Fluid Power Laboratory
Module 5

Pressure measurement on a pneumatic gripper using a pressure transducer

(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. Consolidate previous learnings in fluid power data acquisition systems, instrumentation, and programming methods
2. Lean how to establish an I2C communication between an LCD screen and Arduino
3. Explore possible hardware factors that can affect a pneumatic system

Discussion

Students can work in **groups of 3 or 4** to complete this lab. The aim is to combine findings from the previous 4 labs to measure the **pressure exhibited at the pump and pneumatic gripper** using the **pressure transducer** and display it on an **LCD screen**. In addition, a pressure gauge is incorporated into the pneumatic system to verify the pressure readings displayed on the LCD.

The lab kit includes all necessary components from previous modules such as the assembled gripper, breadboards with connected components (push button, LED, resistors, transistor), the pressure transducer with the associated fittings, and other components listed in *Table 1*. You shall use 1 Arduino Uno microcontroller or equivalent microcontroller board.

Table 1. Utilized components for system and circuit design

Arduino Uno	
LCD Display with I2C communication	
DC 6V Mini Air Pump Motor	

Jumper wires	
Pressure gauge (0-60 psi)	
Hoses (1/8 diameter or syringe fitting)	
Tee fitting (1/8)	
Assembled pneumatic gripper	
Pressure transducer with fittings	
Rubber bands	

Procedure (Electric circuit design, Coding, Testing)

Note: Before starting the lab, it is required **to download, and it is recommended install Arduino and fritzing provided on bright space.** Fritzing is optional; it is used for developing the circuit schematic.

1. **Electric circuit design**

   In this first section, you will work on building an electric circuit that combines the control of the pneumatic gripper with the data acquisition from the pressure transducer, while using 1 Arduino microcontroller.

   Hint: You may use or refer to previously built circuits if possible.

   Step 1: Connect the air pump to Arduino with the LED, push button, resistors, and transistor on the same breadboard.

   Step 2: On the same breadboard, connect the pressure transducer following the wiring conventions shown in previous labs.

   Step 3: On the same breadboard, Connect the LCD display to Arduino as per the below wiring convention (*Figure 1*).

The CLK pin of the LCD corresponds to the clock pin and the SDA pin refers to the data acquisition pin.

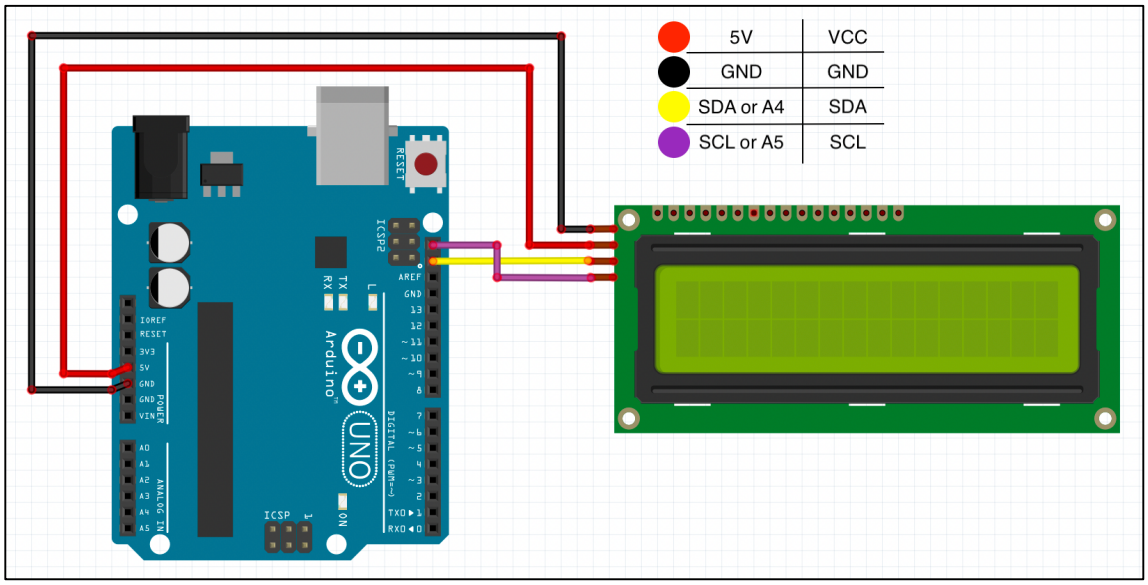

Figure 1. LCD display wiring

1.1. Review your circuit. **Provide a picture of the whole circuit, labeling the basic components in the circuit.**

1.2. **Draw the corresponding circuit schematic.**

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

Note: Keep track of the digital and analog pins you are using for each component on Arduino.

2. Coding

2.1. Retrieve the codes you have developed in the **"Pneumatic gripper actuation and control"** lab and **"Data acquisition and calibration of a pressure transducer – Part 2"** lab.

Note: Change the baud rate to 9600.

2.2. Combine both codes by copying the code initialization, the void setup, and the void loop sections.

Note: Update the analog and digital pins based on your new circuit.

2.3. **Explain the goal of the combined code.**

2.4. Next, you will learn how to display results on the LCD screen.

- In the code initialization, call the functions that allow interfacing with the LCD screen over I2C communication, as such:
  #include "Wire.h"
  #include "LiquidCrystal_I2C.h"

- Also, in the code initialization, set the LCD I2C communication address; format (address, columns, rows) as:
  LiquidCrystal_I2C lcd(0x27, 16, 2);
- In the void setup, initialize the LCD connection by including the following lines:
  lcd.init();
  lcd.backlight();
- In the void loop, add a code that writes and displays the voltage (V) and pressure (psi) values with units and text on the LCD screen. You may use the following functions:
  lcd.setCursor(0,0); //sets cursor to column 0, row 0
  lcd.setCursor(0,1); //sets cursor to column 0, row 1
  lcd.print("Text"); //prints label
  lcd.print(Value, 1); //prints a value to lcd screen, 1 digit on float
  lcd.print(" "); //to clear the display after large values or negatives
  Note: Ensure that the void loop still ends with the sensor delay command.

2.5. After writing the code, compile it to check if there are any errors in the code.

2.6. After reviewing and checking your code, upload it into Arduino to start testing. **Upload your code in the lab submission.**

An example code can be found below for reference (**NOTE: it will need to be modified in accordance with your circuit, experiment values, and specific desired outputs)**:

```
// include necessary libraries
#include "Wire.h"
#include "LiquidCrystal_I2C.h"
// initialize pin locations
int LED = 4;
int pushBtn = 5;
int pump = 6;
int sensorInput = A0;
int sensorReadDelay = 500;
float pressureValue = 5;
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
 // put your setup code here, to run once:
  // define inputs and outputs
  pinMode(LED, OUTPUT);
  pinMode(pushBtn, INPUT);
  pinMode(pump, OUTPUT);
  Serial.begin(9600);
  pinMode(sensorInput, INPUT);
  // initialize LCD connection
  lcd.init();
```

```
  lcd.backlight();
}

void loop() {
 // put your main code here, to run repeatedly:
  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);
  }
   analogRead(sensorInput);
   float pressureValue = analogRead(sensorInput); //set pressure value equal to sensor input and
   convert to volts
   voltageValue = pressureValue/204.6;
   Serial.print(voltageValue);  //display voltage in serial monitor
   Serial.println();   //move to new line
   delay(sensorReadDelay); //delay sensor reading

   // the lines of code written based off the Module 4 results which display the voltage and
      pressure readings will need to be added as well

   // print voltage and pressure readings onto LCD
   lcd.setCursor(0,0);
   lcd.print("Voltage (V): ");
   lcd.setCursor(0,1);
   lcd.print(voltageValue, 1);
   delay(1000);
   lcd.print(" ");
   lcd.setCursor(0,0);
   lcd.print("Pressure (psi): ");
   lcd.setCursor(0,1);
   lcd.print(pressureValue, 1);
   delay(1000);
   lcd.print(" ");
}
```

3. Testing using a pressure gauge

Once the electric setup and the code are completed, you will test the whole system by measuring the pressure at the syringe attached to the pneumatic gripper using the pressure transducer.

3.1. First, start by building a 3-way pneumatic subsystem by connecting the pressure transducer with the pump and the pressure gauge using a Tee fitting and hoses. Connect the other outlet of the pressure gauge to the syringe of the gripper via a hose. Ensure minimal air leaks and secure connections.
3.2. Press the push button to open the gripper. **Provide a video of the working system.**
3.3. **What are the voltage and pressure values displayed on the LCD screen?**
3.4. **Compare the pressure from the LCD screen with the one displayed on the pressure gauge. What do you notice and why?**
3.5. **Explain and justify how changing the following factors affects the pressure measured and the opening of the gripper:**
- **Using a 12V mini air pump**
- **Using longer hoses**
- **Using a larger or smaller syringe at the gripper**
- **Tightening bolts at the gripper joints**

3.6. Attach a rubber band linking the lower joints of the gripper indicated by red arrows on *Figure 2*. **What do you observe? Why?**
3.7. Gradually attached more rubber bands until the gripper fails to open. **What is the maximum number of rubber bands attained? Explain and suggest a possible solution.**

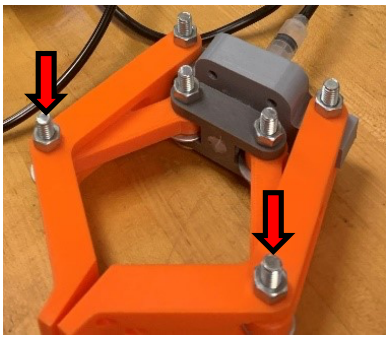

Figure 2. Indication for rubber bands linkage

Report

Please, answer green bolded questions. Turn in one report per group, upload the required photos and videos, and attach your code. Make sure to write your names (first and last name), lab topic, and section date and time.

Thank you