UMassAmherst
The Commonwealth's Flagship Campus

GrowingGreen CDR

Team Members: Austin Hiller, Nate Lemons, Matthew Sargeant, Jason Trainor

Advisor: Professor Kundu

3.23.2020

Who We Are

Austin Hiller CSE

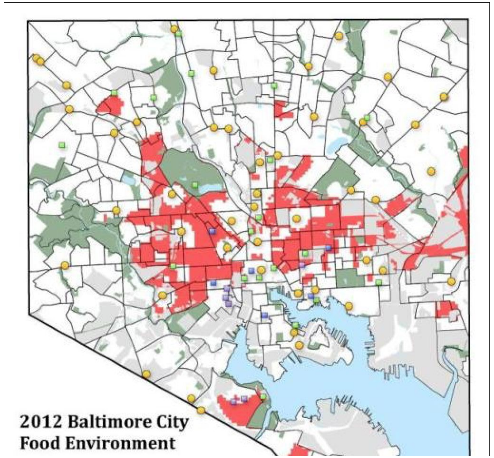
Nate Lemons CSE

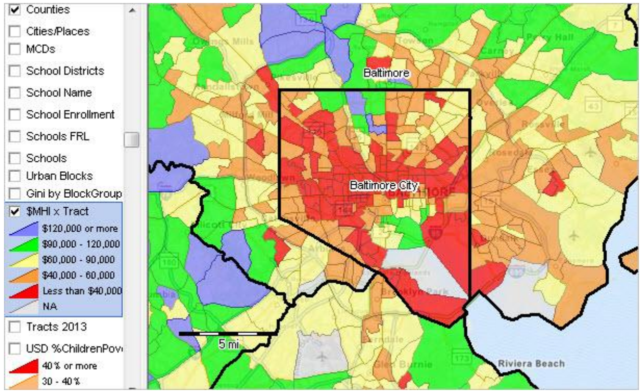
Matthew Sargeant EE

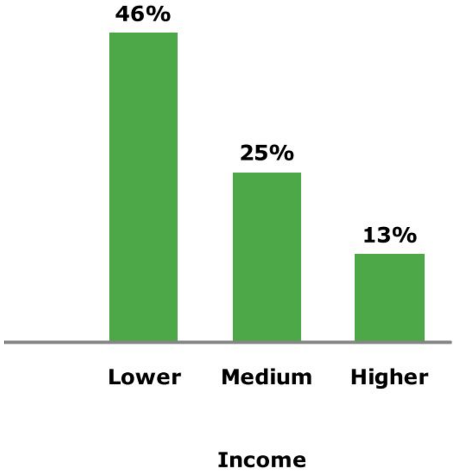
Jason Trainor CSE

Problem Statement

- The Grocery Gap
  - Low income zip codes have 25% fewer supermarkets
  - Rural and urban communities affected
- Proximity to a supermarket is correlated to healthy diet habits
- Low-income neighborhoods have half as many supermarkets as the wealthiest neighborhoods and four times as many smaller grocery stores



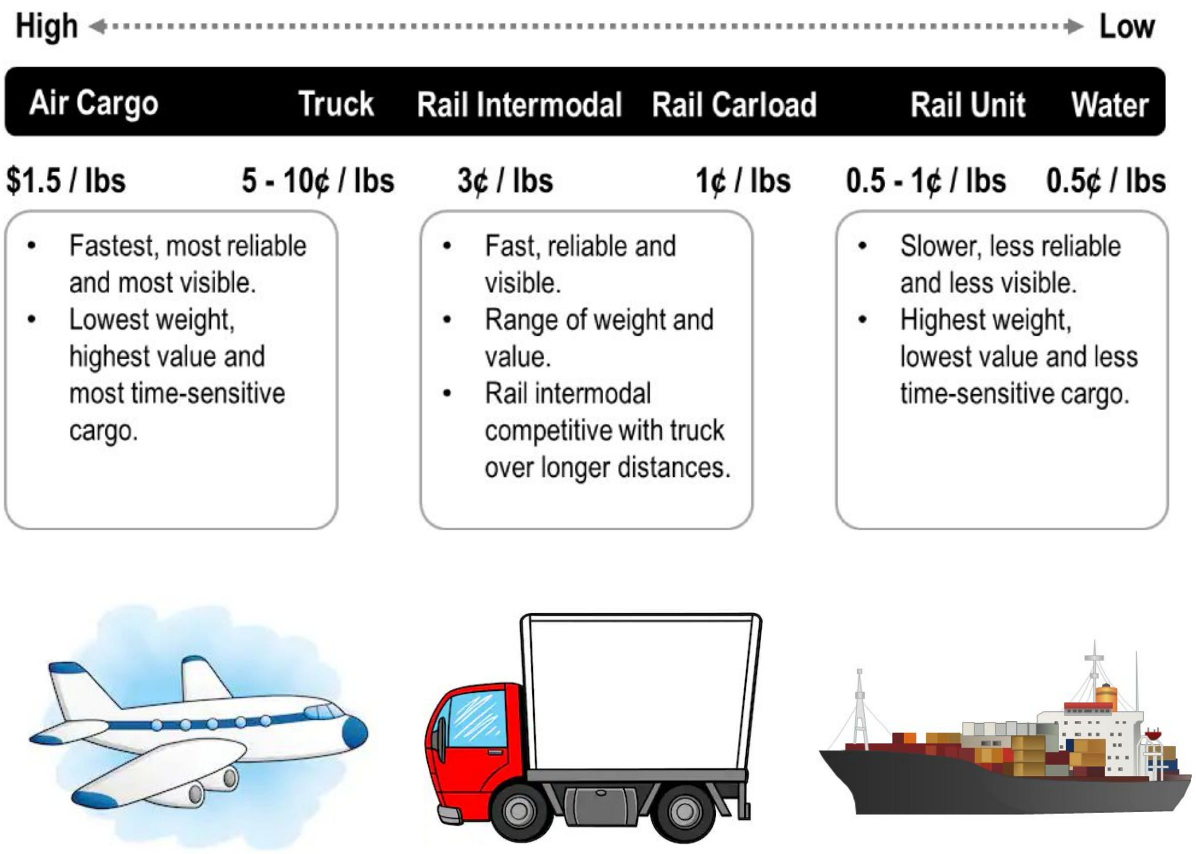




Share of Baltimore neighborhood grocery stores with low availability to healthy food, by income

Problem Statement

- Current food system is very taxing on environment due to transportation
- 10 Kcal of energy from fossil fuel per 1 Kcal of energy from food
- Transport leads to lesser quality, more chemical influence, and higher cost



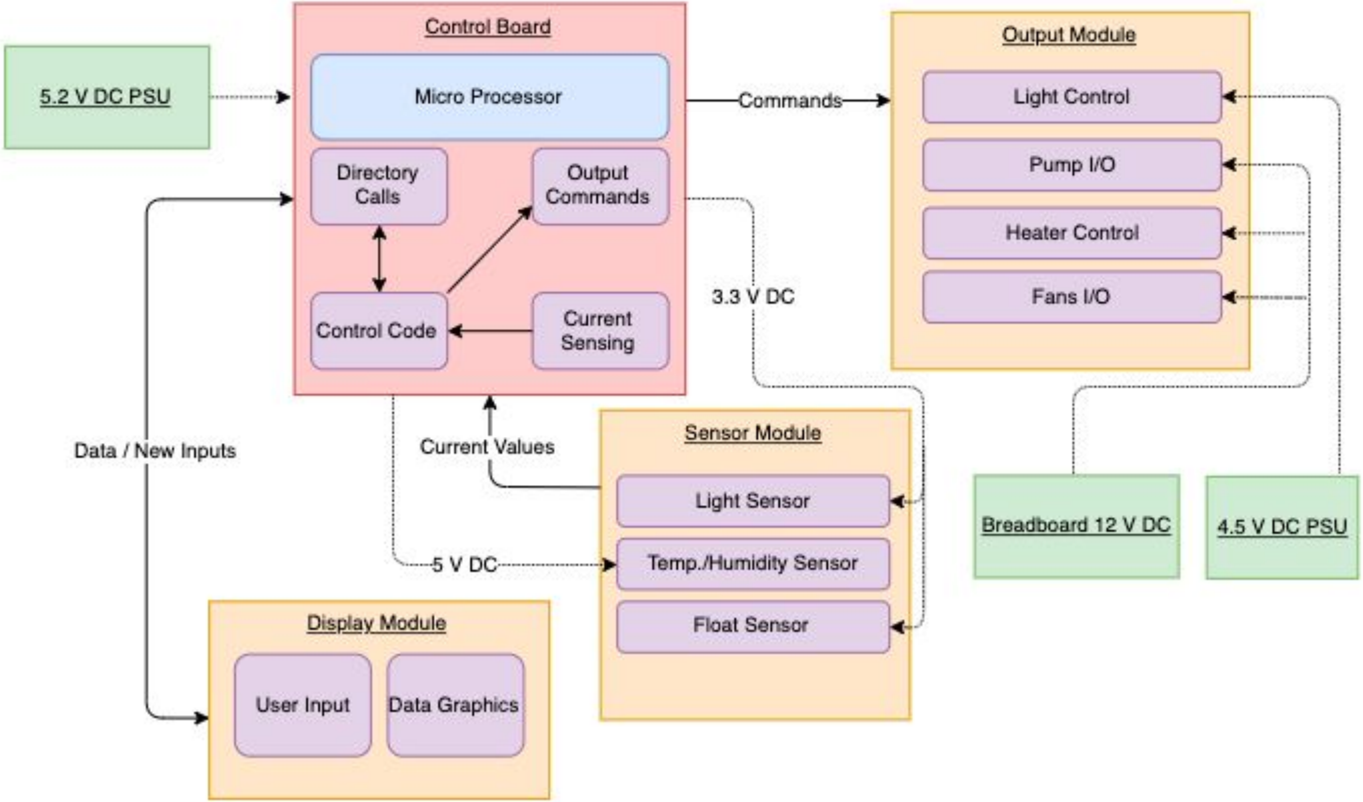
Vision Statement

The GrowingGreen system is a fully automated, energy efficient, in-house growsite with focus on supplying the grower edible vegetation with minimal effort. Our goal is to increase the availability and desire of home growing by simplifying the process through the automation of manual processes, lessening of power consumption, and use of a user console with alerts to keep growers engaged and on schedule. By growing in-house, users will decrease their environmental impact by reducing their carbon footprint and pesticidal use on plants.

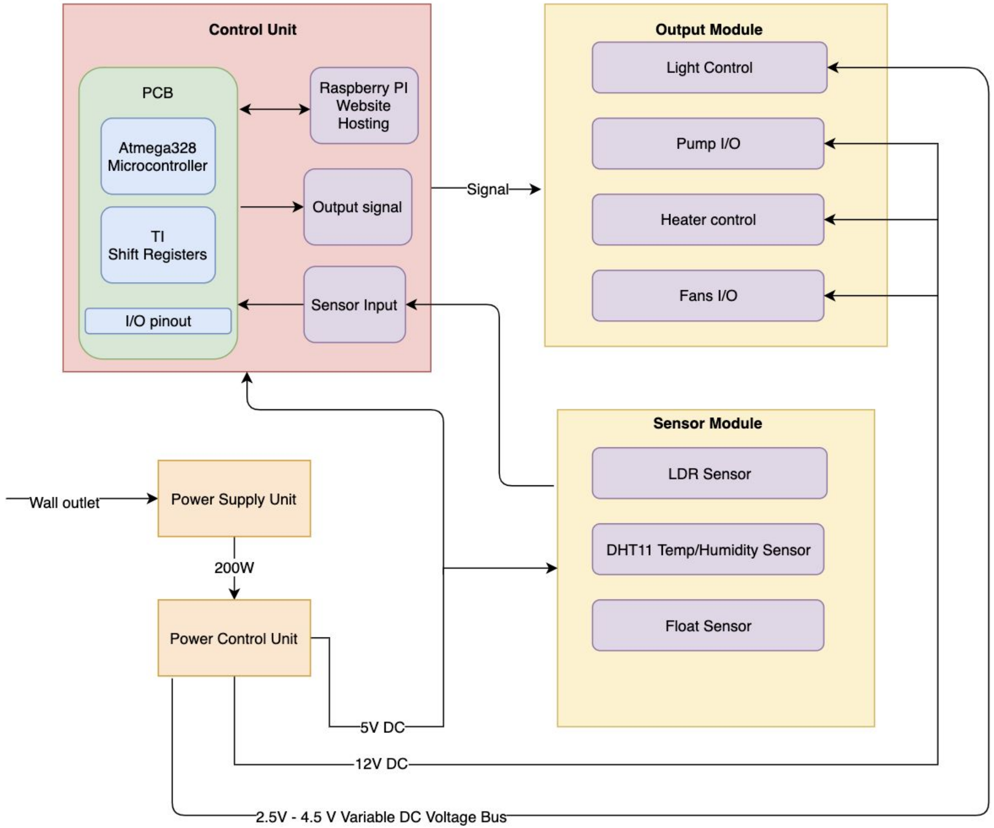
System Specifications

- Reduce power consumption by 3 times the standard
- Produce 24 ounces of product per cycle
- Simplify process of growing so even engineers can grow plants
- Automation through feedback control:
    - Lights
    - Temperature/Humidity
    - Irrigation
- Functional year round
- Data available to user at all times
- Must fit against typical window frame

MDR Block Diagram

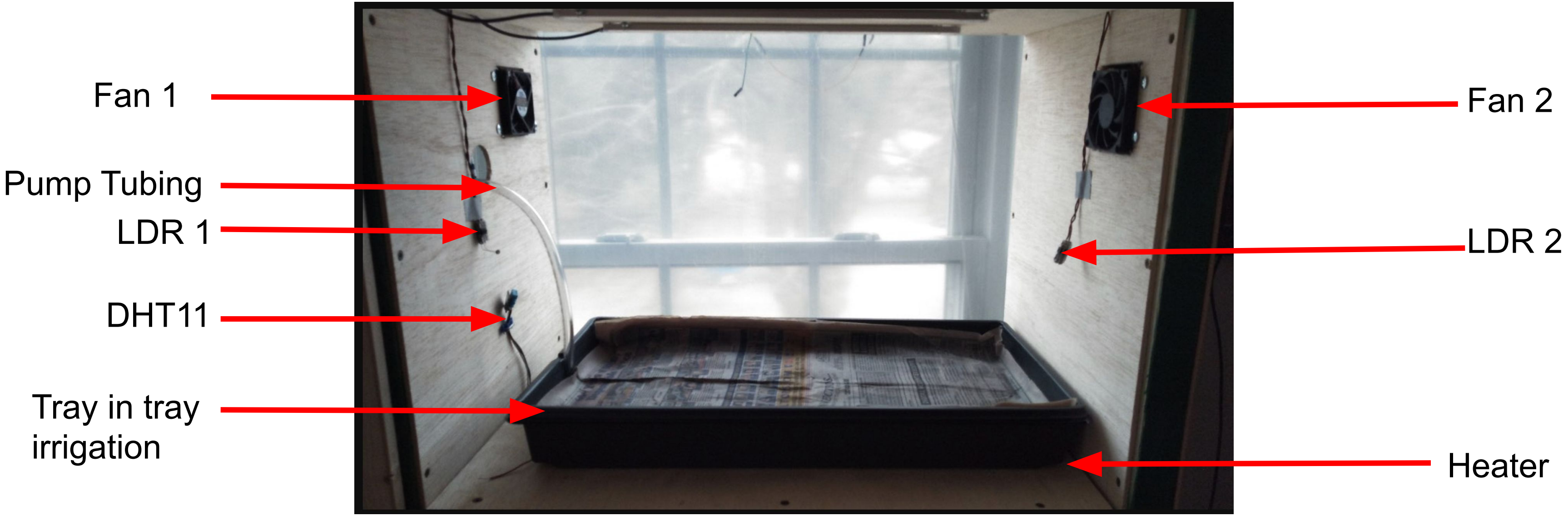


Updated Block Diagram



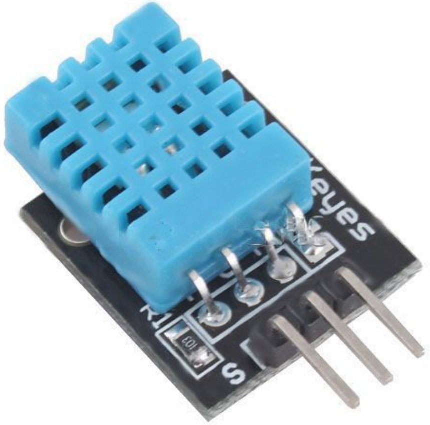
Completed CDR Design

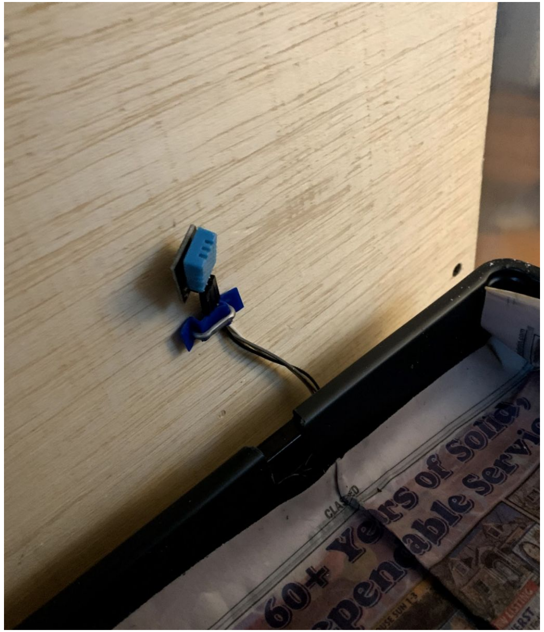
System Overview



Subsystem 1: Temp./ Humidity Sensor

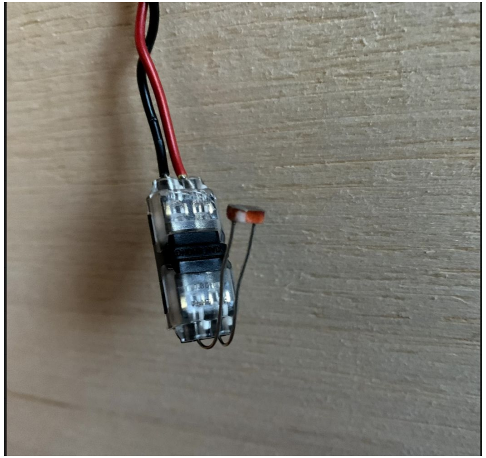
- Temperature and Humidity will be measured with a single sensor. Keyes KY-015 DHT11
    - 20-90% RH range
    - 5 V DC Supply
- Each tier will have a DHT11 located on the wall of the unit





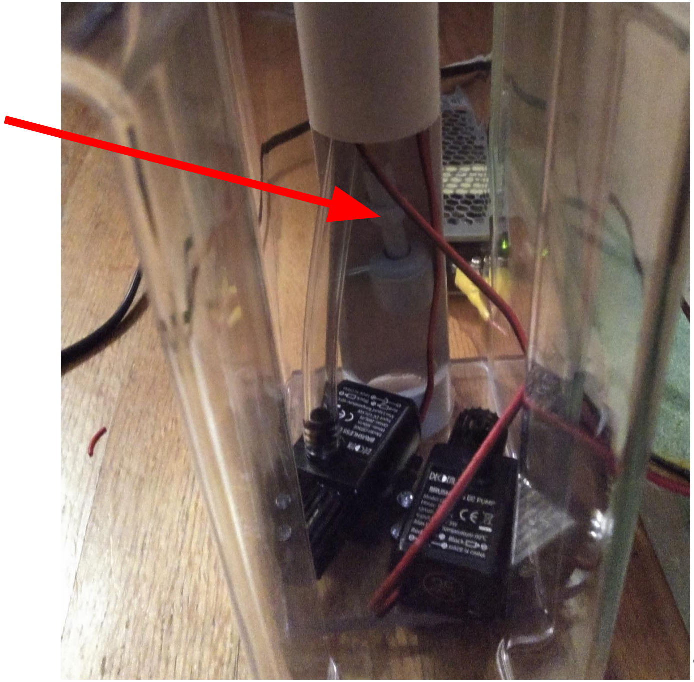
Subsystem 1: Light Sensor

- LDR will be used for our light sensors
- Each tier will have 2 LDRs located on the outer walls of the box



Subsystem 1: Float Sensor

- Anndason PP Float Switch
    - User is notified reservoir is “empty” when button is pressed
        - Pump is turned off

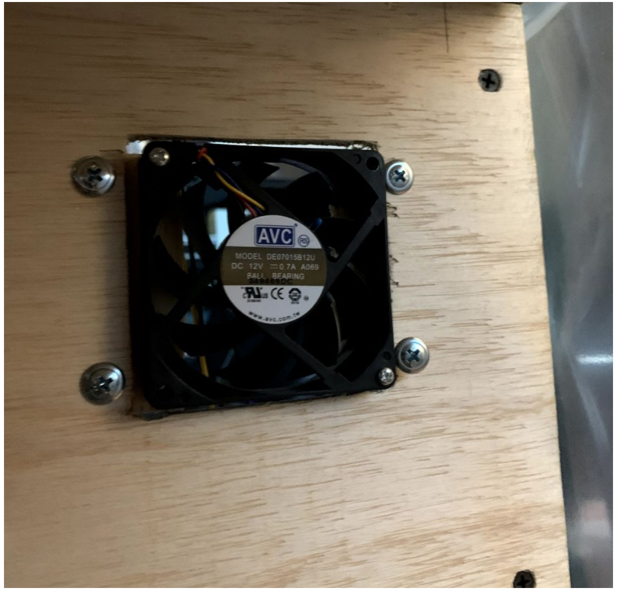


Subsystem 2 Irrigation

- Tray-in-tray system
  - Water supplied from reservoir containing:
    - 12V Decdeal Submersible Water Pump
    - Anndason PP Float Switch
  - Watering cycle is executed each morning

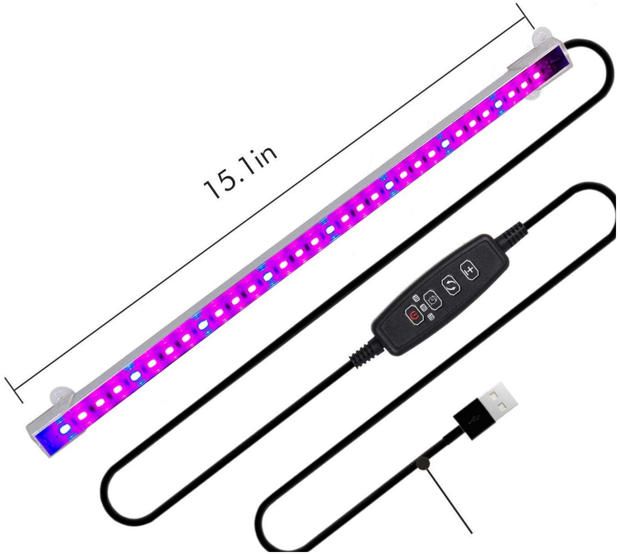
Subsystem 2: Air Flow

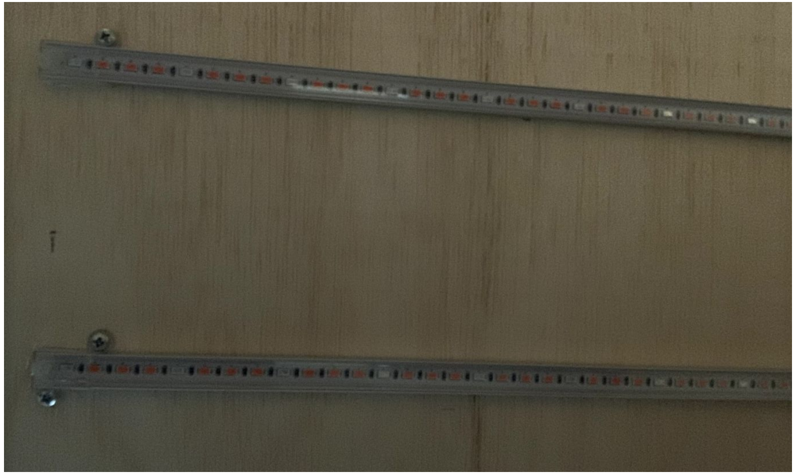
- Four 12V fans used to circulate air if temp/humidity threshold is exceeded
  - 2 fans per tier located on both sides of the greenhouse
    - One is used for intake and the other is used for exhaust



Subsystem 2 LED

- Sondiko LED Grow Light Strip
  - Using 2 LED strips per tier to supplement 200 sq inches of microgreens
  - Using red and blue light spectrums to supplement direct sunlight





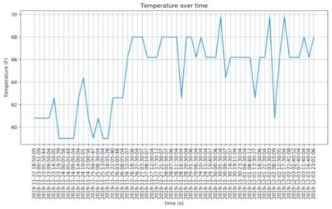
Subsystem 2: Heating coil

- Controlled using 12V DC Freeze Stop Heating Cable w/5 Watts/ft output
- ~16 in = 6.25 W output
- Self-regulated to 85 F to prevent burning out
- One 16 in. strip per tier generates enough heat to control a 75 degree environment

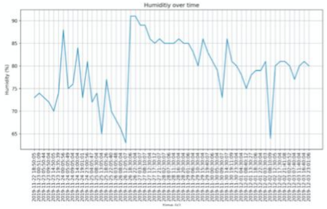
Subsystem 3: Website Interface

- Raspberry Pi will host a custom website for the user to interface with real time greenhouse data
- Because data is stored on the Raspberry Pi, we will also compute real time plots for the user
- We have attached a pdf document in our EMail with an in depth view of our website interface

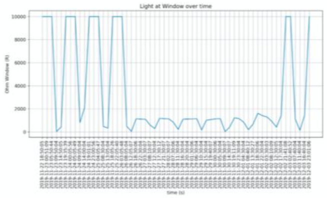
Environment Condition	Current Value
Temperature	68 degrees F
Humidity	55%
LDR Value	649
Reservoir Status	Empty!



Temperature



Humidity



LDR Value

Subsystem 3: LCD Interface

- We will implement a LCD screen on the side of the Greenhouse to show current conditions

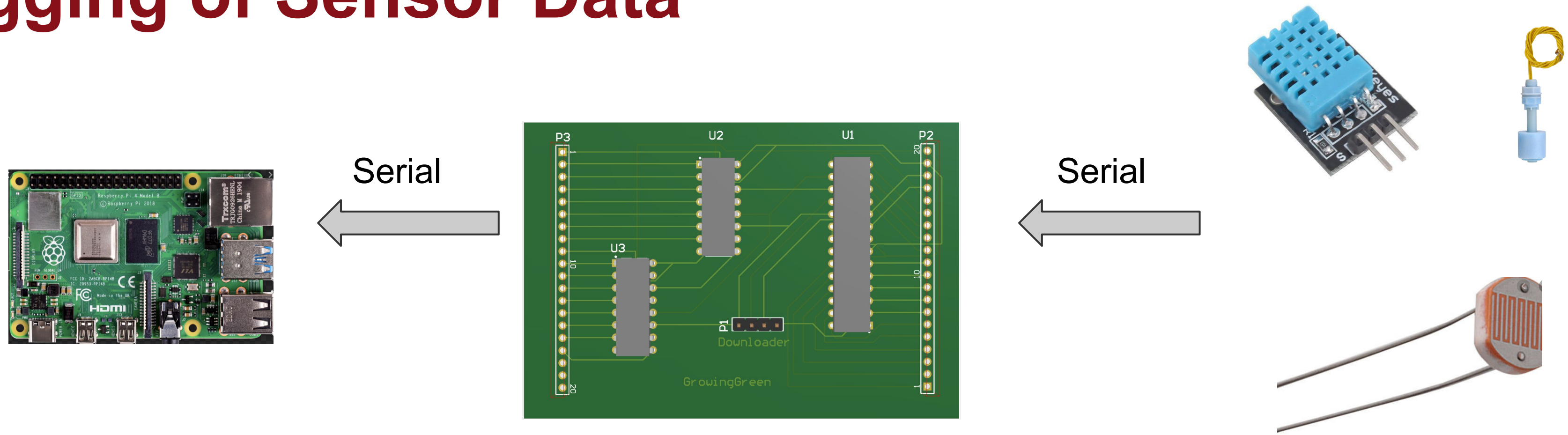
Control Code

- Self-correcting control code that runs on continuous loop
- Make "decisions" for both enclosures at the same time
  - Threshold based with buffers to eliminate over correction
  - Plant chosen can be different for every enclosure
- Method based calls allow for easy expansion of additional enclosures
- Takes signals from Pi to determine which function it should be running
- Assignments gets shifted down the shift register (currently a shift per tier)

```
//method declaration
void assign_register(unsigned int binary_counter);
void push_stack(unsigned char byte);
void controlTemp_Humidity(char enclosure[]);
void offTempControl(char enclosure[]);
void controlLight(char enclosure[]);
void offLight(char enclosure[]);
void controlPump(char enclosure[]);
void offPump(char enclosure[]);
void checkTempHumidity(char enclosure[]);
void convertArr_toDecimal();
void checkFunction();
void LDR_Init();
void checkLight(char enclosure[]);
```

```
//determine which enviornment we are using and assign the proper environment
if(enclosure == "top")
{
    sensor = sensor_top;
    ideal_plant = ideal_plant_top;
    //set to second half of array
    device = 8;
}
else if(enclosure == "bot")
{
    sensor = sensor_bot;
    ideal_plant = ideal_plant_bot;
    device = 0;
}
```

Logging of Sensor Data



1. Atmega328 decodes the data from DHT11 and LDR
2. Atmega328 sends data to raspberry pi via Tx/Rx serial communication channel
3. Raspberry pi decodes the data and updates logs stored on system
4. Raspberry pi computes plots on the logs that are available
5. Raspberry pi updates website automatically and hosts website for user

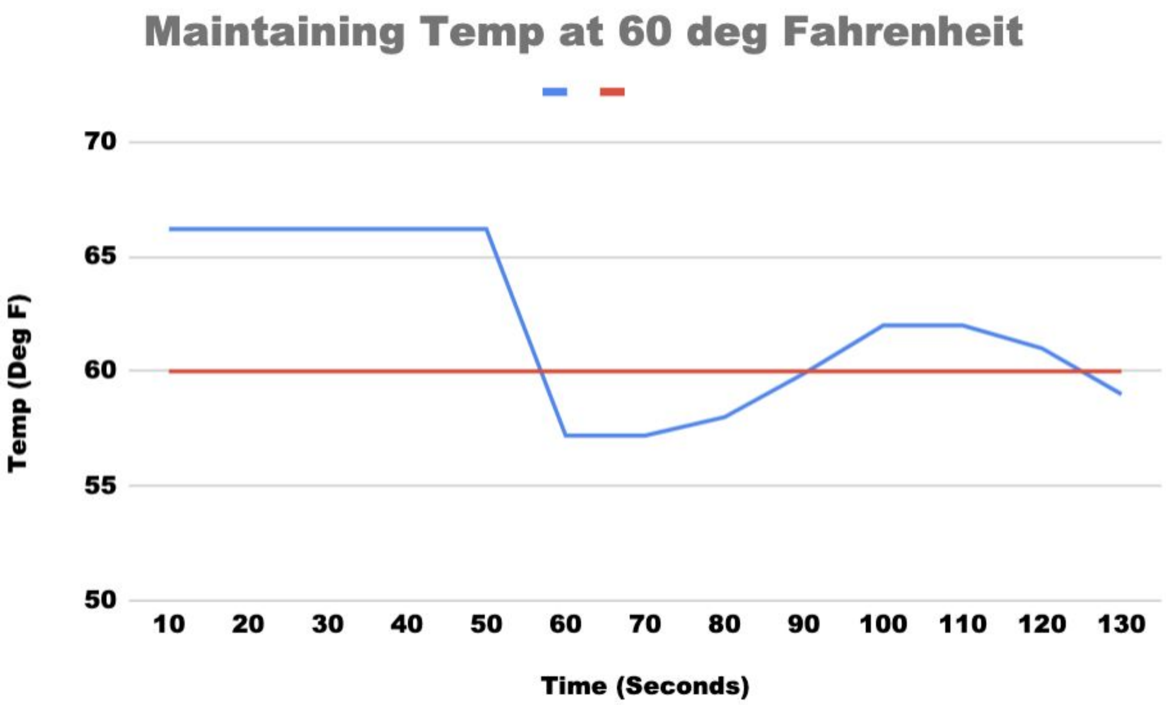
Logging Arrangement

- 3 Log Files per grow cycle
  - data_read.log
  - data.log
  - device.log

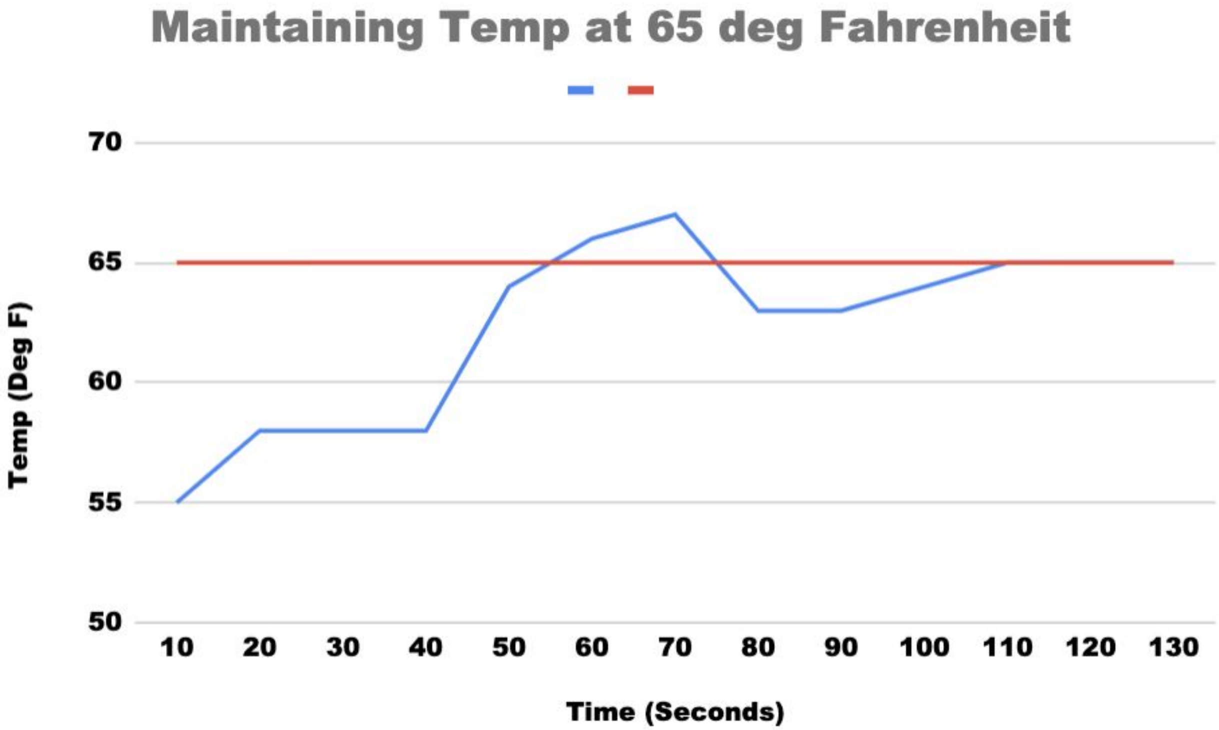
```
2019-12-03 17:30:14,905 - DEVICE - DATA LOGGED
2019-12-03 17:40:02,108 - DEVICE - LIGHT SETTING LOW
2019-12-03 17:40:15,075 - DEVICE - HEATER ON
2019-12-03 17:40:15,076 - DEVICE - OPTION 3
2019-12-03 17:40:15,077 - DEVICE - FANS OFF
2019-12-03 17:40:15,085 - DEVICE - DATA LOGGED
2019-12-03 17:50:01,905 - DEVICE - LIGHT SETTING LOW
2019-12-03 17:50:30,110 - DEVICE - HEATER ON
2019-12-03 17:50:30,112 - DEVICE - OPTION 3
2019-12-03 17:50:30,113 - DEVICE - FANS OFF
2019-12-03 17:50:30,118 - DEVICE - DATA LOGGED
2019-12-03 19:00:05,020 - DEVICE - HEATER ON
2019-12-03 19:00:05,022 - DEVICE - OPTION 3
2019-12-03 19:00:05,023 - DEVICE - FANS OFF
2019-12-03 19:00:05,026 - DEVICE - DATA LOGGED
```

```
2019-12-04 06:30:29,280 - DATA - {'ohm_inside': 3729664, 'ohm_window': 2623315, 'temp': 66.2, 'humidity': 81.0}
2019-12-04 06:40:11,782 - DATA - {'ohm_inside': 485752, 'ohm_window': 320864, 'temp': 66.2, 'humidity': 80.0}
2019-12-04 06:50:04,681 - DATA - {'ohm_inside': 85701, 'ohm_window': 49664, 'temp': 66.2, 'humidity': 80.0}
2019-12-04 07:00:06,672 - DATA - {'ohm_inside': 21606, 'ohm_window': 12343, 'temp': 66.2, 'humidity': 80.0}
2019-12-04 07:10:06,561 - DATA - {'ohm_inside': 8551, 'ohm_window': 4711, 'temp': 66.2, 'humidity': 80.0}
2019-12-04 07:20:09,127 - DATA - {'ohm_inside': 4461, 'ohm_window': 1907, 'temp': 66.2, 'humidity': 80.0}
2019-12-04 07:30:09,525 - DATA - {'ohm_inside': 3259, 'ohm_window': 1796, 'temp': 66.2, 'humidity': 79.0}
2019-12-04 07:40:07,350 - DATA - {'ohm_inside': 2287, 'ohm_window': 1268, 'temp': 66.2, 'humidity': 79.0}
2019-12-04 07:50:04,666 - DATA - {'ohm_inside': 1520, 'ohm_window': 870, 'temp': 66.2, 'humidity': 80.0}
2019-12-04 08:00:04,561 - DATA - {'ohm_inside': 1085, 'ohm_window': 619, 'temp': 66.2, 'humidity': 78.0}
2019-12-04 08:10:04,523 - DATA - {'ohm_inside': 1018, 'ohm_window': 552, 'temp': 66.2, 'humidity': 77.0}
2019-12-04 08:20:03,979 - DATA - {'ohm_inside': 1477, 'ohm_window': 842, 'temp': 66.2, 'humidity': 78.0}
2019-12-04 08:30:06,886 - DATA - {'ohm_inside': 1154, 'ohm_window': 628, 'temp': 66.2, 'humidity': 80.0}
2019-12-04 08:40:04,260 - DATA - {'ohm_inside': 652, 'ohm_window': 400, 'temp': 66.2, 'humidity': 80.0}
2019-12-04 08:50:04,589 - DATA - {'ohm_inside': 625, 'ohm_window': 369, 'temp': 66.2, 'humidity': 79.0}
```

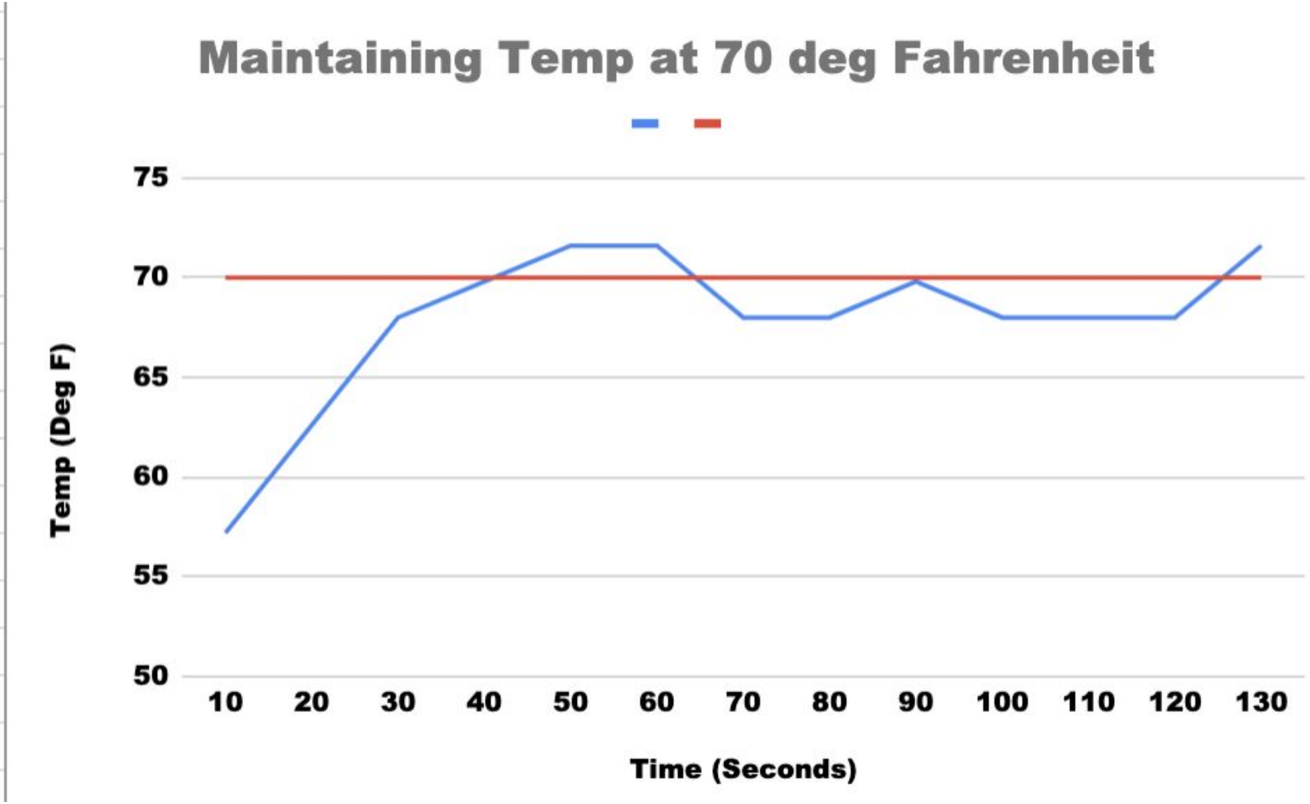
Temperature plotting in new system



Temperature plotting in new system

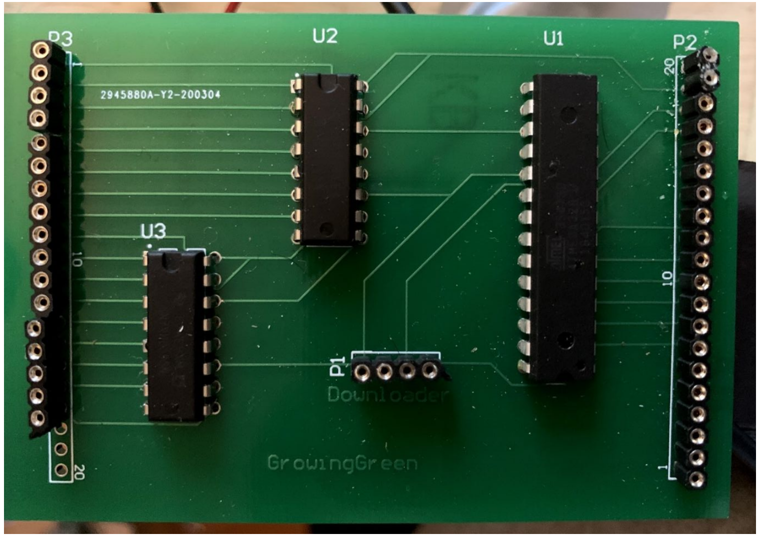


Temperature plotting in new system

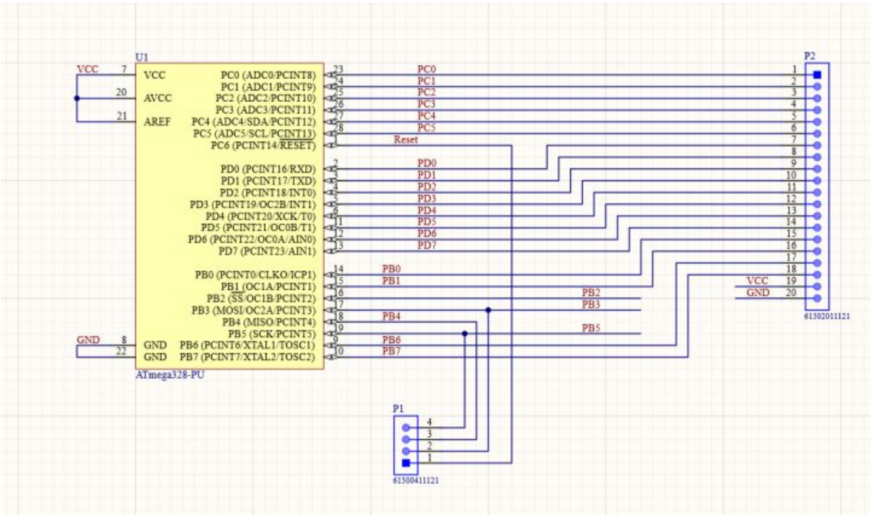


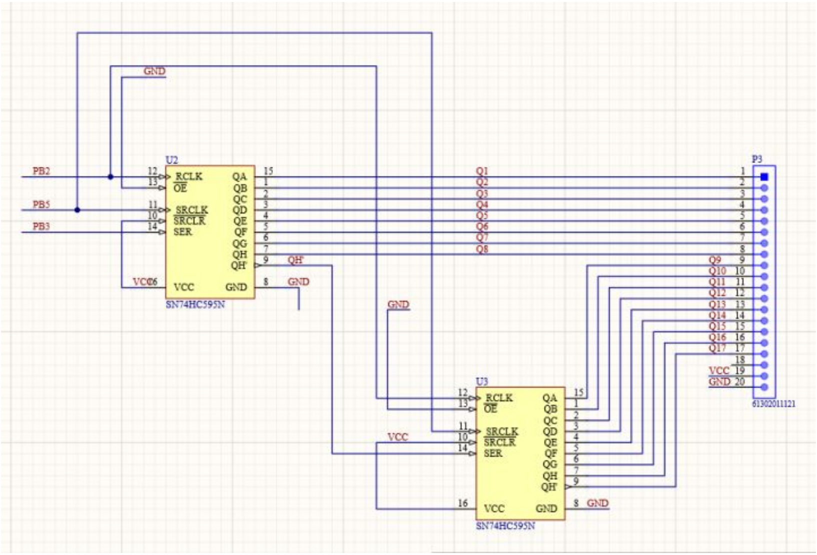
Printed Circuit Board

- Designed to allocate space for all I/O components; i.e. sensors, outputs
  - All sensor will have wired connections
- Atmega328 Microcontroller with connections to TI shift registers

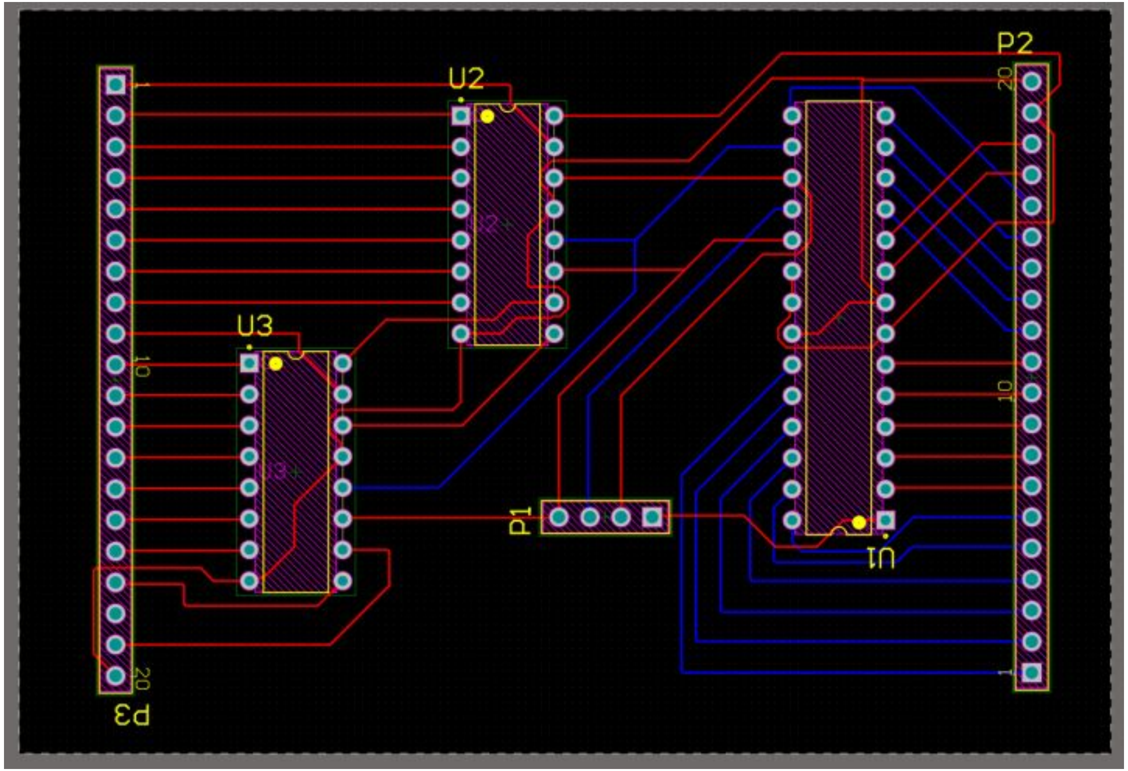


Printed Circuit Board Schematic

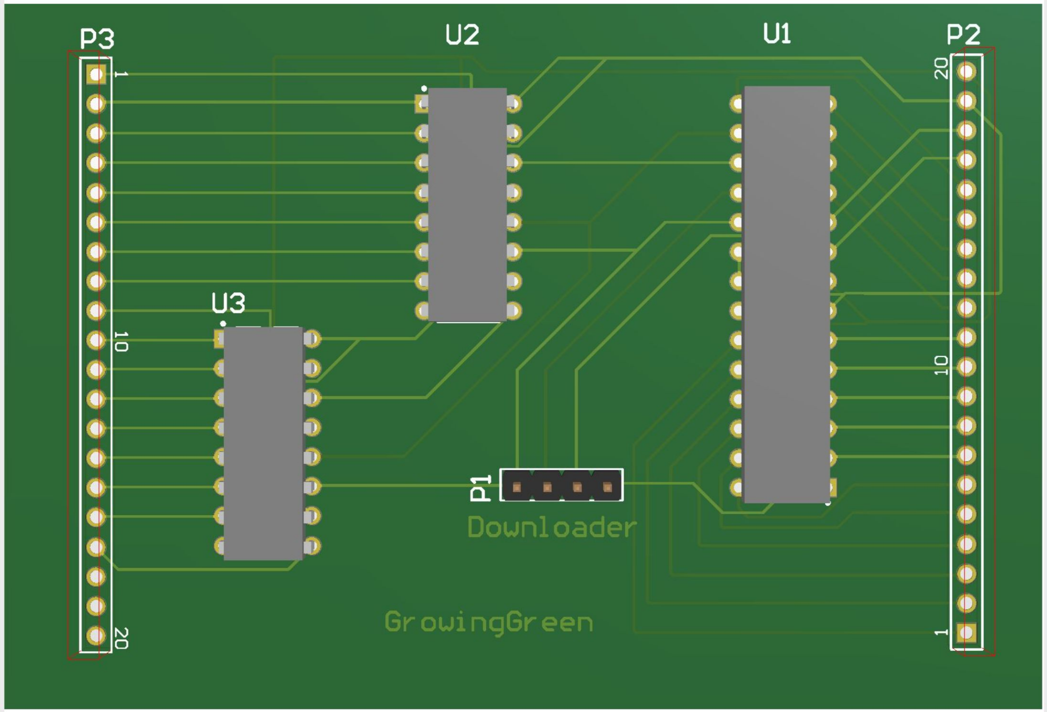




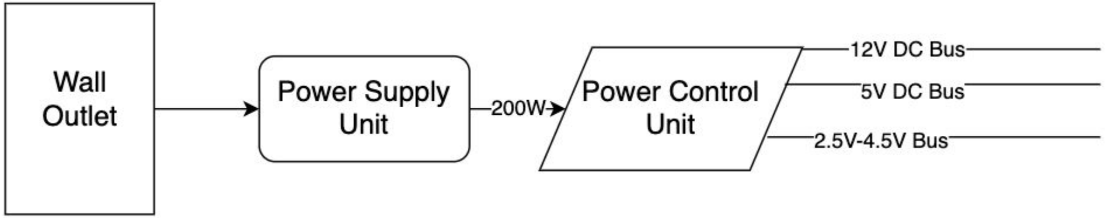
Printed Circuit Board Layer Diagram



Printed Circuit Board Preview



Power Control Unit



- 12V Bus: Fans, Heaters, Pumps,
- 5V Bus: PCB, Raspberry Pi, DHT11, LDR, Float Sensor,
- 2.5V-4.5V Variable Bus: LED Light settings

Problems We Faced for CDR

- Integration from raspberry pi to atmega328
  - Python to C conversion and loss of console for debugging
    - Set up LCD screen to debug data
- Integrating our sensors onto the atmega328
  - loss of helper libraries and simple Raspberry Pi interface
- PCB integrations
  - As factories shut down in China, it was difficult to find a manufacturer to meet our needs
- Connecting multiple perf boards into a circuit
- Dealing with heat dissipation for large voltage drops

Challenges Moving Forward

- Providing continuous delivery of sensor data and output control
- Debugging
- Measuring power and maintaining voltage at specific nodes
- The unknown of how we will get together to work on completing our design

CDR Deliverables

- Completely built prototype with 2 levels for 2 different plants
  - insulated material, UV protected greenhouse
- Fully autonomous greenhouse for 2 seperate grow cycles

  We were unable to complete 2 seperate grow cycles
- Additional sensors and controls for both levels of greenhouse
  - additional LEDs, Fans, pump, sensors
- Bluetooth communication of data to computer or application

  Updated to serial Tx/Rx data transfer

Austin Hiller

PCB

- Designed and manufactured PCB
- Tested and populated PCB

Plotting and Logs

- Designed process to store logs and compute graphs on raspberry pi
- Created hierarchy to supply website with correct and most recent data and graphs

Programming

- Built Process to continuously integrate and test code on Atmega328
- Designed and helped integrate LDR Code
- Designed and helped integrate LCD Code

Nate Lemons

- Chose parts based on requirements found through research
- Determined ideals for environment variables
  - Worked with Jason to ensure thresholds resulted in desired environmental conditions
- Designed and built irrigation and air flow systems

CDR:

- Designed and assembled new irrigation system
- Designed and made necessary cutouts for components in side panel
- Worked with Jason to design bjt switch circuit and then soldered out its perfboard for outputs (shorted - was replaced)
- Designed/developed greenhouse UI

Matthew Sargeant

- Built prototype enclosure with Jason
- Wired and soldered perfboards for final enclosure
  - 12 V bus; 5.2 Vbus; 2.5/3.5/4.5 V bus; switching networks;pcb (decommissioned)
- Determined heating element
- Researched power distribution

Jason Trainor

- Wrote and tested control code (Python -> prototype and C -> current enclosure)
- Created cron job to automatically run programs
- Integrated other’s parts to raspberry pi / PCB
- Made calls to plant directory (information on .csv) / (on “.h” file for final)
- Worked with Matt to construct prototype and final enclosure
- Updated Block Diagram
- Working on Anomaly Detection
- Demo Codes (bench sides, MDR, CDR)

NEW SINCE MDR

- Integrated shift registers, atmega328, LCD for testing
- Implemented the DHT sensors and LDR (w/ Austin) and all outputs with Atmega328 (PCB)
- Designed/made functioning breadboards for others to perf out / to make a PCB of
- Tested/Integrated other member’s parts to PCB
- Created a way to send signals from Pi to Atmega328 (PCB)

Gantt Chart

Due to the volatility of the future of SDP we will provide our next Gantt chart at a future date.

Budget

- Greenhouse Unit box ~ $ 75 - 125
- Sensors
  - Hygrometers ~ $15
  - Photosensitive elements ~ $5
- PCB ~$60
- microcontroller 3 x 10 ~ $30
- Materials for growing 15 cycles x ~ $3 =~ $45
- Heating cable ~$9/ft x 2 = $18
- Fans ~ $5
- Lights ~ $30
- Irrigation ~ $50
- Miscellaneous ~ $50
- Total estimate ~ $385-435

Budget Status

- Fan x 4 - $9.59
- LDR x4 - $6.45
- Vinyl Tubing x1 - $5.99
- Sondiko Grow Light x 4 - $63.96
- Sub pump x 2 - $15.98
- Float sensors x 1 - $10.99
- MicroSD card x1 - $12.59
- Heat cable x 2 - $65.50
- Growing Trays $33.37
- Plastic Cover $32.99
- PCB x 5 $19.55
- PSU x 1 $47.50
- Water Reservoir x 1 $13.99

Total - $336.45
Remaining - $163.55

Questions?

UMassAmherst
The Commonwealth's Flagship Campus