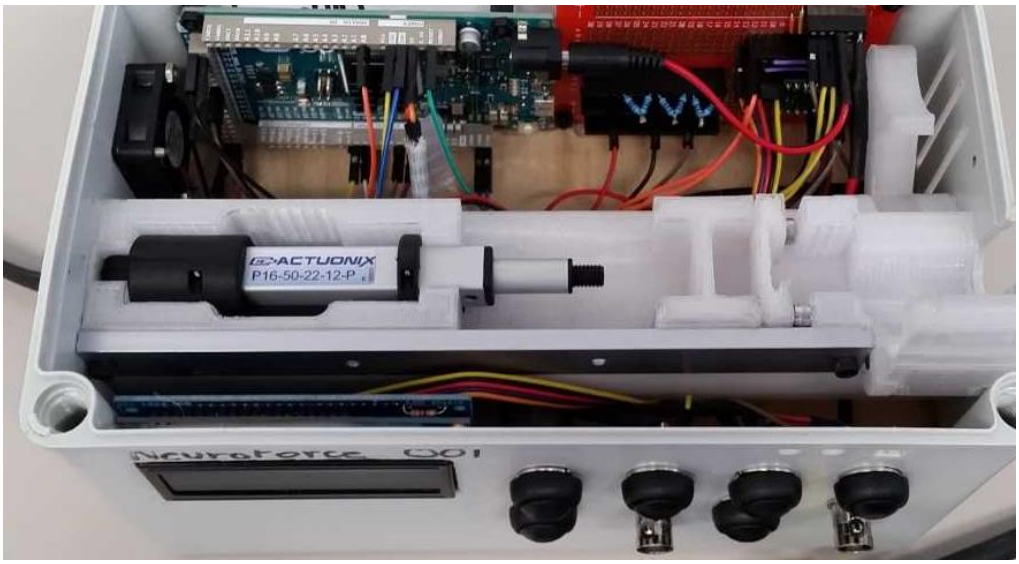
Project Learning – Dr. Martin/CSF System Example – Motors and Microcontrollers

Stephen Schoonen & Mitch Williams

Tour of Dr. Martin CSF research lab to view motor systems.



This first system utilized a DC linear actuator to create back and forth motion. This system was complex in the monitoring and adjusting the actuator's position and speed to be highly accurate. It utilizes a push button and display run through Arduino microcontrollers to accomplish this. This system does not have an encoded DC motor but gives an idea of how we may want to set up the screen and shows that a large number of pins were used to create a fairly basic display (similar to what ours would need).

This second system is a stepper motor attached to a ball screw linear actuator. This is the current system that is being incorporated because the last system failed after nearly 48 hours of operation. This system is much more robust and accurate as well. It utilizes a stepper motor controller that micro-steps the motor for higher accuracy and smaller steps. The system runs primarily from an Arduino right now, but there is work in progress to create a touch screen display run by a Raspberry Pi. This is an option; however, it requires us to know Linux and code in python which none of the team members can do that right now. The button system had no issues so that would also work for our application. The stepper system does not shake badly when run normal or micro-stepped and does not create a large amount of noise.

Microcontrollers:

Arduino Uno – Small footprint, fast enough for most basic applications 16 MHz, 14 digital input/output pins, 6 of which are PWM outputs, 6 analog inputs. Cost = $10-15

Arduino Mega – Slightly larger than UNO, 16 MHz speed, 54 digital input/output pins, 15 PWM capable pins, 16 analog pins. This board is highly compatible and well supported. It has plenty of pins to run a display although the speed may take a hit due to the overall tax of the system. Cost = $15-20

Arduino Due – Same size as Mega, 32-bit ARM core microcontroller, 84 MHz, 54 digital input/output pins, 12 are PWM outputs, 12 analog pins, extra USB port. This controller is better for faster applications, it has many pins and the processor to back them up. The input/output pins on this device are only able to tolerate 3.3V which might prove problematic especially when the DC motors with encoders have 5V requirements to power the encoder. Cost = $35-40

Raspberry Pi – Small footprint, high processing power and easily integrated into the touch screen. Requires a digital to analog converter. Requires Linux and python knowledge. For the project we were asked to keep the technology and knowledge requirement to a minimum for simplicity sake. Cost = $30-40

Motors:

DC encoded motors – Two types, brushless and brushed motors. The function of an encoder is to track the speed of the motor by spinning with it and sending an output to be read by the controller. We can use that information to control speed if needed.

Brushless motors – Tend to last longer and make less noise over all. Tend to be a little more expensive but should not need to be replaced anytime soon. More efficient than a brushed motor because there is no contact of friction.

Brushed Motors – Last 1000-3000 hours of operation. For this application that would be long enough to be safe, they will only need ~250-350 hours of operation if they have a large group of animals and run them for one hour every day from P1-P15. It would gradually make more noise as the brushes wear out and eventually it would need replaced which costs more money in the long run. This motor tends to be just a little less expensive but also less efficient.

Stepper Motor – Highly controllable and precise. Relatively simple to control and will maintain a constant speed with a varying load. Choosing a robust motor controller for operation will also reduce heat produced versus a simple chip controller. Downside is that the stepper motor will run choppy at lower speeds, be nosier than a DC, and vibrate more.