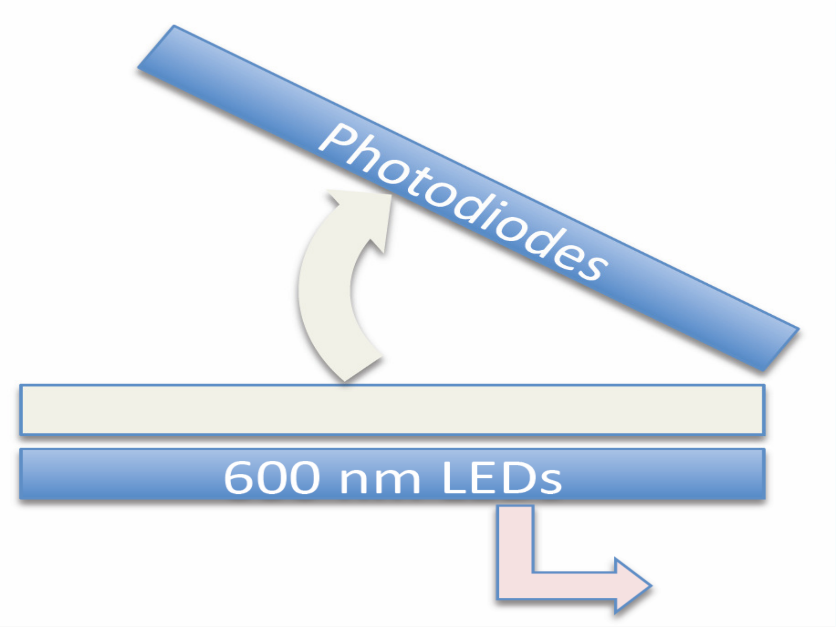
Plate Reader Turbidostat Project Review



Maxwell Holloway
Peter Harker
Evan Dreveskracht

Contents

- Project Description
  - Customer needs

- Literature and Related Work
- System Model
- Hardware and Software Design
- Code, Diagrams, Photos

Project Description

- Customer Needs:
  - Replacing previous expensive multifunction equipment with simpler plate reader for reading turbidostatic* behavior.

* Turbidostat: a device in which a bacterial culture is maintained at a constant volume and cell density (turbidity) by adjusting the flow rate of fresh medium into the growth tube by means of a photocell.

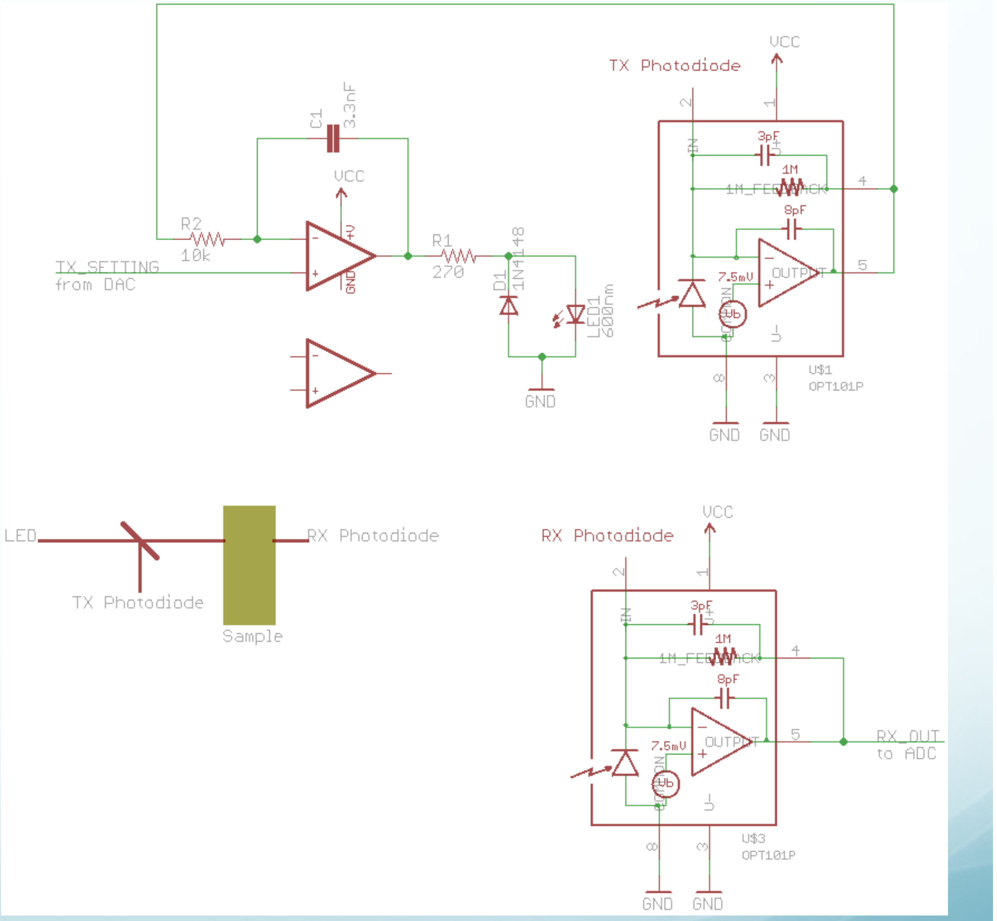
Related Work

Alex Leone
Self Organizing Systems (SOS) Lab
Undergraduate Research Assistant

- Created Turbidostat that worked well
- Wanted to Expand on his idea

Limitations of Original:
- One sample at a time
- Laser was noisy

Advantages of Original
- Two photodiodes



System Equations

$$\dot{x} = \frac{vnx}{k+n} - ux$$

$$\dot{n} = -\gamma \frac{vnx}{k+n} + u(n_0 - n)$$

$$y = kx$$

For our simulation:

v = 2 generations per hour

k = 1

$\gamma = 0.5$

$n_o = 1$

$x_o = 0.5$

PARAMETERS

v = maximum growth rate (generations/hour)

k = half saturation constant (g/L)

Υ = nutrient mass used per bacteria mass grown (unitless because it's a ratio)

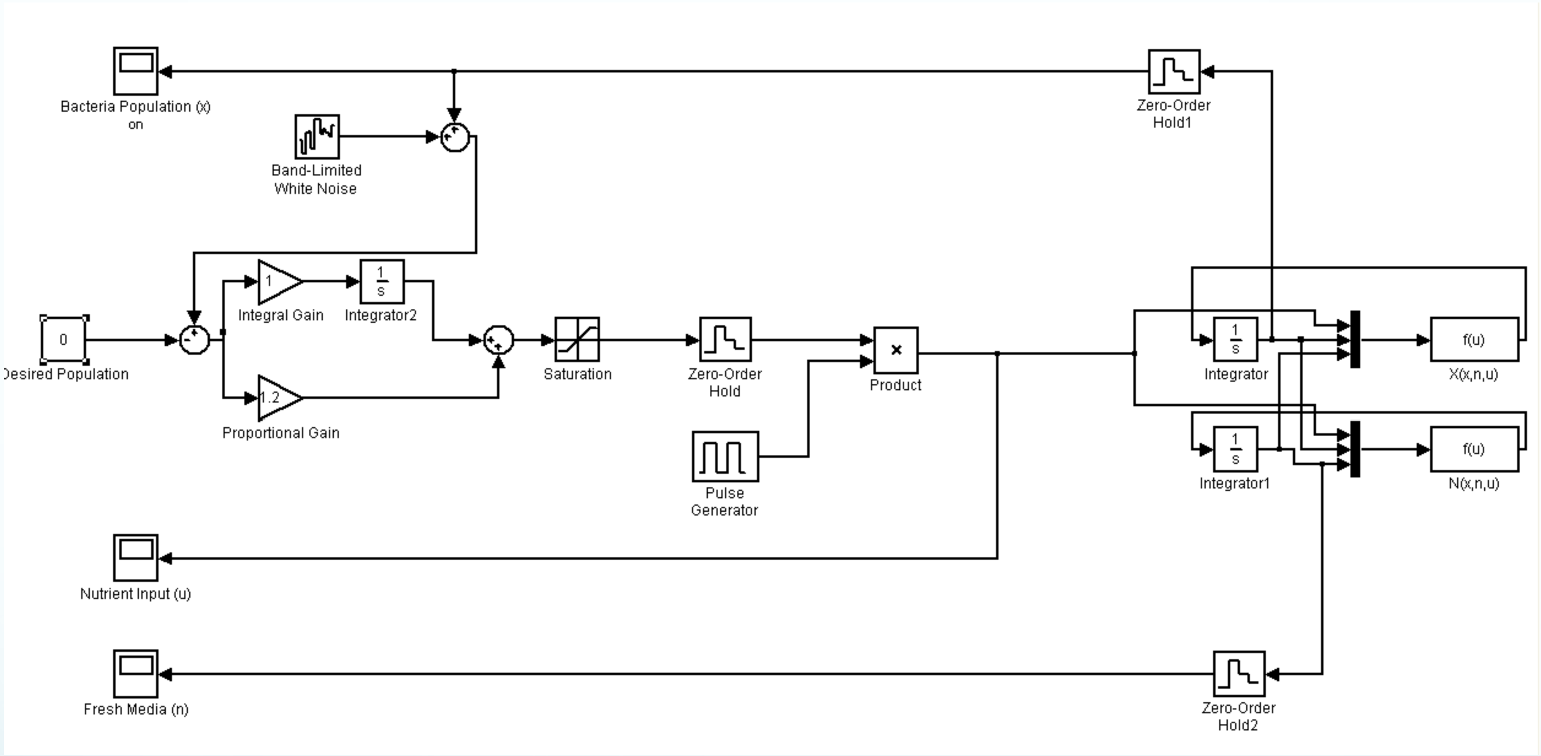
n_0 = nutrient concentration in fresh media (g/L)

STATES

x = amount of bacteria (g/L)

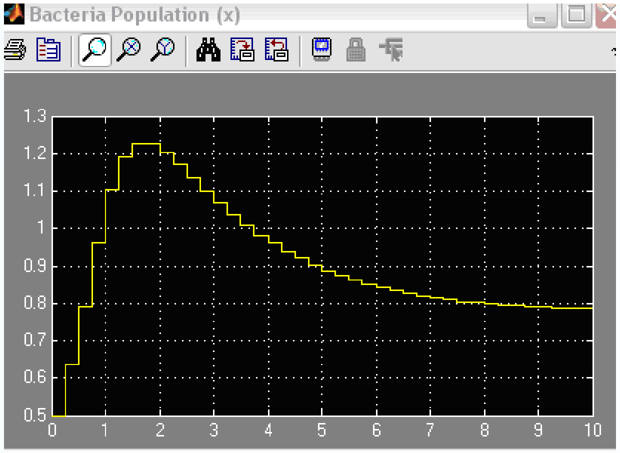
n = amount of nutrients (g/L)

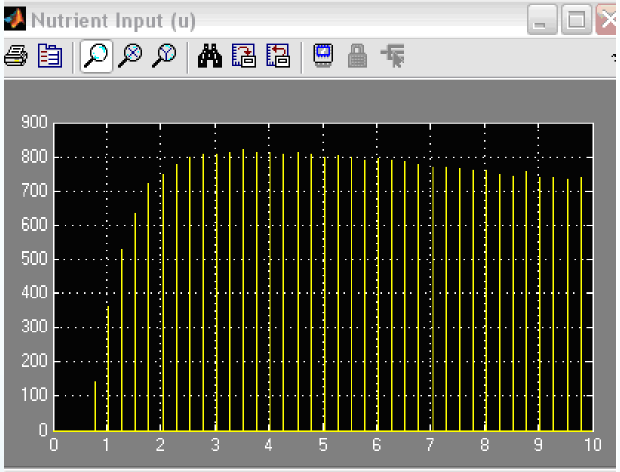
System Model

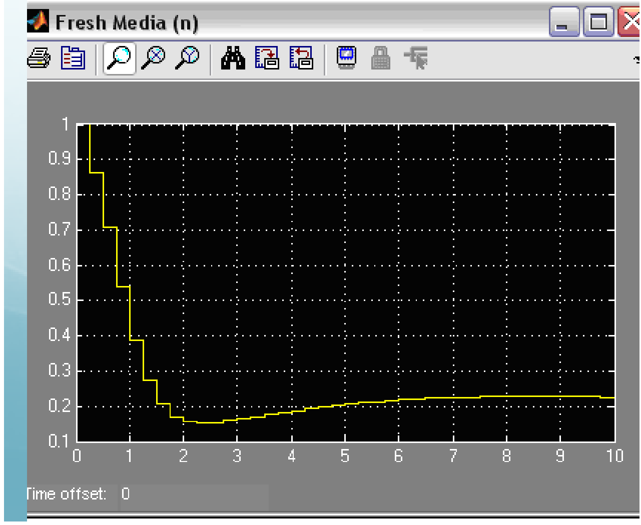


Simulink Model with Discrete Characteristics

Discrete Simulation Model







Time offset: 0

Amplitude = T / Ton = Period / Time on of period, 900 / 2 sec = 450

Period is 15 min. Units on simulation are relative to hours but simulink reads them as seconds. This was the mistake made in the presentation that I was not aware of.

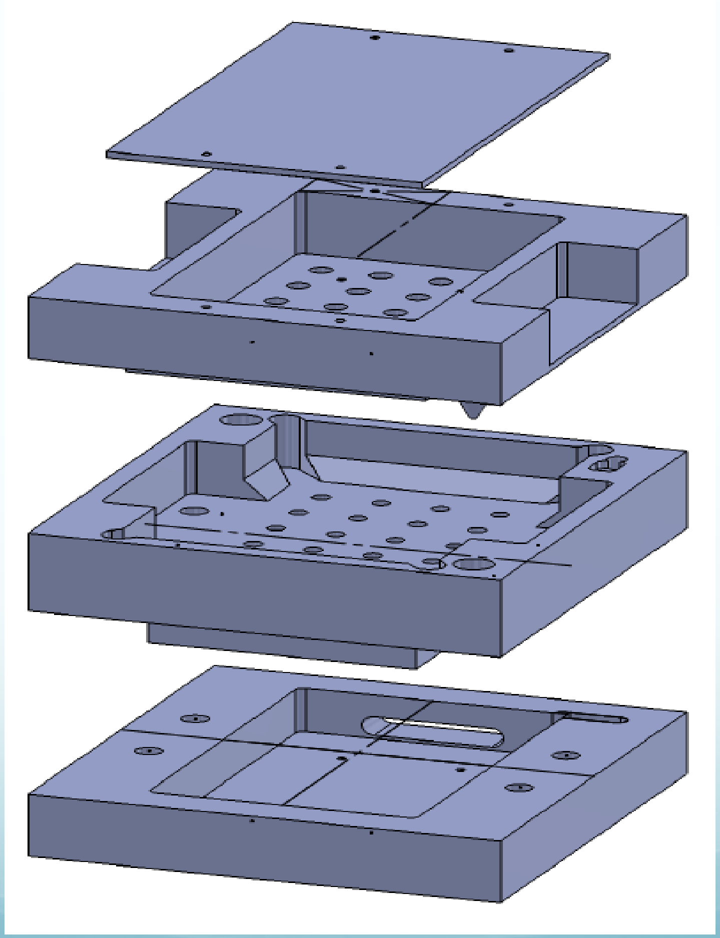
Pulse Width is .0556% or about 2 seconds of period.

Phase delay is 2 min (not sure about this yet since we haven't finished the hardware)

PI Controller Code

```
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
```

The Making of the Plastic Housing



Step 1: Designing

- Bacteria Culture Plate
- 5.035 x 3.365 in
  with .844 in tall
- Primary goal: shine light through wells and read optical density

- Created fully dimensioned drawings on Solidworks
  - Generated a detailed visual of design
  - Made easy to get feedback from customer and group
  - Used SW drawings to actually make it.

Step 2: Learning g-code

- The programming language of a computed numerically controlled (CNC) machine tool.
- Sherline Model 5400 Deluxe mill with 12" base fitted with stepper motor mounts on X, Y and Z axes.
- Located in SOS Lab in EE 359

- Example g-code:
- Want to create a 2x2 inch square w/.375 bit with .2 depth
  - %
  - g20 g01 x.375 y0 z0.1 f10
  - z-0.2 f4
  - y2
  - x2
  - y.375
  - x.375
  - z1 x0 y0 f1000
  - %

Step 3: Making the Design

- CNC Mill Video

http://www.youtube.com/watch?v=quh_tLKN4Yk

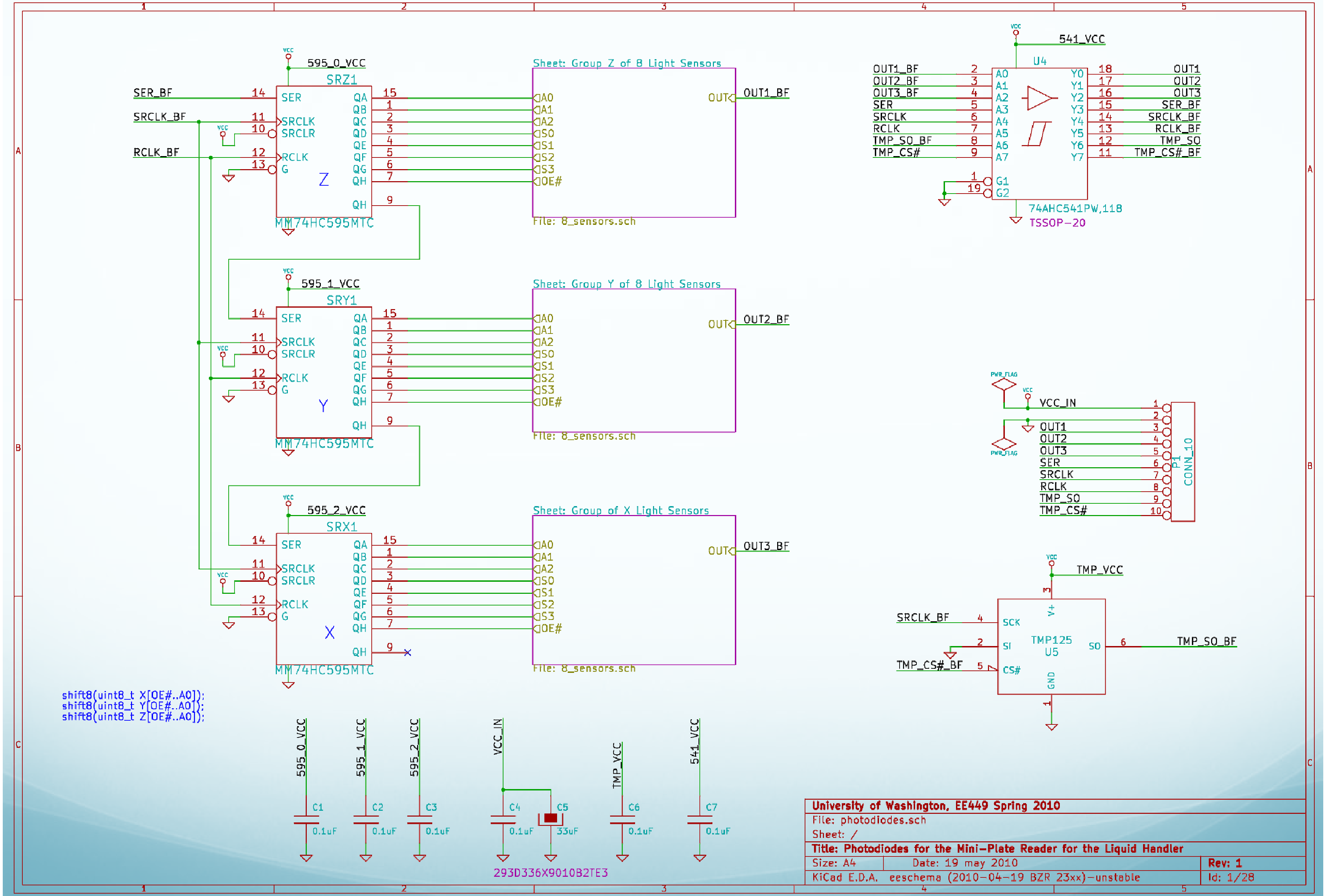
Hardware

Two Main Sections:

- Housing Unit
  - Design
  - Building
- Circuit Design
  - Photodiode
  - LED

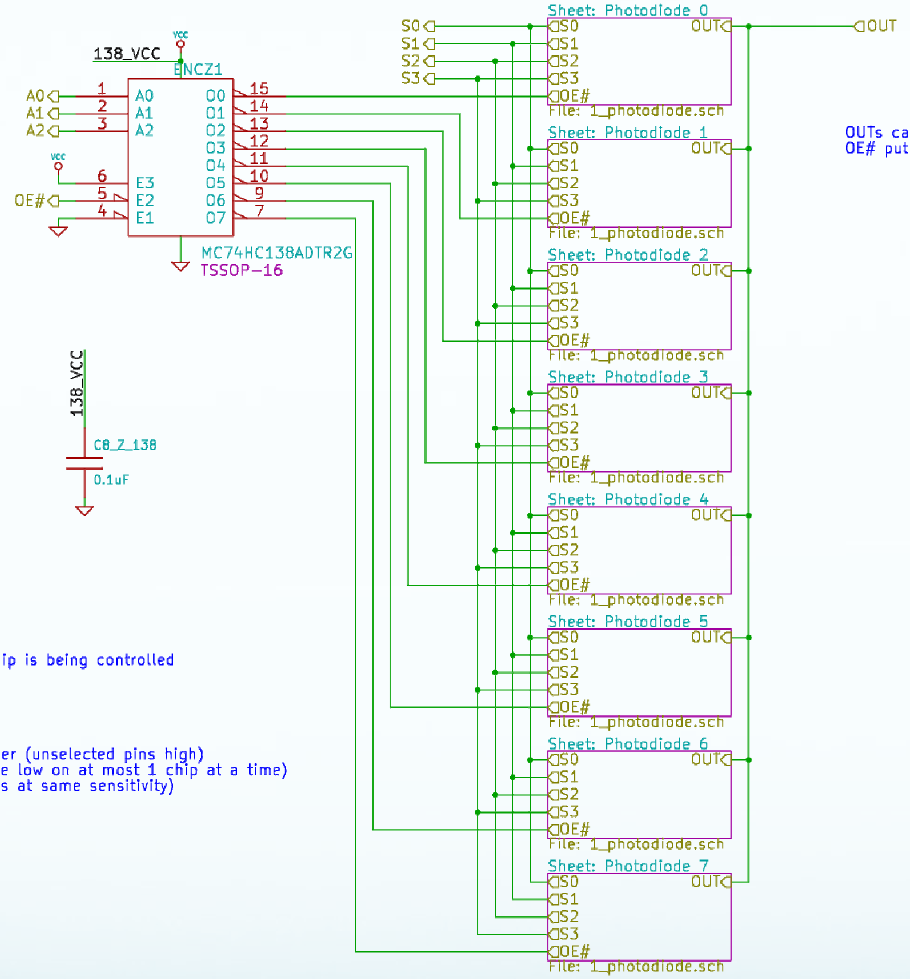
Photodiode Board

- 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
  - Used Bit Banging to control
- Decoders (3, one for each “block”)
  - Used to control the output enables on the photodiodes



```
shift8(uint8_t X[OE#..A0]);
shift8(uint8_t Y[OE#..A0]);
shift8(uint8_t Z[OE#..A0]);
```

University of Washington, EE449 Spring 2010
File: photodiodes.sch
Sheet: /
Title: Photodiodes for the Mini-Plate Reader for the Liquid Handler
Size: A4 Date: 19 may 2010 **Rev: 1**
KiCad E.D.A. eeschema (2010-04-19 BZR 23xx)-unstable Id: 1/28



OUTs can be tied together because
OE# puts OUT in a high-Z state

3 groups of 8 chips

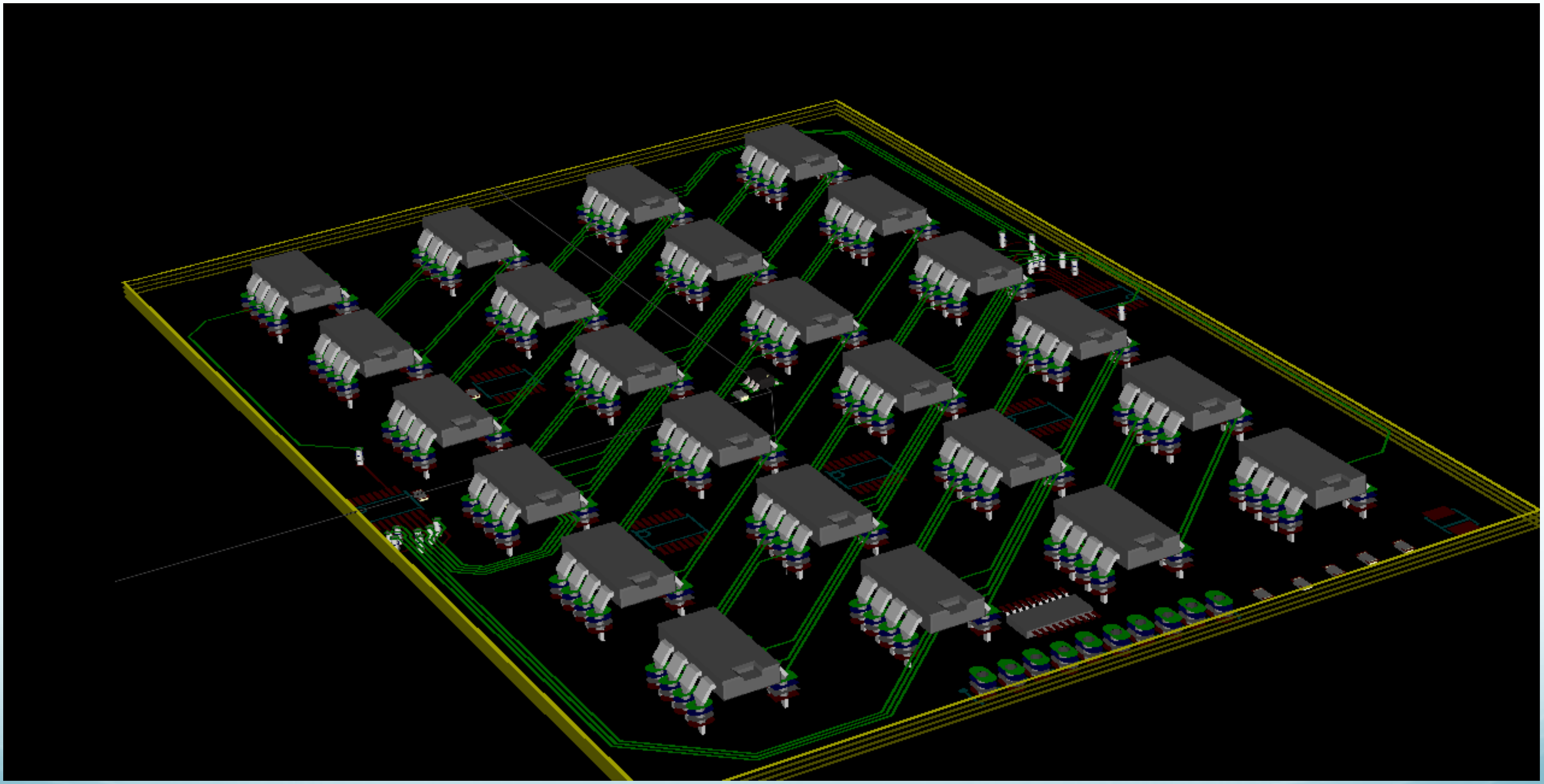
For each group:
5 inputs, 1 output, 3 lines to select which chip is being controlled
total: 9*3 = 27 lines to Arduino + pwr/gnd

OE# = 1 for de-selected chips

ICs:
1. For OE# pins, 74xx138 - active low decoder (unselected pins high)
2. For outputs, connect together (OE# must be low on at most 1 chip at a time)
3. For S0/S1 pins, connect together (all chips at same sensitivity)
4. For S2/S3 pins, connect together.

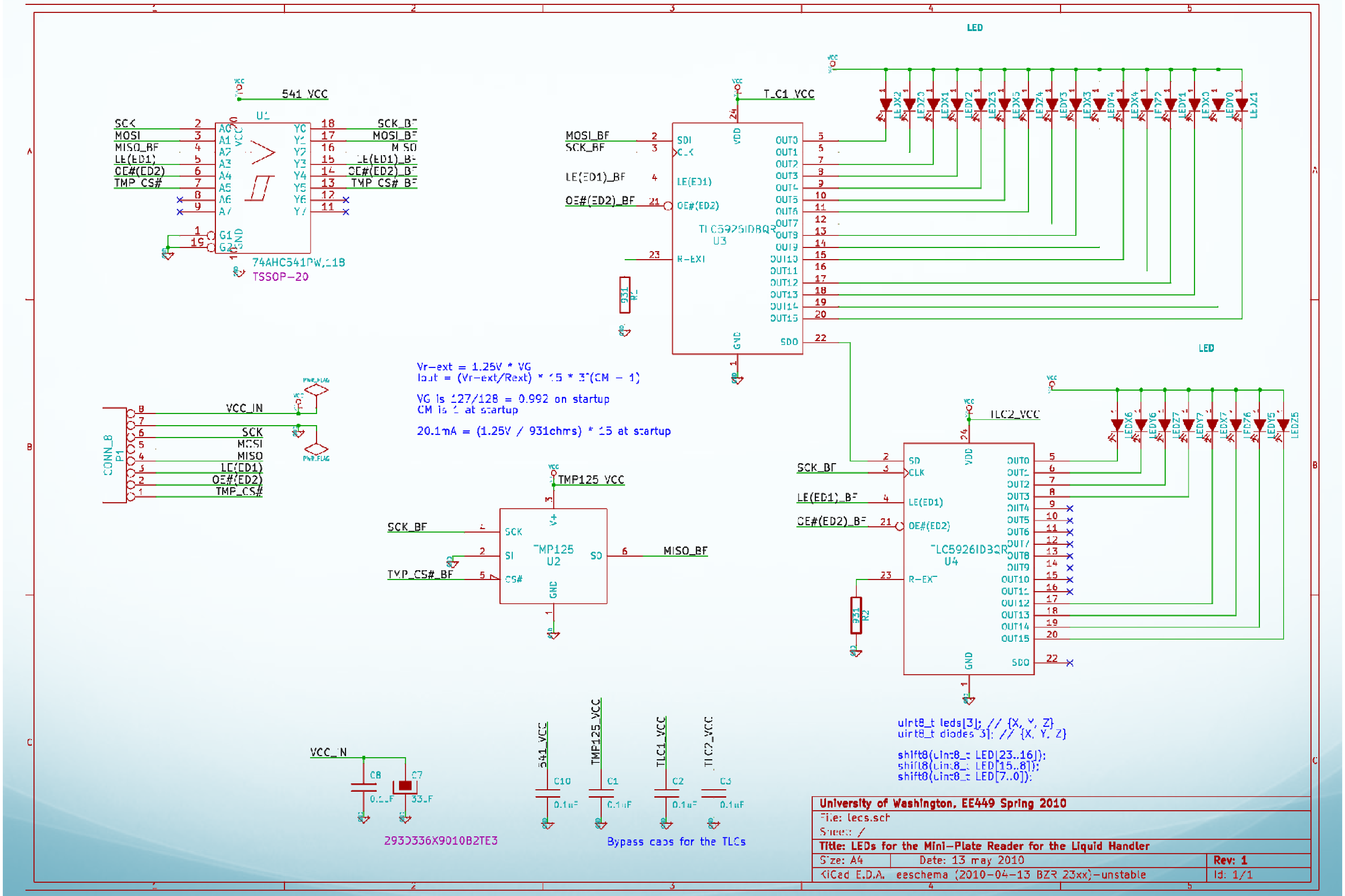
File: 8_sensors.sch
Sheet: /Group Z of 8 Light Sensors/
Title:
Size: A4 Date: 19 may 2010 **Rev:**
KiCad E.D.A. eeschema (2010-04-19 BZR 23xx)-unstable Id: 20/28

Photodiode 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 and sends out parallel output to 16 different LEDs
  - Used SPI (Serial Peripheral Interface) to connect



Vr-ext = 1.25V * VG
Iout = (Vr-ext/Rext) * 15 * 3^(CM - 1)

VG is 127/128 = 0.992 on startup
CM is 1 at startup

20.1mA = (1.25V / 931ohms) * 15 at startup

```
uint8_t leds[3]; // {X, Y, Z}
uint8_t diodes[3]; // {X, Y, Z}

shift8(uint8_t LED[23..16]);
shift8(uint8_t LED[15..8]);
shift8(uint8_t LED[7..0]);
```

University of Washington, EE449 Spring 2010

File: leds.sch

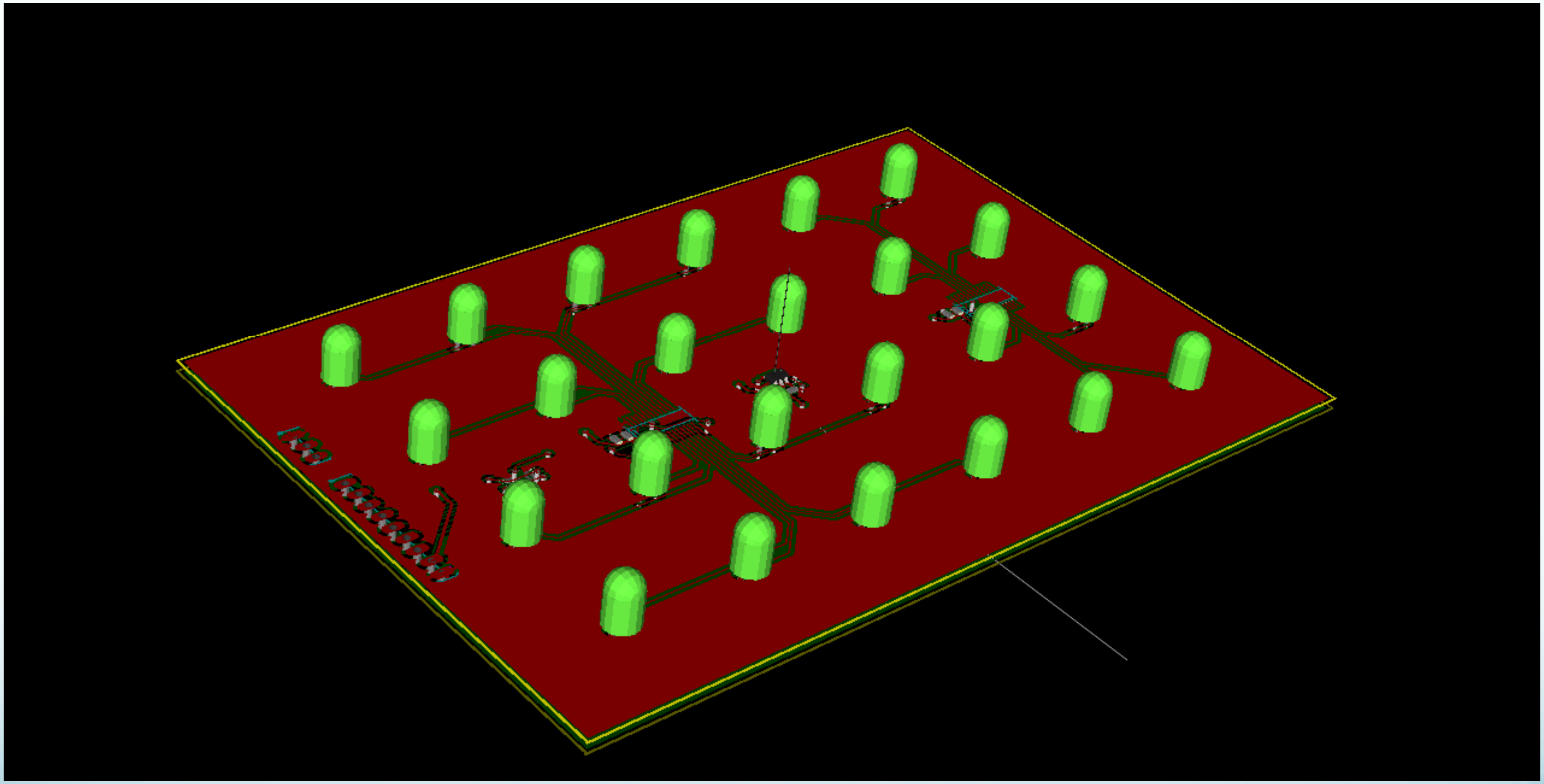
Sheet: /

Title: LEDs for the Mini-Plate Reader for the Liquid Handler

Size: A4 Date: 13 may 2010 **Rev: 1**

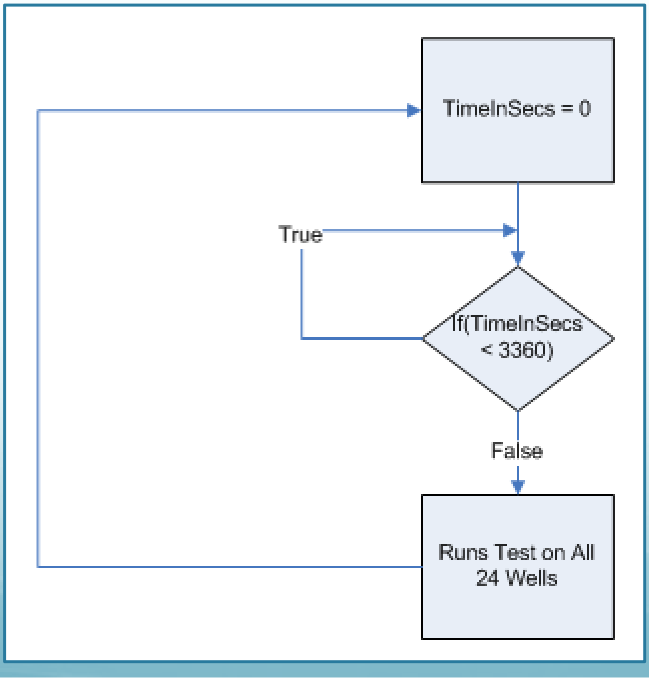
KiCad E.D.A. eeschema (2010-04-13 BZR 23xx)-unstable Id: 1/1

LED Board



Software Design

- Use an interrupt that fires each second
  - Basically establishes operating system that checks the optical density of the wells every hour
  - Give each LED 5 seconds to warm up
  - Takes 240 seconds to read all 24 wells (10 seconds/well)



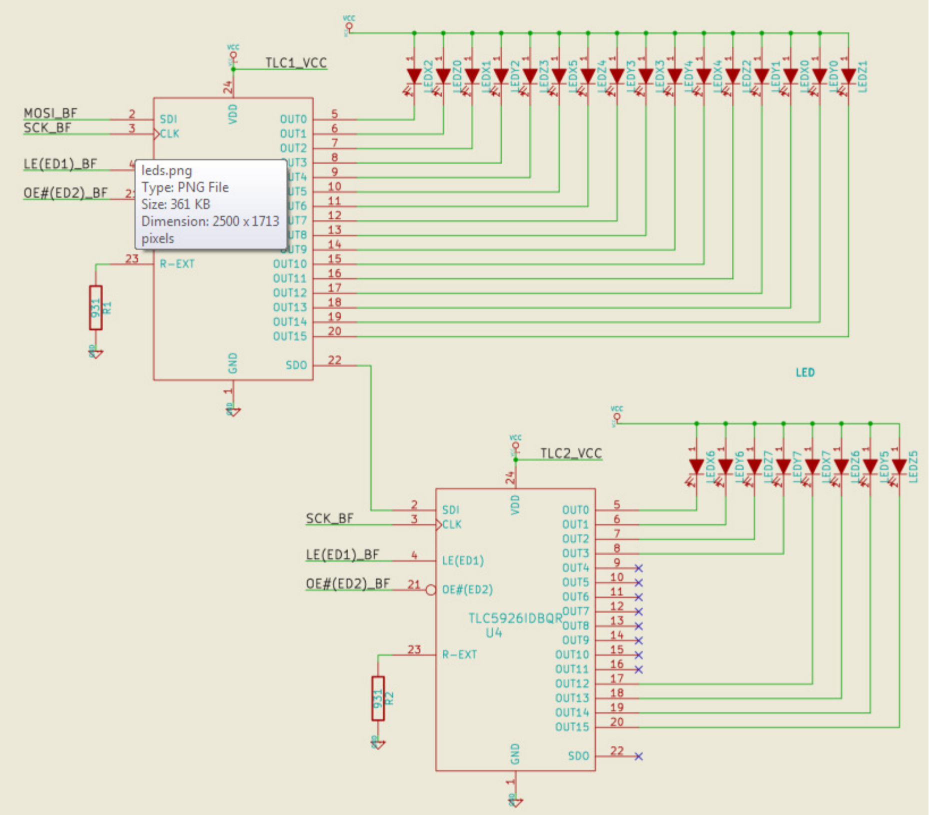
Software Design (photodiodes)

- Originally were going to use 3 UARTs (Universal Asynchronous Receiver/Transmitter) in SPI mode to control each of the blocks for photodiodes
- Arduino breakout does not have all pins of Atmega1280 available (100 pins on Atmega, only 54 on Arduino)
- Had to use “bit-banging” instead

```
for (uint8_t bit = 0x80; bit; bit >>= 1){
    if (bit & data) {
    }else{
    }
    PULSE_CLK();
}
```

Software Design (LEDs)

- Used SPI to communicate with current drivers
- Current drivers that are on board can only run 16 LEDs, had to daisy chain current driver ICs
- SPI Command was made to send 8 bits at a time
- Used original, just sent it four times to send out all 32 bits



Goals From Last Time

- 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

Videos of Software and Hardware

- Current Driver Test: http://www.youtube.com/watch?v=zd7TAv1FRSw
- PhotoDiode Test: http://www.youtube.com/watch?v=UsX3_8-IDD4