Experiment C5
Actuators
Procedure

Deliverables: Signed score sheet, motor mount with holes drilled

Overview

In this lab, you will learn how to implement several common mechanical actuators: DC motors, servos, stepper motors, solenoid valves, and pneumatic cylinders.

Station A: DC Motors

Background

There are many different types of DC motors. In this lab, you will use a 12V geared motor.

Part 1 – Mechanical Assembly

1. Unscrew the two screws from the front of the motor, and remove the cover off gearbox. Place the screws somewhere safe. Be careful not to get grease everywhere!
2. Examine the gears. Will they increase the speed, torque, or both the speed and torque?
3. Replace the gearbox and tighten the two long screws.
4. Locate the 12V DC power supply; it looks like a laptop charger. Connect the motor to the 12V DC power supply to test that it still works.
5. Attach the mounting plate to the 80-20 rail using screws and T-nuts provided.
6. Mount the DC motor to the plate.
7. Mount the pulley to the shaft. Tighten the set screw onto the flat part of the shaft.

Part 2 – MOSFET PWM Circuit

1. Construct the MOSFET PWM circuit shown in Fig. 1. This is the same circuit from last week, but you will replace the resistive heater with the DC motor. You should use the external 12V DC power supply—not the one on the breadboard. Connect the negative terminal of the power supply to ground.
2. Use the function generator to produce the PWM signal via the "Pulse" function with a frequency of 500 Hz, a high value of 5V and low value of 0V. (Use the BNC T to monitor the PWM signal on the oscilloscope.)
3. Turn on the output from the function generator. Vary the duty cycle or pulse width on the function generator, and observe how it affects the motor speed.
4. Use the optical tachometer to measure the angular speed of the pulley. In your lab notebook, record the motor speed in RPM as a function of the % duty cycle.
5. Repeat the measurement using the Arduino to produce the PWM.

6. Note that the motor only spins in one direction. What if you need it to spin in both directions?
7. Disconnect the motor and disassemble the MOSFET circuit from the breadboard.

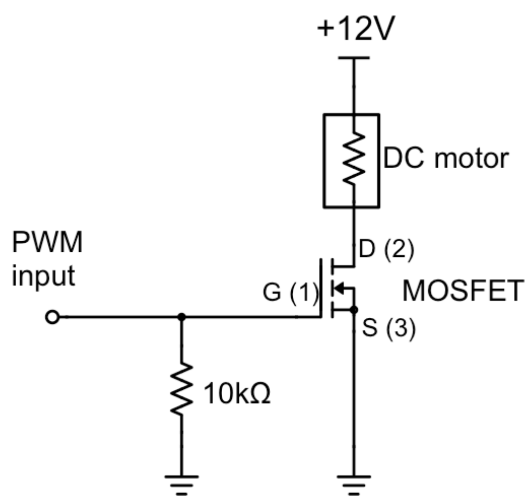


Figure 1 – The MOSFET PWM circuit from last week can be used to control the speed of a DC electric motor.

Part 3 – H-bridge Circuit

The H-bridge circuit shown in Fig. 2a is similar to the single MOSFET PWM circuit. By using four MOSFETS, it allows the motor to spin in either direction with speed controlled by PWM. This circuit is so widely used that it comes as an IC chip, the L298N, which is commonly sold pre-mounted to a "breakout board".

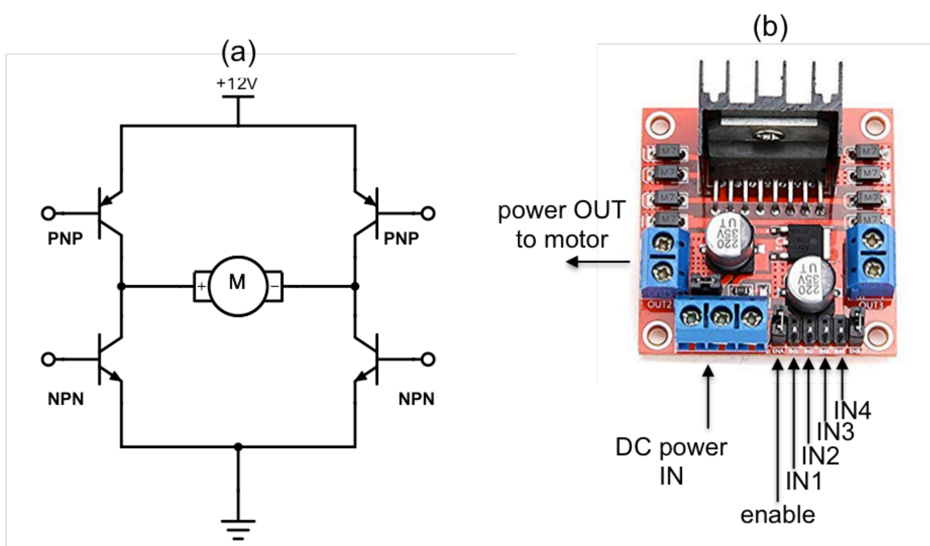


Figure 2 – (a) An H-bridge circuit uses four transistor to control which direction the motor will spin. (b) An L298N IC chip mounted to break-out board contains a more sophisticated version of the H-bridge circuit that also allows for speed control.

1. Locate the 12V DC power supply (not the one on the breadboard); it looks like a laptop charger.
2. Make sure the 12V DC power supply is unplugged from the AC wall outlet. Refer to Fig. 2b, and connect the DC motor to L298N.
3. Remove the plastic jumper connecting “enable” to “5V”. (A previous student might have removed them all ready.)
4. Use a couple strips of 22 gauge wire to connect the 12V DC power supply to the + and – bus lines on the small portable breadboard.
5. Connect the 12V DC and GND power input pins on H-bridge board to the + and – bus lines on the small breadboard.
6. Make the following connections with the Arduino and small breadboard:
   a. Arduino GND → – bus line on the small breadboard
   b. Arduino Pin 1 → IN1 on L298N
   c. Arduino Pin 2 → IN2 on L298N
   d. Arduino Pin 3 → Enable on L298N
7. Download and open the Arudino sketch “L298N_HBridge_template.ino”. Save it to your C5 folder with a new filename.
8. Open the code and examine it line by line. Identify which variables correspond to which pins on the Arduino and L298N.
9. Refer to the truth table (Table 1) for the L298N. Fill in the “***” in the main loop as either HIGH or LOW in the code.
10. Plug the 12V DC power supply into the wall outlet and run the code. When you have the code working, **demonstrate it to the lab instructor to receive credit**.
11. Make sure you have the Arduino sketch saved in the C5 folder of your code library. You will need it for future labs.
12. Return the lab bench to its initial state. Disconnect the Arduino, L298N, breadboard, and DC power supply.

Table 1 – The truth table for motor output A on the L298N.

Enable	IN1	IN2	OUTPUT
LOW	-	-	Stop
HIGH	HIGH	HIGH	Stop
HIGH	LOW	LOW	Stop
HIGH	LOW	HIGH	Forward
HIGH	HIGH	LOW	Reverse

Station B: Stepper Motors

Background

A stepper motor is very different from the previous DC motors. It uses multiple coils of wire that are sequentially switched ON and OFF by a "stepper driver" circuit. A permanent multi-pole magnet is fixed around the motors shaft. Turing a coil ON will cause the multi-pole magnet to rotate and align with the coil. Turning that coil OFF and its neighboring coil ON causes the magnet to rotate.

Part 1 – Mechanical Assembly

1. The stepper motor is a standard "NEMA 23" stepper motor, which means the motor case is 2.3" wide. Write this down in your lab notebook.
2. Make sure everything is unplugged. Remove the four long screws from the front of motor using the short, fat screwdriver. Place the screws somewhere safe.
3. Remove the aluminum cover to reveal the coils and multi-pole magnet on the shaft.
4. Remove the shaft and magnet assembly. Use the screwdriver to push on the shaft from the backside. Be careful not to lose the "wave spring" on the bottom.
5. Sketch the shaft assembly in your lab notebook. How many teeth are on the magnet? How does this compare to the number of steps per revolution?
6. Reassemble the motor. Replace the shaft assembly. (Don't forget the wave spring!) Replace the aluminum cover and screws.

Part 2 – Basic Operation

1. This particular stepper motor has two sets of coils. Each coil has two wires connected to it. Determine which pairs of wires are connected to a given coil. Use the handheld DMM to measure the resistance between two wires. A coil will have a resistance around 35 Ohms.
2. Write down which two color wires share a coil in your lab notebook.
3. Refer to the "Hook-up Guide" for the SparkFun Big Easy stepper driver. (http://learn.sparkfun.com/tutorials/big-easy-driver-hookup-guide/all) It will tell you what the various pins do.
4. Connect the motor wires to the stepper driver. (Refer to the Hook-up Guide for details.)
5. Locate the 12V DC power supply; it looks like a laptop charger.
6. Make sure the 12V DC power supply is unplugged from the AC wall outlet. Connect it to the GND and M+ screw terminals on the stepper driver.
7. Wire up the + and – bus lines on the large breadboard its +5V power supply and ground.
8. Connect the following pins on the stepper driver to ground on the breadboard: ENABLE, MS1, MS2, MS3, GND, DIR.
9. Use the function generator to drive the STEP pin. Use a 100 Hz "Pulse" with a high value of 5V and low value of 0V.

10. Put a piece of masking tape on the shaft of the stepper motor so you can see it spin.
11. Plug the 12V DC power supply into the wall outlet. Turn on the output from the function generator. The shaft should begin to rotate.
12. If the rotation is choppy and inconsistent, use the small flathead screwdriver to turn the white plastic "trim pot" on the Big Easy driver board. This will adjust the amount of current going to the coils.
13. Try different frequencies between 10 – 1000 Hz. Does the motor operate more smoothly at high frequencies or low frequency? Why do you think this is?
14. Change the DIR (direction) pin from ground to 5V on the breadboard. This should change the direction of rotation.
15. Disconnect the function generator and turn it off. Leave everything else connected to the breadboard.

Part 3 – Microcontroller Implementation

1. Disconnect the STEP pin from the function generator.
2. Disconnect the DIR pin from the breadboard.
3. Make the following connections with the Arduino:
    a. Arduino GND → grounded bus line on the small breadboard
    b. Arduino Pin 8 → DIR on stepper driver
    c. Arduino Pin 9 → STEP on stepper driver
4. Download the Arduino sketch "EasyDriver_Example1.ino". Examine the code and try to understand what it will do.
5. Run the sketch on the Arduino. You should see the stepper motor spin.
6. Modify the code to make the stepper spin in the opposite direction. (Change the value assigned to the DIR pin.)
7. Modify the code to make the stepper spin slower. Save it to the C5 folder in your code library.
8. Write a program that will take 5 seconds to move the shaft one full rotation clockwise, then 5 seconds to go one full rotation counter-clockwise. **Demonstrate it to the lab instructor to receive credit.**
9. Save all of your codes to the C5 folder in your code library.
10. Disconnect everything and return the set-up to its initial state.

Station C: Pneumatic Actuators

Safety glasses must be worn for this portion of the lab!

Background

A pneumatic cylinder uses pressure from a gas to drive a piston and rod, resulting in linear motion. Shown in Fig. 3, a 5-way solenoid valve is used to switch the compressed air P_{IN} between two outlets, P_1 and P_2. When $P_1 > P_2$, the rod will extend to the right. The rod will retract to the left when $P_1 < P_2$. The valve also has two vents that allow the pressurized air to quickly leave the cylinder when switching occurs.

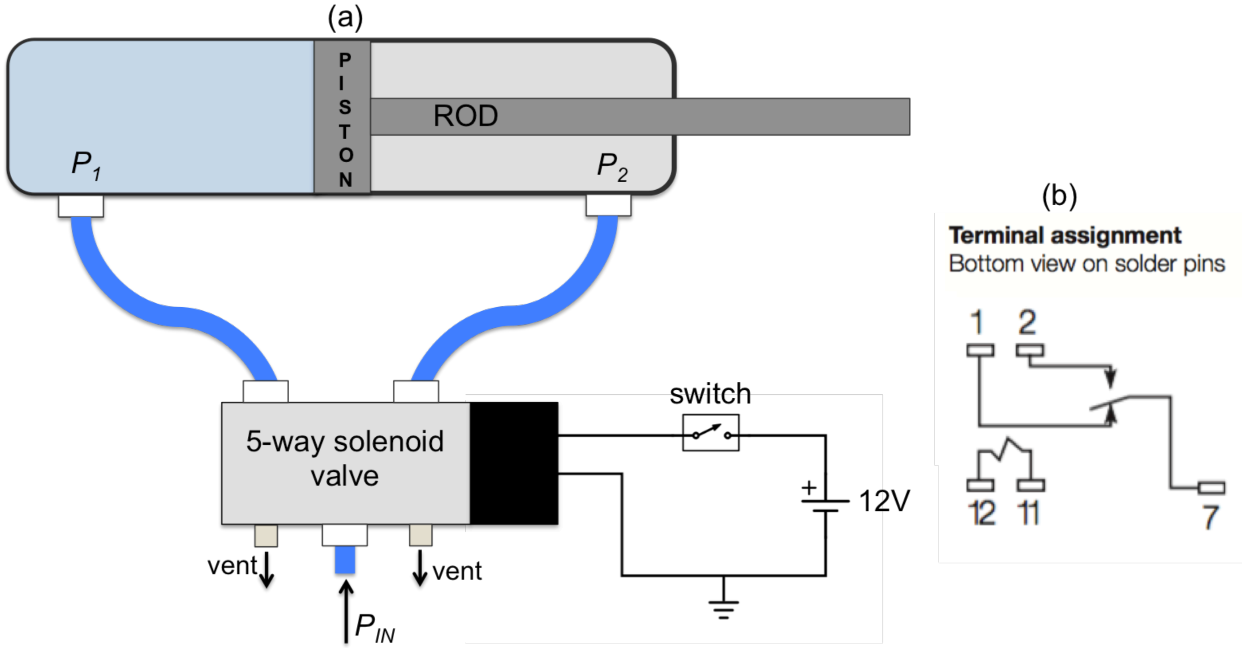


Figure 3 – (a) A pneumatic cylinder is controlled by a 5-way solenoid valve. (b) The pin-out for the T81 SPDT relay. Pins 11 and 12 are connected to the relay "coil".

Part 1 – Assembly

1. **SAFETY FIRST:** You must wear safety glasses during the portion of the lab. Keep the cylinder a safe distance away from you head or body.
2. Connect the blue 6mm OD tubes using the push connectors, as shown in Fig. 1.
3. Mount the cylinder in the beaker clamp close to the base. Make sure it is very snug, because the recoil of the rod and piston will cause it to move quite a bit.
4. Open the quarter-turn ball valve to pressurize the system.
5. Press the blue button on the solenoid to manually actuate the cylinder. Pretty cool, huh?
6. Use the breadboard and toggle switch to construct the circuit shown in Fig. 1a.
7. Test the circuit. Actuate the cylinder by flipping the toggle switch.

Part 2 – Microcontroller Implementation

You will now use the Arduino and an electrical **relay** to switch the solenoid ON and OFF. The relay is similar to the MOSFET transistor: it is a three terminal device that allows large amounts of power to be controlled using a small voltage. The small voltage drives a magnetic **coil**, and the magnetic field of the coil mechanically actuates a switch to turn the current ON and OFF.

1. Replace the toggle switch with the relay. Refer to the pin-out for the relay, shown in Fig. 3b. (Pins 2 and 7 on the relay are equivalent to the two wires on the toggle switch.)
2. Connect the Arduino ground to the breadboard ground.
3. Use the Arduino to "energize" the relay coil. Connect a digital pin on the Arduino to one end of the relay coil (pin 11). Connect the other end of the coil (pin 12) to ground.
4. Create a new Arduino sketch and save it in the C5 folder of your code library.
5. In the Arduino code "setup", declare the pin connected to the relay coil to be an output. In the main loop, write code to periodically actuate the cylinder every 3 seconds. That is, the cylinder should be extended for 3 seconds, then retracted for 3 seconds.
6. Demonstrate the code to the lab instructor to receive credit.
7. Make sure the Arduino sketch is saved to your code library.
8. Turn off the breadboard and disassemble the circuit.
9. Close the quarter-turn ball valve to depressurize the system. Remove all the blue tubes by pressing down on the black plastic ring at the base of the connectors.

Data Analysis and Deliverables

You do NOT have to write a tech memo for this lab. Rather, you have to go to the machine shop and create a motor mount using the bracket provided, and assemble a structure using 80-20.

1. Refer to the CAD for the DC motor in Appendix B. Drill holes in the 80-20 bracket to mount the motor, similar to how the DC motor at Station A was mounted. Assemble the 80-20 to make the inverted T structure shown in Appendix C. **You will use this apparatus for the C8 inverted pendulum lab.**
   - You will NOT be given the motor—only the drawing in Appendix B.
   - You must make a CAD drawing or a hand sketch on engineering paper of the hole pattern you plan to drill with dimensions in inches.
   - You may drill the holes yourself in the machine shop (if you are certified), or you may have the machinist do it for you.
   - Bring your machined motor mount, drawing, and 80-20 assembly to lab next week. You will be graded based on how well the motor fits.

Appendix A

Equipment

Station A

- 1ft x 1.5” 80-20 slotted rail
- Mounting plate with 5/16” T-nuts and bolts
- 12V DC motor
- 12V DC power supply (looks like a laptop charger)
- BNC to minigrabber cable
- M3 screws for motor mount
- Pulley w/ reflective tape
- Tachometer
- L298N H-bridge break-out board
- Arduino Microcontroller

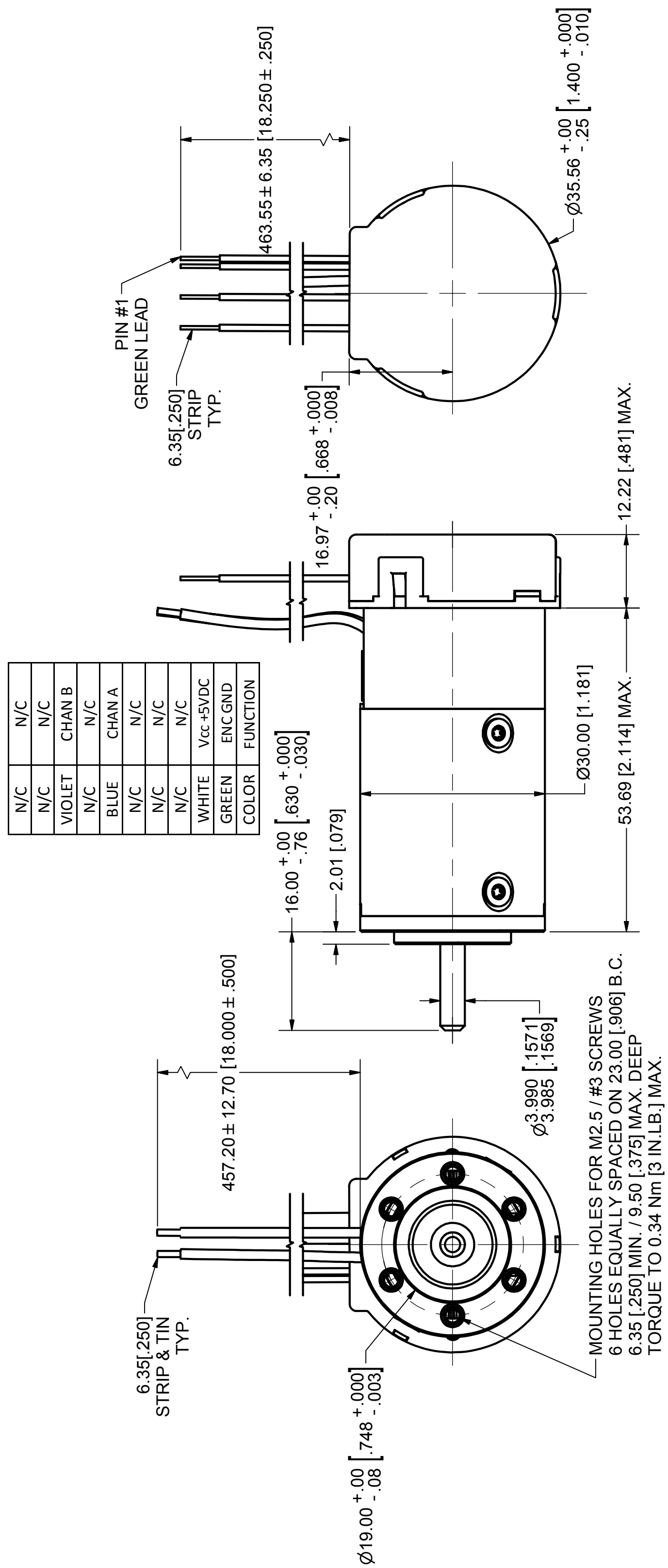
Station B

- Large breadboard
- 12V DC power supply (looks like a laptop charger)
- BNC to minigrabber cable
- Arduino Microcontroller
- SparkFun Big Easy stepper driver
- NEMA 23 stepper motor

Station C

- Large breadboard
- T81 SPDT relay
- Arduino Microcontroller
- Pneumatic cylinder
- Blue tubing
- Quarter turn ball valve

Appendix B



COLOR	FUNCTION
GREEN	ENC GND
WHITE	Vcc +5VDC
N/C	N/C
N/C	N/C
N/C	N/C
BLUE	CHAN A
N/C	N/C
VIOLET	CHAN B
N/C	N/C
N/C	N/C

Appendix C