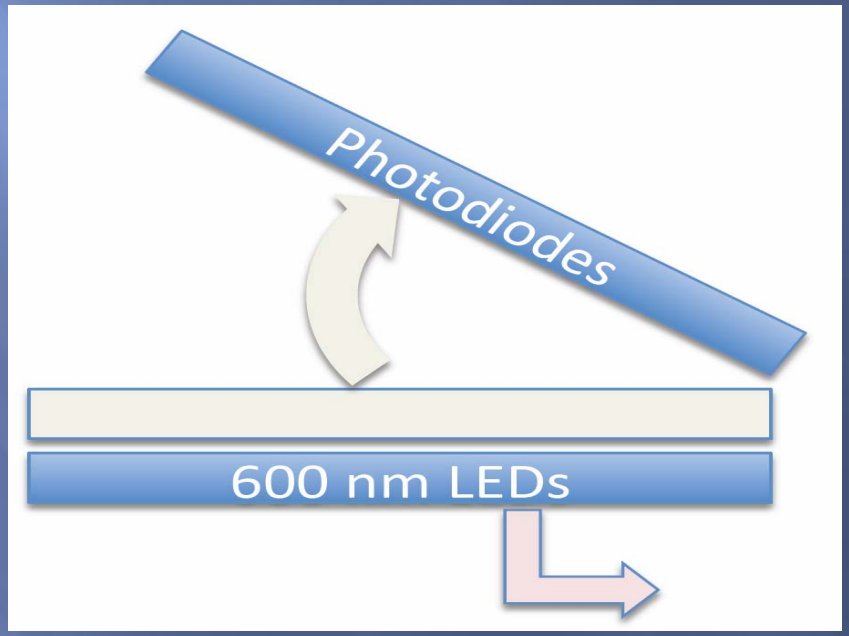
Turbidostat Control Design

Milestone 4



By:

Peter Harker

Evan Dreveskracht

Max Holloway

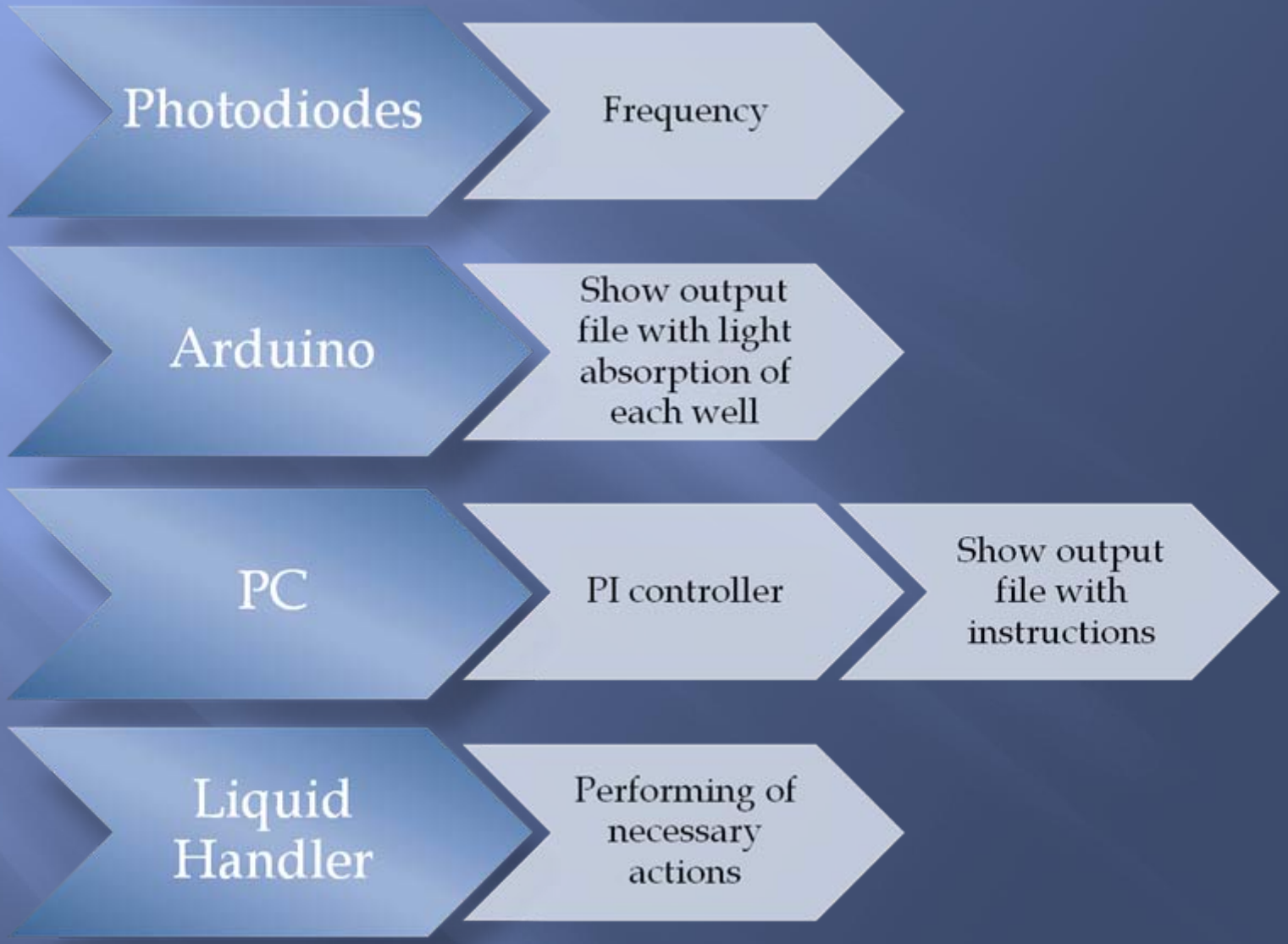
Table of Contents

- 1: Current Project Plan Update
  - Plan for Demo
- 2: Hardware
  - Scanner Assembly Design
  - Drawings for Scanner Housing
  - Gripping Mechanism
  - Circuit Board (LED Board)
  - Circuit Board (Photodiode Board)
- 3: Control Model Discretized
- 4: Software
  - PI Controller Implementation
  - Arduino Software Testing

Current Project Plan

Week	Goal	Action
8	Hardware assembly completed	-Solder circuit components onto boards*. -Combine plastic housing with board*. -Connect Arduino and test output frequency (OD) ✓ * Waiting on PCB and housing materials
9	Interfacing hardware with liquid handler	-Calibrate liquid handler for Housing unit dimensions -Install and fine-tune readings from LEDs to Photodiodes inside closed housing. -Incorporate PI controller with liquid handler software.
10	System Tuning	-Make necessary adjustments to achieve greater accuracy. -Collect data for report.
11	Demo	-Show functionality of plate reader by it's ability to read light density of each well, and incorporate that information into a PI controller which controls the Liquid Handler actions.

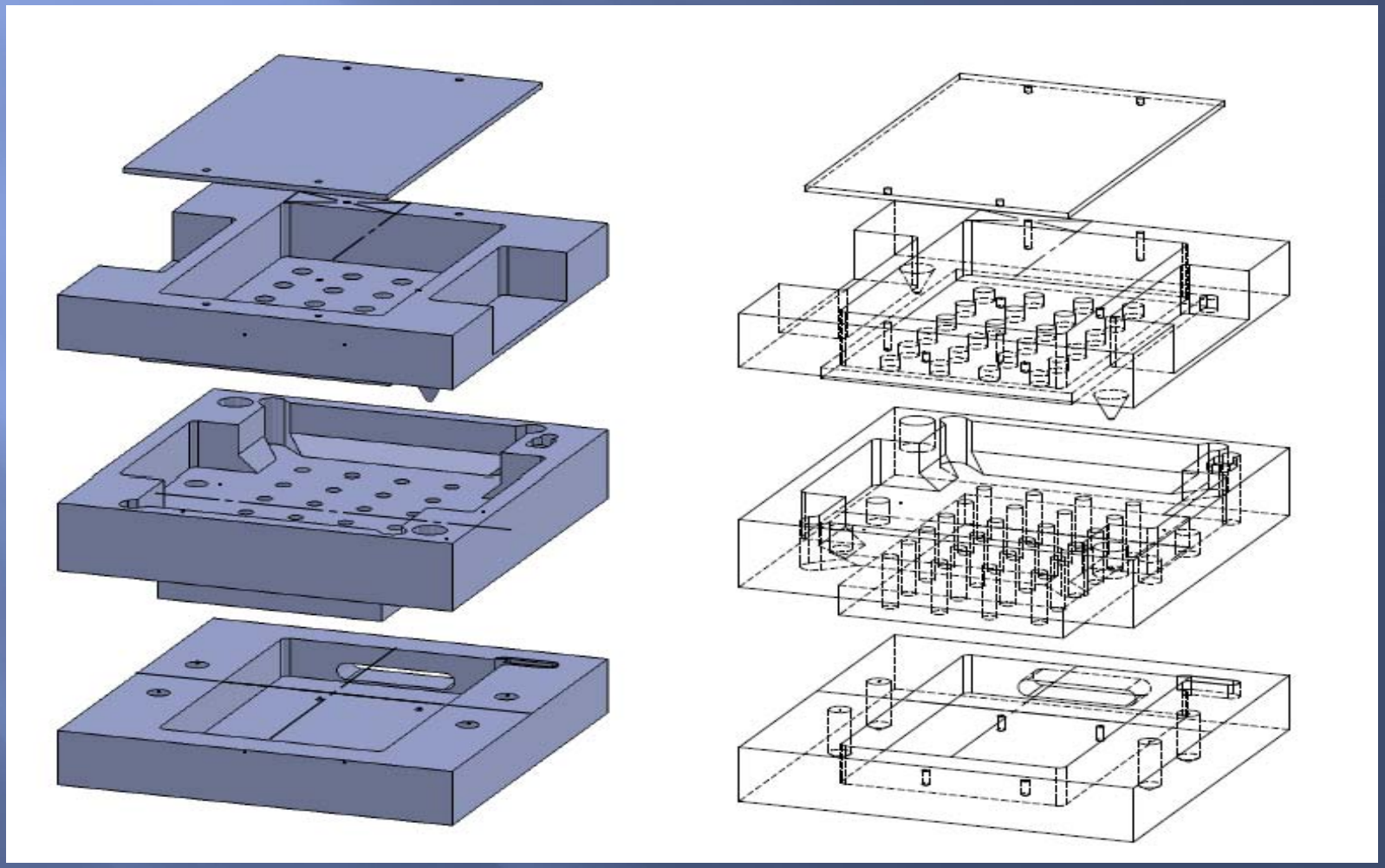
For Demo…


Photodiodes
Frequency
Arduino
Show output file with light absorption of each well
PC
PI controller
Show output file with instructions
Liquid Handler
Performing of necessary actions

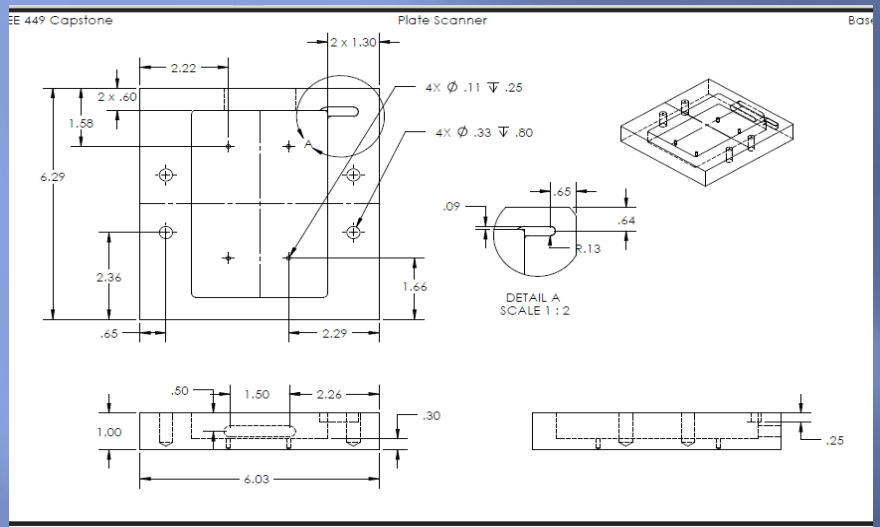
Hardware

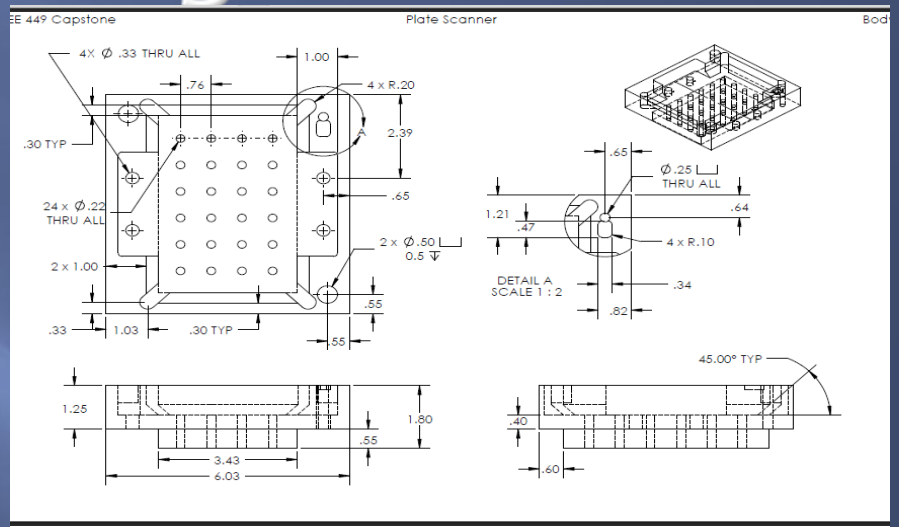
Design: Housing unit (image, highlighting important sections and explaining design [lid with sensor for closed, lid design for grabber functionality, plate holder design for exact placement, Holes for LEDs giving minimal reflection]), LEDs (Wavelength for bacteria absorbtion and size for positioning in the housing), Photodiodes (Ease of use with Arduino)

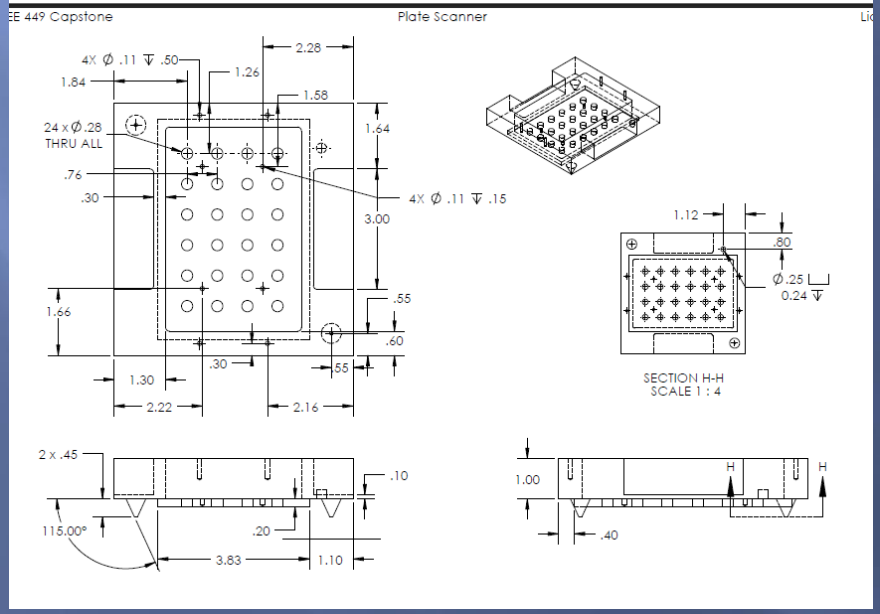
Scanner Assembly Design

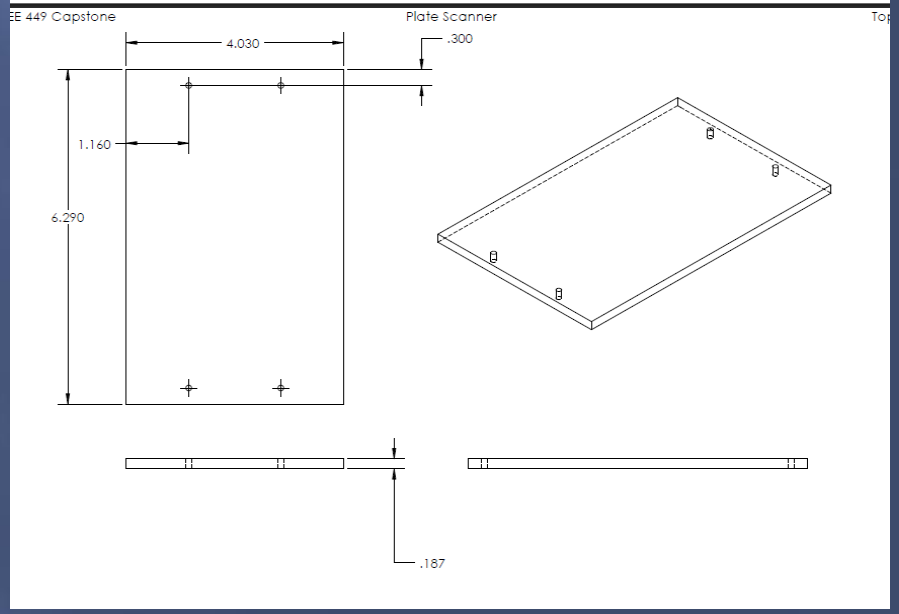


Drawings for Scanner Housing



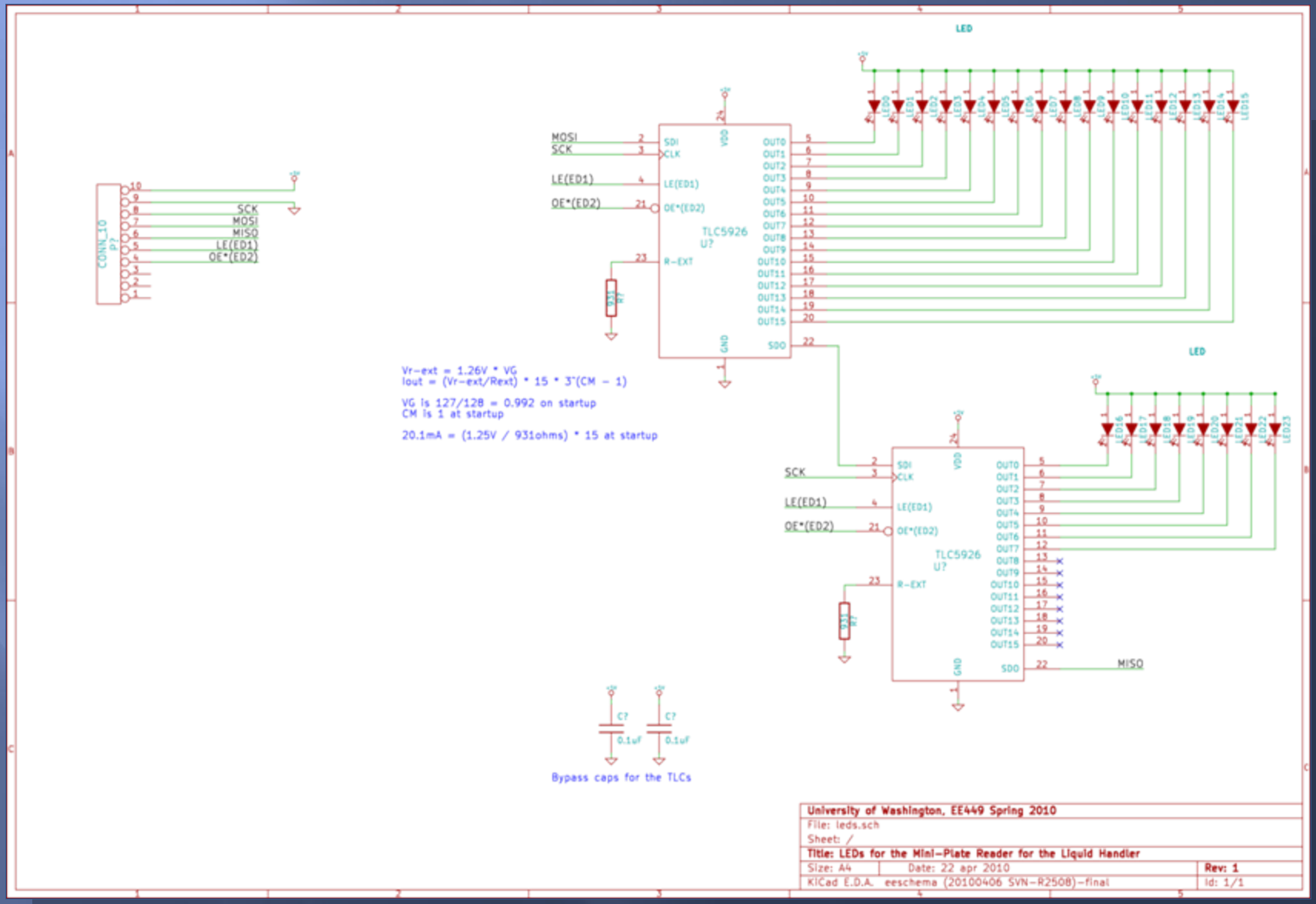






Gripping Mechanism

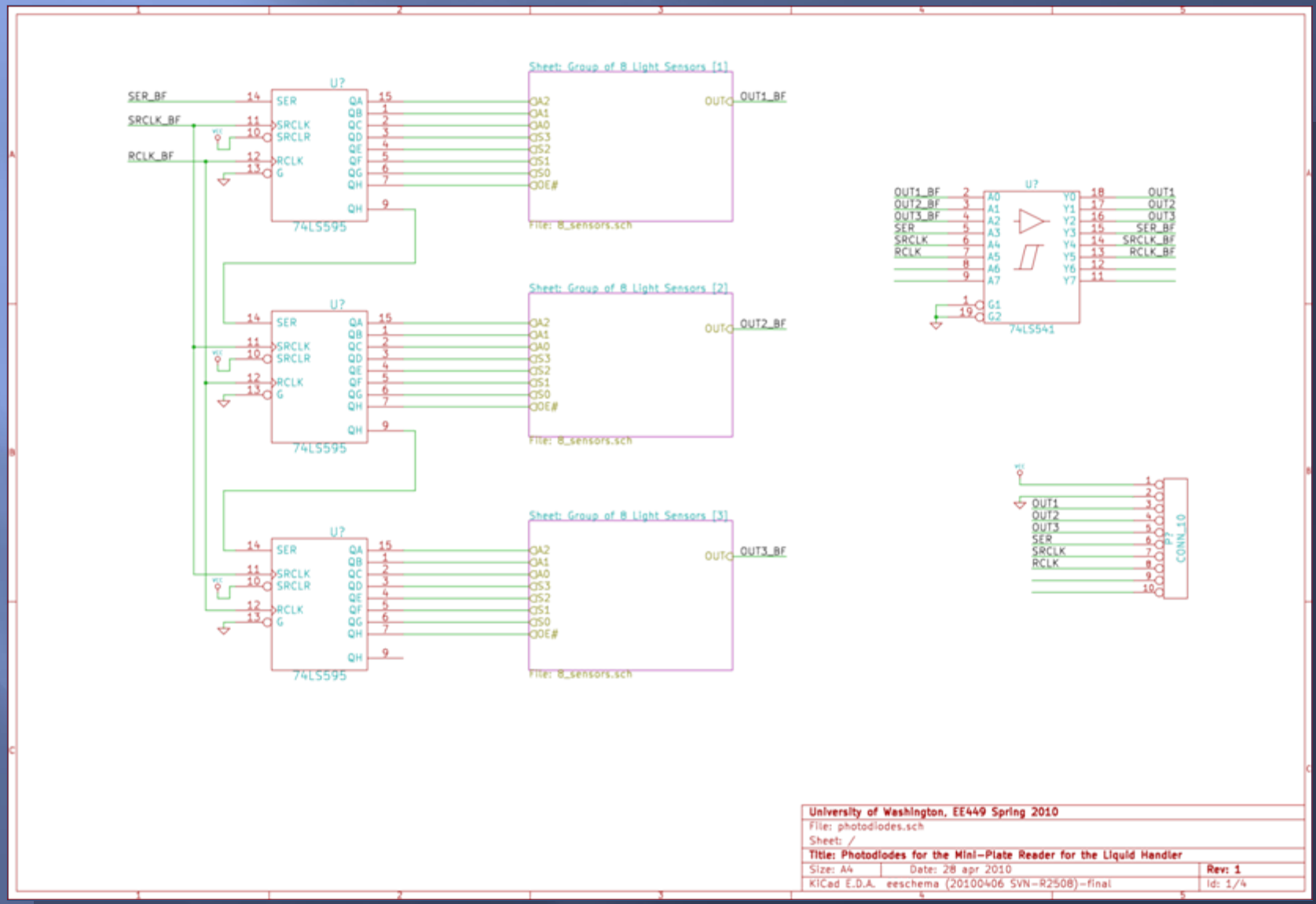
Circuit Board (LED Board)



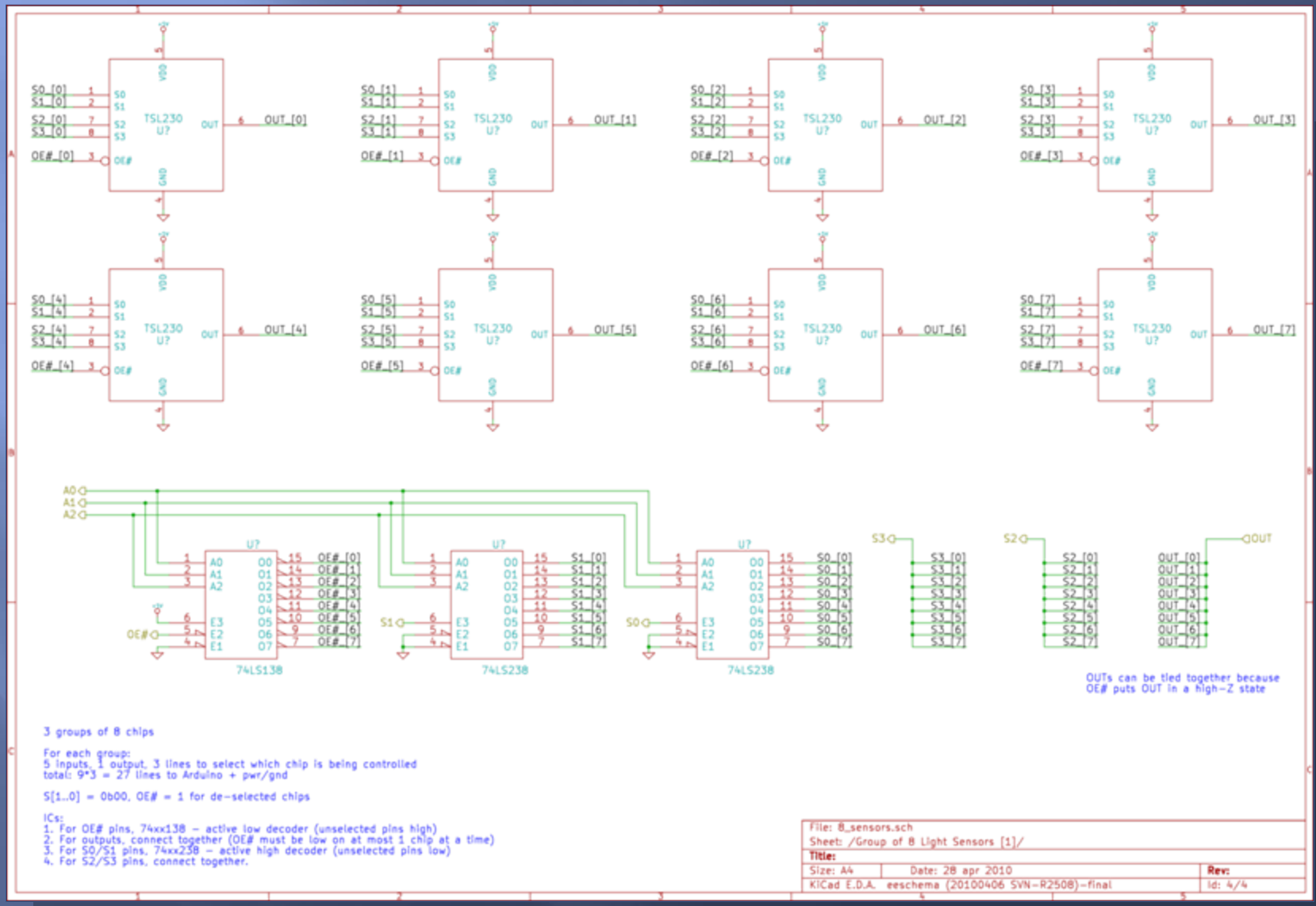
Circuit Board (LED Board)

- LEDs (24)
  - Primary wavelength output is 600nm (bacteria population absorbs this wavelength)
- Current Drivers (2)
  - Gives constant current to LEDs
  - Takes serial input to select output
  - Used SPI (Serial Peripheral Interface) to connect

Circuit Board (Photodiodes)



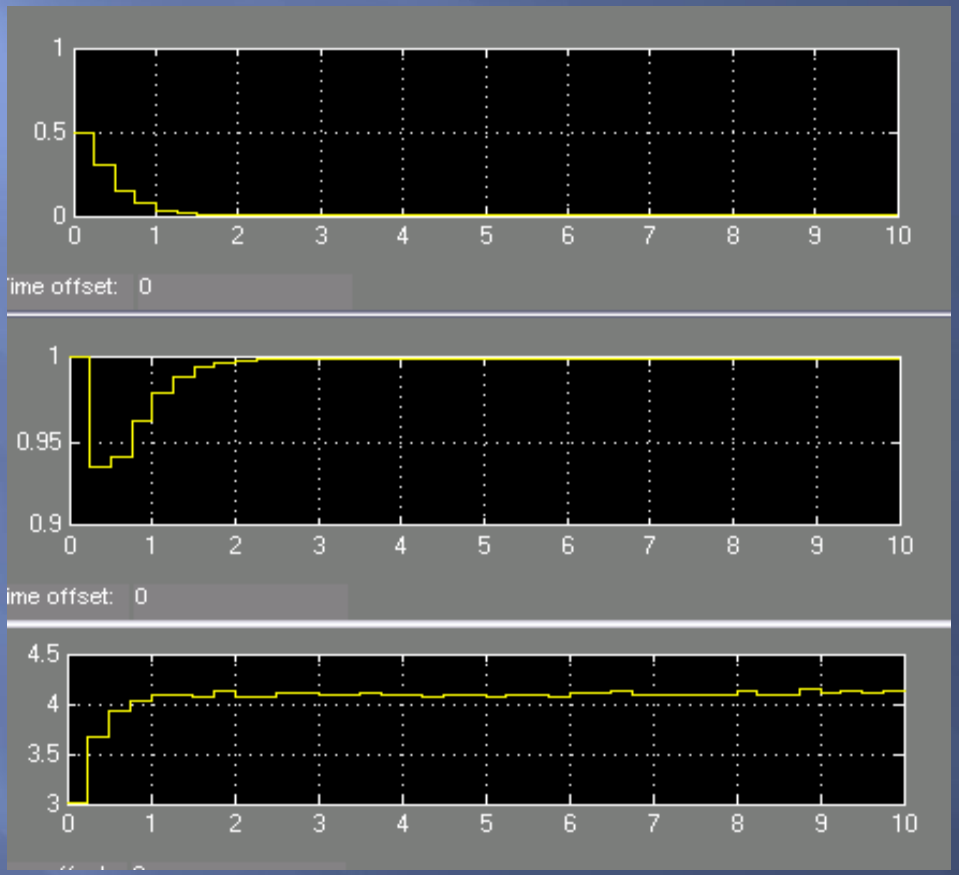
Circuit Board (Photodiodes)



Circuit Board (Photodiodes)

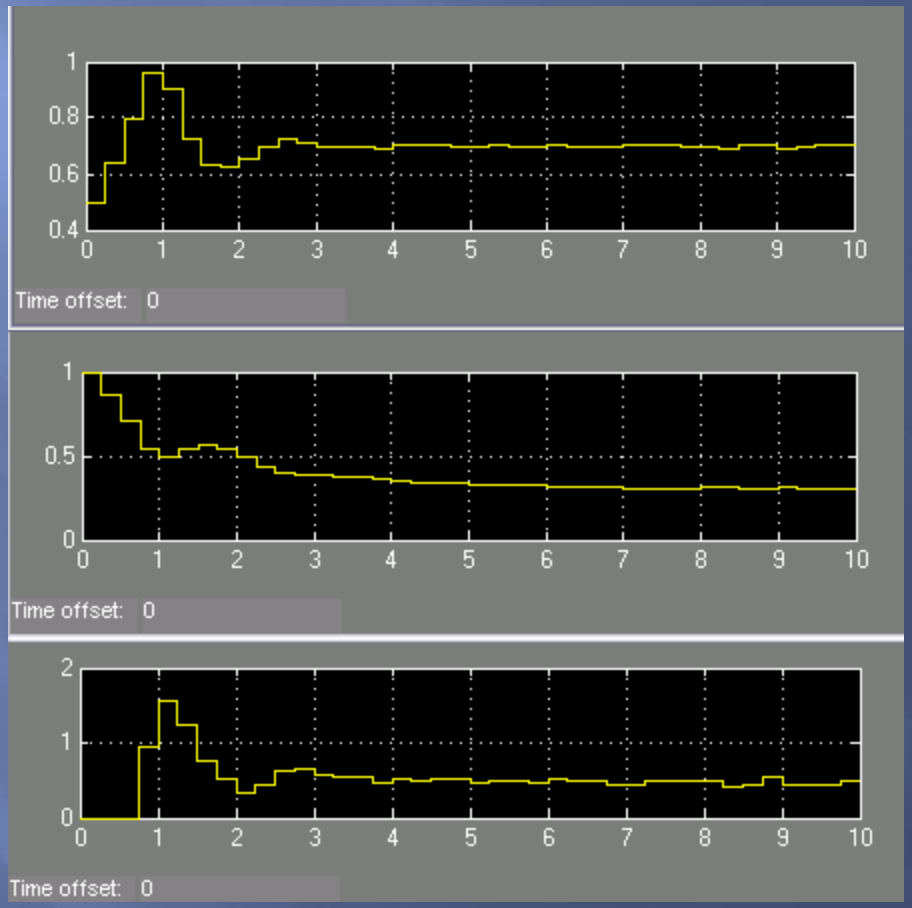
- Photodiodes (24)
  - Light to Frequency Converters
  - Used because there aren't many ADC's on microcontrollers
- Shift Registers (3)
  - Used for control signals of each "block" of eight photodiodes
  - Also used SPI to communicate with this block
- Decoders (3, one for each "block")
  - Used to control the output enables on the photodiodes

Discretized System Simulation



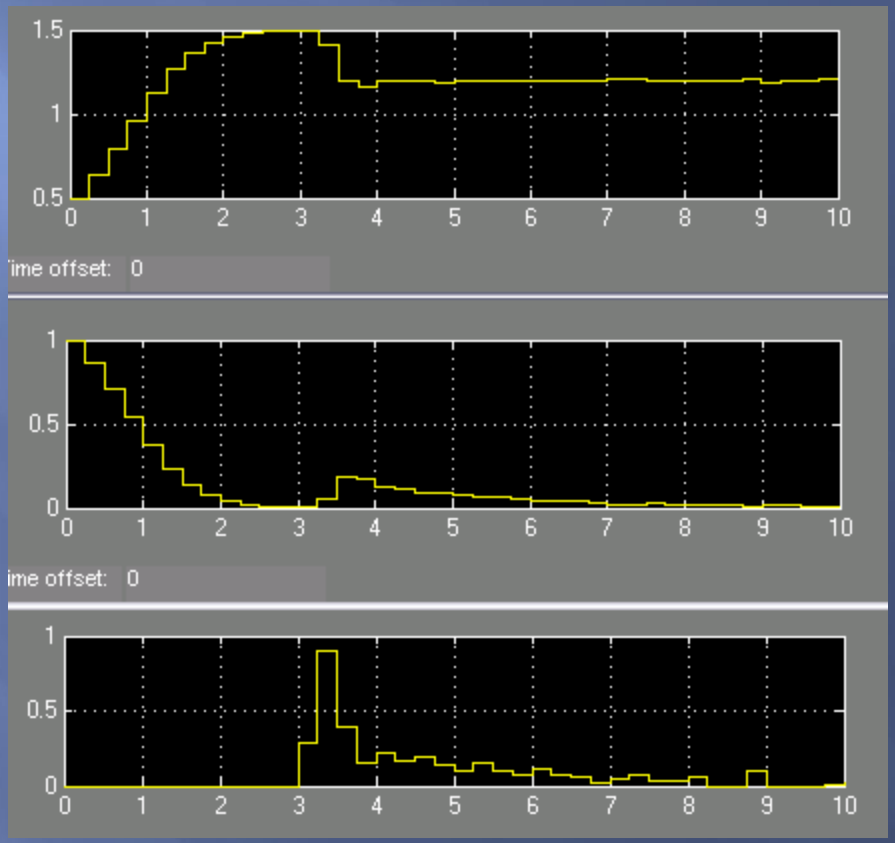
- A more accurate representation of our system since our plant is continuous and our controller is discrete.
- Sample time is every 15 min
- Desired population is 0
- Bacteria Population (top), Fresh Media (center), Nutrients (bottom)

Simulation (cont…)



- Bacteria population (top), Fresh media (center), Nutrients (bottom)
- Desired population is 0.7

Simulation (cont…)



Desired population is 1.2

PI Controller Implementation

- Code to be written in C which includes
  - Read in values saved to a file from arduino and create an array
  - Use a while loop to run each value through the PI controller and create a new array of values for how much liquid to extract per well and how much to add back in
  - Write these to file for liquid handler to read

PI Code in C

- Basic code to be used for implementing PI controller into system

```
integrator_state[num_wells] := 0

while 1
  for each well W
    error := read_well(W)-ref
    integrator_state[W] := integrator_state[W]+error
    u[W] := kp*error+ki*integrator_state[W]
  output(u)
  wait
```

Arduino Software Testing (as is)

- Used external counter to count the low to high transitions on pin 47 of the Arduino
- After one second, an interrupt fires and prints the count value to the serial console (also resets the count value)
- Can put in combination of ‘o’ or ‘f’ followed by a number into serial console to turn LED’s on or off using the LED current driver (with SPI interface)

Arduino Software Testing (soon to be done)

- Need to modify to add two additional counters so there are a total of three (one for each "block" of photodiodes)
- Need to send outputs to decoders to enable outputs
- Need to add another current driver chip in "series" with current driver chip being used and modify SPI function to send data to second chip
- Need to interface with Shift Registers