Fluid Power Laboratory
Module 1

Pneumatic Gripper Actuation and Control

(using Arduino Microcontroller)

Report: Answer the green bolded questions for report submission. Turn in one report per group and upload a video of your working system, and make sure to write your names (first and last name), lab topic, and section day and time.

Lab objectives

1. Develop an understanding of the mechanical design and assembly of a light-duty gripper
2. Get introduced to electric circuit design and electromechanical hardware for the gripper's control
3. Learn how to control the opening of the gripper by controlling a pneumatic pump (6V DC mini air pump motor) using Arduino Microcontroller

Discussion

Students can work in **groups of 3 or 4** to **assemble the light-duty gripper** shown in *Figure 1*. After assembling the gripper, students are required to **code and design an electric circuit** to control the opening of the gripper. This will be achieved by controlling a mini air pump using a push button and an **Arduino microcontroller**. The goal is to actuate the gripper (*Figure 2*) by extending and retracting the syringe/piston via the control of an air pump directly connected to the syringe.

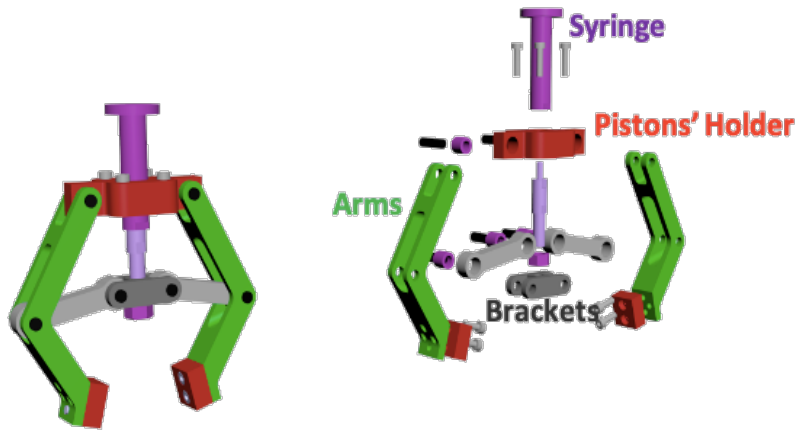


Figure 1. CAD model of a light-duty gripper

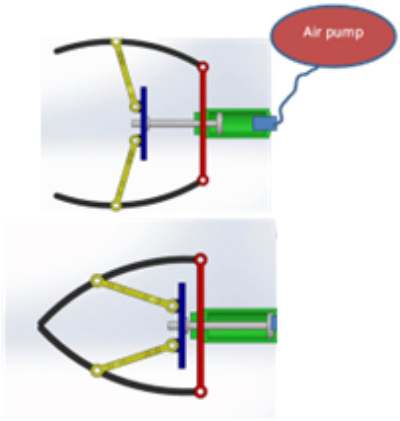
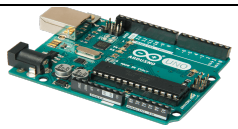
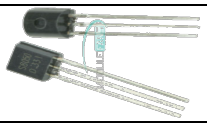
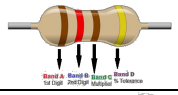
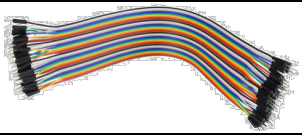
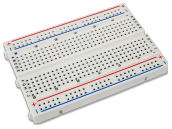


Figure 2. Actuation of the gripper via air pump

The components presented in *Table 1* will be provided for designing the circuit.

Table 1. Utilized components for circuit design

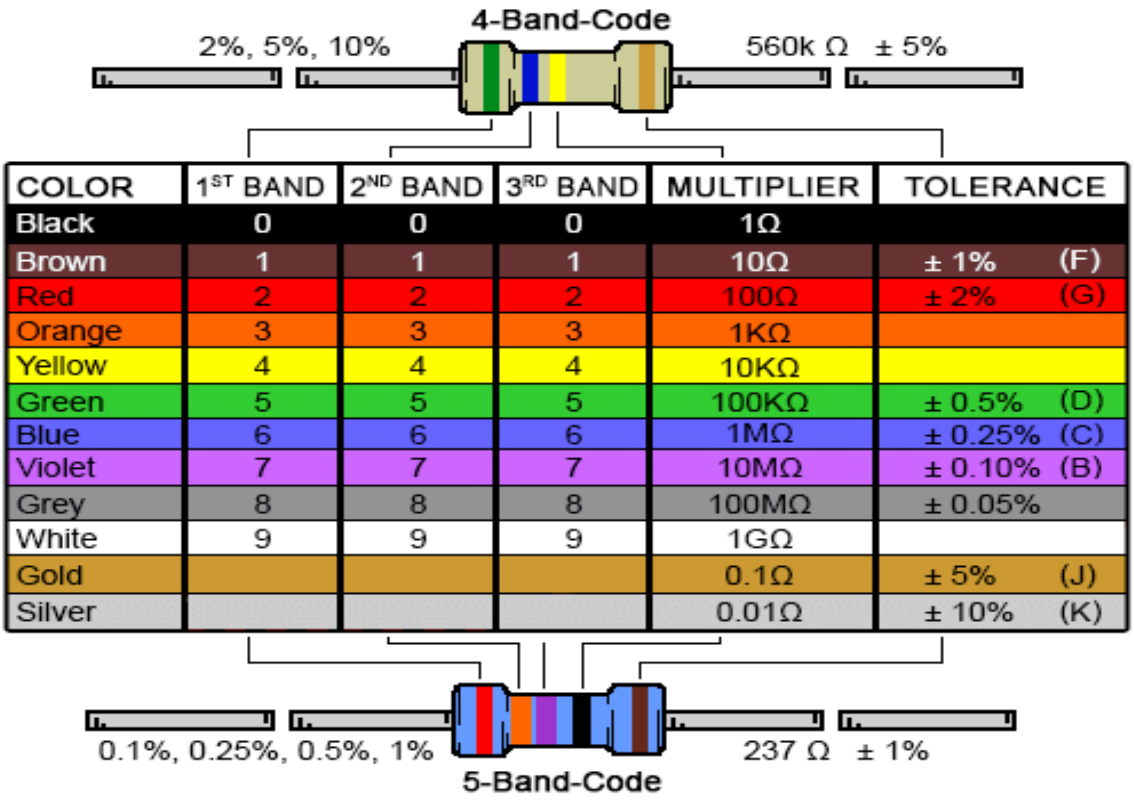
Arduino Uno	
DC 6V Mini Air Pump Motor	
Push Button	
LED	
S8050 NPN Transistor 0.5A Transistor	
1 kilo-ohm and 220-ohm Resistors	
Jumper wires	
Breadboard	

Identifying Resistors

Table 2 shows a resistor identification chart. Typically, resistors have either four or five colored bands on them, which can be used to identify their resistance. For a four-band resistor, the first band's color represents the 100-place value, the second band represents the 10-place value, and the one place value will always be zero. The third band's color represents the multiplier, and the final band represents the tolerance. For example, a 1kΩ resistor with a 5% tolerance would have the bands black-brown-red-gold.

For five-band resistors, the first two bands have the same aforementioned meaning, but the third band represents the ones-place value, which in this case can be any value, not just zero. The fourth band is the multiplier and the final band remains the tolerance. Thus, a five-band 1kΩ resistor with a 5% tolerance would have the color bands brown-black-black-brown-gold. Ultimately, five-band resistors can have more precise resistance values than four-band resistors.

Table 2: Resistor Identification Chart [1]



COLOR	1ST BAND	2ND BAND	3RD BAND	MULTIPLIER	TOLERANCE	
Black	0	0	0	1Ω		
Brown	1	1	1	10Ω	± 1%	(F)
Red	2	2	2	100Ω	± 2%	(G)
Orange	3	3	3	1KΩ		
Yellow	4	4	4	10KΩ		
Green	5	5	5	100KΩ	± 0.5%	(D)
Blue	6	6	6	1MΩ	± 0.25%	(C)
Violet	7	7	7	10MΩ	± 0.10%	(B)
Grey	8	8	8	100MΩ	± 0.05%	
White	9	9	9	1GΩ		
Gold				0.1Ω	± 5%	(J)
Silver				0.01Ω	± 10%	(K)

Procedure (Assembly, electric circuit design, and coding)

Before starting the lab, it is required **to download and install Arduino and Fritzing provided on Brightspace.** Fritzing is optional; it is used for developing the circuit schematic. **Please, read carefully.**

1. **Mechanical assembly**

1.1. Assemble the gripper using the provided components (arms, brackets, screws, syringe)

1.2. Test the system by manually actuating the syringe (testing if the gripper is opening)

1.3. Provide a photo of the assembled working gripper with the components' names labeled (see *Figure 3)*

Figure 3. Photo showing the gripper after assembly

2. Electric circuit design

2.1. Understand the breadboard configuration (*Figure 4*).

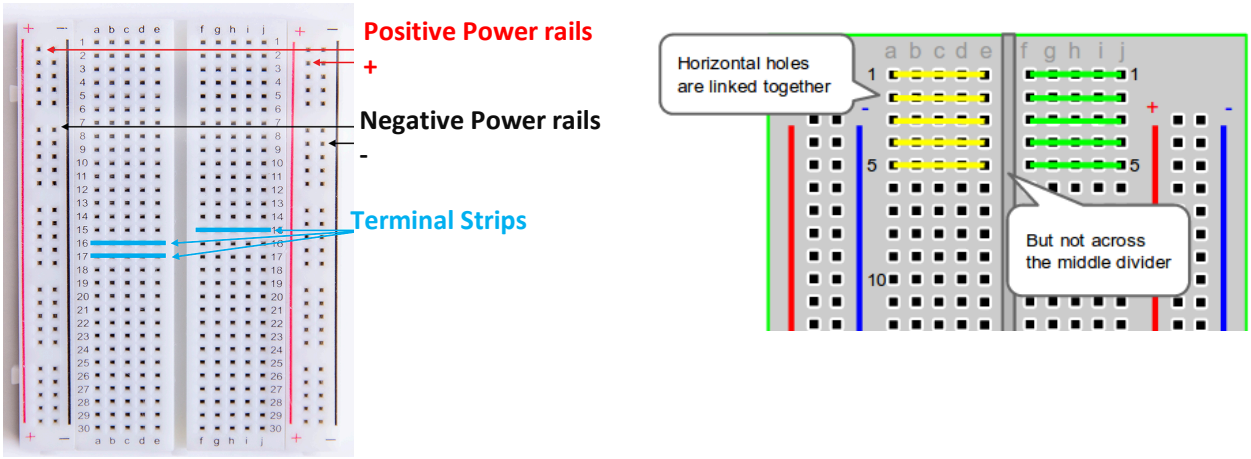


Figure 4. Breadboard configuration

2.2. In two stages, build the electric circuit using the provided components (see *Table 1*).

Stage 1: Connect the LED and the push button to Arduino according to *Figure 5* (for internal wiring and terminal configuration or polarity of the LED and push button, see *Figure 6*).

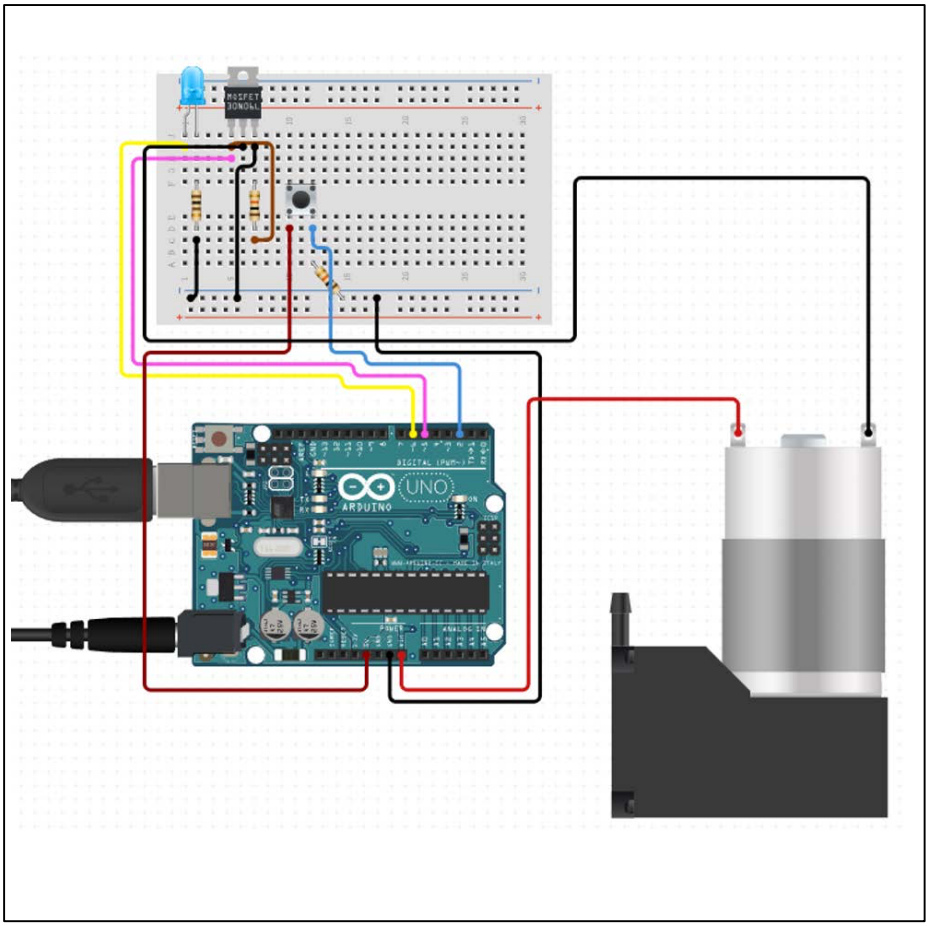


Figure 5: Module 1 Circuit Diagram

Provide a photo of the first part of your circuit (LED and push button connected to Arduino), and draw the corresponding circuit schematic.

Hint: You could draw the schematic on paper or using fritzing software (provided within the Lab folder).

P.S: You can use other digital pins on the Arduino micro-controller.

Figure 6: LED and push button wiring

Stage 2: Connect the air pump to Arduino on the same Breadboard with the LED and control button.

The air pump will be connected to Arduino via an NPN transistor (*Figure 5*) to control its on/off signal through the Arduino's digital pin that will be controlled via the push button, i.e., without a transistor, the pump will always be on.

Provide a photo of the second part of your circuit (pump connected Arduino via a transistor), and draw the corresponding circuit schematic.

2.3. Review your circuit. **It is advised to get it checked by your instructor to avoid overloading or damaging parts. Provide a picture of the whole circuit, labeling the basic components in the circuit.**

3. Coding and testing

3.1. Start by defining your input (button) and outputs (led and pump) as integers.
Example:
int LED=(specified digital pin)

3.2. In the void setup, define the mode of those integers (input/output).
Example:
pinMode (LED, OUTPUT);

3.3. In the void loop, write a code for achieving the following:

Press the push button to turn the LED on and actuate the air pump, i.e., (when push button is on, the pump and the LED should be on and vice versa).
Hint: Use *digitalRead()* and *digitalWrite(),* e.g., *digitalWrite (LED, HIGH)* activates the LED.

Table 2. Table showing the output of the coding exercise

Input	Push button	ON	OFF
Output 1	Air pump	on	off
Output 2	LED	on	off

3.4. After writing the code, compile it to check if there are any errors in the code.
3.5. After reviewing and checking your code, upload it into Arduino to start testing. **Upload your code in the lab submission.**
3.6. Connect the syringe to the pump through a hose and try to open the gripper using the push button. You can close the gripper manually or using a rubber band.
3.7. **Upload a video with the report showing how the gripper is operating.**

Note: An example code can be found in the Appendix.

More information about how to use and code Arduinos can be found here:
https://www.arduino.cc/en/Guide

Report

Please, answer green bolded questions. Turn in one report per group, attach your code, and upload a video of your working system. Make sure to write your names (first and last name), lab topic, and section date and time.

Thank you

Citations

[1] "Resistor Color Code Calculator - 4 band, 5 band, 6 band | DigiKey Electronics," *www.digikey.com*. https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code#modal

Appendix

```
int LED = 4; // specifying LED as pin 4
int pushBtn = 5; // specifying push button as pin 5
int pump = 6; // specifying motor as pin 6

void setup() {
  pinMode(LED, OUTPUT); // define LED as output
  pinMode(pushBtn, INPUT); // define pushBtn as input
  pinMode(pump, OUTPUT); // define pump as output
}

void loop() {
  int btnState = digitalRead(pushBtn); // define button state as a variable
  if (btnState == HIGH) { // determines if button is being pushed
    // LED and pump turn on
    digitalWrite(LED, HIGH);
    digitalWrite(pump, HIGH);
  }
  else {
    // LED and pump are off
    digitalWrite(LED, LOW);
    digitalWrite(pump, LOW);
  }
}
```