schematic.”

It assumes you have a basic idea about electric circuits and common components such as resistors and capacitors. If not then you will want to review a basic circuit tutorial, such as <https://learn.sparkfun.com/tutorials/what-is-a-circuit>, and let us know of any questions you

may have.

## Creating a New Project



After loading, EAGLE always starts with the “Control Panel” window. The control panel provides a quick access point to all previously opened projects as well as parts libraries, documentation and user scripts. The following image shows the control panel with a new project selected. You can access the built-in user manuals by opening the “Documentation” tap in the left-hand list.

- Create a new project by using **(File → New → Project)**
  - EAGLE projects are simply folders that will contain all of your schematic and board design files. You should give the new project a simple name to describe its contents.
  - After creating the new project, notice that there is a green circle next to the list entry for it. This indicates that the project is loaded and any new schematics or board design files will be associated with it.
- Add a schematic to the new project either by right-clicking on the project entry and selecting **(New → Schematic)** or by using the **(File → New → Schematic)**
- Once the schematic window opens, you can save the new empty schematic and it should appear under the project entry in the control panel.
  - Schematic files have the extension “.sch” but may not automatically identify with EAGLE when clicked in a folder. You will always want to load your files through the control panel window.
  - Name your schematic such that we can identify it once you submit the assignment. A good name would be “HW1\_NAME\_UNID.sch”

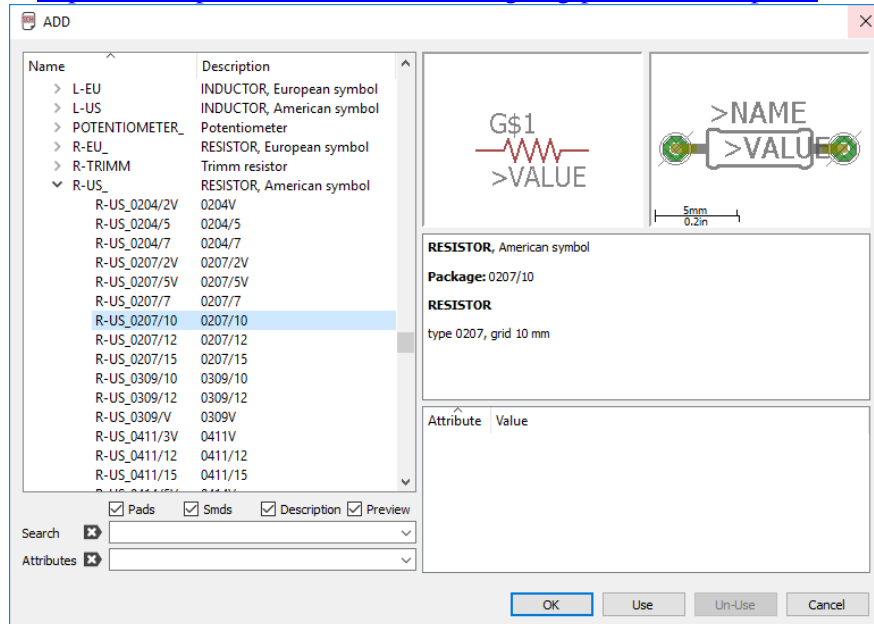
## Adding Parts to a Schematic

- An empty schematic isn’t all that useful. You will want to add components to the schematic by selecting the “Add Component” button on the left-hand toolbar. **(Edit → Add)**
- The first time you add a component to a schematic may take a while as EAGLE loads all of the built-in part libraries.
  - These part libraries contain hundreds of existing schematic symbols and PCB footprints for use. Unfortunately this makes them somewhat difficult to navigate.
  - Many of the online tutorials use their own component libraries. Companies such



as Adafruit or Sparkfun often publish libraries containing the parts they sell.

- **For basic parts such as resistors, capacitors and connectors you are allowed to use existing component libraries.**
  - For parts such as the motor driver and temperature sensor you should create your own library and part symbols.
  - Sparkfun has a good tutorial for creating your own parts.  
<https://learn.sparkfun.com/tutorials/designing-pcbs-smd-footprints>



- For including libraries from other sources, see the setup tutorial.  
<https://learn.sparkfun.com/tutorials/how-to-install-and-setup-eagle>
- Once the ADD window opens, navigate in the left-hand part of the window until you find a parts library called “**rc1**”
  - Once you expand “**rc1**” you will find a number of sub-categories, expand the “**R-US**” category and select a random entry.
  - Once selected, the right-hand parts of the window show the schematic symbol, PCB footprint and other attributes of the selected part.
  -
- Notice that all the parts in the “R-US” category are resistors and share the same schematic symbol? Why so many duplicate parts?
  - Although all these parts are resistors and behave identically in the schematic, they are all physically different when placed on the actual PCB.
  - When selecting parts from a library, you will need to make sure to select the appropriate version to match the physical parts you’ll be using.

### Selecting Parts offered in the Stockroom

The stockroom offers kits containing all of the parts you’ll need to build the motor driver. These kits use mostly surface-mount parts so you’ll need to select SMT (surface-mount) PCB footprints that match the size of parts offered.

Here is a list of footprints that you will want to use from the built-in EAGLE libraries. You can also find similar devices in libraries from Adafruit or Sparkfun.

#### Resistors (1206 SMT)

- Use R-US\_R1206 or R-US\_R1206W
- Found in the rc1 → R-US library.

#### Capacitors (1206 SMT)

- Use C-USC1206, C-USC1206K, C-EUC1206, C-EUC1206K

- Found in the rcl → C-US and C-EU libraries.

### Electrolytic Capacitor (Radial, through-hole)

- Use CPOL-USE2-5
- Found in the rcl → CPOL-US library.

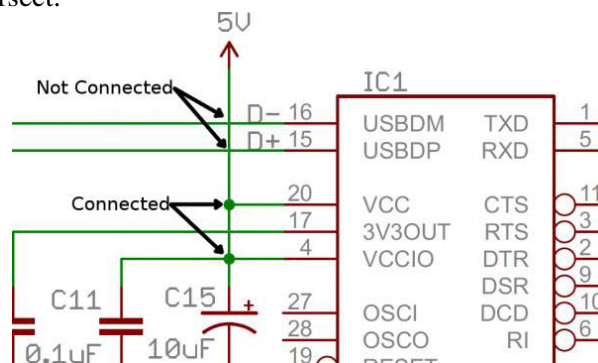
### Pin Header (100mil, through-hole)

- Female pin headers can be found in the **con-1sta** library.
- Male pin headers can be found in the **con-1stb** library.

You'll need to create your own symbols and footprints for the diodes, motor driver and temperature sensor.

## Making Schematic Connections

- After selecting a part in the ADD window, press OK to return to the schematic.
  - You should have a copy of the selected part on your cursor, you can rotate the orientation of the part before placing with the right-mouse button.
  - You can continue to add copies of that part by clicking until you press ESC or click the red stop button on the top toolbar.
- To create circuit connections between the pins of the parts you placed, click the "Net" button on the left-hand toolbar.
  - Using the "Net" tool you can click on a pin of a component and then a net/wire connection will follow the cursor.
  - You can click on the schematic to make corners and other geometry for routing.
  - Click on another pin to end the current net.
- You can change the routing style (right-angle, 45-degree, or direct) by clicking on the right-mouse button while routing.
- You can join multiple nets by clicking on the middle of an existing line while placing a new net.
  - When joining nets, EAGLE should always place a junction "node" on the intersection point.
  - Without a junction, there is no electrical connection in the schematic even if the nets intersect.



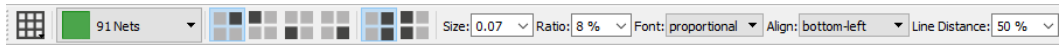
## Preparing The Schematic for Submission

- While not strictly necessary, a page frame makes the schematic look a lot neater.
  - Page frames are added like a circuit component, for a standard letter-size paper use the **frames**→**FRAME\_A\_L** component.
- You can document the value of passive components such as resistors with the "Value" tool. This doesn't have any real effect on the schematic, but makes it much clearer to anyone attempting to understand it.
- You can place arbitrary text using the "Text" tool. The default text size is pretty small, after typing your text in the edit box, press enter or click OK. Your text should be at the cursor, waiting for you to place it.





- While the text tool is active there is a secondary top toolbar that allows you to select the text layer, size and font style.



- When submitting the first part of the assignment, you will want to upload the “.sch” file. Eventually in the later parts of the assignment you will create board layouts which are contained in “.brd” files.

## 5. Adding Peripherals

You can add whatever peripherals, chips, or sensor you like to the board. The following are some standard ones that you will need on almost any embedded system.

Add at least one 0.1 $\mu$ F and one 1  $\mu$ F capacitor between your power rail. These are called decoupling capacitors (decap) and are used to smooth the voltage rail (so connect it between VCC and GND). If you use LEDs, then add another 10  $\mu$ F and 1  $\mu$ F capacitor onto the power rail for the LEDs.

Also make sure of adding test pins. It will be extremely useful during debugging to grip a scope probe or a pin of your logic analyzer.

## 6. PCB Layout

Once you are happy with your schematics, it is time to layout your PCB. Use the user manual, or other tutorials linked earlier to get started. You can generate a board layout file from a schematic with the “Generate/Switch to Board” button on the top toolbar.

If you generate a board file from a schematic, you will see the empty outline of a large rectangular PCB with all of the components you included in the schematic placed randomly on the outside edge. You’ll be arranging these and routing traces to complete the virtual “airwires” that indicate connections you made in the schematic.

You don’t have to complete a board layout for the first part of the assignment, just finish the schematic and submit the “.sch” file.

## 7. Deliverables

- EAGLE schematic file (.sch)
- PDF of the Schematic:
  - Make sure that the following information clearly appears on your Schematic sheet:
 

First name, Last name

UNID

Class section
  - Write a couple of sentences to comment on your design choice (like part selection for the capacitors and sense resistor)

**Submit your printed document through Canvas.**

**All files you submit to canvas should follow the convention**

**Section\_AssignmentName\_FileName.**