

Compact Hydroponics System

GEEN 1400-060 First Year Projects Fall 2016

Supervised by: Prof. Michael Walker, Ph.D



Team: THE CLUB

Grant Novota, Brady Barkemeyer, Garrison Lee, Meshari Alqeryan, Luke

Lastoczy

UNIVERSITY OF COLORADO BOULDER

TABLE OF CONTENTS

CHAPTER	PAGE
Abstract.....	2
Introduction.....	3
Design Requirements, Alternatives, and Process.....	4
Testing and Analysis.....	17
Power and Cost Analysis.....	20
Conclusion.....	21
Budget.....	22
Timeline.....	23
APPENDIX.....	24
LIST OF REFERENCES.....	28

Abstract

The goal of our project was to build a hydroponics garden that was efficient enough, has a great aesthetic, and is compact enough for it to be integrated into the home. We also wanted our system to be fully-automated, requiring little manual maintenance to function. The reason for this was to make growing plants in a hydroponics system as effortless as possible so that it would appeal to a broader audience. Our target audience is people who want to grow plants inside of their home or apartments. The design consists of many arduino elements in order to achieve the goal of creating an automated system in addition to woodworking and laser cutting to make the project look aesthetically pleasing.

The most important aspects of our design were the electrical components because the system relies on them the most to function successfully. The electrical components consisted of an arduino microcontroller, pH sensor, peristaltic pump, servo motor, speaker, water level sensor, power relay, and water pump, with a separate grow light incorporated into the design if natural sunlight becomes unavailable to the system. All of these parts needed to be connected to the arduino and controlled with the code that we wrote, which told each component how we wanted it to behave. After many rounds of testing we were able to get each of the electronic components to function the way we wanted to and operate simultaneously with each other. We achieved this by testing each part individually before combining them into a single, complex system.

We were able to achieve all of the things that we set out for this project. If we were to recreate this system in the future, we would solder some of the wires and apply other methods of

permanently attaching components, so that this system would have a greater ability to function in the long-term without the need for maintenance.

Introduction

Hydroponics is a system of farming that eliminates the use of soil. Instead, the roots are held in a different medium, often clay pellets, which give the roots a place to hold onto to avoid tearing them in turbulent waters. The nutrients are added directly to the water, which is then given to the plants through various techniques, depending on the type of system. In our design, the nutrients become available to the plants when they are transported into the upper container by the water pump. Studies show that hydroponics systems use significantly less water, less space, grow faster, waste less water, and have less of an impact on the environment around them than traditional farming techniques. Unfortunately, hydroponics also usually uses more power and has a very high entry cost. Our project was not aimed at fixing these problems. Instead, we hoped that by building a system that can be brought into the home, we could foster care about the alternative farming method. Already, hydroponics gardens are able to reduce several problems integral to traditional, soil-based farming. The reason that it has not grown to be used more widely is a high entry cost and lack of demand by consumers. As more people learn what hydroponics farming is and begin to care about it, the more widely spread it will become. The following report will examine the development of our compact hydroponics system as well as our research and experimentation.

The Design Requirements, Alternatives, and Process

The Ebb and Flow System:

For this part of the project, we required a durable and compact way to store and move the reserve water, while also protecting and providing water to the plants. Plastic containers were used to allow ease of work and quick fixes to

the design of the system. The larger plastic container on the bottom is the main water reservoir is where the automated system adds nutrients and controls the pH level. The pump is also located in the bottom plastic container with tubing that is approximately twice as long as the distance from the pump's top to the drain on the top container to allow for easy access of removing the lid. The top basin is



Figure 1; the water reservoir and plant basin

connected to the reservoir by the ebb and flow drains and by the latches that are on the reservoir container (Figure 1). We cut 2.5 inch holes on the plastic lid that are between the top basin and the plastic water reservoir lid to connect the ebb and flow drains to them. The top water basin is what houses the plants and supplies them with water, so we cut holes into the lid of the top basin that were 4.5 inches in diameter to hold the plastic pots and allow the water to cover the plants' roots completely when the top container is

flooded (Figure 1). The bottom basin holds anywhere from 4.5 gallons (the minimum height to cover the pump) to about 8 gallons, which will then be pumped into the 2.5 gallon upper basin. The water pump that was used has a flow rate of 396 gallons per hour and with the use of the potential energy equation we have calculated that the maximum height of the water that is going to be pumped will be 8.85 meters, and that distance is more than what is needed for the project (Figure 2). The drain time to empty the upper basin is twice the time it takes to fill it up. The

$$PE = P \cdot V \cdot g \cdot h \rightarrow \text{Watts} \cdot \text{Pump}$$

$$h = \frac{W_{\text{Pump}}}{P \cdot V \cdot g} = \frac{36}{1000 \times 4.35 \times 10^{-4} \times 9.8} = 8.85 \text{ meters}$$

$$V = 396 \text{ gal/hr}$$

$$P_{\text{water}} = 1000 \text{ kg/m}^3$$

$$g_{\text{gravity}} = 9.8$$

$$V = \frac{396 \text{ gal}}{1 \text{ hr}} \times \frac{3.78 \text{ liters}}{1 \text{ gal}} \times \frac{1 \text{ hr}}{60 \text{ minutes}} = \frac{24.9 \text{ liters}}{1 \text{ minute}}$$

$$V = \frac{24.9 \text{ L}}{1 \text{ min}} \times \frac{1000 \text{ mL}}{1 \text{ L}} \times \frac{1 \text{ m}^3}{1000 \text{ L}} \times \left(\frac{1 \text{ m}}{100 \text{ cm}}\right)^3 \times \frac{1 \text{ min}}{60 \text{ sec}} = \frac{4.35 \times 10^{-4} \text{ m}^3}{1 \text{ sec}}$$

Figure 2; The calculations for the maximum height of the water that the pump could reach

water movement is very smooth which will not interrupt the roots, and the pump moves the water without reaching excessive noise levels.

There are many design alternatives that could have been implemented into this system that would have made the process of caring for the plants and the construction of the wood housing, among other, much easier. For example, the top basin could have been realigned to

allow for more area space in the front, which would have led to a more symmetrical look when viewed from certain angles. Also, the use of a smaller water reservoir could have allowed for a more aesthetically pleasing design, and a more compact system. If a smaller reservoir were used, then the use of wooden materials would have been much less, decreasing the cost of the system. Furthermore, a weaker pump could have saved money for our budget and would have been sufficient for the scale of our current project.

The type of hydroponic farming we used was an ebb and flow system which is basically a flood and drain system. The plants are submerged in water for set amounts of time and then after that time passes, the water is drained back into the bottom container, thus allowing the plants to aerate. However, there are other systems that could have been used, such as a deep water culture system (Figure 3) or a drip system (Figure 4) which might have been easier to work with and design for, but we agreed to use the ebb and flow system because of its moderate difficulty and integrating the many different automated systems was easier to apply to the ebb and flow system.



Figure 3; the deep water culture system, where the plants grow directly in the water reservoir



Figure 4; the drip system where the plant nutrients are dripped slowly into the top basin

The final design of the ebb and flow system hasn't changed from what we intentionally planned to do except for a few additions that made troubleshooting and automating the system much easier. For example, the extra space that was available on the top of the water reservoir, due to the trapezoidal side design, allowed for the wires of the pump and the pH sensor to fit inside with ease. Also, dropping the nutrients into the water reservoir was made very easy with the extra space in the back of the table. Due to the fact that we used plastic as our material for our water reservoir, it made manipulating and cutting it very easy.

Wooden Housing:

The first design that we tried resembled a lotus flower that contained the plants in the top basin in their own habitat. It would open from all sides and allow a lotus flower design with the appearance of blooming and flourishing (Figure 5). This design would have allowed for a much more aesthetically pleasing piece of furniture. However, the design, although beautiful, didn't allow for much arduino application or extra functionality that we had

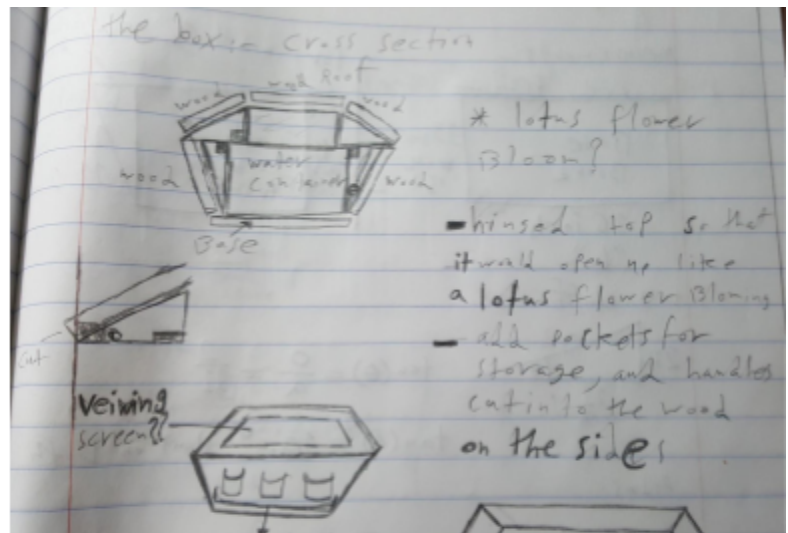
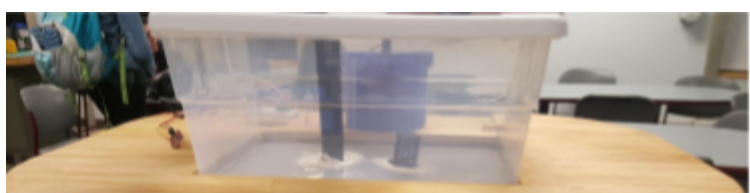


Figure 5 ; The lotus flower design idea and notes on it

intended to implement into our final design. We had to scrap the idea of the lotus flower and went with the idea of a table, which was a much easier to implement into a home, and it would

Compact Hydroponics



save costs on the wood materials that was needed.

For the scale of our project, the wooden housing needed to be aesthetically pleasing and easy to implement into a home, so we designed it to be similar to furniture in a normal house (Figure 6). The design of the wooden housing needed to have a dual function of being a complex system of farming while also not looking too out of place next to other furniture. Also, a strong infrastructure was needed to allow for some impact resistance and the protection of the inner basins. This way, when people are moving the portable system, it has the durability to withstand collisions as well as its own weight (several gallons of water).

The assembly process of the box involved two large pieces of wood that were 40 by 40 inch pieces and were an eighth of an inch thick. The first piece of wood was used to create the

base and the sides of the box. At first, the base of the box was supposed to be larger and rectangular in shape, however due to constraints with the amount of wood that was being used for the sides, the base had been shrunk to a square shape that is 16 inches on all sides. This modification allowed for the ease of maneuverability of the side pieces around the water



Figure 7 ; The inside of the box where the arduino was being stored on the sides and some of the wooden pieces on the base where hollowed out from the inside.

reservoir. Furthermore, the sides needed to be raised because the plastic water reservoir did not allow for a lot of room for the arduino to be stored inside the box. We used 4 pieces of wood that were about 2 inches in width. Two of the pieces were 16 inch squares in length, while the other

two were 14 inches, and those are the pieces that have been hollowed out by an inch to fit the water reservoir inside tightly while also raising the sides, creating more room for the electronics (Figure 7).

The sides of the wooden housing were made using the rest of the first piece of wood. The rectangular sides were 16 inches by 11 inches and they were attached to the larger trapezoidal shaped pieces with the dimensions of a 16 inch base, 11.5 inch sides, and 22 inch top side. We calculated the angle to cut the wooden pieces at using simple trigonometry (Figure 8). One of the larger side pieces of the wooden housing had been laser cut with a design, while the other piece was hollowed out to fit an acrylic piece with an additional design that was laser cut (Figure 9), (Figure 10).

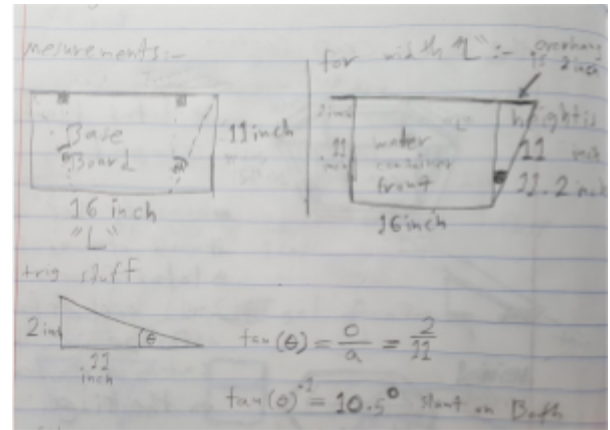


Figure 8 ; the calculations for the trapezoid's side angles using trigonometry



Figure 10 ; the drawing for the larger side piece of the wooden housing



Figure 9 ; The design that was laser cut onto the acrylic piece on one of the trapezoid shaped sides of the box

The other large piece of wood was used to construct the table top. However, due to the irregular placement of the top plant basin, the dimensions that were used when cutting the

tabletop were very hard to work with. For example, the backside of the tabletop is much larger than the front side, because that was where we planned to attach the nutrient dispenser. In order to center the tabletop, while still allowing for the nutrients dispenser to be attached safely, we agreed to have the tabletop extend out 3 inches extra on each side, while having the front side of the piece be 1 inch smaller than the backside. The final tabletop dimensions from the sides of the

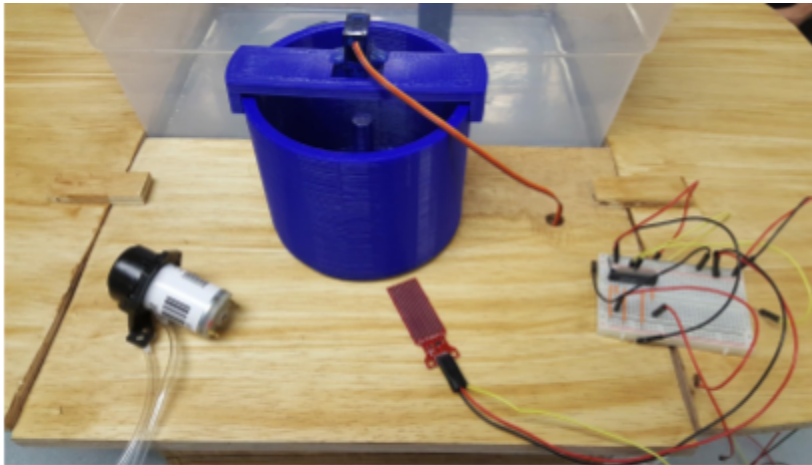


Figure 11 ; The back panel that holds the nutrients dispenser

top basin are 19.5 inches, and 15 inches from the front to back, which was cut using a power jigsaw. Furthermore, in order for the nutrient dispenser to be secured safely on the tabletop, wooden holders were attached to the back of the

tabletop piece. However that removed the extra functionality of having a sliding tabletop for easy access to the water reservoir. For the back panel that the nutrient dispenser is secured to, a small hole was drilled to allow for the arduino wires to attach to the servo (Figure 11).

After the main components were cut, we glued and clamped all the sides. Finally, in the last couple of weeks we stained all of the housing and sanded it, making it look the best we could.

Nutrient Dispenser:

In this hydroponics system, the plants' roots are contained within plastic pots that are filled with clay pebbles and therefore, the nutrients that are normally absorbed from soil are unavailable to these plants. To solve this problem, we created a nutrient dispenser to provide the plants with the nutrients needed for them to survive (Figure 12).

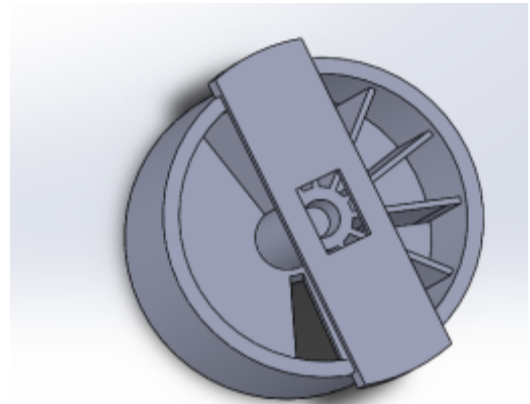


Figure 12 ; Nutrient Dispenser Assembly

The nutrient dispenser was designed in Solidworks and constructed with 3D printers. The design for this dispenser consisted of four components: a cylinder base, rotating dividers, a servo holder, and a servo motor. The cylinder base holds all of the dispenser components; it has a rod in the middle to hold the dividers in place and allow them to rotate (Figure 13). The rotating dividers create 6 slots for nutrients to be placed in (Figure 14). This means that the plants may be fed up to 6 times automatically before the dispenser needs to be manually refilled. Each slot should have enough nutrients to feed the plant for eight days until the next one is used. We

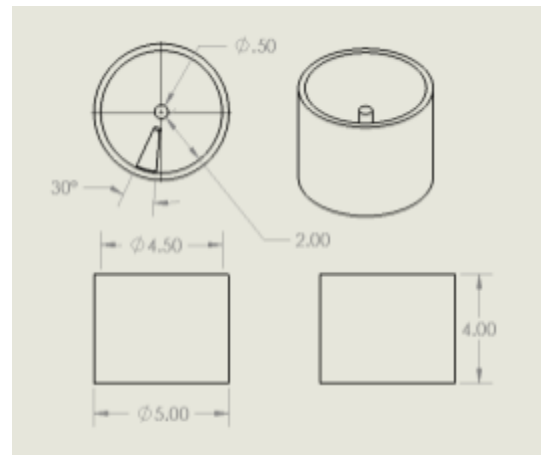
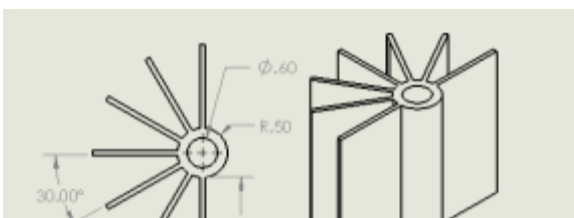


Figure 13 ; Cylinder Base



system by being dropped into the lower water reservoir. This happens when the dividers rotate 30 degrees so that the slot containing the nutrients slides over the hole in the base of the dispenser, allowing the nutrients to fall into the water below (Figure 15). The dividers are rotated with an arduino servo motor. This servo connects to the top of the dividers and is held in place by the servo holder component which attaches to the sides of the dispenser base and has a rectangular hole in the middle where the servo motor was placed and secured with hot glue (Figure 16).



Figure 15 ; Plant nutrients

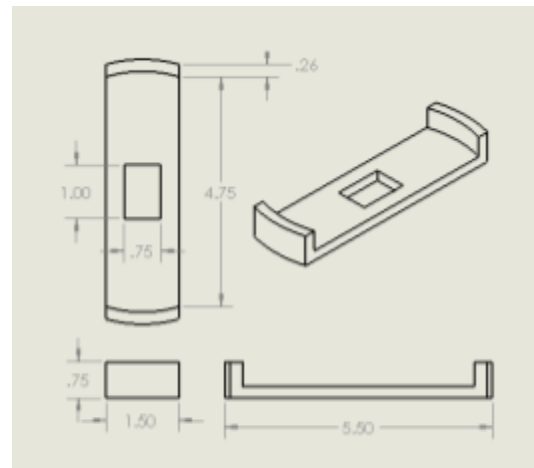


Figure 16 ; Servo Holder

pH Sensor and Peristaltic Pump:

The ideal water pH for most plants is 5.5 to 6.5 which is slightly acidic. As plants grow the pH of the water will rise over time. pH has a substantial influence on the solubility and absorbability of several nutrients in the water that the plants need to absorb. As you can see in

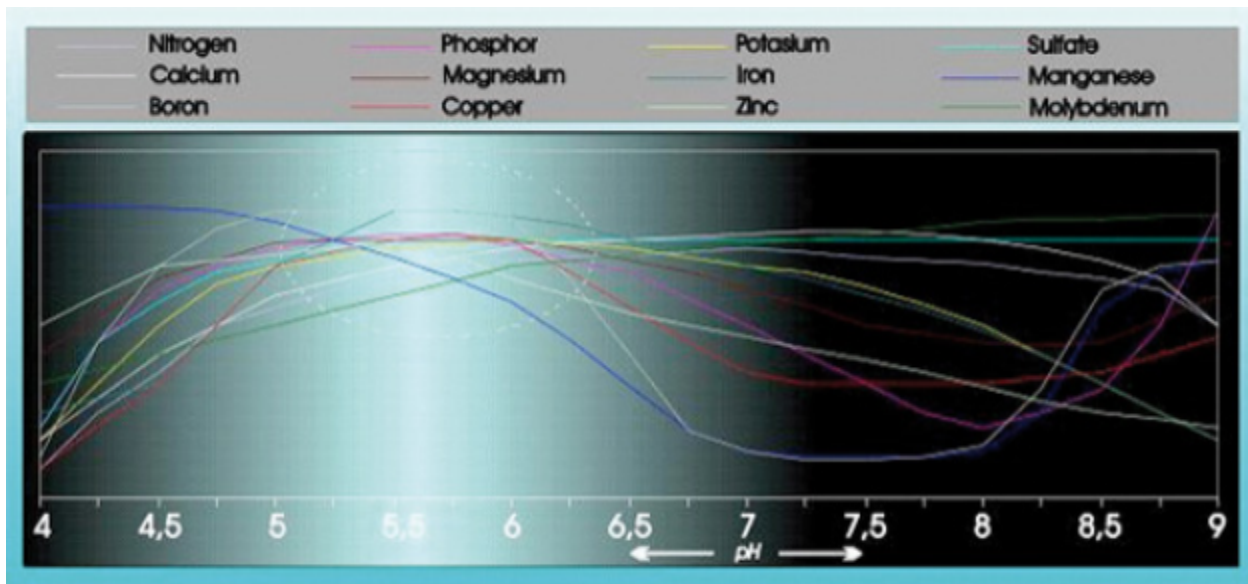


Figure 17 ; the graph of the ideal pH that plants need

the graph below, the ideal range where the majority of the nutrients have high relative absorbability/solubility properties is in the just slightly acidic range (Figure 17).

Various plants thrive in different ranges. Regardless of the ideal range for the given plant, regulating the pH is important as allowing it to rise beyond this range limits the plant's ability to easily extract the necessary nutrients.

To do this, we purchased a pH sensor probe and integrated it, and a separately purchased peristaltic pump, with the arduino. The pH sensor works by measuring the potential difference in the surrounding fluid. This is proportional to the amount of hydrogen ions in the water.

Hydrogen ions are also what control the pH. The code for the pH sensor essentially read the analog input value given by the pH sensor based on the potential difference, reordered the values from least to greatest, excluded the outliers on either side, then averaged the values, which were converted into the pH scale's number of 0-14, and recorded that value as the pH. To make sure

that the probe was indeed accurate, we tested it against litmus paper. A further discussion of this is in the **Testing and Analysis** section below.

For ease of use and in order to make the system more user friendly, a variable at the top is named idealpH. The user can simply set this equal to the ideal pH for their plant and the system will maintain it at that value. If the pH value returned by the probe rises above the idealpH value, the peristaltic pump will begin dropping eight drops from a separate reservoir containing an acidic solution every 30 seconds. The reason for the delay is to avoid lowering the pH too far below the idealpH since there is no way to raise the pH afterwards. To further prevent this, the tube feeding the acidic solution into the main reservoir was fixed near the location of the pH sensor to avoid an issue we were having where it would take minutes for the acid to disperse through the large lower basin, meaning that far too much acid was released before the probe registered the decrease.

Water Level Sensor:

Although there is a lid on both reservoirs that hold water, some evaporation can still occur through the clay pellets holding the plants. In addition, water will be absorbed through the roots of the plants. This water will then evaporate through the pores on the leaves during photosynthesis. As such, some water loss will occur. To avoid plant death resulting from this, a water level sensor was included (Figure 18). The sensor has several parallel metallic

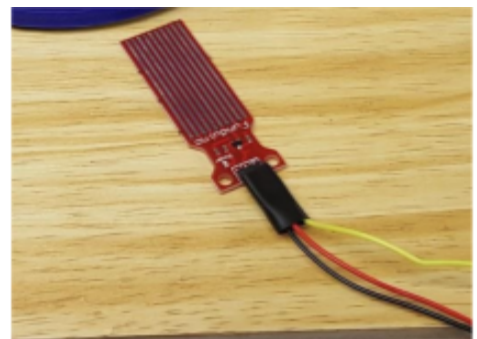


Figure 18 ; water level sensor

rods that do not touch. And since they do not touch, they do not complete the circuit and thus return a value of 0 to the arduino when dry. However, when wet, the water itself completes the circuit so a non-zero value is returned. If the value being returned is ever zero for greater than half an hour, an alarm will begin to buzz once an hour for one second each time. This is simply to notify the person they need to add more water without being too annoying.

The reason that the sensor must be dry for a full half hour prior to beeping is because it is fixed in the lower basin where the water will periodically drain away from before filling. The whole emptying and refilling cycle should never take more than half an hour though, so if it is dry for longer that means the proper level of water did not return to the lower basin.

Relay Control for Main Pump:

The larger of the two pumps (the former being the peristaltic pump and the latter being the ‘main’ pump) drew too much power to directly draw from the arduino. Instead, a relay control was purchased. The relay control is essentially an extension cord that can be turned on and off via a simple digital output signal from the arduino (Figure 19).

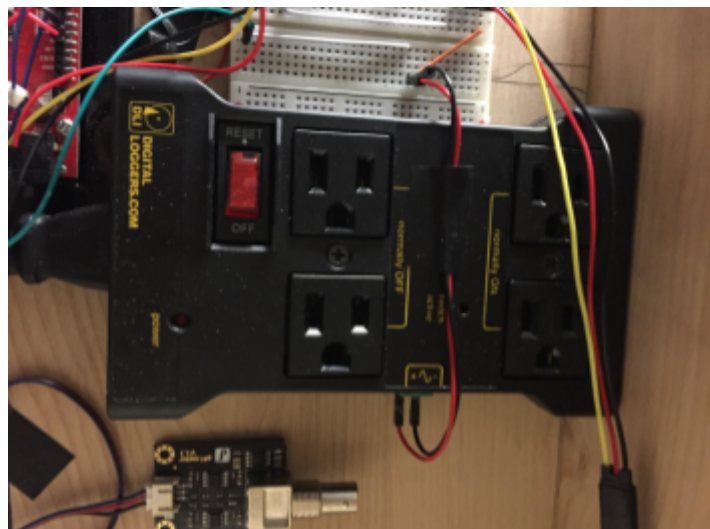


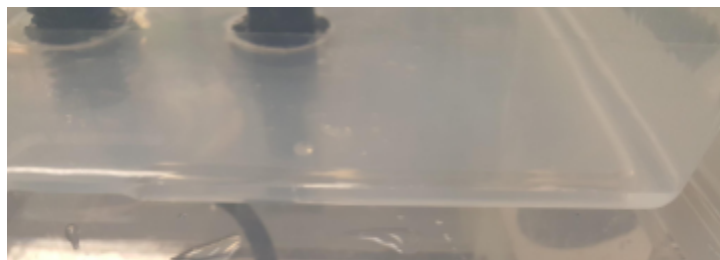
Figure 19 ; Relay

Using the milliseconds function that is inherent to arduinos, the relay control was programmed so that, from when it was plugged in, it would wait twelve hours, come on for a flooding cycle of twenty five minutes, drain, then wait for six hours, flood, wait for another six hours, flood, and then repeat the process. This schedule of flooding and draining was necessary to give the plant access to water without suffocating the plant with too much water.

Testing and analysis

Testing:

Water Proofing: When assembling the ebb and flow system and the pump access to the top basin, we had a lot of problems regarding water leaks. That leaking would have caused a lot of wear on the wooden housing and might have caused problems to the electrical components. In order to alleviate this problem, some leak testing needed to be done. We had to test the connections of the system on three separate occasions using three different methods. The first time we tested it was when we first assembled the system, and at that point the system wasn't leak proof. We applied some caulk on the top basin to minimize leaking, however the caulk was not as resistant to water as we



wanted it to be and that layer wore away (Figure 20). The second round of testing went much smoother with the application of a thicker layer of caulk while allowing it to dry for a longer amount of time. We also strengthened the connectors in between the top basin and the water reservoir. Testing the second round resulted in much more promising results. The leaking was extremely minimized and the caulk withstood a long amount of time underwater, but the system was still leaking. The final round of testing included the use of a temperature activated, clear adhesive to fix the leaking instead of the caulk. The clear adhesive worked much better than the caulk at preventing any water from leaking.

Sensors: To ensure that our pH sensor was indeed accurate, we tested it against litmus paper. We were limited by the fact that litmus paper uses a qualitative system to measure --

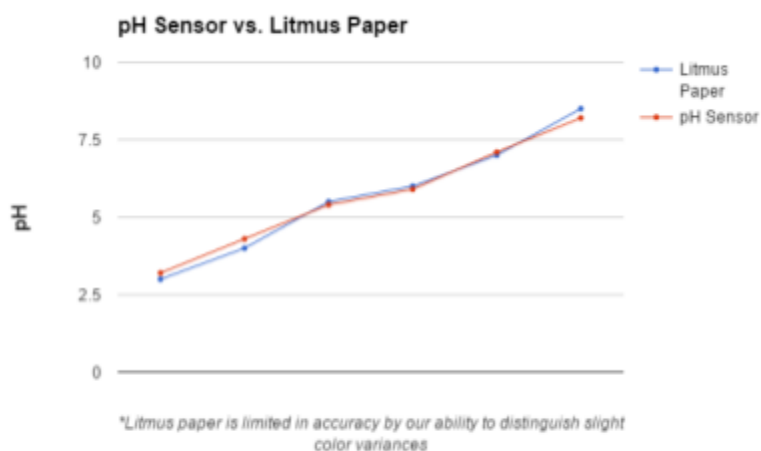


Figure 21; pH sensor accuracy testing graph

color.

The testing did, however, reassure us that the probe was at least as accurate as litmus paper (Figure 21). The water level sensor was far easier to test. Using an LED instead of the alarm to

avoid being obnoxious, we simply observed the light come on whenever the sensor was dry and go out whenever the sensor was wet. Interestingly, a finger is conductive enough when placed across the parallel metal rods that a low value is returned. This likely means that the water level sensor will have to fully dry before it begins returning a value of zero; dampness will likely return a low value.

Nutrient Dispenser: The nutrient dispenser initially had too much friction for the servo to overcome meaning that it simply wasn't spinning. To combat this, we tried several methods. We first tried sanding the inner sides and bottom. This yielded moderate success though the system was still sticking part of the time. We then tried elevating the inner blades so as to not rest on the bottom. Again, this helped but was not perfect as the blades would sometimes slip downwards and rest on the bottom anyways. Finally, we laser cut an acrylic base that reduced the friction enough that, if placed lightly, the inner blades would spin unimpeded (Figure 22).

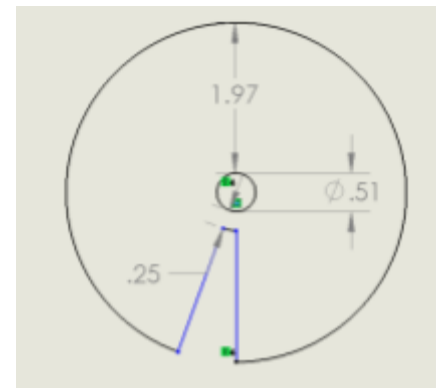


Figure 22 ; Acrylic Base

Relay Control: Due the extreme compactness of the design, the relay control could only be fit in one orientation meaning the pump is plugged into “normally on”. This means that if no input is being received from the arduino, the outlet is providing power. This requires the arduino to provide power almost continuously thus draining the battery faster. Second, if the arduino dies or becomes disconnected for any reason, the plants will flood. They will drown more quickly than they would dry out which means ideally the pump would be attached to “normally off”. A

second revision would certainly take this into consideration. Because the arduino can count but not tell time, for the cycle of feeding to be thrice daily, you must start the system in the evening.

Power/Cost Analysis

Although hydroponics is an efficient and sustainable alternative to conventional agriculture, cost is the one area where hydroponics has a disadvantage to traditional farming. Our hydroponics system requires a light, pump, containers, pH solutions, a peristaltic pump, sensors, nutrients, and a nutrient dispenser. These requirements for a hydroponics system can add up to be a little more expensive than growing plants in soil. Although the entry cost for hydroponics is high, once the system is set up and running the cost is very minimal. Plants only use a small percentage of water and this type of system includes a recirculating reservoir that allows it to use up to 90% less water than regular farming. This system also limits evaporation and prevents water run-off. The average cost of water in the United States is \$1.5 for 1,000 gallons. Using a five gallon average for our system, and even a large estimated evaporation rate of .5gal/month due to the limited air exposure, the total cost of water over a year would be \$0.0165. Given the average water cost of a plant is a cup per day, over the course of a year, it will cost \$0.03375 to water a plant in a pot every day. This means our system uses 49% less water than if you watered a plant in your home regularly. For this small application, it doesn't seem like there's too much of a difference, because the cost is small. However, if you upscale it to the size of feeding a small

community, it could save thousands of dollars in water costs. Granted that the initial cost of water for our system is \$0.0075 compared to \$0.00009375 of watering a plant in a pot, but down the road it is very efficient. This 8,000% higher entry cost seems amazingly high, but over the course of not even half of a year, the cost is equalized.

Conclusion

Our goal for this project was to create a compact hydroponics system for people that want to grow plants and herbs in their own home. Our hydroponics system is a sustainable alternative to conventional agriculture that uses less water, less space, and requires less maintenance than traditional soil-based farming. In addition, hydroponic farming yields more growth in a shorter amount of time and has less of an impact on the environment than other farming techniques. We created an automated system that consists of many arduino elements along with aspects of woodworking, laser cutting and 3D printing (Figure 23). This complex system consisted of many different components including a pH sensor, a peristaltic pump,



Figure 23 ; The fully functioning hydroponics system

water pump, water level sensor, a relay control, and a grow light. Throughout the design process we encountered a few problems that we had to troubleshoot such as water proofing the system and fixing the nutrient dispenser. After conducting multiple rounds of testing we were finally

able to successfully complete our goal of creating a compact hydroponics system. Upon testing we found that our system uses 49% less water than a comparable soil-based farming system. All of our goals were met for this project. For the future, creating a design to minimize the cost as much as possible would greatly benefit this system of hydroponic farming and make it more appealing to the public for widespread use.

Budget

GROUP MEMBER	ITEM	COST
Brady	3D printing material	\$5
	Containers	\$15
	Wood	\$50
	PH Up	\$10
	Plants	\$10
	Water Sensor	\$5
Luke	Nutrients	\$13
	Ebb and Flow drains	\$10
	Plastic Pots	\$14
	Water Pump	\$35
Meshari	PH Sensor	\$57
	PH Down	\$12
	Peristaltic Pump Tubing	\$3
Garrison	Peristaltic Pump	\$12
	Clay Pebbles	\$16
	Light	\$33

	Black Tubing	\$16
	Acrylic	\$15
Grant	Poster	\$53
	Power Relay	\$27
	TA-7291P Semiconductor	\$14
Total		\$425

Timeline

Week/ progress	week 1	week 2	week 3	week 4	week 5	week 6	week 7	week 8	week 9
Brainstorming and story boards									
Material research and budgeting									
Acquire the materials									
Build the main water reservoir/ ebb and flow system									
Start work on housing and 3D printing									
Start on arduino and continue work on other components									
Finish housing, 3D printing, and laser cutting									
Finish arduino coding, and table top piece									

Putting it all together, testing and expo prep									
--	--	--	--	--	--	--	--	--	--

Appendix

Arduino Code:

```

//for ph - (6-7 range) **
//pPump (how strong should the acid be)
//dispenser (once a week real life, 30 seconds) **
//waterLevel (how long does it take to fill AND drain)
//mainPump (how long it takes to fill) add the code for this *

//pHSensor Stuff
#define pHSensorPin 0 //pHSensor input A0
unsigned long int avgValue; //Store the average value of the sensor feedback
float b;
int buf[10],temp;
unsigned long phTimer = 0;
int sensorPin = 12; // do you need this? pHSensor to 12

//PPump Stuff
unsigned long timerPPump = 0;
int pPumpPin = 5; //pPump to 5
boolean pPumpOn = false;
float pH = 0.0;
int idealPH = 6.6;

//ServoDispenser Stuff
#include <VarSpeedServo.h>
VarSpeedServo servol;
int servolPos = -30;
unsigned long servoTime = 0;
long nutesTime = 30000;
int servoPin = 2; //servo to 2

//WaterLevel Stuff
unsigned long wTimer = 0;
unsigned long buzzTimer = 0;
int WaterLevel = 0;
int buzzPin = 9; //buzzer to 9
boolean fill = false;
boolean buzzOn = false;
#define wLevelSensor A5 //water level sensor input A5

```

```

//Relay for main pump control
int relayPin = 13; //relay to 13
unsigned long mPumpTimer = 0;
unsigned long sleepTimer = 90000;//43200000;
unsigned long dayTimer = 60000;//21600000;
unsigned long floodTime = 20000;//1500000;
unsigned long floodTimer = 0;
boolean mPumpOn = false;
boolean night = false;
int fed = 0;
int changeInitialPump = 0;

void setup(){
  Serial.begin(9600);
  //pHSensor Stuff
  pinMode(sensorPin,OUTPUT); //do you need this??
  Serial.println("Ready");
  //PPump Stuff
  pinMode(pPumpPin, OUTPUT);
  analogWrite(pPumpPin, 0);
  //ServoDispenser Stuff
  servo1.attach(servoPin); //DISPENSER TO PIN 2
  //WaterLevel Stuff
  pinMode(buzzPin,OUTPUT);
  //Relay for main pump control
  pinMode(relayPin,OUTPUT);
}

void loop(){
  //pHSensor Stuff Sensor goes to A0
  for(int i=0;i<10;i++)
  {
    //Get 10 sample value from the sensor for smooth the value
    buf[i]=analogRead(pHSensorPin);
  }
  for(int i=0;i<9;i++) //sort the analog from small to large
  {
    for(int j=i+1;j<10;j++)
    {
      if(buf[i]>buf[j])
      {
        temp=buf[i];
        buf[i]=buf[j];
        buf[j]=temp;
      }
    }
  }
  avgValue=0;
  for(int i=2;i<8;i++)//take the average value of 6 center sample
  {
    avgValue+=buf[i];
  }
  float phValue=(float)avgValue*5.0/1024/6; //convert the analog into millivolt
  phValue=3.5*phValue; //convert the millivolt into pH value
  pH=phValue;
  if (millis()-phTimer>=1000)
  {
    Serial.print(" pH:");
    Serial.print(phValue,2);
    Serial.println(" ");
    digitalWrite(12, HIGH);
    digitalWrite(12, LOW);
    phTimer = millis();
  }
}

```

```

//PPump Stuff
if (pH>=idealPH)
{
  if ((millis()-timerPPump>=2000) && (pPumpOn == false))
  {
    analogWrite(pPumpPin, 255);
    pPumpOn = true;
    timerPPump=millis();
  }
  if ((millis()-timerPPump>=20000) && (pPumpOn ==true))
  {
    analogWrite(pPumpPin,0);
    pPumpOn = false;
    timerPPump=millis();
  }
}
else
{
  analogWrite(pPumpPin, 0);
}
}

//ServoDispenser Stuff
unsigned long currentMillis = millis();
if((currentMillis-servoTime >= nutesTime))
{
  servolPos += 30;
  servoTime=currentMillis;
  servol.slowmove(servolPos,50);
}
if((servolPos == 180)&&(currentMillis-servoTime>=5000))
{
  servolPos = -30;
}
}

//WaterLevel Stuff
if (millis()-wTimer>=1000)
{
  wTimer = millis();
  WaterLevel = analogRead(wLevelSensor);
  if (WaterLevel <= 300)
  {
    fill = true;
  }
  else if (WaterLevel > 300)
  {
    fill = false;
  }
}
if (fill == true)
{
  if ((millis()-buzzTimer>=10000) && (buzzOn==false))
  {
    digitalWrite(buzzPin,HIGH);
    buzzTimer = millis();
    buzzOn = true;
  }
  else if ((millis()-buzzTimer>=500) && (buzzOn==true))
  {
    digitalWrite(buzzPin,LOW);
    buzzTimer = millis();
    buzzOn = false;
  }
}
else if (fill == false)
{
  digitalWrite(buzzPin, LOW);
}
}

```

Yellow goes to 5/6

Servo goes to 2

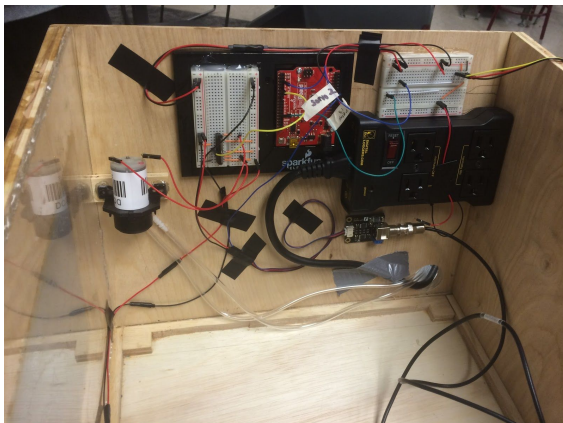
Sensor goes to A5. Buzz goes to 9

```

//Relay for main pump control
if (changeInitialPump == 0)
{
    changeInitialPump = 1;
    digitalWrite(relayPin,HIGH);
}
if ((millis()-mPumpTimer>=sleepTimer)&&(mPumpOn==false)&&(night == true))
{
    mPumpTimer=millis();
    mPumpOn=true;
    digitalWrite(relayPin,LOW);
    floodTimer = millis();
    night = false;
}
if ((millis()-mPumpTimer>=dayTimer)&&(mPumpOn==false)&&(night == false))
{
    mPumpTimer=millis();
    mPumpOn=true;
    digitalWrite(relayPin,LOW);
    floodTimer=millis();
    fed = fed+1;
    if (fed==2)
    {
        fed = 0;
        night = true;
    }
}
if ((millis()-floodTimer>=floodTime)&&(mPumpOn==true))
{
    mPumpOn=false;
    digitalWrite(relayPin,HIGH);
}
}

```

Other pictures:



References

Walker, Michael, Dr. - General Project Supervision

Montoya, Elijah - Teaching Assistant

Kahlia Hogg - Teaching Assistant

Nebbs - Arduino Assistance