Handpiece Cleaner/Lubricator Prototyping at M3 Design

Siddharth Kurwa
skurwa.github.io

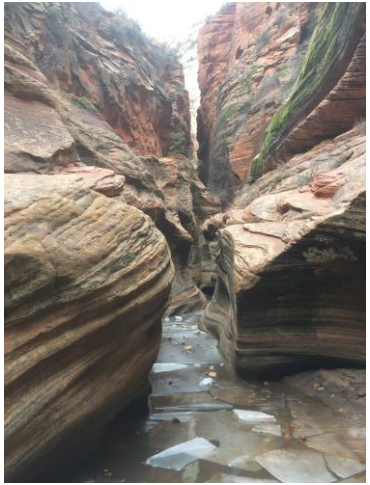
Agenda

- About Me
- Project Background
- Design Process
  - Hardware Selection
  - Low-Resolution Prototype
  - Functional Prototype
  - Looks-Like/Works-Like Prototype
- Outcome
- Questions

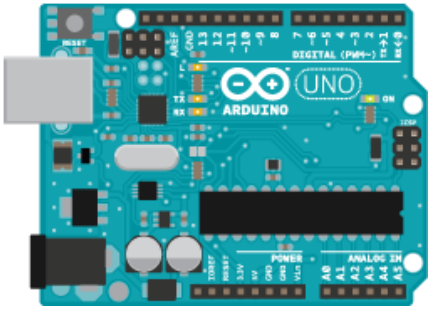
A Few of My Favorite Things

Mediterranean Food

Zion National Park



Tinkering

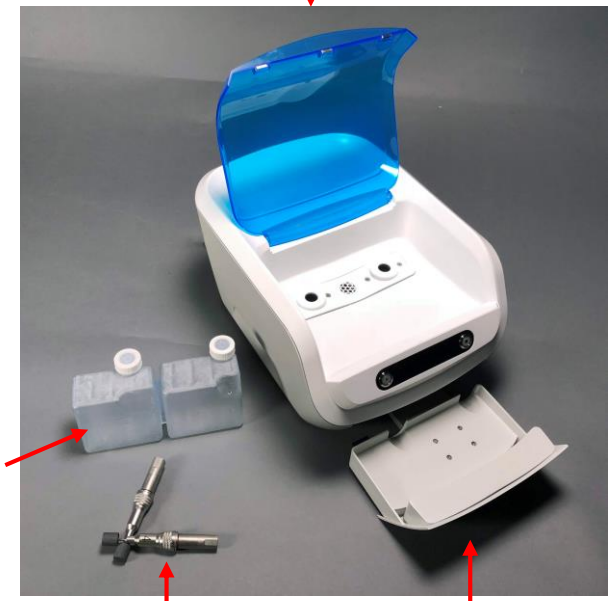


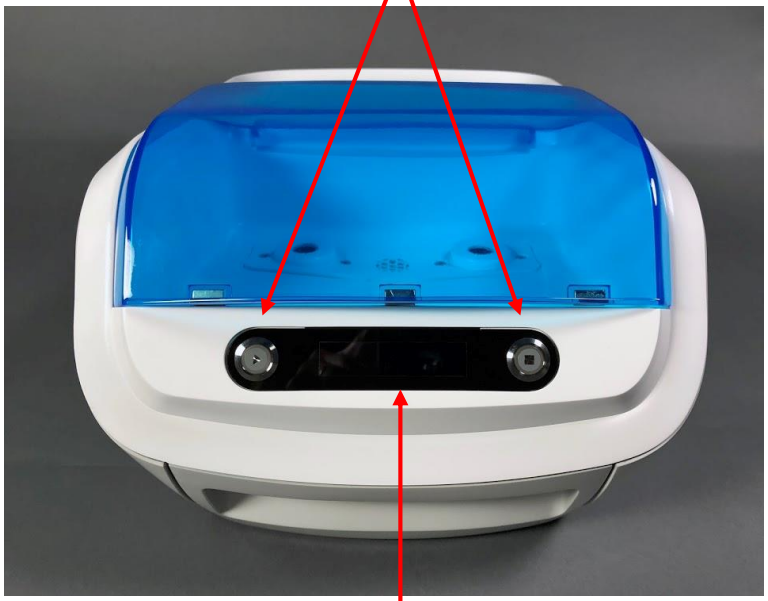
Project Background

- Client Need
  - Field support for failed medical handpieces is expensive
  - Handpiece cycle life reduced by poor manual maintenance
- Problem Statement
  - Design and build a cleaning/lubricating system to simplify handpiece maintenance
- Timeline
  - 1.5 months
- Team Roles
  - 2 ME – fluid system design, CAD, mechanical prototype
  - 1 ID - user workflow, product aesthetic
  - 1 EE/ME - verification, support, debug
  - **1 EE - hardware selection, software development, system testing**

Client's surgical handpiece products.

Product Externals





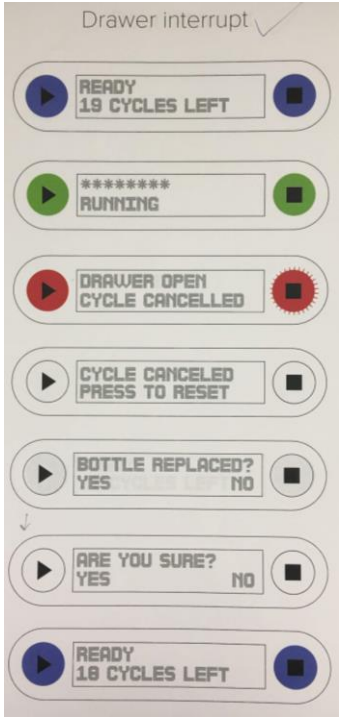
My Scope

Select, design, and build the electronics and control system to meet mechanical and industrial design requirements in a 1.5 month timeline.

Requirements

Table of design requirements.

Source	Type	Requirement
Industrial Design	Sensing	Detect open/closed door
	Sensing	Detect open/closed drawer
	UI	OLED character display
	UI	2 pushbuttons
	UX	10 user workflow maps
Mechanical	Sensing	Detect tank overflow
	Controls	Satisfy pump pressure/flow operating conditions
	Volume	Fit within 12” x 4” x 6” volume



User workflow map.

Process

- Identify possible hardware options.
- Bench test components to verify viability.
- Integrate subsystems to verify interaction.
- Build and test client deliverable.


Hardware
Selection

Sensor Survey

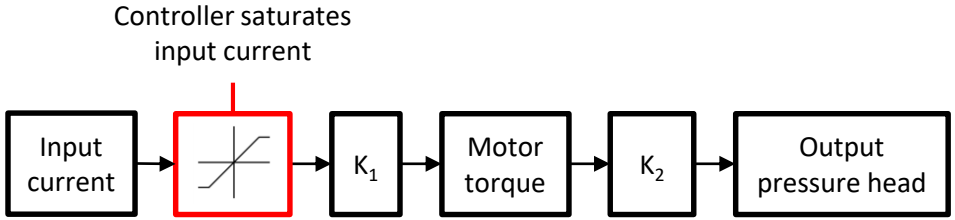
- Open/closed door and drawer
  - Considered detent and reed switches – quickly chose reed switches for smaller package size
- Liquid level switch
  - Researched sensing principles (ultrasonic, capacitive, optical, pressure, magnetic) and found several options – narrowed down to below list

Comparison of liquid level switch solutions.

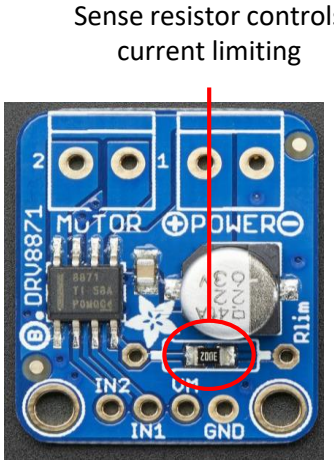
Working Principle	Source	Lead Time	Pros	Cons
Capacitive	DFRobot	1-2 weeks	Cost; non-contact; package size	Limited resources; lead time
Capacitive	Carlo Gavazzi	1 day	Non-contact; lead time; support	Cost; package size
Optical	Sparkfun	1 day	Cost; lead time; package size	Contact

Pump Control Solution

- Pump operating condition requirements
  - Open handpieces: 30 GPH, <4 psi
  - Restrictive handpieces: 0 GPH, <9 psi
- Tested pump characteristics without control
  - Open flow: 30 GPH, 3.6 psi
  - Choked flow: 0 GPH, 16 psi
- Made physical system model to identify control need
  - Motor driver to satisfy requirement



Open-loop control model of pump.

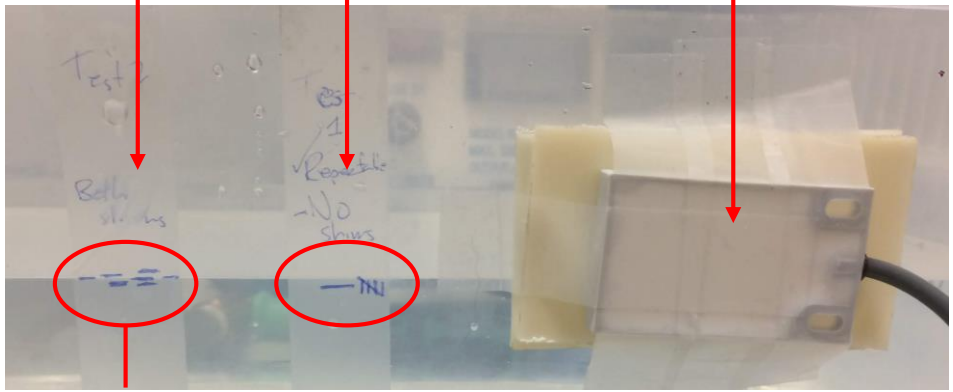


DRV8871 motor driver.


Hardware Selection
Low-Res Prototype

Sensor Function Checking

Level sensor bench test setup.



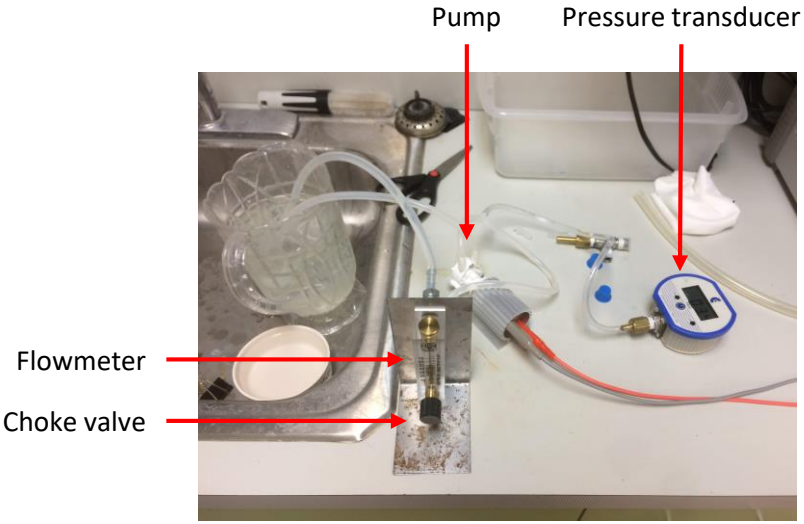
0.1" repeatability range

Test results.

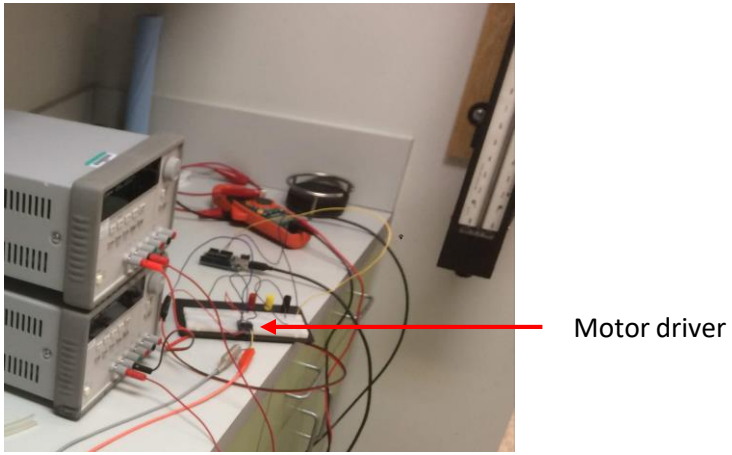
- 0.1" repeatability acceptable with margin based on tank CAD and projected sensor position

Pump Control Testing

- Iteratively determined current sense resistor to meet pressure requirements
- Provided empirical model: $R_{sense} = \frac{64k}{I_{lim}}$



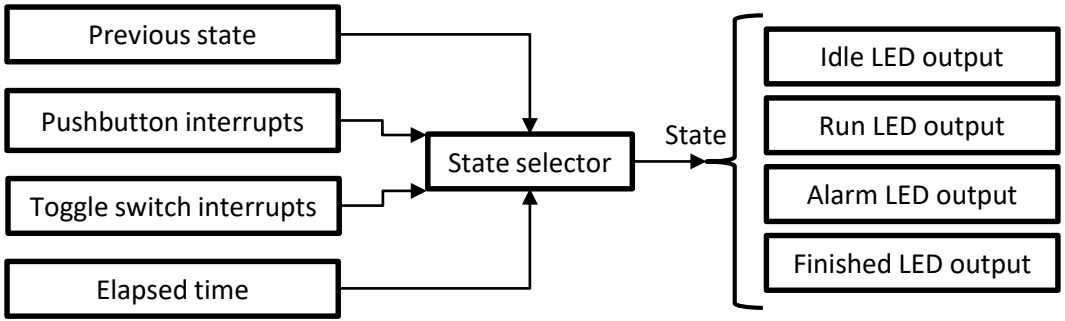
Pump measurement setup.



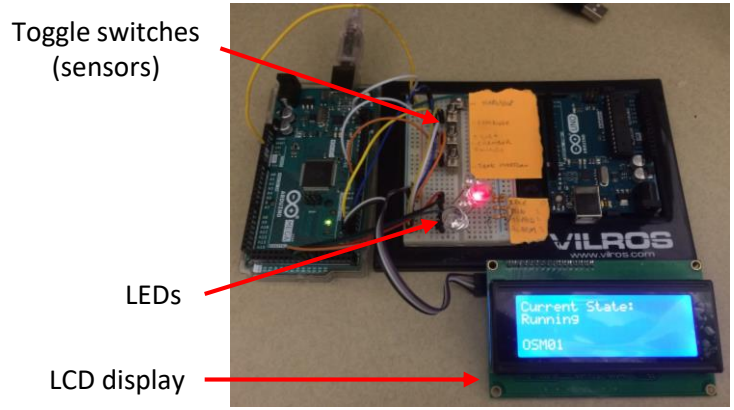
Motor driver outputs.

State Machine Software

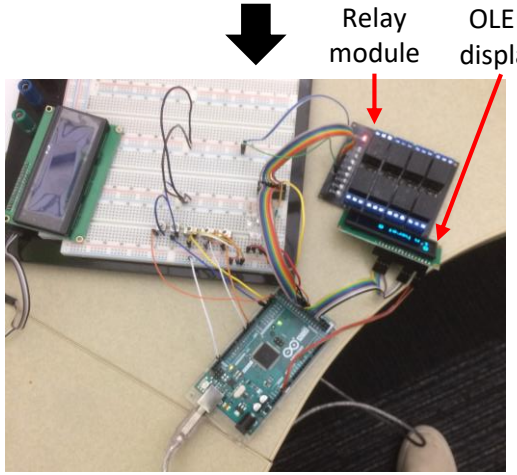
- Arduino microcontroller for digital IO
- Used simple components to simulate physical system



Foundational state machine architecture.



Initial prototype to drive LED output.

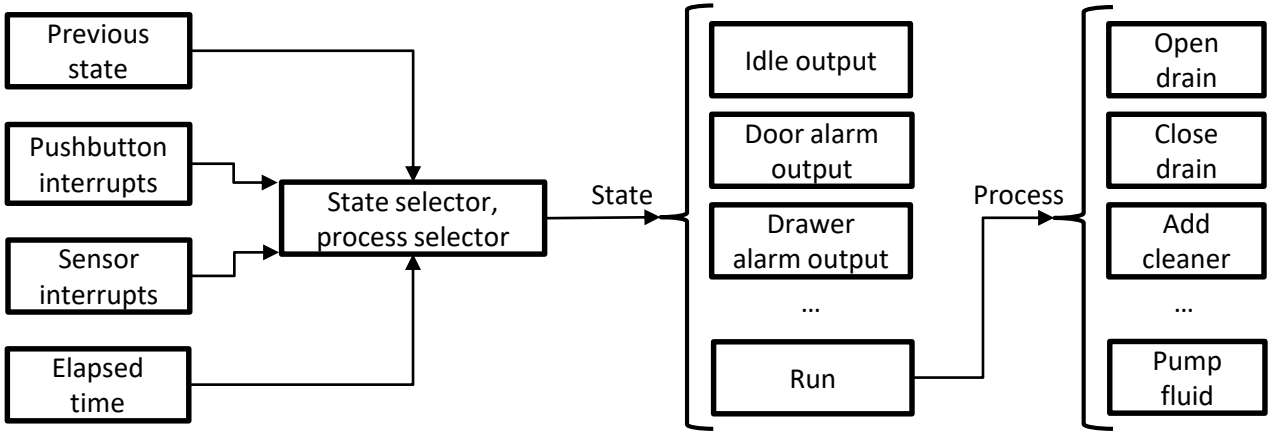


Incrementally added components.


Hardware Selection
Low-Res Prototype
Functional Prototype

State Machine Expansion

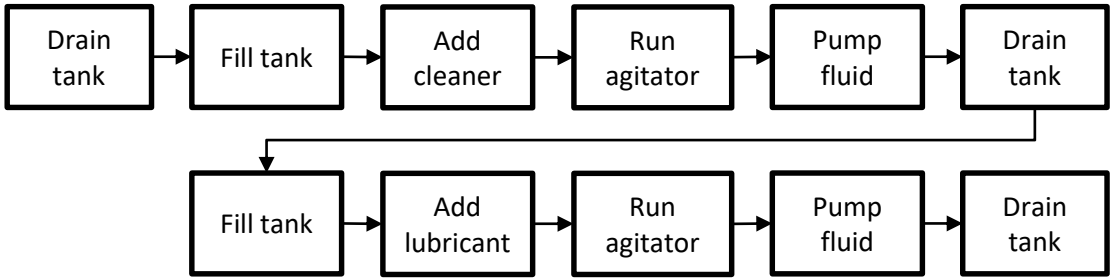
- Hierarchical state machine to control sequencing of run-time processes
- Handled nominal and edge cases with 11 states and event/interrupt hierarchy
- Ex. edge case: enter run, trip door alarm, trip drawer alarm



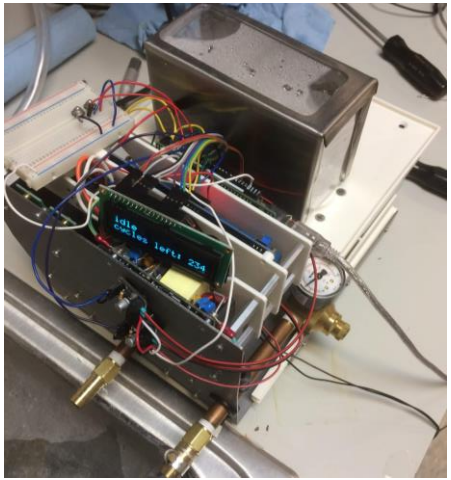
Robust state machine architecture.

System Integration

- Validated pump performance in actual system
- Supported ME in tuning process function timing and sequence



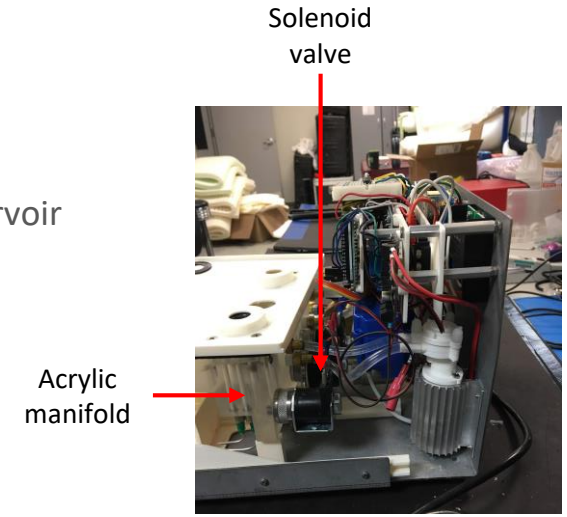
Determined process sequence.



Functional prototype.

System Debug

- Issue
  - Solenoid valves overheating – melting risk to acrylic manifold/reservoir
- Context
  - Solenoid directly connected to power supply, bang-bang control
- Root Cause Analysis
  - Found solenoid valve power rating on spec sheet (7W)
  - Estimated power dissipation (16W) based on winding resistance
- Solutions
  - Limit solenoid valve current draw
    - Add in-series resistance – cheaper but slower
    - Swap relay to motor driver with current limiting feature – immediate but more expensive

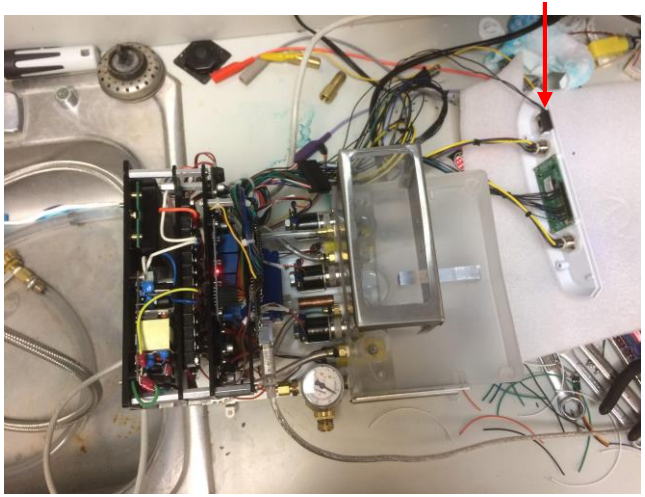


Solenoid valve and acrylic manifold interface.

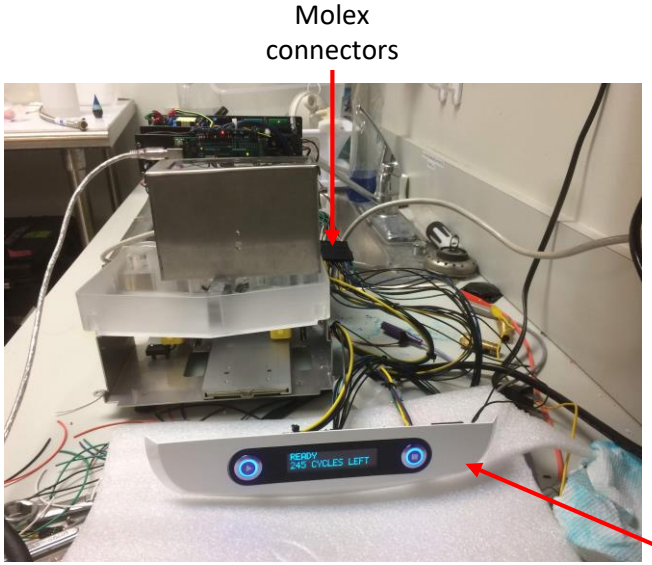

Hardware Selection
Low-Res Prototype
Functional Prototype
Looks-Like, Works-Like Prototype

Looks-Like, Works-Like Prototype

- Incorporated sensors, UI features, and reliable connections

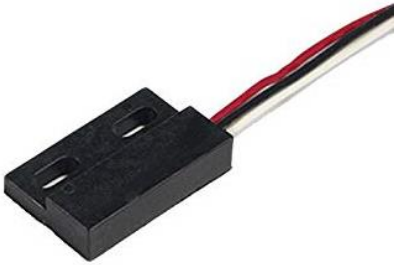


Adding final electronics features to system.



Powering on system with finalized components.

System Debug



Reed switch.

- Issue
  - System freezes when pumps turn on
- Context
  - 3 sensors and 2 pushbuttons use interrupt service routines (ISRs) with simple delay debounce routine
    - If interrupt: wait 25 ms, then check state: ISR if high, exit if low
- Root Cause Analysis
  - Removed sensor response from logic – timing worked reliably
  - Added interrupts back incrementally – timing froze again with both reed switches
  - Possible causes: weak magnet, EMI signal noise, mechanical vibration
- Solutions
  - Software Patches
    - Counter-based debounce routine to minimize time spent in ISR
    - Change reed switches to polling architecture rather than event-based
  - Hardware
    - Stronger magnets
    - Twisted-pair cabling of high amperage lines to minimize EMI

Outcome

- Project
  - $200,000 billable project delivered on schedule
  - Client demoing prototype to customers since mid-Sep without issue
  - Client returned with 2 new projects
- Personal
  - Owned critical components and deadlines of multidisciplinary project (ME/EE/ID)
  - Built tuneable controls architecture to handle every case
  - Systematically debugged issues to identify root cause and corresponding solution

Questions?