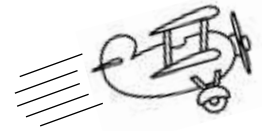## Lab Assignment 2—*"Prepare for Takeoff!"*
H-Bridge DC Motor Control

Overview:

This week you'll be using the **NodeMCU ESP-12E** to control a propeller attached to a DC motor. Specifically, you'll be interfacing the NodeMCU with an L293D H-bridge motor driver chip, and the L293D will actually control the motor. The propeller, motor, and L293D are all part of your lab kit.

Prelab – Cloning the Remote Lab02 Assignment Repository

You should have received an email about this assignment from your instructor. Accept the assignment and GitHub Classroom will create a private repository for you. Clone this repo into a local repo directory. Here's a reminder on how to do this:

1. From within the `EECE4263-S2022/lab02-YourGitHubUserName` repository, click the green ⬢ Clone or download ⬢ button. In the dropdown that appears copy the web URL that is shown (easy via the 📋 icon).

2. Open Git Shell and navigate to the `D:\EECE4263` directory on your removable drive.

3. Type `git clone` followed by the URL from Step 1., followed by `Lab02`. The command line should look like this:

```
D:\EECE4263> git clone https://github.com/EECE4263-S2022/lab02-YourGitHubName.git Lab02
```

This will create a new directory named `Lab02` inside your `D:\EECE4263` directory, make the new directory a Git repository, and populate it with the contents of the remote repo.

Background:

As you know, the NodeMCU ESP-12E board contains an ESP8266 microcontroller. And like most microcontrollers, it can only supply limited amounts of voltage and current via its output pins (definitely not enough to operate even a small DC motor). So for this lab you'll be using an **H-bridge** to control the motor. In general, an H-bridge is an electronic circuit that allows low voltage/current signals to cause large amounts of voltage/current to be delivered to a load. For this lab, the load will be the DC motor with propeller. The H-bridge comes as a single IC (part number **L293D**) manufactured by STMicroelectronics. Use the **L293D STMicroelectronics** datasheet (found in the Technical Documents folder in Canvas) to answer the following questions. When specifying operating parameters, **remember to include proper units!**

1. How many pins does the L293D have? _____

2. Assuming two channels per motor, how many motors can the L293D control? _____

3. What's the max <u>continuous</u> (not peak) current that can be supplied to *one motor*? _____

4. What's the minimum supply voltage, **V$_S$**? _____   The maximum? _____

5. What's the minimum logic supply voltage, **V$_{SS}$**? _____   The maximum? _____

6. The L293D is suitable for switching applications up to what frequency? _____

Lab Procedure:

Develop a system to control the **speed** and **direction** of a propeller attached to a DC motor given the following design specifications:

> DESIGN SPECIFICATIONS: Upon power-up or processor reset, the motor must be off (i.e. not rotating). A pushbutton should cycle the motor between three states: Off, Fast, and Slow (in that order). A second pushbutton should change the direction of spin. The default direction should cause air to flow *from the propeller toward the motor*. Pressing the direction button *whether or not the motor is spinning* should reverse the direction. Pressing **both buttons simultaneously** must stop the motor and reset its direction to the default. The L293D **load supply** (V_S, pin 8) must come from a **separate 5V source** (i.e. not the USB-supplied 5V). The L293D **logic supply** (V_{SS}, pin 16) can come from the NodeMCU's Vin (USB 5V) pin. Of course, **everything should have a common ground**.

SUNFOUNDER provides a manual of lessons with their "Super Kit" and the code is available on their website. Since the NodeMCU boards can be programmed using the Arduino IDE, the SUNFOUNDER code will make a great starting place. *But ONLY a starting place*, since the NodeMCU pins and capabilities are different from an Arduino. Refer to Lab01 for details on programming the NodeMCU using the Arduino IDE, and to SUNFOUNDER Lesson 7. Also, your assignment repo has an example program for **debouncing switches in software**. Very helpful!

Begin your design by creating a schematic. Include the L293D, the NodeMCU, the switches and the power sources. Properly annotate your schematic with pin numbers, voltage source names and values, etc. Remember to use the same 5V source to power the NodeMCU and the L293D *logic*, **but use a separate 5V source for the motor supply**. Schematically, this will just be two different voltage source names, e.g. V_{logic} and V_{load} , but will be the NodeMCU's USB power for the logic and the breadboard power supply (shown at right) for the motor supply. And don't forget that common ground!

Your schematic must be included with this lab sheet when you turn it in. I expect a **professional looking schematic**—something you could give a lab tech and have it built correctly **without coming back to you with questions**. You may use any schematic capture tool you like (even pencil and paper), but **I recommend Autodesk Eagle**. The Pro version is free for students and Eagle is used by electrical engineers world-wide. (Also, it will probably be useful for your senior design project!) Learn more at www.autodesk.com/products/eagle/overview.

Finally, since H-bridge motor control often depends on pulse-width modulation (PWM), here's a good reference for doing PWM on a NodeMCU ESP-12E using the Arduino IDE:

https://randomnerdtutorials.com/esp8266-pwm-arduino-ide/

Lab Report

This is a **one-week lab** and **no formal lab report required**. Instead, you must turn in a PDF of this sheet (including your professional-looking schematic) via Canvas, and store your final code in the Lab02 assignment repo on GitHub. Also, you must **demonstrate your working code** to your instructor at or before the beginning of our next lab. Be prepared to answer questions about your design (e.g. why you chose the pins you did, challenges you faced, what worked and what didn't, etc.)