
Experiment M2

Actuators

Procedure

Deliverables: Signed score sheet

Notes: Every lab bench will have the Station A: Solenoid Linear Actuators. Half the lab benches will have the Station B: DC motors and the other half will have the Station C: Pneumatic Cylinders. Students will have to switch lab benches to do either Station B or C.

Overview

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

Station A: Solenoid Linear Actuators

Background

As we saw in the first lab, a solenoid electromagnet can be used to move a ferromagnetic piece of metal. This effect can be used to make a linear actuator, and many applications including valves and electronic door latches use solenoid linear actuators. The Arduino UNO cannot output enough current to power the solenoids, so we will need to use transistors to amplify the digital signal from the Arduino.

Procedure

The solenoid will be powered by a 12V DC power supply.

Part 1 – Manual Testing

1. Connect the 5.5mm barrel connector on the 12V DC power supply to the inline kill switch. Connect the other end of the kill switch to the screw terminal adapter.
2. Insert red and black dupont pin jumper wires into the + and – screw terminals on the 12V output of the DC power supply. Use the screwdriver to tighten down the screw terminals.
3. Plug in the DC power supply. Use the handheld DMM to verify that the power supply works by measuring its voltage output. Flip the kill switch ON and OFF to verify that it works, as well.
4. Measure the electrical resistance of each of the solenoids. Record the values in your lab notebook.
5. Connect the 12V DC power supply to the + and – vertical bus lines on the breadboard, as shown in Fig. 1.
6. Individually test each solenoid. Plug each pair of wires into the + and – vertical bus lines on the breadboard. Briefly turn on the 12V DC power supply, and the solenoid should click when energized.

Part 2 – MOSFET Transistor Switching

The digital outputs from the Arduino cannot produce enough current to drive the solenoid, so we must use a metal-oxide-semiconductor field effect transistor (MOSFET) to amplify the signal.

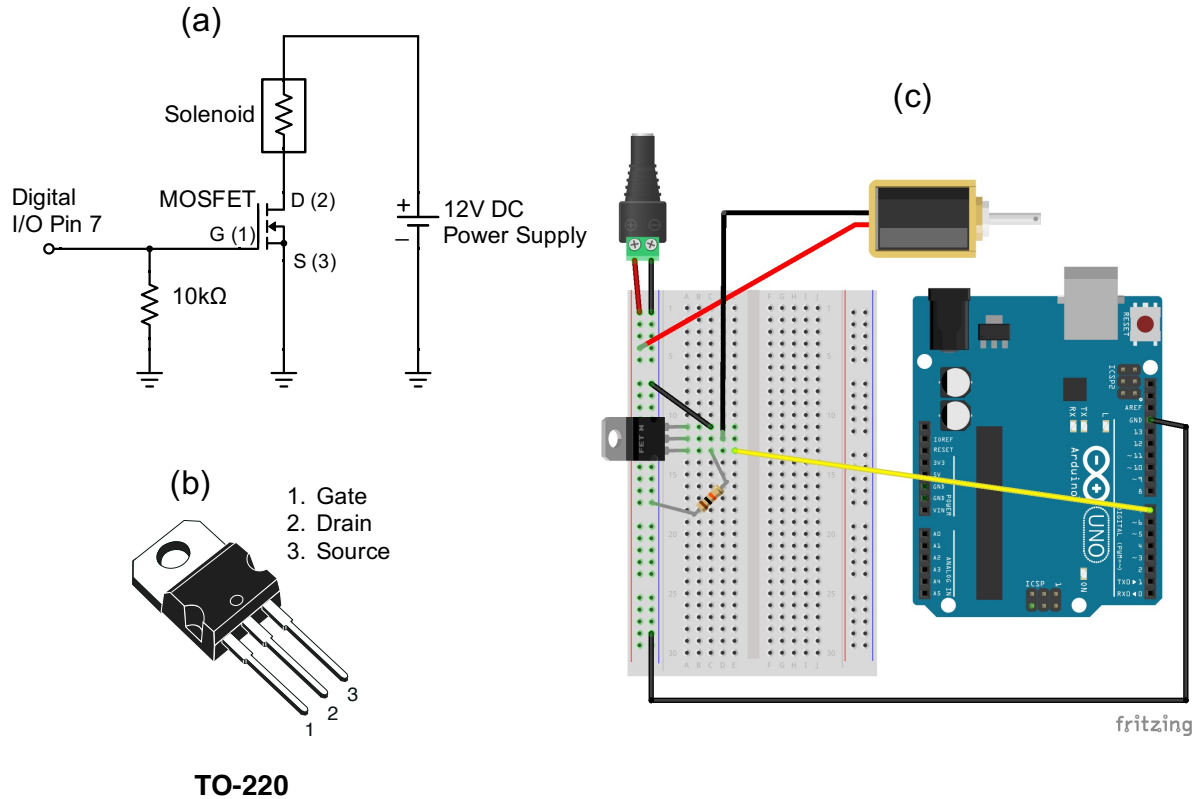


Figure 1 – (a) A circuit drawing for the MOSFET switch. (b) The pinout for the TO-220 N-channel MOSFET. (c) A Fritzing drawing of the complete electronic system.

1. Sketch the circuit in Fig. 1(a) and the pinout in Fig. 1(b) in your lab notebook.
2. Construct the MOSFET switching circuit shown in Fig. 1.
3. Manually test the MOSFET by connecting the gate (yellow wire in Fig. 1c) to 5V on the Arduino. This should switch ON the MOSFET and energize the solenoid.
4. Reconnect the gate to Digital I/O Pin 7 on the Arduino.
5. Download and open the Arduino sketch “Solenoid_TEMPLATE.ino”. Save it to your code library with a new filename.
6. Fill in the “***” in the code, so that the solenoid/MOSFET are repeatedly switched ON for 3 seconds, then switched OFF for 3 seconds.
7. Save the code and upload it to the Arduino. If it works, disassemble the circuit and move on to Part 3. If it does not work, think about what you did wrong.

Part 3 – Darlington Array

You will now use the Arduino to control two separate solenoids. Instead of constructing two separate MOSFET circuits, you will use a single integrated circuit (IC) chip that contains 8 individual transistor circuits on it. This chip, known as a Darlington driver, can drive up to 8 separate actuators.

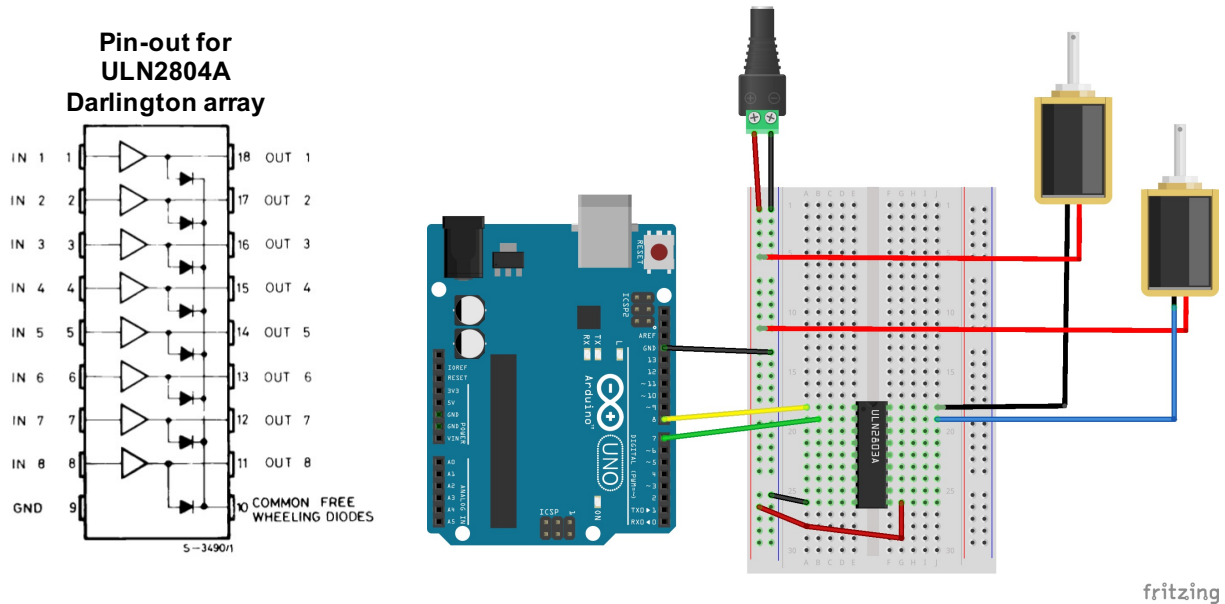


Figure 2 – (Left) The pinout for the ULN2804A Darlington array, which is used in the circuit (Right) to drive the solenoid linear actuators.

1. Use the breadboard to construct the circuit shown in Fig. 2. Make the following connections:
 - a. Arduino GND → GND bus line on breadboard
 - b. Darlington Pin 9 → GND bus line on breadboard
 - c. Darlington Pin 10 → +12V bus line on breadboard
 - d. Darlington Pin 1 → Digital I/O Pin 8 on Arduino
 - e. Darlington Pin 2 → Digital I/O Pin 7 on Arduino
 - f. Darlington Pin 18 → Solenoid 1 Black wire
 - g. Darlington Pin 17 → Solenoid 2 Blue wire
 - h. Solenoid 1 Red wire → +12V bus line on breadboard
 - i. Solenoid 2 Red wire → +12V bus line on breadboard
2. Modify your code from Part 2 to make the solenoids repeat the following sequence:
 - a. Solenoid 1 should be ON for 3 seconds, while Solenoid 2 is OFF for 3 seconds.
 - b. Both solenoids are OFF for 2 seconds.
 - c. Solenoid 2 should be ON for 3 seconds, while Solenoid 1 is OFF for 3 seconds.
 - d. Both solenoids are OFF for 2 seconds.

- e. Don't forget to setup both digital I/O pins to be outputs.
3. When you have the system working, demonstrate it to the TA to receive credit on your score sheet.
4. Disassemble the circuit. Save the code to your code library with an intelligent filename.

Station B: 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. 3. This is the same circuit you used with the solenoid, but you will replace the solenoid with a DC motor.
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.

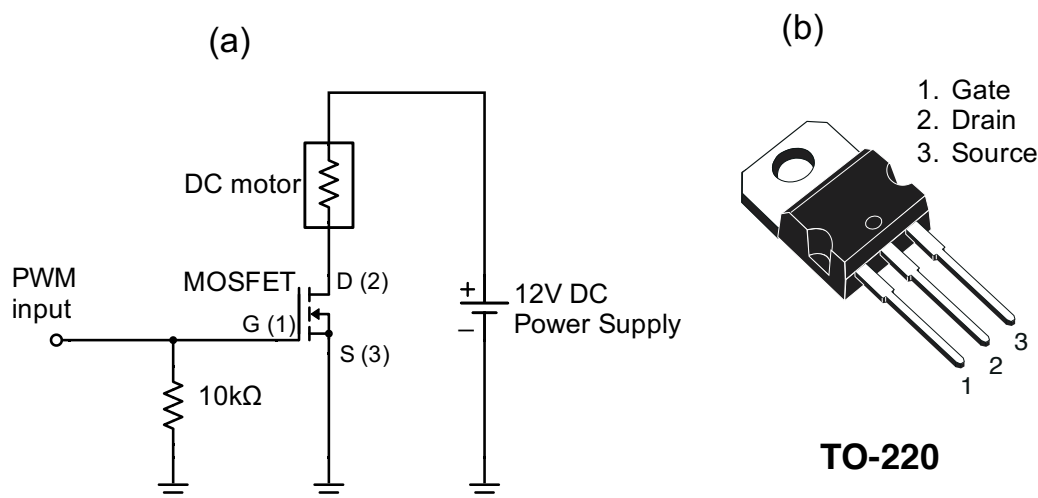


Figure 3 – The MOSFET PWM circuit from last week can be used to control the speed of a DC electric motor.

Part 3 – H-bridge Circuit

The H-bridge circuit shown in Fig. 4a is similar to the single MOSFET PWM circuit you just created. By using four MOSFETS, it allows the motor to spin in either direction with speed controlled by PWM. This circuit is the basis of any DC motor driver board, such as the Cytron MD10C shown in Fig. 4b.

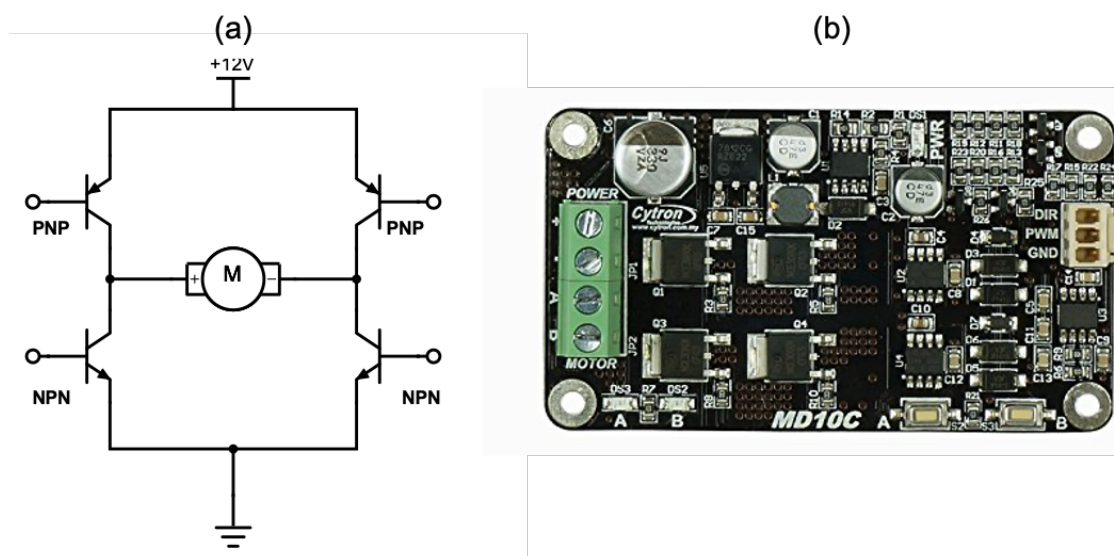


Figure 4 – (a) An H-bridge circuit uses four transistors to control which direction the motor will spin. (b) A Cytron MD10C motor driver board contains an H-bridge circuit that allows for direction and speed control.

Procedure

1. Locate the 12V DC power supply (not the one on the breadboard); it looks like a laptop charger. Make sure it is NOT plugged in to the AC wall outlet yet.
2. Use red and black jumper wires to connect the 12V DC power supply to the + and – screw terminals on the Cytron motor driver board.
3. Connect the DC motor wire to the A and B screw terminals on the Cytron.
4. Make the following connections with the Arduino and Cytron:
 - a. Arduino Digital GND → Cytron GND
 - b. Arduino Digital Pin 5 → Cytron DIR
 - c. Arduino Digital Pin 3 → Cytron PWM
5. Download and open the Arduino sketch “Cytron_template.ino”. Save it with a new filename.
6. Open the code and examine it line by line. Fill in the “***” in the main loop as either HIGH or LOW in the code.
7. 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.**
8. Make sure you have the Arduino sketch is saved in your code library. You will need it for future labs.
9. Return the lab bench to its initial state. Disassemble the motor mounting bracket and circuit.

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.

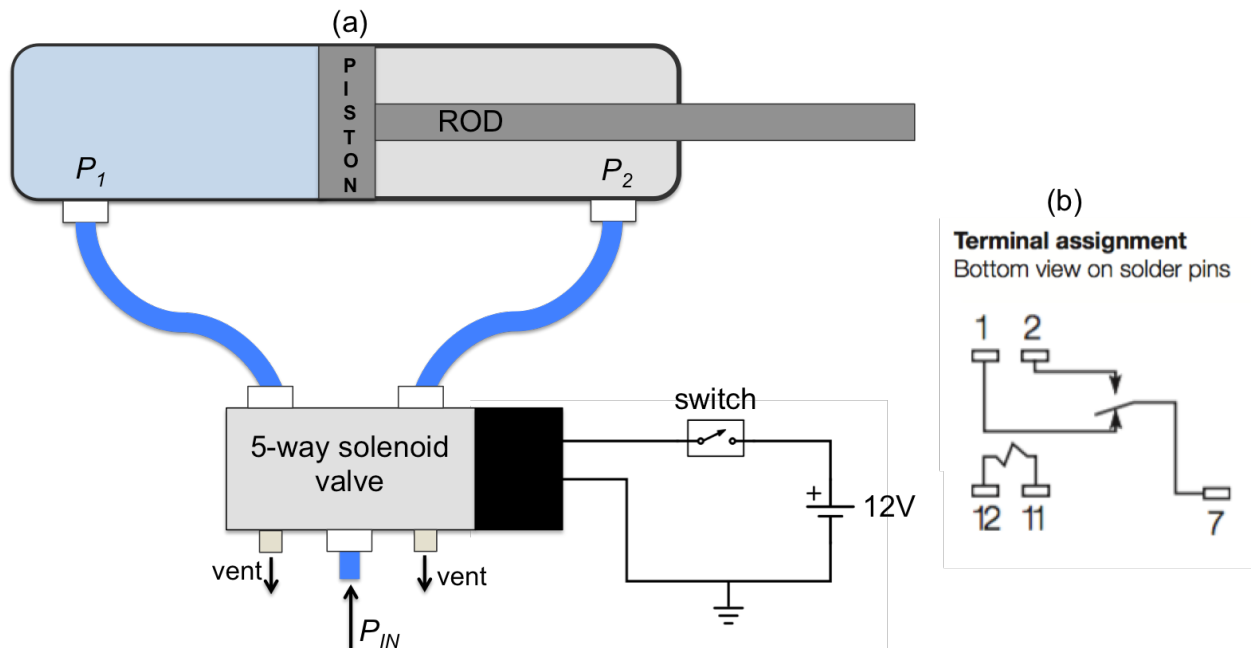


Figure 5 – (a) A pneumatic cylinder is controlled by a 5-way solenoid valve. (b) The pin-out for the T81 SPDT relay. Pins 11 and 12 are connected to the relay “coil”.

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. 3.
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, 12V DC power supply, and in-line kill switch to construct the circuit shown in Fig. 5.
7. Test the circuit. Actuate the cylinder by flipping the 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. Implement the relay into the circuit, so it switches the solenoid valve ON and OFF. Refer to the pin-out for the relay, shown in Fig. 3b.
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 to 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.

Deliverables

You do NOT have to write a tech memo for this lab. Just make sure the TA fills out the score sheet as you complete each part of the lab.

Appendix A

Equipment

Station A

- Intermittent Duty Solenoid Open Frame (Pull) Type 0.394" (10.00mm) Stroke 12VDC Chassis Mount – Digikey Part # 1528-1552-ND
- Intermittent Duty Solenoid Closed Frame (Pull) Type - Stroke 12VDC Chassis Mount – Digikey Part # 1528-1191-ND
- Darlington transistor (BJT) array – Digikey Part # 497-2357-5-ND

Station B

- 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 C

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