UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

HomeGrow

ECE 445 - Senior Design

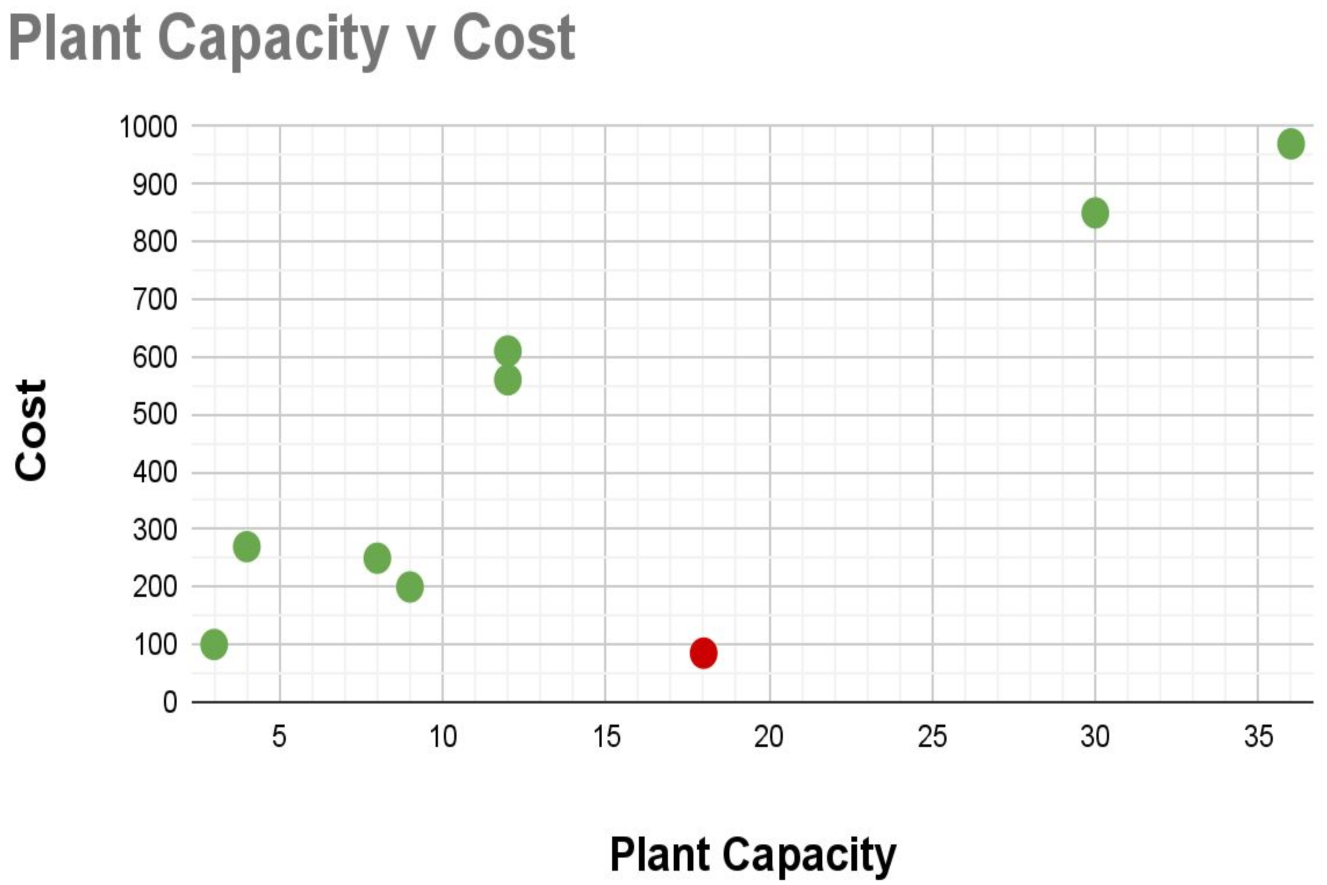
Ciara Ward, Sanjana Sastry, Stephanie Sieben

April 28, 2022

Problem Statement

Lack of Proper Access to Healthy Foods

- Healthy & organic food is difficult to find and often expensive
- At home gardening is expensive, time consuming, and requires extra space
- The current automated watering systems come at extremely high costs
- All of these factors further drives the food gap between high and low income communities



Market Average = $33/Plant

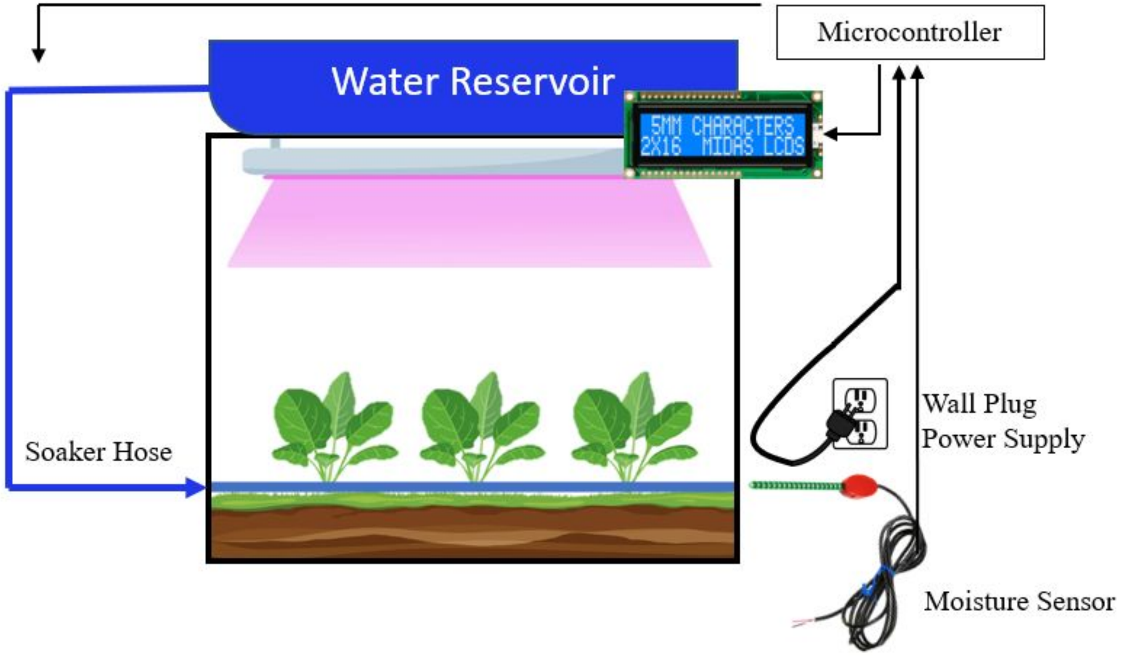
Our Cost = $4.72/Plant

Project Goal

Create an in-home vertical gardening system for people to grow their own produce and herbs for a more sustainable lifestyle

Design

High Level Requirements

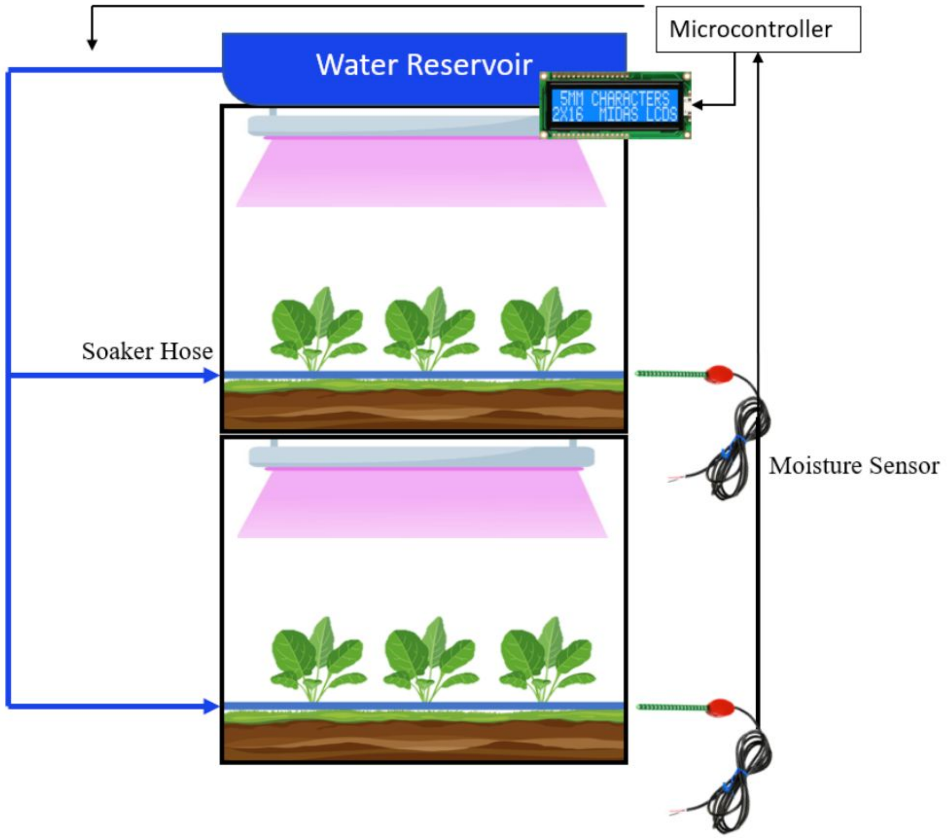


.

- User interface notifies user 10 minutes before scheduled watering cycle with either:
    - Watering cycle is being skipped
    - Solenoid valve will be opened
- 15-minute watering every 24 hours
    - Exception: when moisture levels > 60% saturation
- Control system instructs LED grow lights to stay on for 12 hours and off for 12 hours.

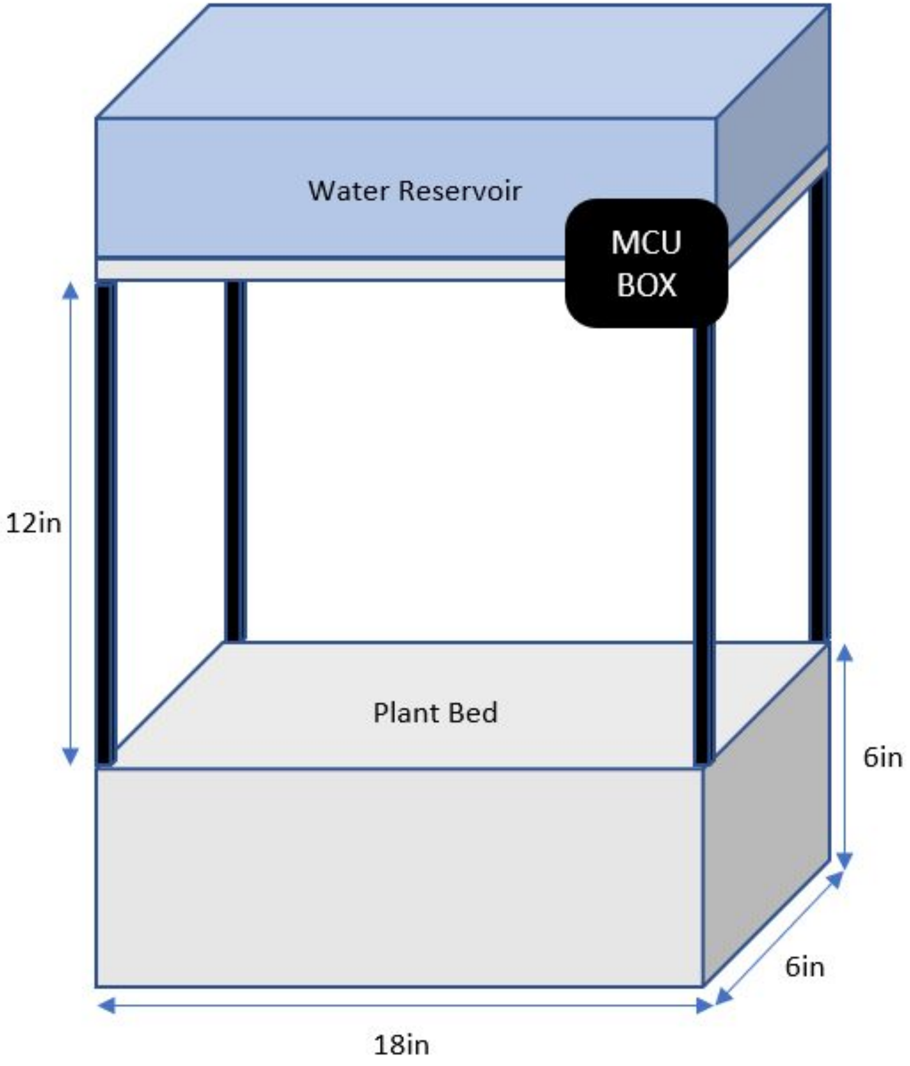
Process and Changes

- Originally planned a multi-tiered device
  - Sized down for efficiency and cost
- Originally planned for 1 gallon water jug as reservoir
  - Refined model with casing and put eyelet at bottom to see water level
- Display discussion
  - LCD vs. LEDs

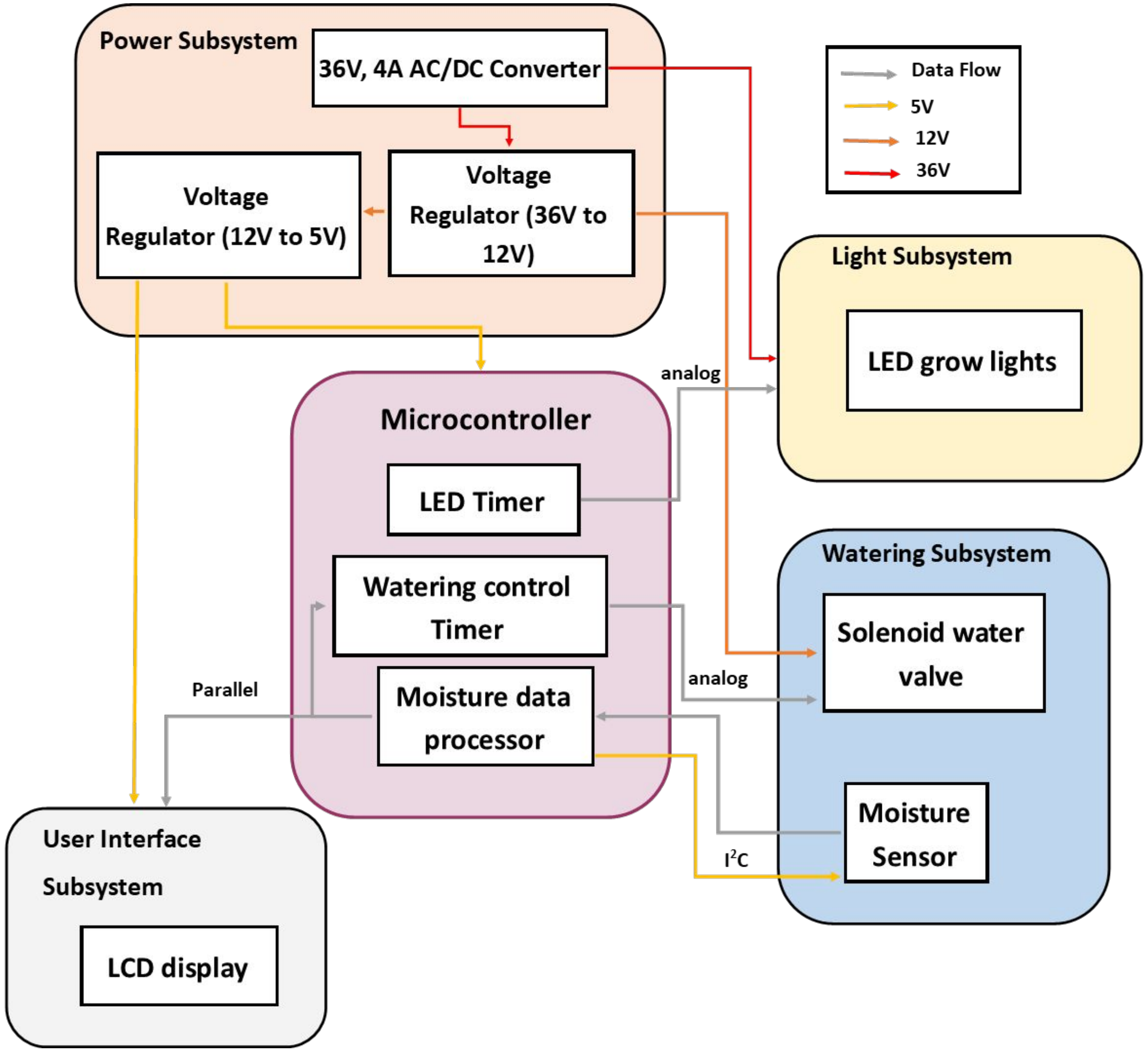


Original Design

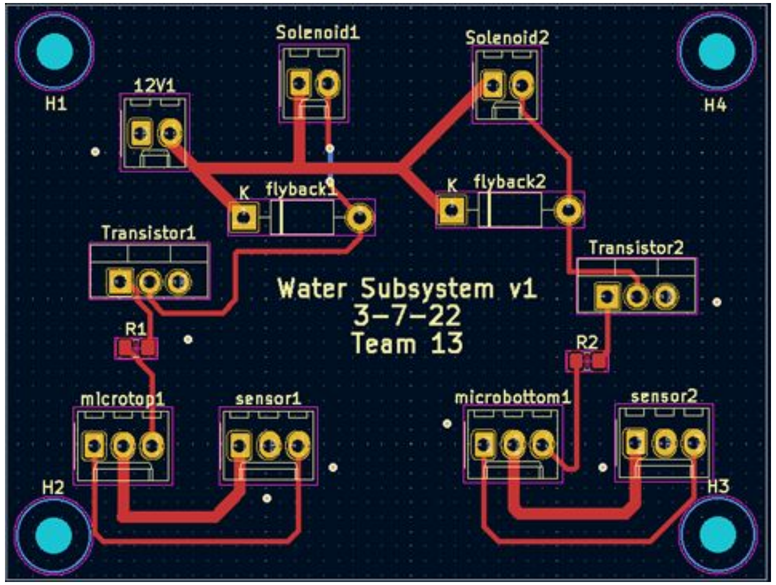
Design Mock-up

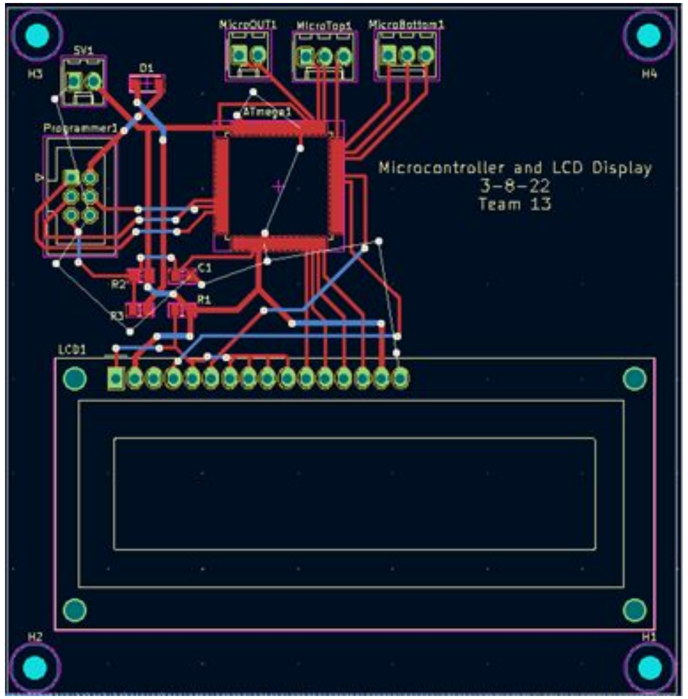


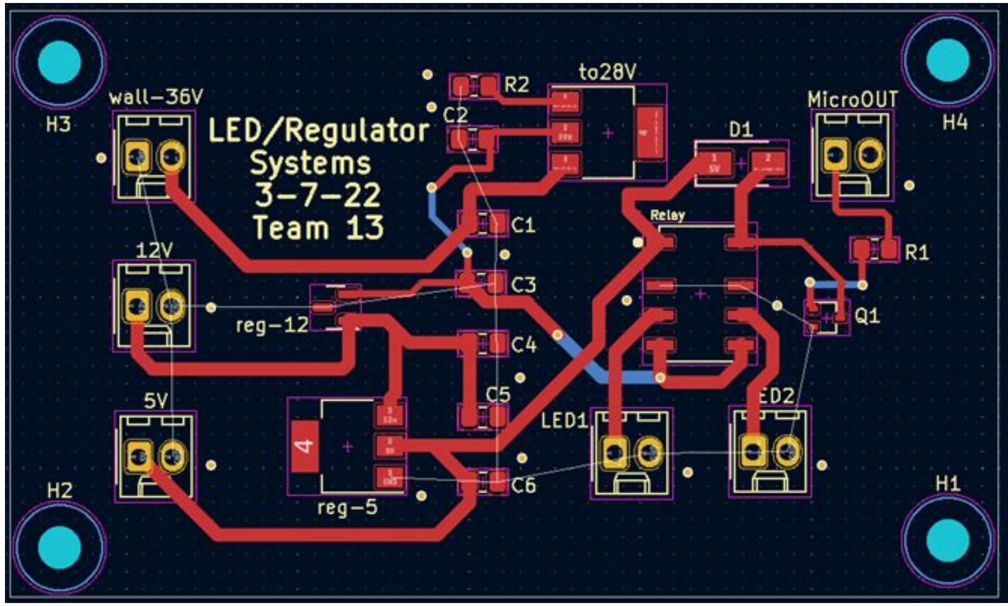
Block Diagram



PCB Layouts

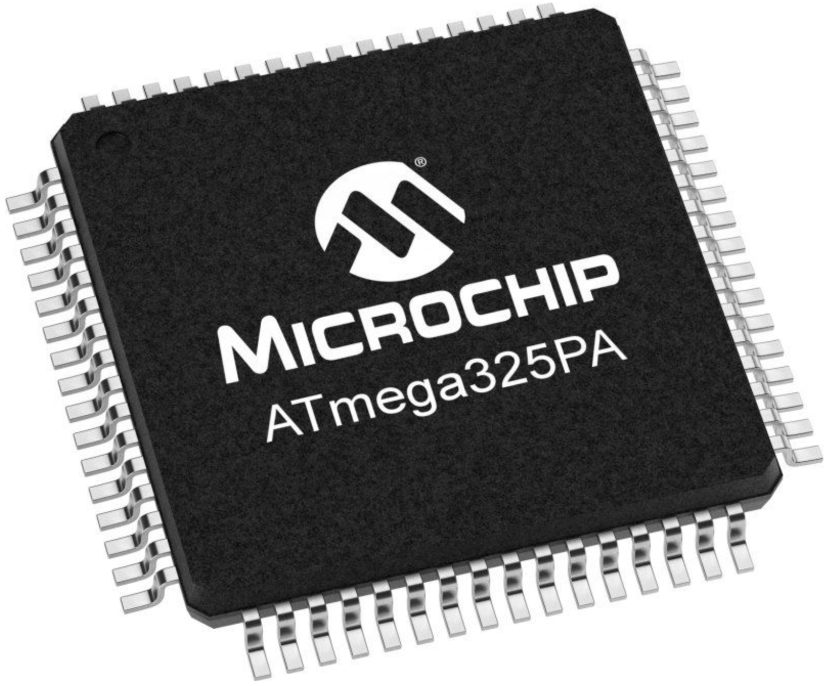

Solenoid1
Solenoid2
12V1
H1
H4
flyback1
flyback2
Transistor1
Transistor2
Water Subsystem v1
3-7-22
Team 13
R1
R2
microtop1
sensor1
microbottom1
sensor2
H2
H3


Microcontroller and LCD Display
3-8-22
Team 13
LCD1


wall-36V
to28V
R2
C2
LED/Regulator
Systems
3-7-22
Team 13
MicroOUT
H3
H4
D1
Relay
C1
R1
12V
C3
reg-12
Q1
C4
5V
C5
LED1
C6
reg-5
H2
H1

Microcontroller Subsystem

- Microcontroller is able to be programmed by the computer
  - Either using Arduino IDE or Microchip Studio
  - Using the SPI programmer



Testing

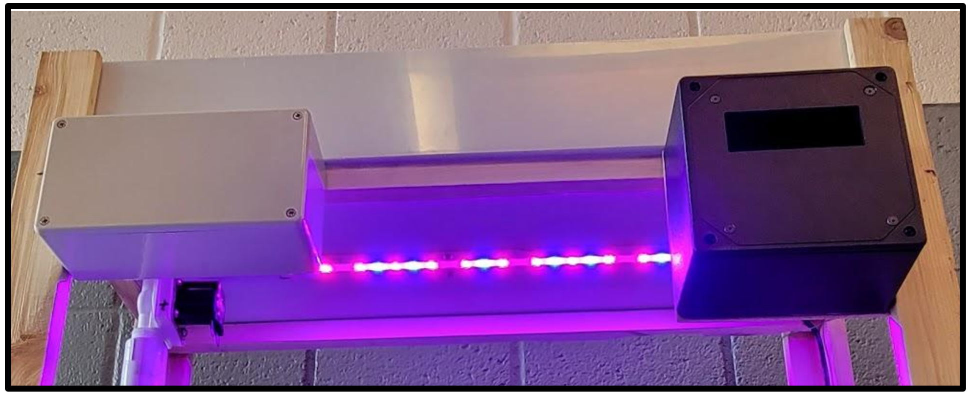
User Interface Subsystem:

- Control system alerts user when water cycle is being skipped or not (10 min prior)

Lighting Subsystem:

- LED grow lights turn on when supplied with 36 volts

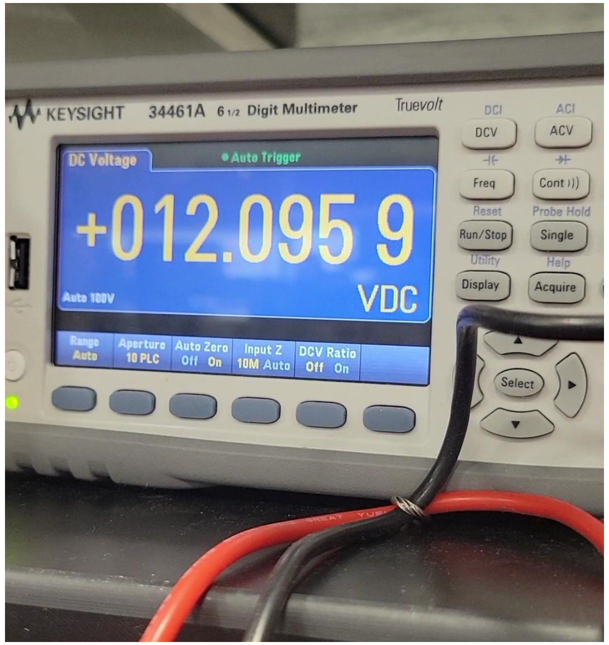
- Lights subsystem adheres to designated 12 hours on, 12 hours off cycle every 24 hours



Testing

Power Subsystem:

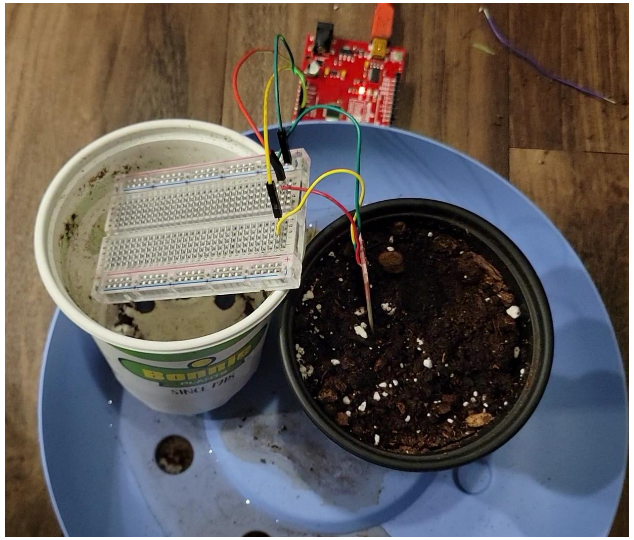
- Regulates 36 volts to 12 volts
- Regulates 12 volts to 5 volts

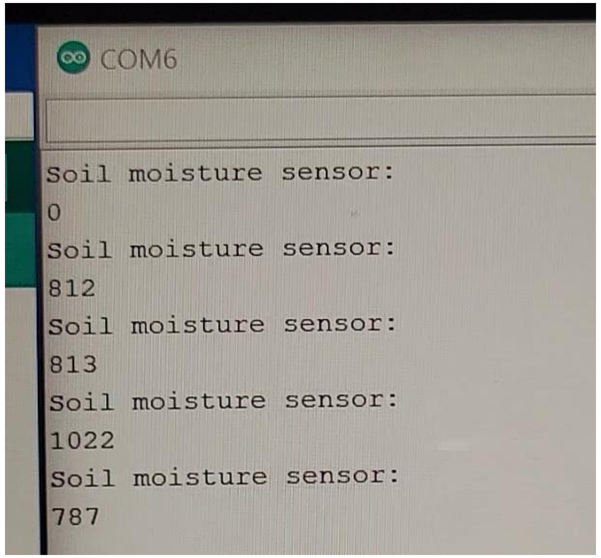


Testing

Watering Subsystem

- Water valve opens when 12V applied with the power supply
- Moisture sensor gathers data on soil moisture level
- Water valve follows the desired schedule
  - If moisture level is below 60%, the valve opens
  - If moisture level is above 60%, the cycle is skipped
- Water valve remains open for 15 minutes to provide sufficient water to plants
- Solenoid valve provides at least 4oz of water in under 30 minutes

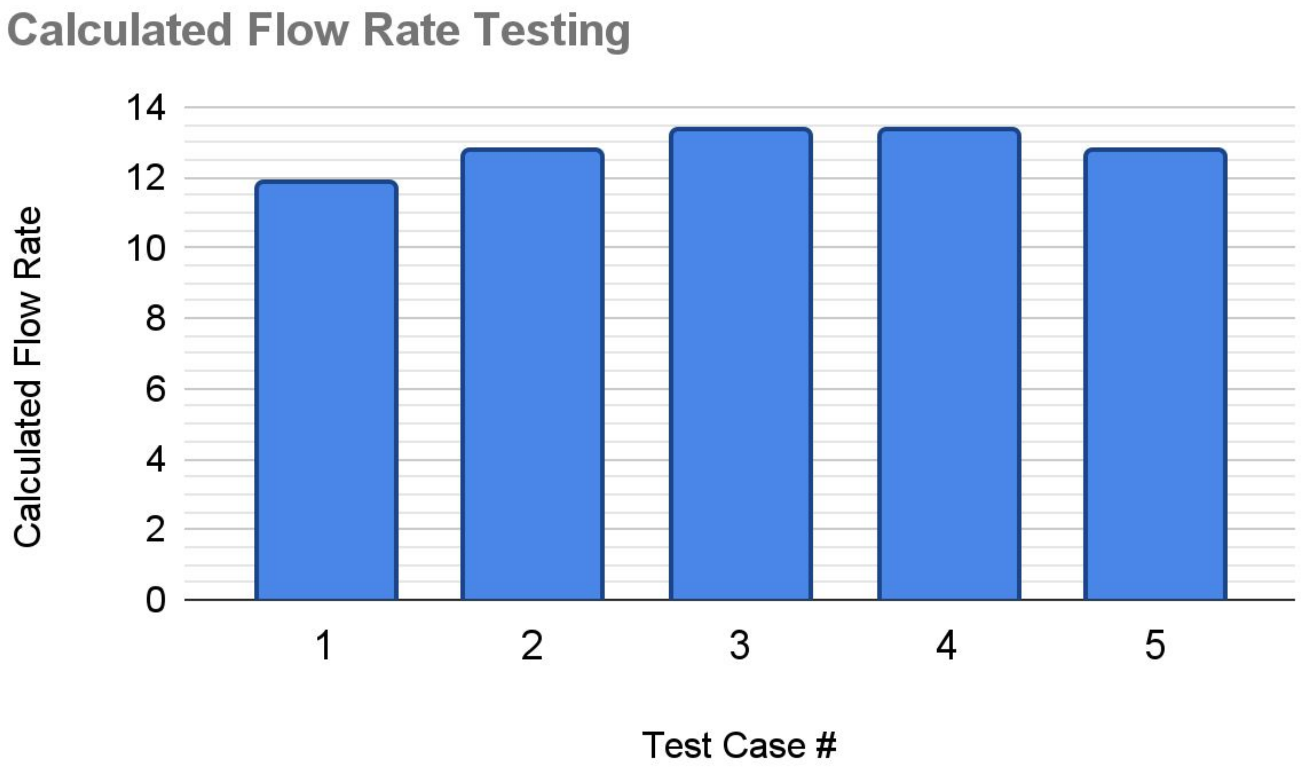




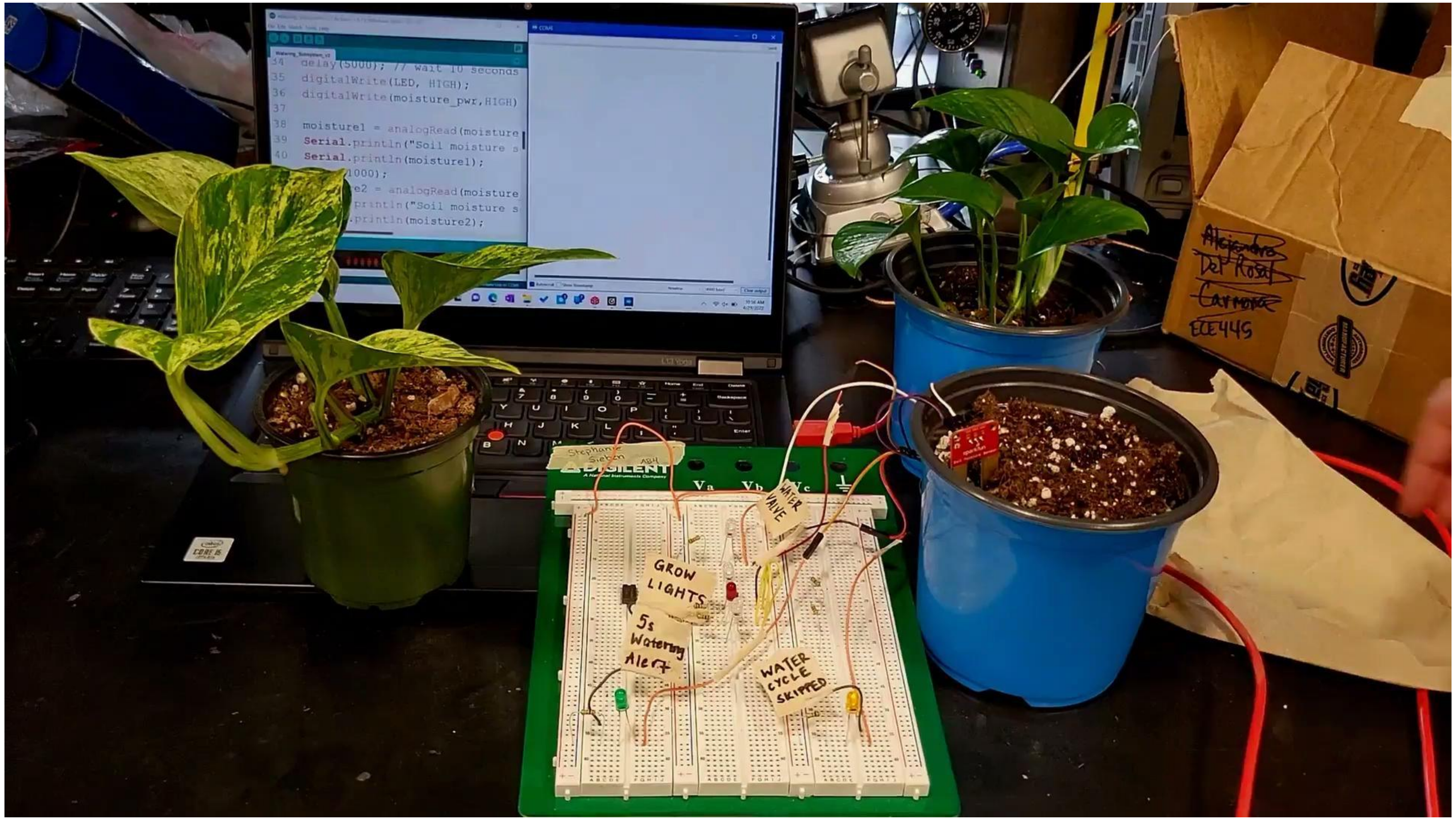
Water Flow-Rate Testing

Average Flow Rate: 12.78mL/min

Test #	Duration(min)	Water Collected(oz)	Calculated Flow Rate (oz/min)	Calculated Flow Rate (mL/min)
1	10	4	.4	11.83
2	15	6.5	.43	12.72
3	15	6.75	.45	13.31
4	15	6.75	.45	13.31
5	15	6.5	.43	12.72

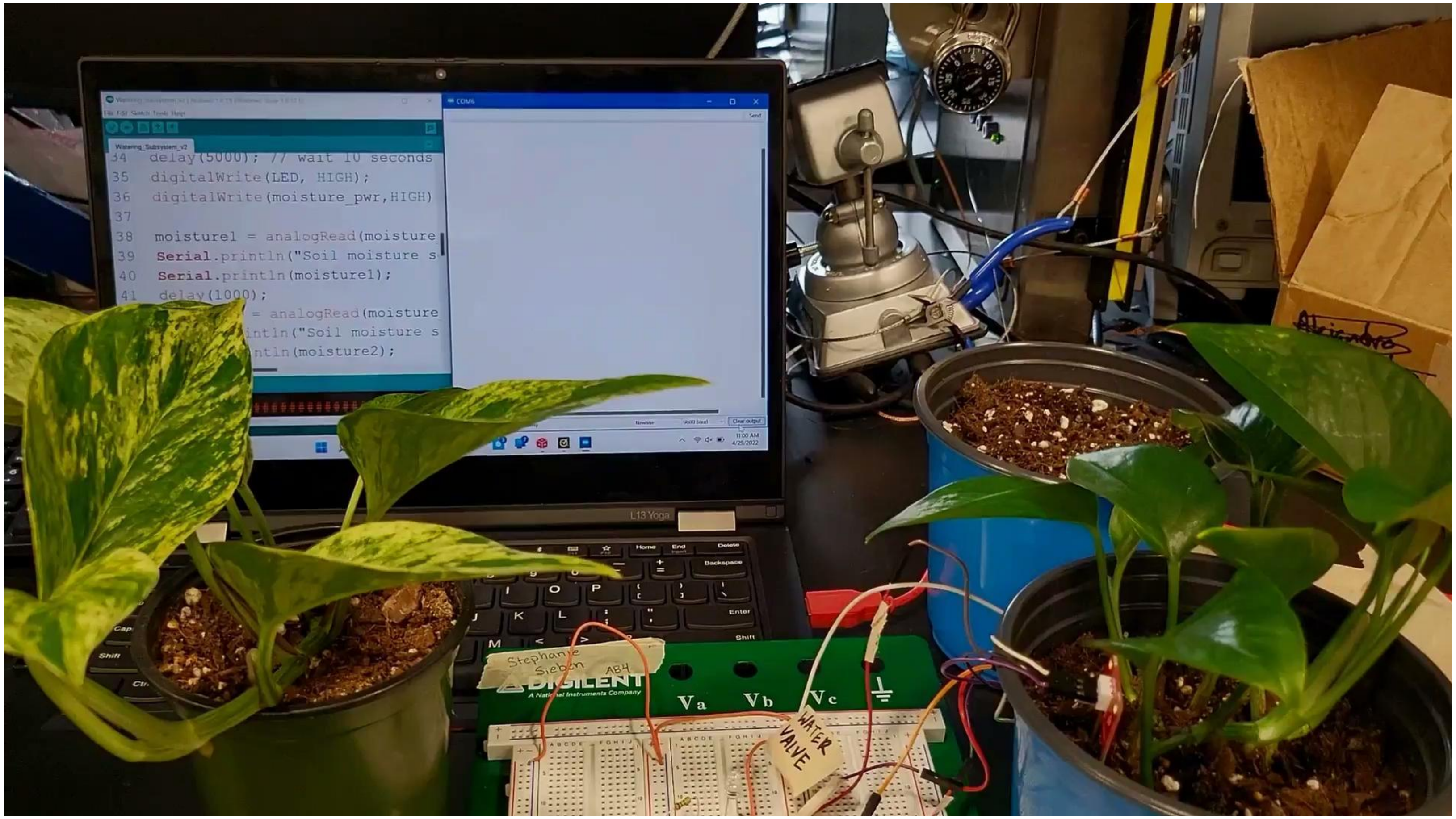


Functional Test of the subsystems



Video 1:Operation when plant needs to be watered

Functional Test of the subsystems



VIdeo 2:Operation when plant does not need watering

Final Product

Challenges

Challenges

LCD

- Arduino shut off with 5V power - potential short somewhere in the wiring
  - Changed current on potentiometer but only managed to get backlight turned on (3.3V power input)
- Further testing with a power supply overheated the LCD (5.4 V input, accidentally too high) and made it unusable
- Solution: We decided to go with a simpler and more universal solution of LED Indicators on a user interface

Microcontroller

- Programming the ATmega325PA proved to be our biggest roadblock
- Could not configure into Microchip Studio with the USBasp from the lab
- Prevented us from full system integration

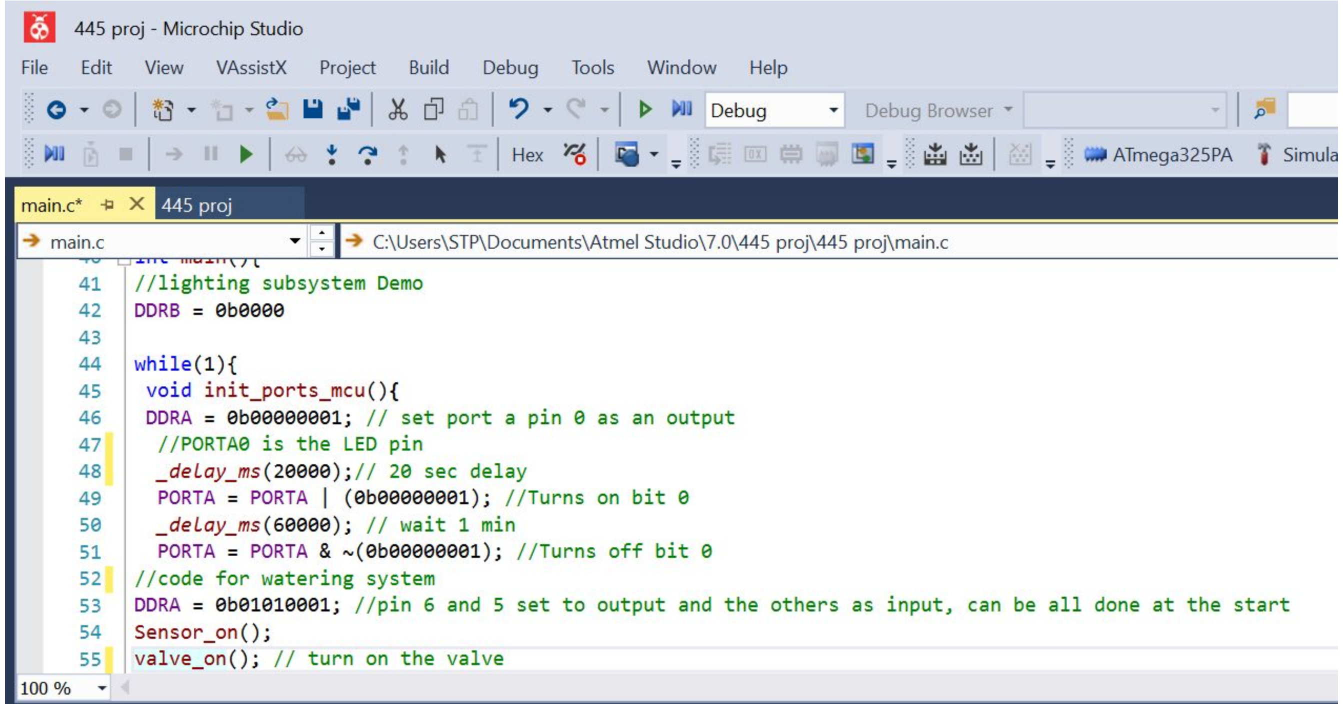
Conclusion

Changes & Improvements to Design:

- Add a lid and wheels to increase portability
- MCU that has only the number of I/O pins needed
- Use real-time-clock built into a microcontroller rather than programing timers

What we learned along the way

- Integrate subsystems
- Coding ports rather than pin numbers in Microchip Studio
- PCB and hardware design in KiCad



The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Appendix

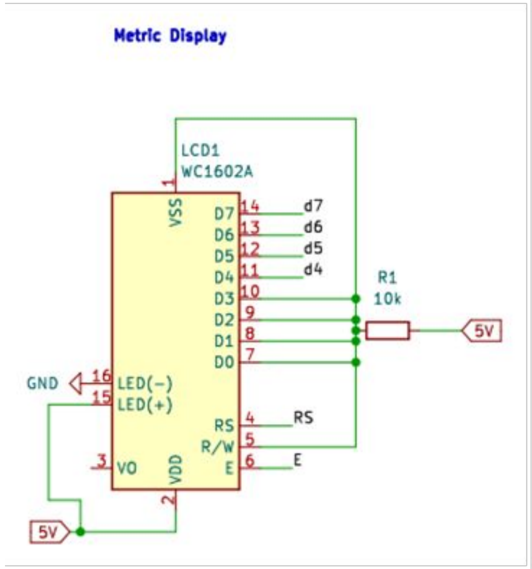
What's next?

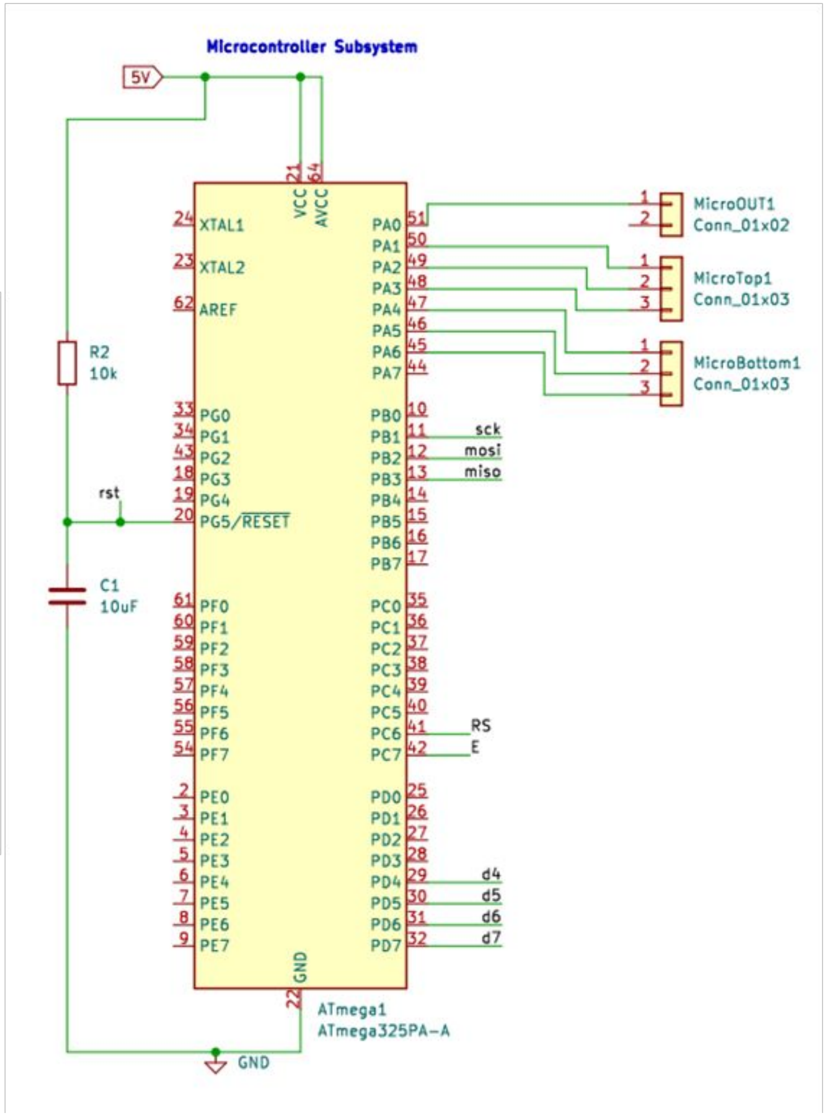
What we would do differently

- Conducted more upfront microcontroller research
- Worked more as a team to avoid specialization

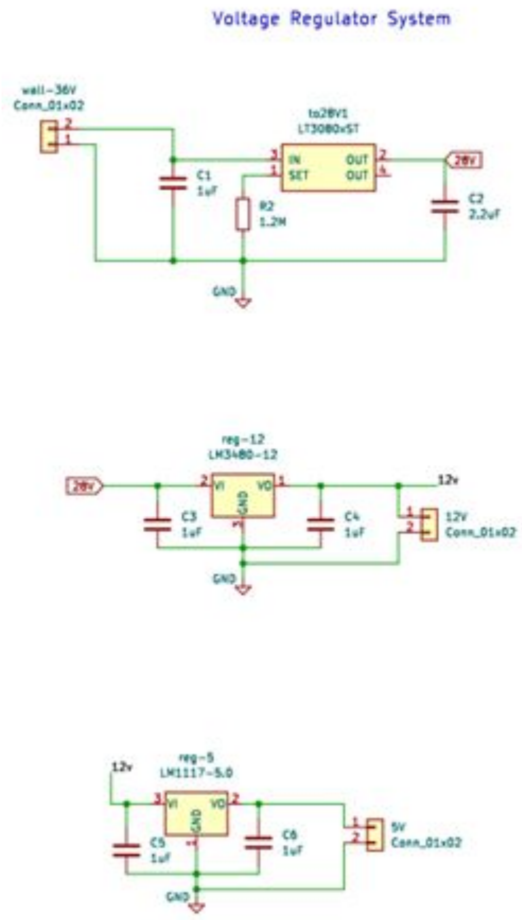
Hardware Design

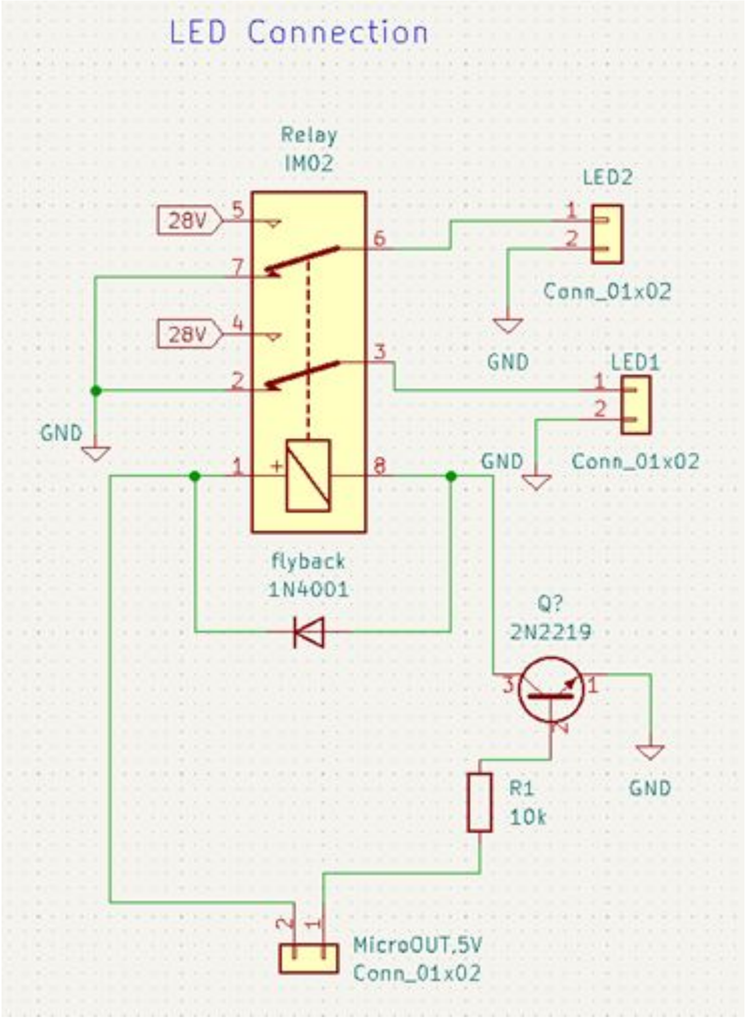
User Interface and Microcontroller Subsystem Schematics





Power and Lighting Subsystem Schematics





Watering Subsystem Schematics