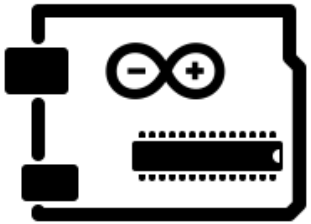
Pneumatic Flow Rate Regulation System

Consultants: RJ Weld & Kishan Patel

Client: Dr. Peter Galie

December 10, 2018

Overview



Scope and Motivation

Project Constraints and Requirements

Budget and Resources

System Design

System Operation

Future Work

Conclusions

Scope and Motivation

Started with a Previously Broken Project

Original open-source code converted for Dr. Galie's use but broken in the process

Create GUI in Python for User Controls

Allows user to adjust output waveforms

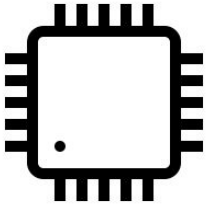
Code Firmware on Arduino for System Control

Takes user input data and adjusts signals on system

System Controls Air Flow Rate from Lab Benches

Enables more controlled lab experiments

Project Constraints and Requirements



Had to abide by Arduino and other hardware restrictions

Very user friendly setup and usage

System was already bought, did not want to make too many changes to components needed

Budget and Resources

Base System Components Already Bought
All components from open-source bill of materials

Had to Replace Industruino
Did so with Arduino UNO

Simple Circuit to Increase Arduino Output
Because transducer input requirements previously met by Industruino specs

Appended Original Bill of Materials
All original components already bought, new required hardware recorded

System Design

Create Easy-to-Use GUI

With Python 3 which sends data via serial port to Arduino

Communication via Serial Port

Python writes to Arduino read buffer and vice versa

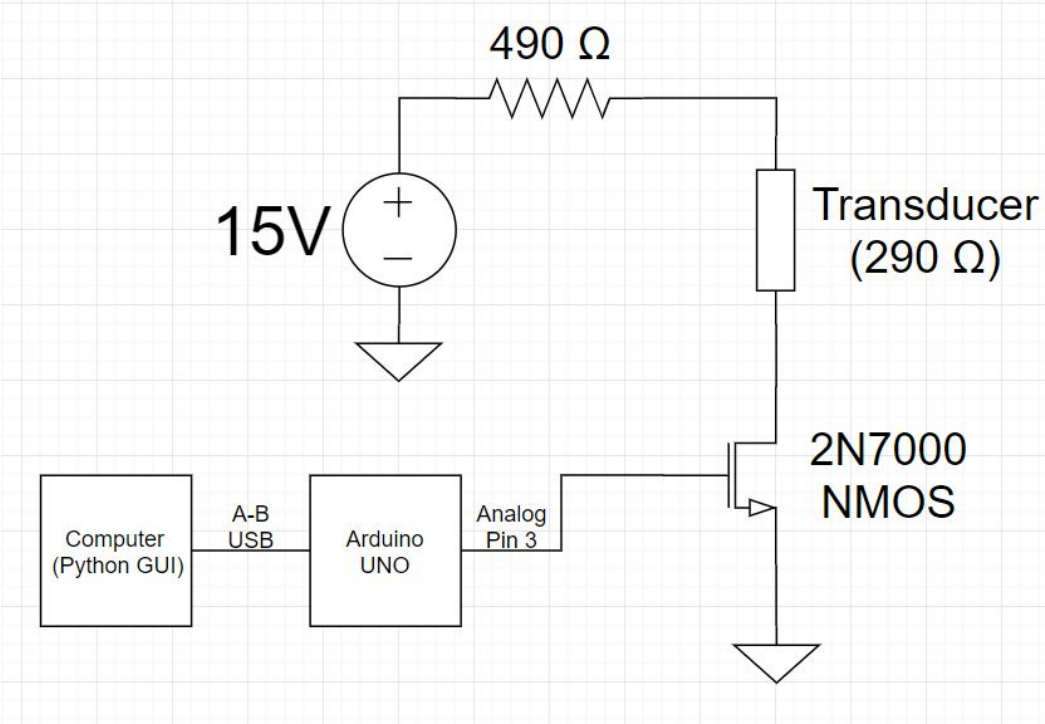
Utilize an Arduino to Control Pump

Arduino takes python data, and produces a pressure via the transducer

Amplification of Voltage

MOSFET low-side switch design to step up voltage

System Design



System Operation

Multiple Output States

Can create constant (DC), pulse (50% PWM) and ramp waveforms

High and Low DAC Outputs

8-bit DAC (0 - 255 on GUI) mapped to 4mA - 20mA output

Variable Output Period

For pulse and ramp outputs, period can be changed from 0s - 1s

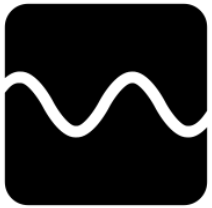
System Operation

First-Time and One-Time Configuration Setup

User is prompted to input serial COM port Arduino is found at

GUI Pops Up in Default State

Default state is constant output set to 0V DAC



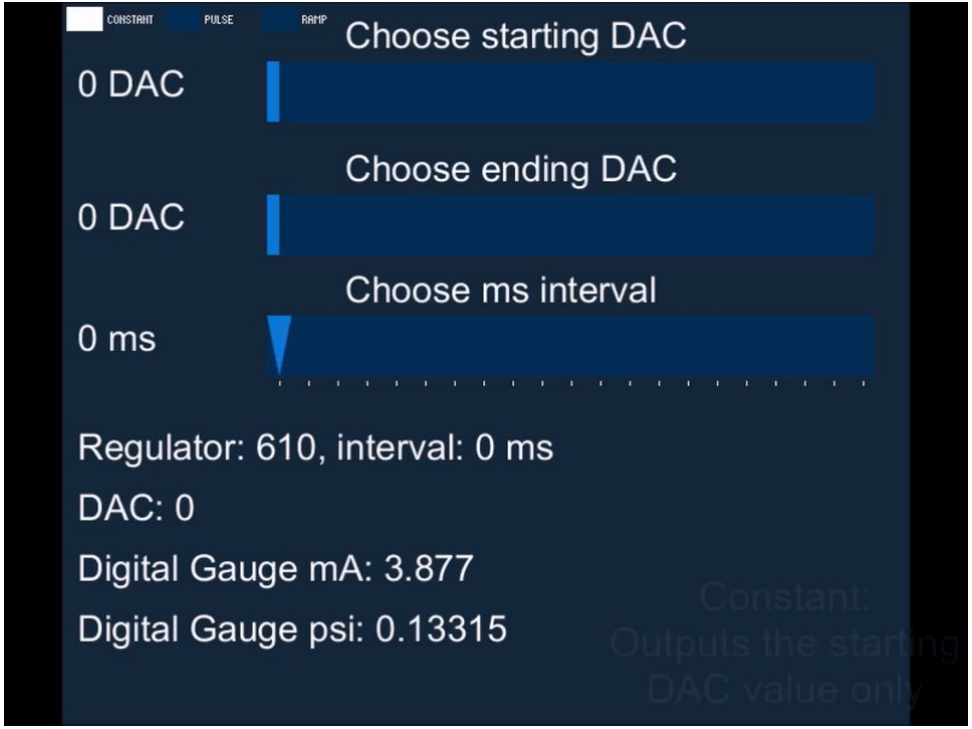
User Modifies Output with Slider Bars

Python read/writes serial port at rate of 10Hz

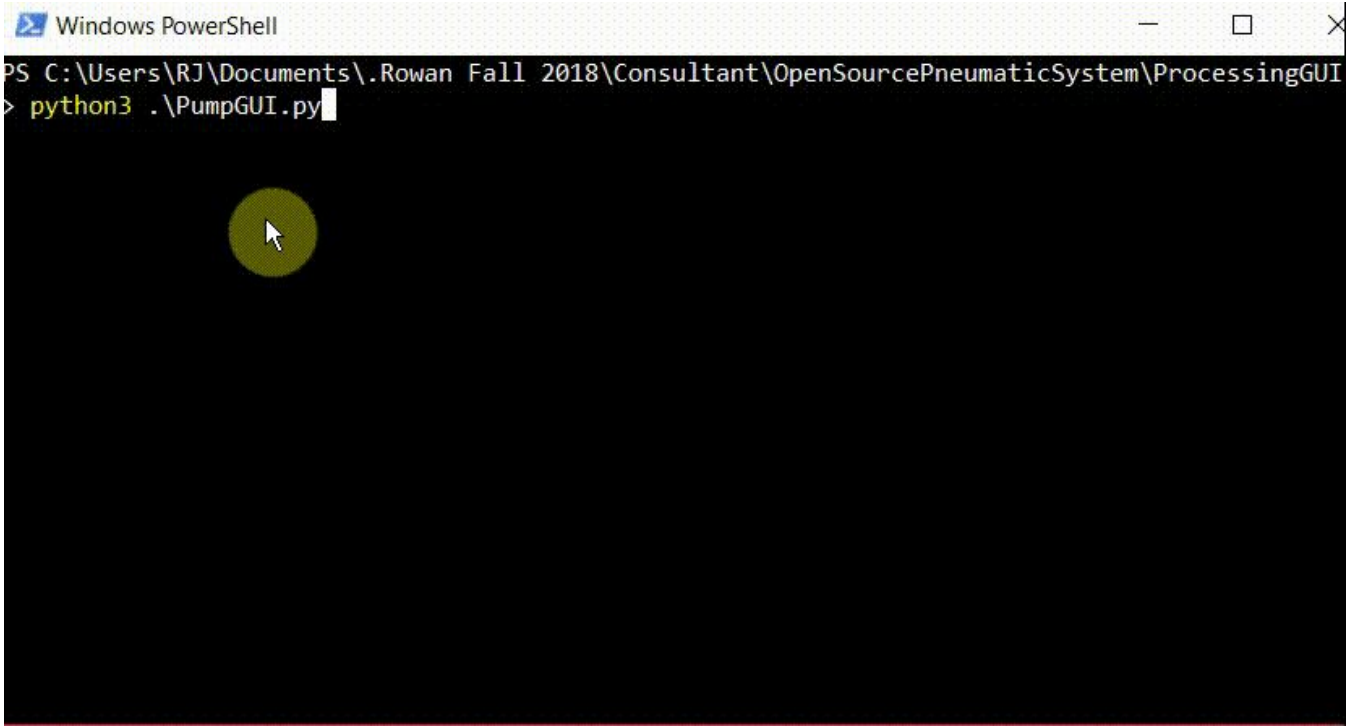
Output Feedback to User

Arduino reads digital gauge and writes data to serial port

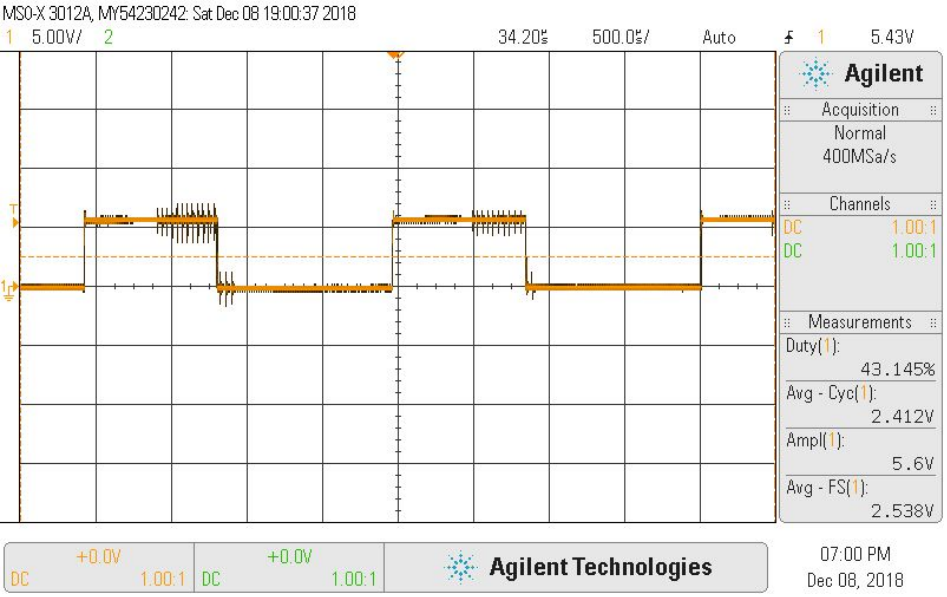
Old GUI Operation

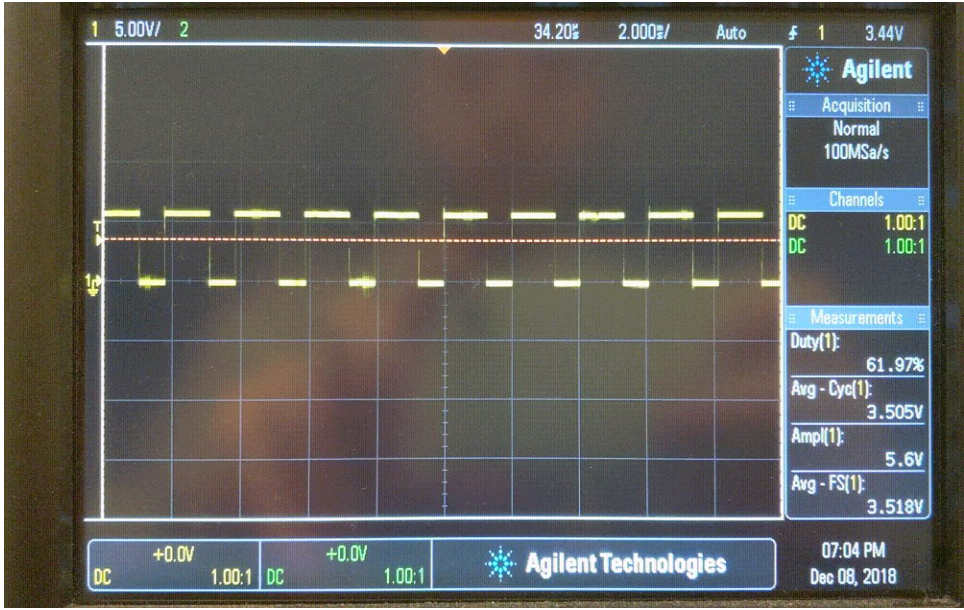


New GUI Operation



System Operation

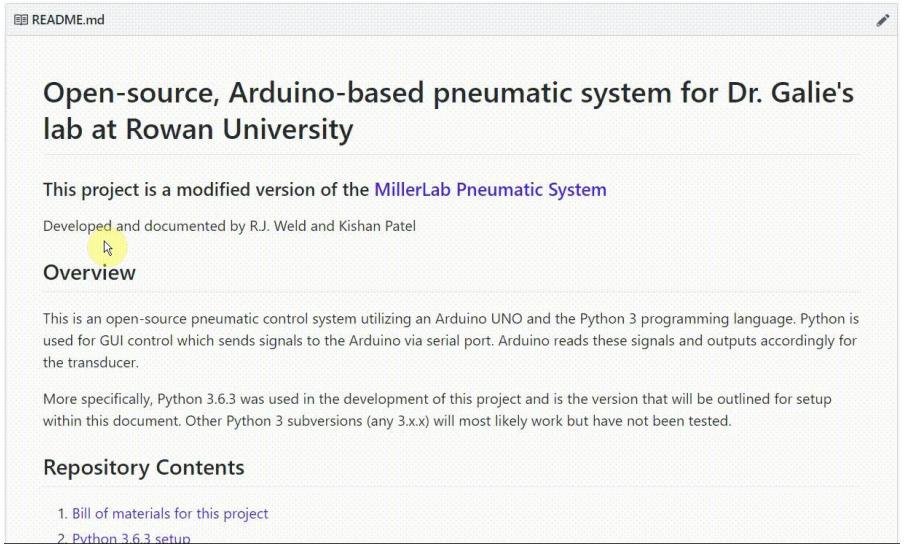




System Documentation

README Files Accessible on GitHub

Thorough documentation for easy setup, usage and troubleshooting



Future Work

Feedback Signals and Control System

Utilize a closed loop system for better control output and user feedback

Embed circuit onto PCB

Increases circuit stability and durability

3-D Printed Enclosure

Enclose circuit in 3-D printed box to better protect it and its users

Conclusions

Avoid tunnel visioning
Always try to have a backup plan

Efficient Planning is Key
Having a plan of attack, and setting deadlines helps

More Work Needed to Create Optimal Functionality
Implement PID controller on Arduino with feedback signals

Questions?