Agricultural Modulation

A Major Qualifying Project Report

Submitted to the Faculty of

Worcester Polytechnic Institute

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

in Electrical and Computer Engineering

By

Molly Clem & Wenting Li

05/06/2021

Approved by:

Professor Stephen J. Bitar, Advisor

Electrical and Computer Engineering, WPI

Abstract

Nitrogen fertilizer overuse in fields causes extreme damage to local river ecosystems and poses an environmental danger worldwide[1,2,3]. Modulating and testing the exact amounts of fertilizer needed to grow crops can benefit these local ecosystems, optimize plant growth, and increase the overall knowledge about fertilizer use in the industry. Utilizing servos, valves, seven different types of sensors, a microcontroller, and a pump, this scale model project can be implemented into farms to home in on precise values of nitrogen, phosphorus, and potassium needed for plant growth, as well as takes data on the conditions that plants thrive in. Wiring the whole system together and connecting all the tubing for the water dispersion system, as well as programming it, correctly ensured its functionality. We were able to take preliminary data, to prove its operation. As a result, this device can benefit our current agricultural practices, improve what is known about agriculture, and have a positive environmental impact.

[1] Kim, Seungdo, and Bruce E. Dale. "Effects of Nitrogen Fertilizer Application on Greenhouse Gas Emissions and Economics of Corn Production." *Environmental Science & Technology*, vol. 42, no. 16, 2008, pp. 6028–6033., doi:10.1021/es800630d.

[2] Muratoglu, Abdullah. "Grey Water Footprint of Agricultural Production: An Assessment Based on Nitrogen Surplus and HIGH-RESOLUTION LEACHING Runoff Fractions in Turkey." *Science of The Total Environment*, vol. 742, 2020, p. 140553., doi:10.1016/j.scitotenv.2020.140553.

[3] Brueck, Holger, and Joachim Lammel. "Impact of Fertilizer N Application on the Grey Water Footprint of Winter Wheat in a NW-EUROPEAN Temperate Climate." *Water*, vol. 8, no. 8, 2016, p. 356., doi:10.3390/w8080356.

Acknowledgements

We would like to thank our project advisor, Professor Stephen Bitar, for his extensive guidance and insight through the duration of this project. Without his continual mentorship this project would not have been possible.

Executive Summary

General Design

Our purpose of building this MQP is to distribute chemical fertilizer to plants more reasonably, to maximize the yield of plants and reduce the environmental pollution caused by the composition of fertilizer. At the same time, we also need to pay attention to the impact of environmental factors on plants.

The goal of this project is to build an irrigation system that can control and disperse fertilizer. At the same time, this system needs to analyze the surrounding environment and track the condition of plants. This system can be roughly divided into three main parts. The first part is mainly responsible for the storage and mixing of fertilizer and water. The solution of different fertilizers and irrigation water are stored in four different tanks, and the bottom of the tank is equipped with a valve and a load sensor. These liquids will flow into another tank for mixing, and there is a pump under the mixing tank to send the solution to the soil. The second part is mainly used to collect the environmental conditions near the plants. Considering that light, soil humidity, carbon dioxide concentration and other factors will also affect plant growth, we place sensors next to plants to collect data of the surrounding environment. We also use a TDS sensor and a pH Sensor in the water tank to record the TDS and pH values of the irrigation water. The main component of the third part is a microprocessor. The microprocessor will connect all sensors for data collection and control the switching of valves and pumps. It will transmit the collected data to the computer by Wi-Fi, so that users can more easily monitor the situation of plants. Our main objectives are to demonstrate our ability to create a machine that monitors the environmental factors of the plants, as well as disperse fertilizers.

The scope of this project is to create a model of a system that would be implemented into a field for regular agricultural use. This entails building miniaturized systems that achieve the same function as a larger model, and then being able to scale it accordingly. In the future we could possibly do some more research and create a specialized sensor for monitoring soil moisture, but because of the scale of the project, we do not have time to implement one of these features yet. Our project also originally was designed to implement a machine learning aspect, in which a camera would monitor the field, and take data on what makes a healthy plant, then compare the model of a healthy plant to a model of other plants growing, helping to evaluate which variables impact plant growth the most. Due to multiple CS members moving to other projects, we scaled down the project to just focus on the fertilizer distribution, sensing capabilities, and data collection of the machine.

Some of the constraints of this project include the amount of space we have, our budget, the hardware and software limitations, thinking for scalability, time, and the amount of manpower the project has.

This project could easily be implemented into a large field, however WPI does not have a lot of large agricultural fields around. This entails that we must work within the size they allot us. Our MQP project was given two tables in Atwater Kent Laboratories that were three by six feet long. This means that given the smaller scale model we were able to set up testing conditions of a small gardening bed. This was constructed using plywood, a pond liner/tarp, and common potting soil. This means that our project does not exactly fit all the conditions it would be outside. Additionally, we had to use grow-lights in order to support our crops' necessity for sunlight. This again does not perfectly simulate a field. However, we made sure to pay close attention to the spectrum of light the plants need and buy those lights accordingly.

We also had a limited budget. Capacitive volumetric moisture sensors can cost up to a hundred dollars a meter. Our budget was given at two hundred and fifty dollars per person on the team, bringing us to a total of five hundred dollars. Based on this budget we realized that we could not buy an industrial capacitive moisture sensor. Similarly, a high grade laboratory pH Sensor costs around one hundred dollars as well. Another expensive factor. Due to these we opted to use two different moisture sensors, as well as a calibrated commercial grade pH Sensor in order to get as close as possible to an accurate reading. Most other materials fit inside our budget, though we did do our best to pick the cheapest options we could that still fit our quality needs.

Another constraint we ran into was the limitations of our hardware and software. For instance, we found our TDS sensor to not be sensitive enough to include changes of hydrogen neutral materials being dissolved into water. Dissolving salt had a huge difference on the TDS value, whereas dissolving sugar had little difference to the TDS value. This was because the conductivity of the material is what gave us our TDS value. There are not a lot of other ways to take a TDS value other than a full spectroscopy of the water, as well as a chemical test, leaving us with this as our best guess for the real TDS value of the water. One way we might gain insight to the real TDS value, is if we compare our TDS results with our pH results in order to guess what amount of the TDS value is caused by dissolved conductive material. Furthermore, we have a few software constraints. These being simply that we will not have infinite memory, and instant processing time, though we expect this to not have a huge effect on the end goal of the project.

We also attempted to think about the scalability of our project. If this is to be implemented as a real design, we need to consider how it might be implemented. All the sensors, pumps, and motors need to work if we were to scale the size of the machine. As far as we can tell we have met this benchmark. For instance, we selected our pump as a 12 Volt 3 Amp pump because it will for certain get the water wherever it needs to go. If we implemented this process on a large field. The pump would still be able to do its job and get the water fertilizer mix to the less centralized locations of the field.

We also had to consider time to be a major factor of our project. Given the opportunity I would like to run this project for 5-10 years in many locations. However, since this is just a school project, and we need to graduate, that is not an option. With more time this project could include a larger scope and scale that would allow it to be directly implemented into the real world. Furthermore, we would have more time to refine our process/project and improve on it. There would also be more time/multiple seasons to collect data, which would allow us to draw stronger conclusions about the project.

Lastly, this project has two people working on it. Given a team of 5 people from various stem areas i.e. Biochemical and Computer science backgrounds, this project would be able to incorporate an artificial intelligence section, as well as a fertilizer database selection.

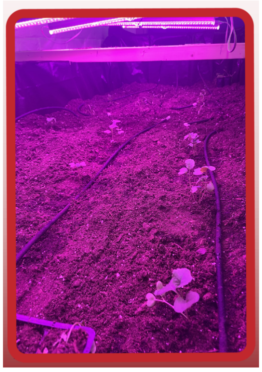


Figure 1, System Setup

The scale and constraints of the project certainly have played a large role in how we have designed this project, and what our next steps would be as a team to allow its implementation from a lab to a real-world setting.

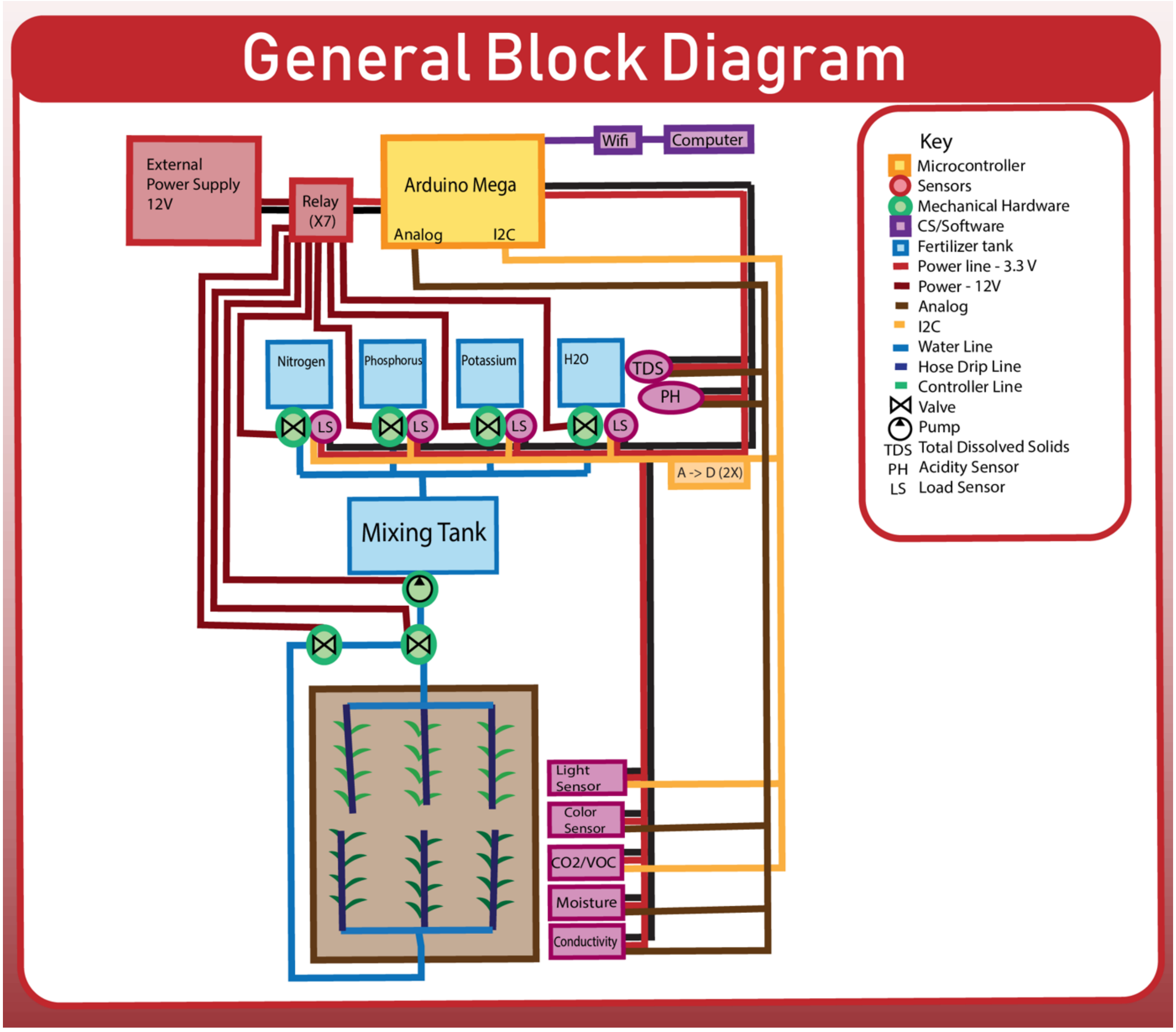


Figure 2, General Block Diagram

Mechanical/Hydraulics

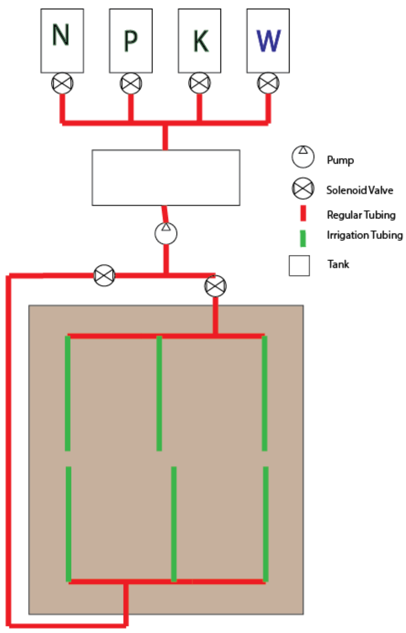


Figure 3, Pump System Schematic

The objectives of the mechanical aspect of this project were to have a working motor and solenoids irrigation system. This was a network of different types of tubing and valves that would allow sections of our bed to be watered, no matter the distance (horizontal or vertical) from the central tank. Some sections of our project used irrigation tubing while other sections used solid tubing. Something our team considered was that the tubing had to be the proper grade and not be clear. The reasoning behind this was so that the tubes did not secrete any plastics into the water, the tubes did not break under hydraulic pressure, and it was not possible for anything to grow within the tubes, leading to contamination. Additionally, the irrigation tubing had to drip a specific number of milli-liters per minute such as to not flood the crops too fast, or drip so slow as to never distribute water. The scale of the mechanical aspects of this project were thoroughly assessed. As stated previously our motor selection works for larger applications. The general design of our tubing layout also allows for any aspect of to be at a larger scale, as well as have as many additional units as needed.

Some of the mechanical constraints of the project were the size of our components. The team did not want to work with a huge pump, or large solenoid values. Furthermore, we had to find these products cheap to stay within our budget. After a lot of hours of searching we were able to find our solenoid valves that fit these constraints.

Figure 4, Pump System

Electrical

The goals of the entire electrical system were to distribute the correct amount of power to each part of the system, as well as correctly take in the environmental data. This meant that each sensor we worked with had its own set of constraints and goals.

For the entire system we had to devise a way to distribute 12V of power to the motors and solenoid values, while controlling them with a 5V out microcontroller. The microcontroller also had to receive power at 5V as well. To meet these goals, we used an external 12V power source and planned to step down the voltage so our microprocessor had the correct amount of power, while also being able to supply voltage to our motor and solenoids.

Next, we also had to consider the electrical objectives and constraints of our sensors. We used four load sensors (strain gauges), but in order to reduce the number of I/O ports we needed we used a premade system that compared the differences in the resistive values caused by the load sensors that were arranged into a full bridge pattern.

This allowed us to free up I/O ports when selecting a microcontroller. Additionally, we had to consider the differences between analog and digital outputs of these sensors. For many of these sensors we were going to end up having an analog output, however for our load sensor, VOC/CO2 gas sensor, and our light sensor, they used I^2C notation which we then had to consider when selecting them. Electrically, this means that the information is being encoded in a different way, which is reflected in our microcontroller selection. We also used a power and ground bar in order to simplify some of the wiring, as well as a breadboard to organize sensors. Some of the electrical constraints of this project included the amount of power that any single sensor could receive had to be reasonably within our microprocessor's output range, the precision of the sensors in terms of output voltage had to be consistent and precise, and all the sensors had to be bought for a reasonable price. We also switched to using twisted shielded pair wires for the sensors in order to reduce the amount of interference/noise in our system. If we were to employ this system into the real world, we would need much more cable. In the future we might also implement a LoRaWAN system in order to not have to be connected to the internet.

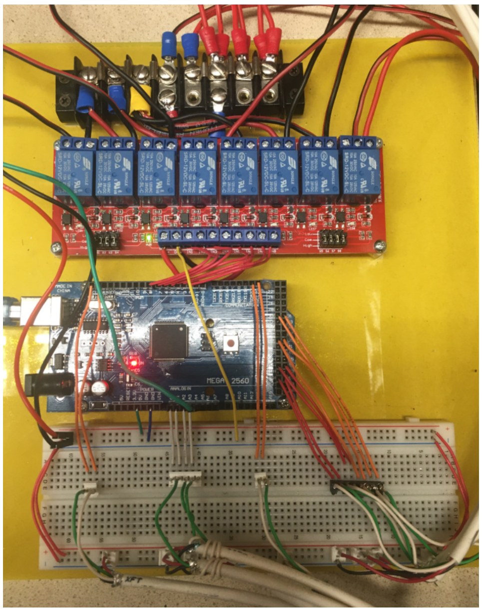


Figure 5, Electrical System

Software

The scope of this project is just to take in all the data given to us. The software objectives of our projects were to have all the sensors running and recording data and be able to send data to the computer through a SIM card. Some of the constraints in this project are that it must be able to handle both digital and analog information, then make all of it digital. On the other end of that we need to have a spreadsheet to dump the information into. Lastly, our program needs to fit and be configured on our microprocessor. This means it needs to be resource efficient. In the future if we find ourselves with more resources and time, we might make a system that applies an algorithm for a neural network system in order to process and draw conclusions about patterns within the data, and have a proper database.

Table Of Contents

Page

Table of Figures

Table of Tables

Introduction

As the society's demand for food production increases, applying fertilizers to crops has become an indispensable part of agriculture. Adding appropriate fertilizers to plants can effectively increase crop yields. However, if excessive fertilizer is applied, it will cause negative impacts on the environment. The chemicals that remain in soil and are not fully absorbed by plants, will cause varying degrees of water pollution, soil pollution and air pollution. This pollution damages the ecosystems of rivers, forests, oceans, by causing algae blooms[4], which often suffocate fish. These blooms can even ruin human drinking water, as they are toxic[5].

Figure 6: Suffocated Fish in Rivers

Figure 7: Algae Blooms in Florida

In order to avert the damage that fertilizers impart on the environment, it is important to know not only how much is needed, but how much is optimal. This agricultural modulation system is designed to not only disperse fertilizer but track the conditions and health of the plants. Through this data collection, the environmental and nutritional factors that sustain plant health can be derived.

This report will cover the environmental impact of fertilizers, their chemical properties in relation to plant growth and previous technology in agriculture. It will also review the electrical, mechanical, and computer programming design processes, as well as its implementation. The results, conclusions, and observations of the monitoring machine will be included at the end of the report. Additionally, this paper addresses the scope, scale, and constraints of our product, along with implications of its scaling, and in-industry use. The overall goal of the project is to mitigate environmental impacts of the agricultural industry and contribute to the human knowledge of the optimization of growing plants.

[4] Fischels, Josie. "At Least 600 Tons of Dead Fish Have Washed up along Tampa Bay's Shore." *Environment*, NPR, 13 July 2021, https://www.npr.org/2021/07/13/1015312707/a-summer-red-tide-has-left-hundreds-of-tons-of-dead-fish-along-tampa-bays-shore.
[5] "The Effects of Fertilizer Runoff." *The Effects of Fertilizer Runoff | Connection Between Agriculture & Drinking Water Contamination*, Multipure, 10 Oct. 2021, https://www.multipure.com/purely-social/science/effects-fertilizer-runoff/.

Background

Historical Context

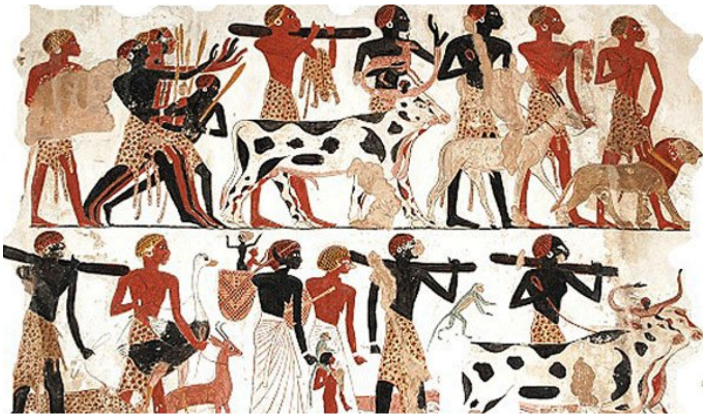


Figure 8 - Agriculture in Ancient Egypt

Humans have been farming for at least 11,000 years. Over this time much has changed. From agricultural practices, to tools, to the plants themselves, the field has grown. Ancient civilizations worldwide created techniques that are used in the modern day, such as terraces, and routine flooding. In the Middle Ages, agriculture stayed largely the same, with some small improvements, such as the domestication of other species, such as sugar. The greatest changes in the history of agriculture have all been recent. In the 17th to mid-19th century, with the British Agricultural Revolution, many processes became mechanized, and the concept of crop rotation was invented, as the demand for food increased with the population. New types of plows were made to accommodate soil types, nitrogen replenishing using clovers was adopted, and fertilizers were experimented with in 1843[6]. The first general purpose tractor was popularized in 1901, and 20 years later saw the decline of using animals in the field. The backbone of all nitrogen fertilizers, ammonium nitrate, was only created in the 1940's, as the 1950's were characterized by gene manipulation and hydroponics. In the 1970's high nutritional cereal was invented by The Green Revolution, which is estimated to have saved nearly a billion people from starvation. In the modern day, many people are still researching how to improve agriculture, and only within the last two decades has there been research done on the environmental impacts of agriculture. In 2000 the first plant's genome was sequenced,

Figure 9 - 18th Century Agriculture

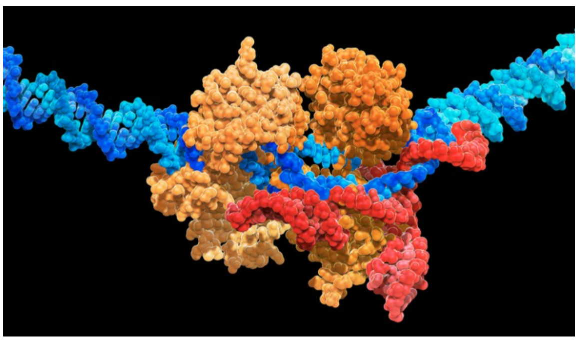


Figure 10 - CRISPR Gene Editing

[6] "History of Agriculture." *Wikipedia*, Wikimedia Foundation, 4 Oct. 2021, https://en.wikipedia.org/wiki/History_of_agriculture.

and in 2012 CRISPR opened the opportunity to gene edit plants[7]. In the scope of the entire history of agriculture, even simple things such as the use of modified fertilizers is relatively novel. Future research, including this project, aims to continue to optimize the field through automation, and creating controlled environmental greenhouses.

Automating Agriculture & Environmentally Controlled Agriculture

A lot of the inspiration for this project came from the two of the three main ways of innovating agriculture in the modern day. The first is by creating better technology in which to care for and harvest the plants, tractor innovation. The second of which is controlling the environment in which the plants reside. Most of the latter has been achieved through greenhouses, and hydroponics. In this sense as many of the factors for plant growth can be individually controlled in order to produce the best results. However, both technological innovations have their pros and cons. Innovating the technology to harvest and care for plants in the fields is entirely complex. Harvesting tomatoes for instance involves an entire AI system to identify the tomatoes that are ready to harvest and building automated arms that can pick the tomatoes. On the bright side, this does save hours and hours of manual labor, though sadly it is currently in development and very expensive. On the other hand, Hydroponic systems and Greenhouse controlled environments show promising results. Farmers can immediately adjust the conditions of the plants within seconds to allow them to get their needs. However, this is also incredibly expensive, as building football sized greenhouses is costly.

Figure 11 - Automation Filled Greenhouse

Furthermore, oftentimes the food grown in greenhouses tends to lack a bit of flavor. This is because the trace minerals usually found in soil, such as manganese, zinc, copper or molybdenum make a difference to the flavor profile of the food and are not usually supplied in various quantities the same way as soil.

Our project actually takes a new approach on deductive reasoning for finding variables for agricultural systems but can also be used in conjunction to all of the current methods for innovating agriculture (Gene Modification, Harvesting/Care Technology, and Environmental Control) Rather than trying to control every aspect of the environment and change one factor

[7] "The Birth of Agriculture ." *Innovature*, ASTA, 2019, https://innovature.com/timeline.

at a time. A problem that could take 2045^2045 individual experiments over every one of each plant's developmental stages, we aim to take data about the plants over time in whatever the current environment is given. Whether the plant is grown outdoors on a farm in Michigan, or in a greenhouse in India, the machine will gather data about the crop. It then will show which changes in variables made the most impact on the plant in every stage of its development. Through this, optimization per environment, and comparisons between each environment can be done. This will allow us to use deduction and optimize plant growth not only in general to the plant, but also per environment, as the conditions in which the plant is grown make a huge impact on its needs.

Motivation

We want to use electronic technology to make agriculture more convenient and intelligent, to globally reduce the amount of environmental impact. Many animals and even a few human lives can be saved by mitigating algae populations. Furthermore, this can financially benefit other industries, such as the fishing industry, and the farming insurance industry, by helping the environment, and optimizing the number of resources needed to ensure plant production. Additionally, the true values for the amount of fertilizer plants need are completely unknown. Finding these exact values along with the changes in the environment's impact on these values will add to the agricultural industry's knowledge of plant development.

Previous work

Before this project was formally announced, both members had started doing research on the Agricultural field. There were also experiments done on hydroponic systems to see what systems would be conducive to growing plants in a controlled vs not controlled environment. The initial idea for the project was created in February of 2019, where one project member experimented with various types of irrigation designs. Following this, research was conducted on what the current knowledge of fertilizer compounds are used in the field.

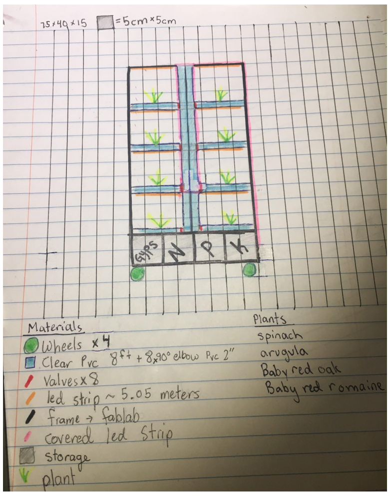


Figure 12: Hydroponics Sketch

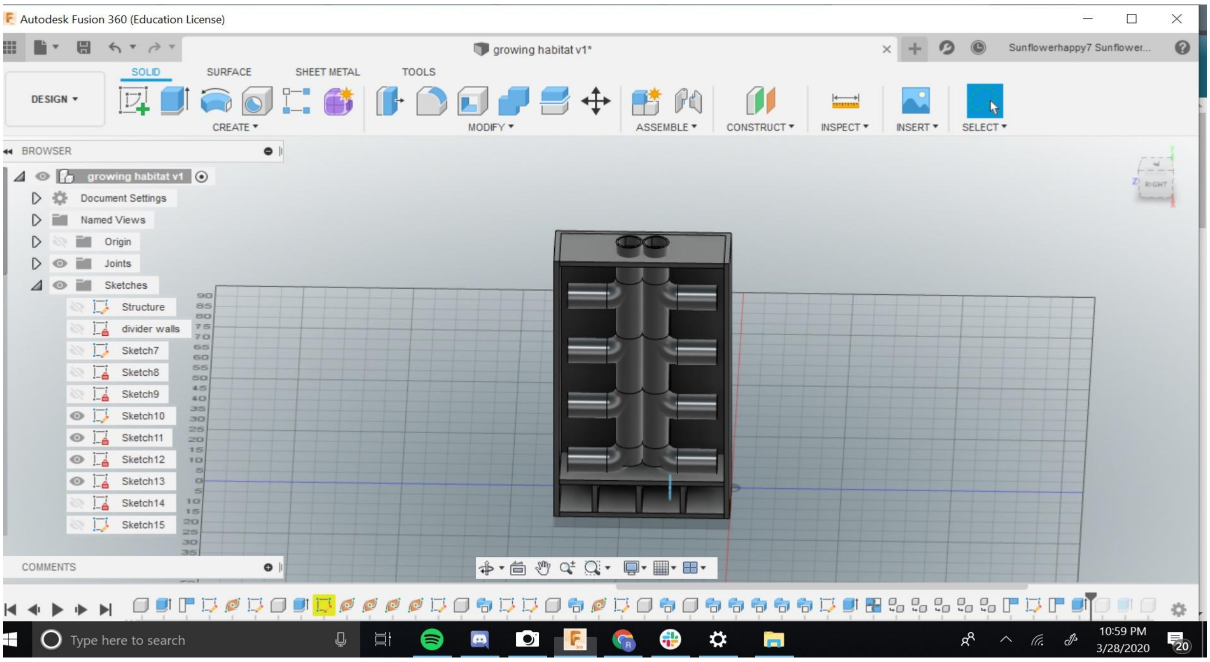


Figure 13: Early CAD Models of a Hydroponics Plant Feeder

Regulations and Standards

The Agricultural Industry in America is regulated by the EPA (Environmental Protection Agency) The EPA places standards on lots of different metrics surrounding farming. Including, but not limited to, pesticide use, livestock and aquaculture regulation, and hazardous chemicals. Searching the official government website showed us which regulations we would be subject to. We found that mostly the hazardous chemicals section applies to us. This involves making sure we meet pollutant limits, and operational standards, and report the use of chemicals.

Crop Production (including nurseries, greenhouses, forestry)

Topic	Type of Farm or Ranch Activity:	Link to Program Area Information	Requirements of Farm
Land Application:	Farms that land apply biosolids or which own land on which biosolids are land applied.	National Pollutant Discharge Elimination System (NPDES) - Biosolids	Federal permit generally not required, but farms must directly meet regulatory requirements for pollutant limits, management practices, operational standards, reporting and other requirements.

Table 1: Crop Production Regulation (EPA)

Goals/Specifications

- The sensors all work
- The mechanical aspects all work
- The system as a whole will function in cohesion
- The system models what a scaled-up version would look like in the field
- We don't kill our plants
- Microcontroller can collect the data and send it to a spreadsheet
- Identify the implications of a scaled up real world application

Constraints

- Cost, some of the sensors and equipment can be very expensive, we need special cable to reduce noise/interference
- Time, this is a fairly ambitious project for a short time frame
- Labor, we have two people working on a fairly large project
- Realistic, we need to make sure that every portion of the projects remains scalable
- Chip Shortage, there was a chip shortage at the time we tried to select our microprocessor
- Size, the components should be able to fit in a relatively small area in the lab.
- There are many more detailed constraints found in the body of the components they relate to.

Methodology

Sensor Selection

Tank Sensors - Load Sensor Research

We needed sensors to be able to sense the values of the tank at any given time, as well as levels of chemicals in each tank, how much we dispensed from each tank, and the volume of the tank in total.

We aim to achieve this by adding a total of five weight/pressure sensors. One per each tank and one on the mixing tank. If we take the weight of the tank, given the approximate weight of an empty tank, gravity, and density of the chemicals, we will be able to calculate the mass and volume of the contents of the tank.

We researched many different types of sensors including pressure sensors, pressure transducers, and pressure transmitters. The pressure sensor has no electronics built-in for signal conditioning and amplified output, and its output signal is in millivolt. The utilization of the devices is around 10ft-20ft from electronics before losing signal because of the low output voltage with wire resistance. For a pressure transducer, it has signal amplification capabilities to boost output signal. Also, it has higher output voltage which is good for battery powered, and greater range from electronics. Pressure transmitter has low impedance current output, and it is analog 4-20 mA with 2-4 wire configuration. It has good electrical noise immunity, but it is unsuitable for battery or high pressure.

There are six main types of pressure sensors we can choose from: potentiometric pressure sensor, inductive pressure sensor, capacitive pressure sensor, piezoelectric pressure sensor, strain gauge pressure sensor and variable reluctance pressure sensor. First is the potentiometric pressure sensor. This one is too crude for this project. Inductive pressure sensor is a sensor sensitive to metals. It can tell how far away metal is, and as the distance changes it keeps track for it. The capacitive pressure sensor is the sensor which will change capacitance as the sensor is pressed due to its structure. It has non-linear output, although this can be reduced in touch-mode devices. However, this may cost greater hysteresis. But it is durable and has a low cost. Piezoelectric pressure sensors will fade under static pressure, and it is not good for our project considering the static weight. Strain gauge pressure sensors is a kind of sensor made to measure the change of something. The resistance of this sensor will vary with applied force. The last one is the variable reluctance pressure sensor. Magnetic sensor is formed, pressure forces spring down, and the difference is calculated. But it can be expensive and may increase our budget.

Taking everything into consideration, we decided to use a strain gauge pressure sensor as our pressure sensor. And the Load Cell is one of the strain gauge pressure sensors. The load cells are incredibly precise. Most seem to be within less than 1% of the load. And the price range tends to be between 1-400 depending on how much you want to carry, and the type/brand. We decided to go with a load sensor because they are incredibly cheap, accurate, and can be made of materials that do not interfere with the chemicals we are using. They don't use a lot of power.
The estimated cost of our load sensor is 20$ for all of them

Products
https://www.walmart.com/ip/50kg-Weigh-Sensor-Thread-Hole-Half-bridge-Resistance-Strain-Body-Load-Cell-4pcs/803040186?wmlspartner=wlpa&selectedSellerId=571

Part Specification

- Power usage
  Maximum Operating Voltage: DC 8V
- Dimensions
  Load Cell Size: 34 x 34 x 8.8mm (L*W*T)
- How long does it last for - a long time
- Analog/Digital
  Output impedance: 1000+/-5 Ohm
  Input impedance: 1000+/-5 Ohm

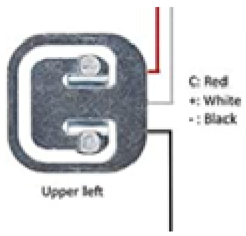


Figure 14: Load Cell

Water Sensors

Water is an essential piece of plant growth. It is imperative that we can gauge what the water is like that is being given to the plants. In particular we care about a few different metrics. These metrics include, quantity of water, number of pollutants in the water, the amount of hydrogen in the water (pH) and possibly the amount/type of minerals dissolved in the water.

Looking at our design the first of these constraints is met by using our load sensor, which tells us the relative quantity or water within the system by measuring the change in volume of the tank over time. The second of these metrics, pollutants can be found using a TDS, or total dissolved solids sensor, which works by estimating the conductivity of the water. To estimate the pH of the water, or the amount of hydrogen in it, we can use a pH Sensor. While getting a pH value for the water may not tell us what exact minerals are in the water. Both sensors together can give us a rough, and useful estimate.

Our team decided that the best place to measure these values would be before the water is able to reach the plants, or while it is still in the initial water tank, as shown on the general diagram.

TDS Sensor

There is only one way to know the TDS of a liquid, and that is through taking the conductivity of said liquid, this means that there is only one type of TDS sensor. However, one possible con to this setup, is that there may need to be an annual check on the TDS sensor to check for salt build up on the probes.

Products

https://wiki.seeedstudio.com/Grove-TDS-Sensor/

Figure 15: TDS Sensor

Part Specification

- Power usage
  support 3.3 / 5V Input Voltage
- Dimensions
  140mm x85mm x11mm
- How long does it last - for 2+ years annual checkup advised
- Analog/Digital
  analog
  0 ~ 2.3V Output Voltage

pH Sensor

Next, we need to use a pH Sensor to take the acidity of the water. This is useful because pollutants in the water tend to change the pH of the water, and we can also use this to gain insight as to if the dissolved solids are salt by deduction, as salt is fairly neutral. Given Salinity, Conductivity, and pH (H+ ions) we can determine the purity of the water being given to plants.

There are a few different types of pH Sensors. These include Combination pH Sensors, Differential pH Sensors, Laboratory pH Sensors, and Process pH Sensors. Within most of these types of sensors, there is also a wide range of specifications, which include but are not limited to the type of setting that they will be applied to, such as Consumer, Lab Grade, Micro pH Sensors.

Combination pH Sensors are the most commonly used sensors, and act as a basis for which lab sensors are built on, they are electrochemical sensors that use a reference and measuring electrode to

Differential pH Sensors work in a similar way to combination pH Sensors; however, they have three probes, the final probe is an additional grounding probe that helps prevent any reference fouling within the sensor.

Consumer probes are a version of the Combination pH Sensor that is simplified and good for taking very general measurements, such as a swimming pool.

Lab Grade probes are probes made of higher quality materials and allow for more precise measurements of the pH of the water.

Micro pH Sensors are good for smaller applications in which the tiniest amounts of water are sensed.

We ultimately decided that we would be looking for a lower end, hopefully durable, Lab Grade sensor, Using the general consumer grade combination pH Sensor would give us an accurate reading of what the pH is, and a micro pH Sensor would be too small for this application as we are working with tanks in the gallons. Seeing as we do not need the pH bounds from 0 -14, and instead 2-13 will be specific enough, we do not need the Lab level pH Sensor.

<u>Part Specification</u>

- Power usage
- Dimensions

  12mm x 150.6mm (0.5" x 6")

  Time before recalibration ~1 year

  Life expectancy ~1.5 years +
- Analog/Digital

Figure 16: pH Sensor

Products

https://www.amazon.com/Atlas-Scientific-Consumer-Grade-Probe/dp/B07VDMNB92/ref=pd_sbs_2/144-2475047-7024531?pd_rd_w=USDtE&pf_rd_p=0f56f70f-21e6-4d11-bb4a-bcdb928a3c5a&pf_rd_r=W080QXNA5T00FBPT77X2&pd_rd_r=724c6cb3-ff70-4fc3-bfe3-823069b02584&pd_rd_wg=rGHeT&pd_rd_i=B07VDMNB92&psc=1

https://atlas-scientific.com/blog/pH-sensor-raspberry-pi/

Light Sensor

It is also important to collect the data on the light received by plants. We want to know the durations, intensity, and the type of light from the light sensor.

In order to collect these data, we can stick a sensor at the top of the machine (pointed up) to read the light, and then place a color filter over the top of the sensor so that it only takes in the light that plants would care about, thus judging the "quality" or type of light we care about.

There are three main types of the light sensors: Photoresistor sensors, Photodiode sensors and Phototransistor sensors. Photoresistor sensors are related to the photoconductivity which senses the light on the surface of the resistor and then increases the resistance as the light decreases. They can experience small amounts of latency, especially when going between absolutely dark to absolutely light situations - as much as a second. They can be intrinsic or extrinsic. They are commonly made of Cadmium. They have a low cost, but they are less sensitive than a Photodiode or Phototransistor. Photodiode sensors can convert light to current. They have slower response time with larger surface areas, and the type of material changes what you are reading. They are sensitive to electromagnetic radiation. Solar cells are used for this kind of sensor. They are relatively inexpensive, but they cannot be used for low-light situations. Phototransistor sensors are bipolar transistors. They are able to detect light better than Photodiodes and have a lower response time. If we use one of the sensors above, we also need a color filter. It is a way to block a specific wavelength of light from reaching a place.

In addition to the above sensors, color sensors are also an option we consider. They can sense color values in RGB+IR/XYZ+IR and output a frequency response or digital output of precise values. They can poll the exact light range wanted with digital logic. They are extremely precise and used commonly in cameras. However, they are more expensive than other sensors.

We ultimately decided to use a color sensor because they are much more specific to the application we would be using them for. Plants need a variety of wavelengths in order to grow, particularly the red and darker blue wavelengths. Getting the measurements for these wavelengths could be useful. Then later we decided to add another sensor to judge the intensity of the light itself.

https://www.seeedstudio.com/Grove-Sunlight-Sensor.html

Part Specification 2X

- Power usage
  Operating Voltage 3.0-5.5V
- Dimensions
  140mm x85mm x11mm
- Lasts a long time
- Analog/I2C

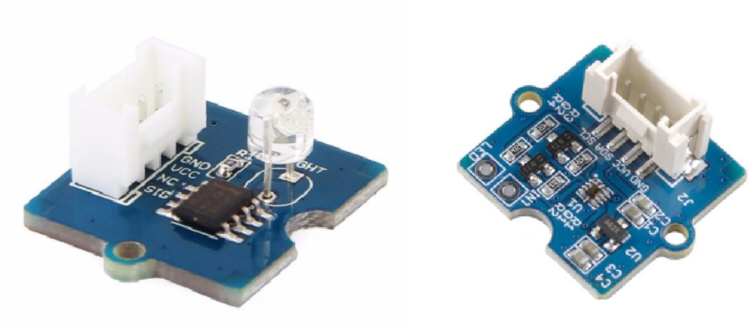


Figure 17: Light Sensors

CO2 Sensor

It is also necessary to collect data on the concentration of carbon dioxide around plants. We will need a sensor to measure CO2. We choose the CO2 Sensor with Arduino Compatible. For this kind of sensor, the output voltage of the module falls as the concentration of the CO2 increases. If the CO2 concentration is high enough, a digital signal will be released. It is highly sensitive to CO2 and less sensitive to alcohol and CO and has low humidity and temperature dependency.

Products

https://www.mouser.com/ProductDetail/DFRobot/SEN0159?qs=Zcin8yvlhnMKsp%2FpSVzC4Q%3D%3D&mgh=1

Part Specification

- Power usage
  0 V to 5 V
- Dimensions
  32 mm x 42 mm
- Uses I2C
- Lasts a long time

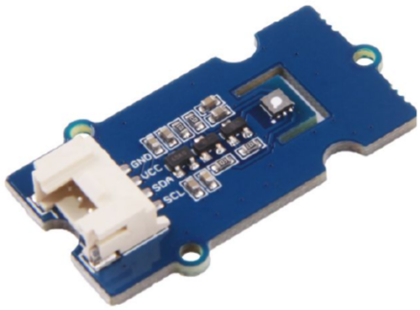


Figure 18: CO2/VOC Sensor

Soil Sensors

Moisture Sensor

There are two main types of water sensors. Volumetric Water Content sensors, and Tensiometers. The prior senses the percentage of a material that may be made of water, and are excellent for a lab setting, where very high precision is necessary. The second of the two works when placed in soil and takes a rough measure of the water content of the soil. For the project we decided to use a low budget capacitive sensor as well as a conductivity sensor in order to compare them and get consistent results.

Conductivity Sensor

A conductivity sensor works by putting two rods into the ground, and running a charge between them, the soil will then act as a slight insulator, only allowing a certain amount of the voltage to pass between the filters. With a resistor in place, we can use this as a voltage divider, which allows us to see how much of an insulator the soil was. This can give us insight into the pH, The moisture level, and the general composition of the soil. We chose this sensor because we could have made this sensor on our own using two nails and a resistor. (Which was done during the prototyping phase) we find that the manufacturer below has provided a very cheap, and slightly more consistent sensor that we can use. The estimated cost for this sensor is six dollars.

https://www.mouser.com/ProductDetail/SparkFun/SEN-13322?qs=WyAARYrbSnberEHb08L3VQ%3D%3D&mgh=1&gclid=Cj0KCQjw18WKBhCUARIsAFiW7JxuKlBQfNPP9uNrgvwZQlnFIVxsJsJNhYgBDAPKvREH-_Gp5Nm8NP4aAgHbEALw_wcB

Part Specification 2X

- Power usage
  0 V to 5 V
- Dimensions
  <5 inches
- One uses I2C
- Analog/Digital

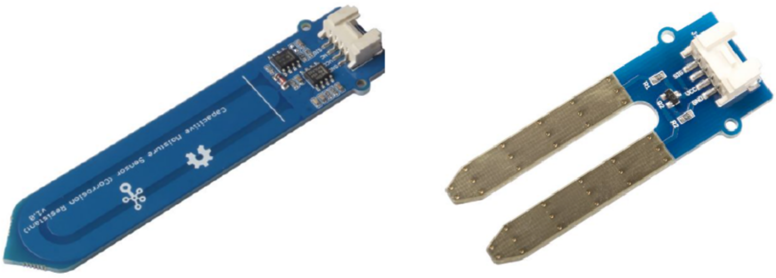


Figure 19: Moisture Sensors

Mechanical Selection

Valves

We need several valves to control and regulate the flow of liquids. This is important for us to control the concentration of fertilizer in the solution and the amount of nutrition applied to the plant.

There are many types and designs of valves. There are some examples of the common types of valves: ball valve, butterfly valve, globe valve, gate valve, plug valve, diaphragm valve, reducing valve, needle valve, check valve, and safety/relief valve. The methods of controlling flow can be divided into four categories. First is moving a disc or plug into or against an orifice. For example, globe or needle valves. The operating principle of the second kind of valves is sliding a flat, cylindrical, or spherical surface across an orifice, like gate and plug valves. The third kind of valve is working by rotating a disc or ellipse about a shaft extending across the diameter of an orifice, for example, butterfly and ball valves. The last kind of valve, such as diaphragm valves, is working by moving a flexible material into the flow passage.

We decided to use solenoid valves in our project. Solenoid valve is an electrically controlled valve, which contains an electric coil with a movable ferromagnetic core (plunger) in its center. The magnetic field is generated by the current of the coil, and the magnetic field pushes the switch of the valve. This is the basic principle of the solenoid valve. It is fast acting and can only be used for clean liquids and gasses. And it can get hot as it requires energy to switch and stay in that position. This kind of valve is often used in heating systems, compressed air, vacuum, irrigation, car washing, etc. We choose to use it because it can be controlled by the microcontroller.

Solenoid valves are generally divided into three types: normally closed solenoid valve, normally open solenoid valve, and bi-stable solenoid valve. For a normally closed solenoid valve, the valve is closed when there is no power supply, and the media cannot flow through. When the current passes through the coil, it will generate an electromagnetic field, forcing the plunger upward against the spring force. This will open the orifice to allow the media to flow through the valve. For a normally open solenoid valve, contrary to normally closed, it is open when de-energized. When the current passes through the coil, the electromagnetic field will force the plunger downwards overcoming the spring force. This will close the orifice to stop the media to flow through the valve. This is more suitable for applications that need to open the valve for a long time, because it is more energy efficient. A bi-stable or latching solenoid valve is switched by a momentary power supply. When there is no power applied, it stays in its current position. They are using permanent magnets instead of springs to achieve that. We chose the normally closed solenoid valve because most of the time we don't need liquid to pass through.

https://megadepot.com/resource/a-guide-to-types-of-valves

https://tameson.com/solenoid-valve-types.html

Figure 20: Normally Closed Solenoid

Pump

We need a pump in order to move the water in the tank through the field. In order to do this, we had to pick a pump that can move water at least 3 meters. These are relatively inexpensive and simple. Just to be careful we picked a 12V DC dredge pump for boats. (It worked well in testing) It costs no more than $25.

Figure 21: 12V DC Motor

Micro Processor Selection

Sensors	Working (V)	Working (I)	Signal (V)	Type	Channels on sensor
load sensors (4)	3.3-5.5			digital	GND, VCC, SIG, DT
pump	DC 12			-	VCC, GND
valves	DC 12			-	VCC, GND
ph sensor	3.3-5		0-3	analog	VCC, GND, SIG
CO2 sensor	3.3-5	-	0-2.3	analog	GND,VCC, SIG, SMCL
Resistive Moisture sensor	3.3-5	35mA		analog	GND, VCC, SIG
Conductivity sensor	3.3-5			analog	GND, VCC, SIG
TDS sensor	3.3-5.5	3mA-6mA	0-2.3	analog	GRD, VCC, SIG
sunlight sensor	3.0-5.5	3.5mA		analog	VCC, GND, SDA, SCL
light sensor	3~5	0.5mA-3mA		analog	GRD, VCC, SIG

SIM card	
GM Sheild from Sparkfun	https://www.sparkfun.com/products/15087
GM Sheild from Arduino	https://www.tinyosshop.com/arduino-gsm-shield

Table 2: Microprocessor Sensor Specifications

In order to choose a suitable microprocessor, we need to consider the following factors. First is the power efficiency. Higher processing power will consume more energy. The second one is the temperature tolerance. It depends on the environment in which your microcontrollers operate. Next is the hardware architecture. Microcontroller's packaging will directly influence its size and performance. For processing power, what we mainly need to consider is we need a single core or multicore microprocessor. Memory is also an important factor. More programs need more random access memory (RAM). In addition, we need to consider security, hardware interface, software architecture and cost.

https://www.microcontrollertips.com/key-factors-consider-choosing-microcontroller/

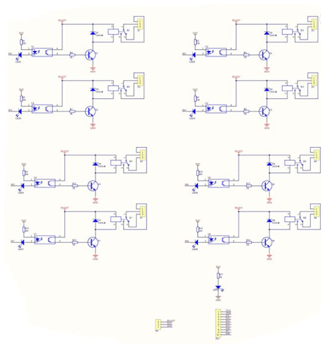
In order to confirm the above requirements and select the correct microprocessor, we need to choose the microprocessor according to the following steps. First, we need to list the required hardware interfaces. There are two general types of interfaces we need to list. One is communication interfaces, such as USB, I2C, SPI, UART. The second type includes digital inputs and outputs, analog to digital inputs, PWM's, etc. These two interface types will dictate the number of pins that will be required by the microcontroller. Then we need to examine the software architecture. We need to know how much processing power will be needed. The computing power will be one of the most important requirements for the architecture and frequency of the microprocessor. Next, we need to select the architecture. Based on the information we get on the first and second step, we can get an idea of the architecture we need. After that we identify memory requirements. Flash and RAM are two very fundamental components of the microprocessors. And it is important to ensure that the program space or variable space is not used up. Then we can start searching for the microprocessor based on the requirements we list before. On this basis, we need to examine costs and power constraints. When we chose the microcontrollers, check the part availability. Then we need to select a development kit and investigate compilers and tools. After all of these are done, we can start experimenting.

https://community.arm.com/arm-community-blogs/b/embedded-blog/posts/10-steps-to-selecting-a-microcontroller

Design/Implementation

Relay

We also decided we needed a relay in order to power the pump and the solenoids. In order to do this, we googled relays that could be powered by the microprocessor we selected, and we bought it for $12



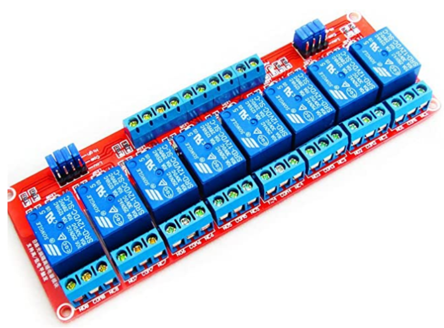
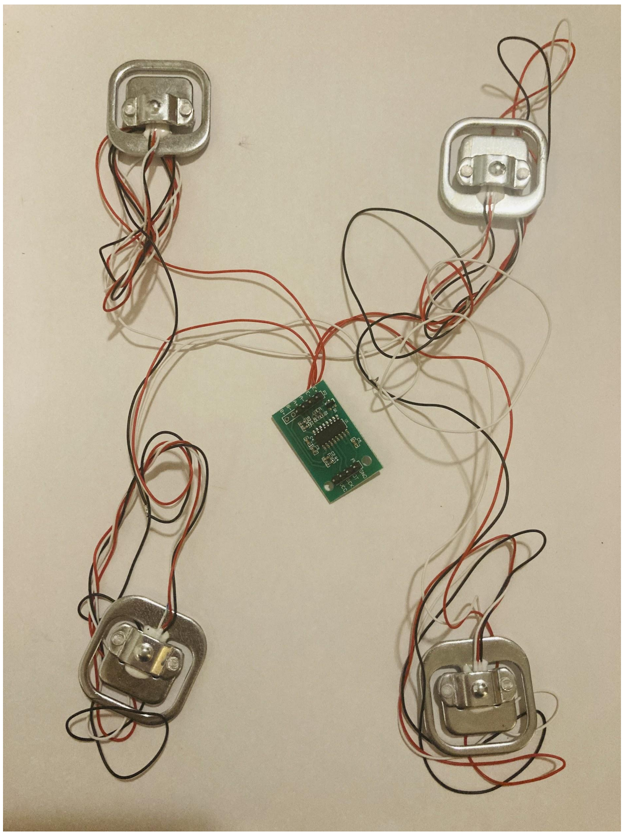


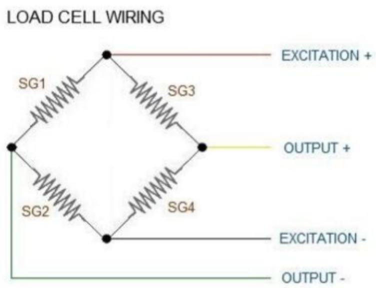
Figure 22: Relay Schematic *Figure 22: Relay Schematic*

Sensor Testing

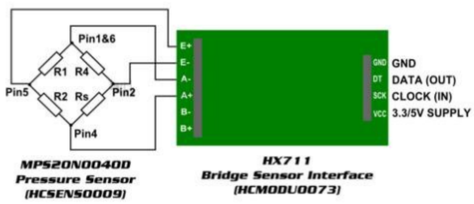
Load Sensor

The load sensors function as a strain gauge. Meaning that when the metal is bent it applies a certain resistance to the circuit. As this happens, we can calculate the amount of pressure being applied on the system, thus calculating the mass of liquid the system is dispensing to the garden bed. Our load





sensors are



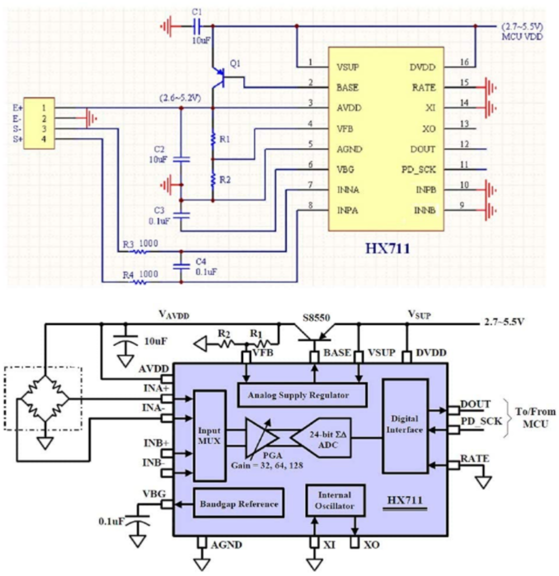


Figure 24: WheatStone Bridge Schematic

Figure 25: Soldered Load Cells

soldered into a Wheatstone bridge arrangement (shown in a schematic below and wired to the right) to judge the resistive values of the strain gauge load sensors using fewer I/O ports than if we had wired them individually. To the right you can also see the analog to digital converter board that came with the load cells. This process is happening by running

Figure 26: Analog to Digital Converter

the raw analog values through a MUX converter, then applying a programmable amplification or gain to the values. Afterwards this becomes a 24bit ADC connection that is sent as a digital signal. Our microprocessor then can process this information when we send it to our digital input ports as shown below.

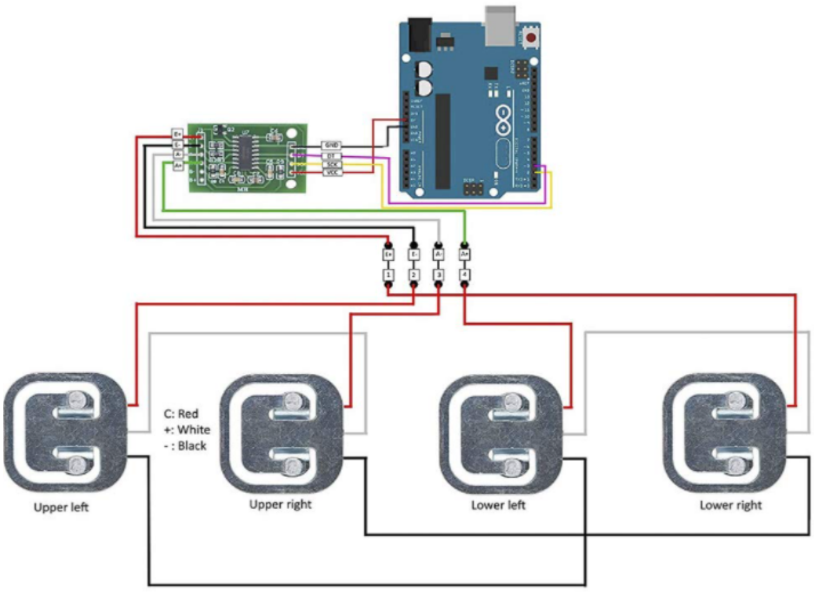


Figure 27: Load Cell System Diagram

In our application we use these under the tubs of solution to know the mass of water/solution that we are depositing into the bed

TDS Sensor

In order to test the performance and sensitivity of the TDS sensor, we first recorded the TDS values of both unfiltered and filtered water. And then we dissolved sugar, salt, urea and potash in water, tested and recorded the TDS values of different solvents at different concentrations. During the test, we immerse the test head of the TDS sensor into the liquid, but we can't touch the edge of the container, as that would short out our system. Once the TDS sensor touches the container, the measurement results will become inaccurate.

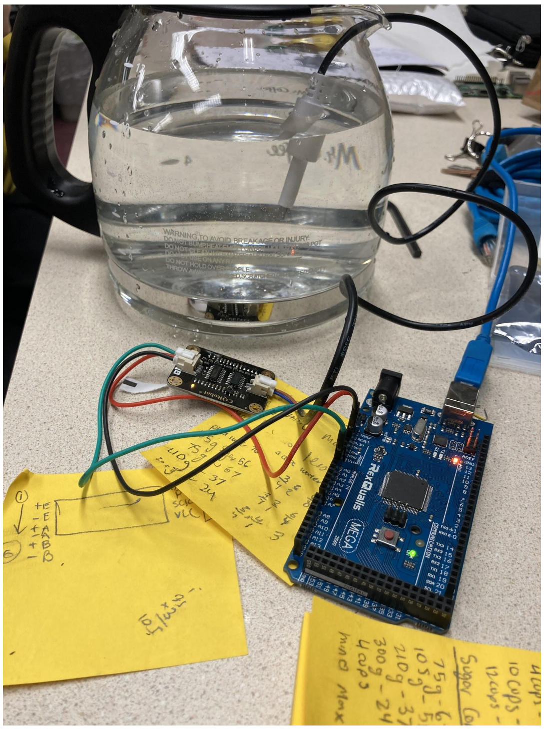


Figure 28: TDS Sensor Testing

In the table below, we can see that the unfiltered water has higher TDS value than the filtered water, but the volume of the liquid will not influence the TDS value. For potash and salt, the change of TDS value is very sensitive to the concentration of the solution. When the solution concentration becomes high, the TDS value will rise sharply. But for urea and sugar, the relationship between TDS value and concentration has different manifestations. The urea solution has lower TDS value than the filtered water, and it almost does not change with the variation of concentration. The performance of sugar is that the higher the concentration, the lower the TDS value. The way that we are applying this into the system is that we will stick the probe into the water source to sense how many particles are dissolved within our water supply, then we will keep a running log of the information. This way we know the general content of our water supply and will be able to assess how that impacts the system.

TDS Sensor Testing Potash (K20)	
Concentration oz/Cup	ppm
0.00006211	190
0.00012261	308
0.00024523	626

TDS Sensor Testing Urea CH4N2O	
Concentration g/Cup	ppm
0.083	70
0.162	72
0.324	72

Control Test Unfiltered H2O	
Volume (cups)	ppm
10	90
8	90
4	90
2	90

TDS Sensor Testing Sugar (C12H22O11)	
Concentration g/Cup	ppm
18.75	67
26.25	56
52.5	37
75	24

TDS Sensor Testing Salt (NACL+NAlO3)	
Concentration g/Cup	ppm
0.025	198
0.05	284
0.075	390
0.1	492

Control Test Filtered H2O	
Volume (cups)	ppm
10	76
8	76
4	76
2	76

Table 3: TDS Sensor Testing

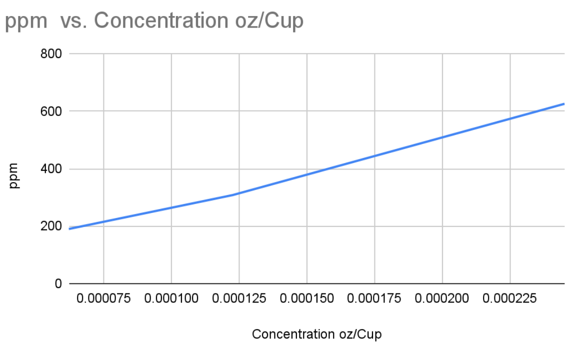


Figure 29: PPM vs Concentration Salt Sugar

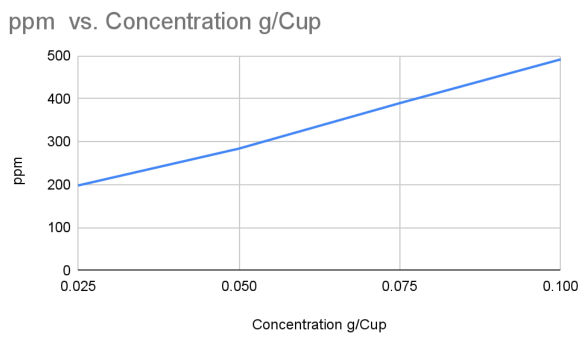


Figure 30: PPM vs Concentration

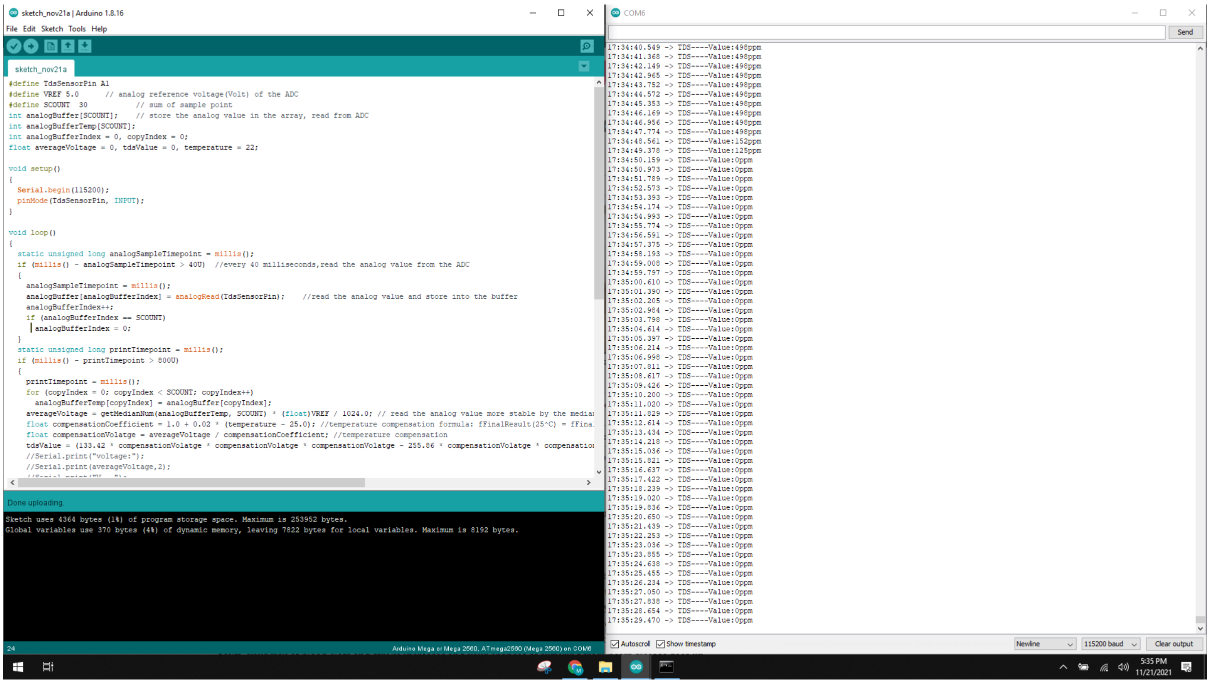


Figure 31: TDS Sensor Code

pH Sensor

Our pH Sensor uses a BNC type connector to interface to its sensing module. This takes the pH of the system and converts it to a usable analog signal. In order to calibrate the pH meter, we used local Worcester City water reports to give us the pH of the water. We then adjusted the pH value of the water using software such that it would output the correct pH. We then continued to test the pH Sensor checking its sensitivity. We found that it was sensitive enough for our use by adding a pinch of salt and watching the pH adjust accordingly. We found that our pH Sensor is adequate for our application. The code for the pH Sensor is as follows. The way that we applied this into our system is that we placed the probe into the water source of the system to actively read the water going into our crops. We then connect that pH Sensor to our I/O ports on our microcontroller through our PCB board. The following equations show us what chemicals we are using as fertilizer.

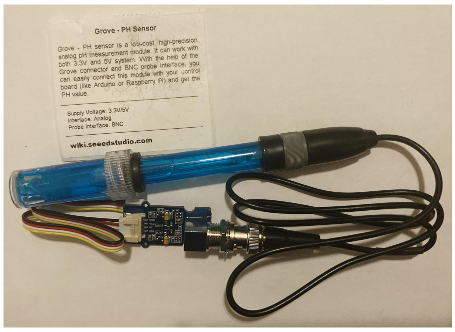


Figure 32: pH Sensor

$$F^-(aq) + H_2O(l) \rightarrow HF(aq) + HO^- \qquad HSO_4^- Na^+$$

Figure 33: Chemical Equations

```
ph_sensor
#define SensorPin A2          //pH meter Analog output to Ardui
#define Offset 41.02740741    //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth  40    //times of collection
#define uart  Serial
int pHArray[ArrayLenth];   //Store the average value of the sens
int pHArrayIndex = 0;
void setup(void)
{
  pinMode(LED, OUTPUT);
  uart.begin(9600);
  uart.println("pH meter experiment!");    //Test the uart monit
}
void loop(void)
{
  static unsigned long samplingTime = millis();
  static unsigned long printTime = millis();
  static float pHValue, voltage;
  if (millis() - samplingTime > samplingInterval)
  {
    pHArray[pHArrayIndex++] = analogRead(SensorPin);
    if (pHArrayIndex == ArrayLenth)pHArrayIndex = 0;
    voltage = avergearray(pHArray, ArrayLenth) * 5.0 / 1024;
    pHValue = -19.18518519 * voltage + Offset;
    samplingTime = millis();
  }
  if (millis() - printTime > printInterval)  //Every 800 millise
  {
    uart.print("Voltage:");
    uart.print(voltage, 2);
    uart.print("    pH value: ");
    uart.println(pHValue+1, 2);
    digitalWrite(LED, digitalRead(LED) ^ 1);
    printTime = millis();
  }
}
```

```
ph_sensor
double avergearray(int* arr, int number) {
  int i;
  int max, min;
  double avg;
  long amount = 0;
  if (number <= 0) {
    uart.println("Error number for the array to avr
    return 0;
  }
  if (number < 5) { //less than 5, calculated direc
    for (i = 0; i < number; i++) {
      amount += arr[i];
    }
    avg = amount / number;
    return avg;
  } else {
    if (arr[0] < arr[1]) {
      min = arr[0]; max = arr[1];
    }
    else {
      min = arr[1]; max = arr[0];
    }
    for (i = 2; i < number; i++) {
      if (arr[i] < min) {
        amount += min;      //arr<min
        min = arr[i];
      } else {
        if (arr[i] > max) {
          amount += max;   //arr>max
          max = arr[i];
        } else {
          amount += arr[i]; //min<=arr<=max
        }
      }//if
    }//for
    avg = (double)amount / (number - 2);
  }//if
  return avg;
```

Figure 34: pH Sensor Code

Based on the schematic of the pH Sensor below we noted that the pH Sensor works by sending two voltages through two non-inverting amplifiers. This is then just directly sent along the signal wire to be our analog output. In the software side of things, we poll this output voltage every few milliseconds and then multiply our voltage by a known constant and divide it by our preset offset value. This then is stored in an array where it is averaged with other values to give us our average output value.

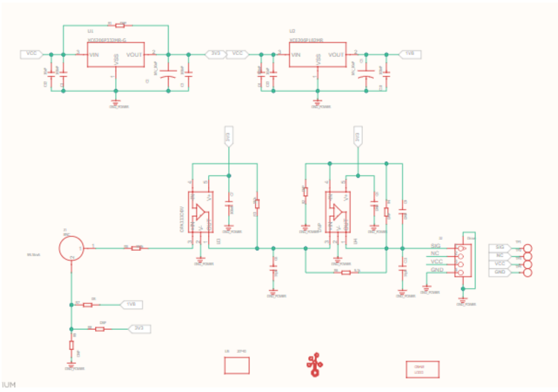


Figure 35: pH Sensor Schematic

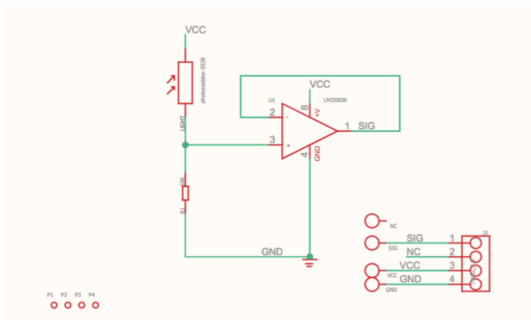
Light Sensors

We tested our light sensor at three different light intensities. We carried out experiments under normal light, flashlight irradiation and complete shielding of the light sensor. The value drops down to zero when we cover the light sensor. At partial coverage we noted that the value changes significantly. And when we use flashlights to illuminate the sensor, the value of light strength has increased significantly. For the purpose of this experiment, we will use the lux units. (Scaling can be seen to the right of this) Seeing as this was an analog device we plugged it into the analog input pin of our microcontroller. In the general sense the way that this is implemented is that it will sit next to the gardening bed under the same light conditions as the bed. It will then give us the intensity of the sunlight that our crops are experiencing. The code simply polls the value and prints that output.

This sensor works by sending the signal by a photoresistor and going through an amplifier. This is then fed directly to the analog port on the Arduino, allowing us to read the analog signal.

light_sensor

```
void setup() {
  // initialize serial communication at
  Serial.begin(9600);
}

void loop()
{

  int Sensorvalue = analogRead(A0);
  Serial.println(Sensorvalue);
  delay(100);
}
```



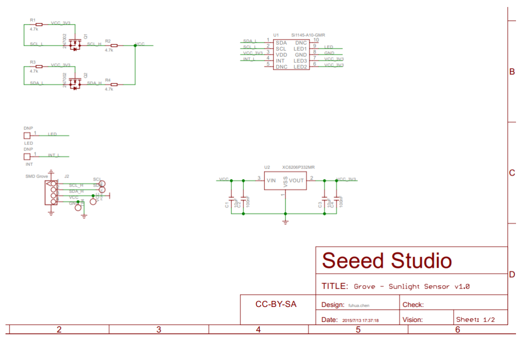


Figure 35: Light Sensor Schematic 1
Figure 36: Sunlight Sensor Schematic
Figure 37: Light Sensor Code

Sunlight Sensor

Unlike the other sensor, this light sensor detects a wider spectrum of light. Recently it has been proven that the UV-Spectrum[8] has a significant effect on plants, and this sensor takes that into account. In sum, this sensor uses a 7-bit I2C protocol in order to send its information. This

8

means that on the software side we cannot use an analog input pin on the Arduino. Similar to the first light sensor, this sensor will also be kept next to the bed to track sunlight.

Resistive Moisture Sensor

moisture_sensor

```
int sensorPin = A0;
int sensorValue = 0;

void setup() {
    Serial.begin(9600);
}
void loop() {
    // read the value from the sensor:
    sensorValue = analogRead(sensorPin);
    Serial.print("Moisture = " );
    Serial.println(sensorValue);
    delay(1000);
}
```

Figure 38: Moisture Sensor Code

The moisture sensor interfaces with the microprocessor by the analog port. We use it to detect the moisture in the soil. We tested it by inserting it to wet soil and dry soil and comparing the results we got. To test our sensor was functioning properly we had a cup where the top of the soil was completely dry, and the bottom was completely wet. As we pushed the sensor into the soil, we noted that the value was zero when we started, and as it plunged deeper into the soil the value went up at the same rate that we layered the damp soil into the cup. Meaning this sensor is sensitive enough for our purposes. Based on the schematic we can tell that this is a simple device. In essence it works by running a voltage through one of the prongs and taking the value of resistance between the two prongs. We can see a transistor that acts as a switch to connect and disconnect the sensor. In our system, we insert it into the soil in the gardening bed and detect the soil condition around plants. We will also compare this value with the capacitive moisture sensor.

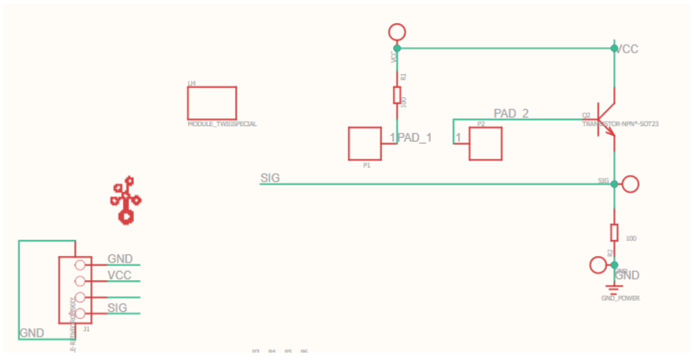


Figure 39: Resistive Moisture Sensor Schematic

Capacitive moisture sensor

Capacitive moisture sensor has similar usage with the moisture sensor, but it is based on capacitive sensing. It is also interfaced by analog. We insert it into the soil in our project and detect the moisture of the soil.

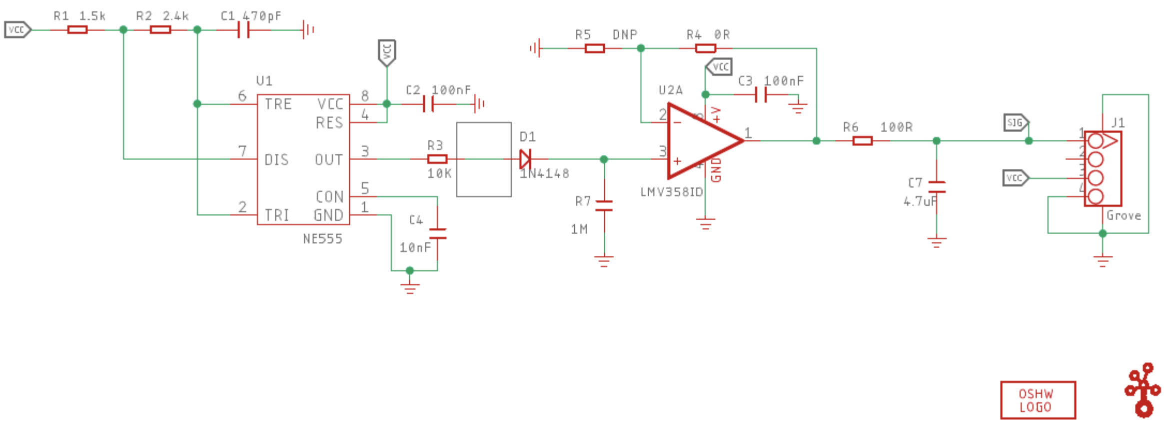
Figure 40: Capacitive Moisture Sensor Code

Figure 41: Sensors in Field

capacitive_moisture_sensor

```
/*
  AnalogReadSerial

  Reads an analog input on pin 0, prints the result to the Serial Monitor.
  Graphical representation is available using Serial Plotter (Tools > Serial Plotter menu).
  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

  This example code is in the public domain.

  https://arduino.cc/en/Tutorial/AnalogReadSerial
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue-500);
  delay(100);        // delay in between reads for stability
}
```

Figure 42: Capacitive Moisture Sensor Schematic



CO2 Sensor

The VOC and eCO2 gas sensor are used to detect the total VOC and H2-based CO2eq in ppm. It features an I2C interface to transform data. In our system we arrange it in the gardening bed and test the CO2 concentration near the plants.

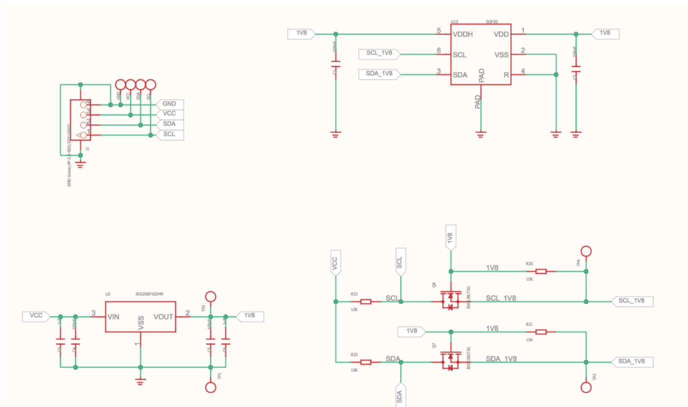


Figure 43: CO2 Sensor Schematic

Mechanical System Testing

In our irrigation system, we have 4 containers to hold the water, nitrogen, phosphorus and potassium, and each container has a solenoid valve to control the flow. The solenoid valves are connected to one tank by tubing. When solenoid valves open, the liquids will flow through the tubing and mix in the tank. To ensure that the liquid can flow smoothly into the mixed tank, we put four tanks containing water and fertilizer on the table and put the mixed tank under the table. The water pipe at the bottom of the mixed tank is connected to a pump. The pump will send the mixed fertilizer to the field with plants. The other end of the pump is connected with two solenoid valves, so that we can deliver different percentages of fertilizer to different plants.

Figure 44: Irrigation System

Power and Control Board

After looking at all the specs we need, we found that an Arduino Mega 2560 did the job. With over 14 Analog IO pins, the ability to use a few I2C, and its availability during the chip shortage, we chose it. In order to send 12V DC power we connected it to a relay. We also used a bar to connect a lot of our 5V signals to the relay from the microcontroller, as well as a breadboard to organize the sensors on.

We used it to control the pump and solenoid valve and record the data of sensors. In order to control the solenoid valves and pump through the microcontroller, we added a relay to the circuit. Connecting the relay to the microprocessor, the microprocessor will send the signal to relay to control the switch of corresponding ports. We also connected all our sensors to the microcontroller. Most sensors are connected in the analog interface. In addition to VOC and sunlight sensors, we use the I2C interface. All the sensors are powered at 5V by microcontroller. The microcontroller is connected to a computer which will record the data and power the microcontroller.

We also powered and ground all the solenoid valves, pump and relay. We use the same DC input and ground for all the solenoid valves and relays. Since the pump has different voltages required, we power the pump separately.

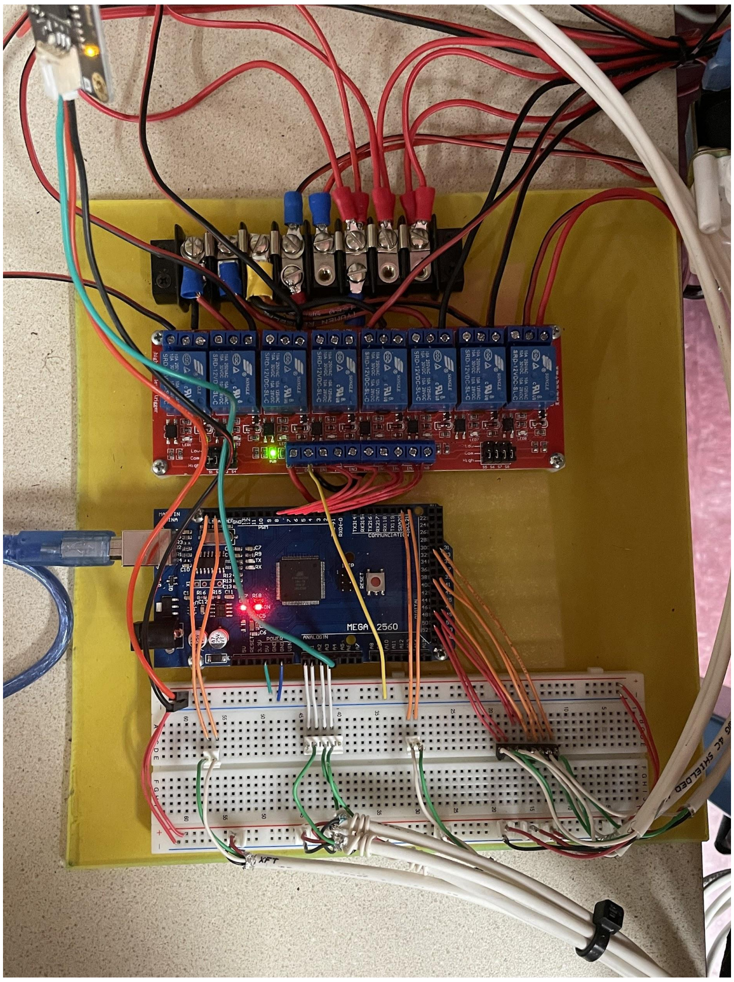


Figure 45: Control Board System

Analysis/Results

Code

We use Arduino code to collect the data from sensors and control the valves and pump. We ran the irrigation system twice a day and collected the data for each 30 minutes. We controlled the value of water and fertilizers by time. For each round of irrigation, we switch on the valves for a certain time, and when we finish all the mixing, we pump the water to the field.

	N	P	K	Water	Pump
time (s)	160	80	80	240	300

Table 4: Fertilizer Proportions and Solenoid Timing

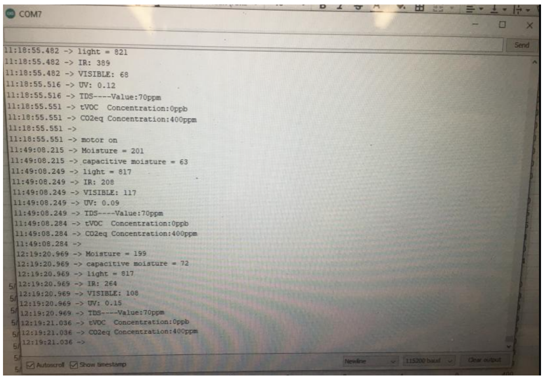
Irrigation System

When we run this system, the whole irrigation system performs well. There is no water leakage in any connections, solenoid valves and pump, and the fertilizer can be transported to the field smoothly. Except for a few plants that were accidentally broken during transfer, the rest of them are healthy.

Figure 46: Field Image

Sensors

Figure 47: Sensor Readout



Since we have been recording data and there is a lot of data, here we take the data of May 3rd as an example. The pH Sensor was unpredictably broken when we assembled, so we didn't record the data of the pH Sensor.

Table 5: Sensor Data Table

Date	C Moisture	moisture	IR	visible	UV	light	tVOC	CO2eq	tds
5/3/22 0:43	213	421	20	5	0.01	436	0	400	86
5/3/22 1:13	211	431	19	6	0.01	441	0	400	86
5/3/22 1:43	209	231	19	6	0.01	441	0	400	86
5/3/22 2:14	211	393	11	4	0	277	0	400	86
5/3/22 2:44	208	382	0	0	0	0	0	400	86
5/3/22 3:14	203	217	0	0	0	5	0	400	86
5/3/22 3:44	212	375	1	0	0	0	0	400	86
5/3/22 4:14	212	366	0	0	0	0	0	400	86
5/3/22 4:45	208	378	0	0	0	4	0	400	86
5/3/22 5:15	209	383	1	0	0	2	0	400	86
5/3/22 5:45	210	348	1	0	0	2	0	400	84
5/3/22 6:15	213	420	2	0	0	1	0	400	86
5/3/22 6:46	198	336	0	0	0	1	0	400	86
5/3/22 7:16	207	338	13	4	0	264	0	400	84
5/3/22 7:46	211	342	1	0	0	0	0	400	86
5/3/22 8:16	210	275	0	1	0	0	0	400	86
5/3/22 8:46	211	339	19	7	0.01	436	0	400	86
5/3/22 9:17	206	320	18	6	0.01	441	0	400	86
5/3/22 9:47	202	330	343	158	0.14	816	0	400	86
5/3/22 10:17	207	333	380	110	0.11	816	0	400	86

5/3/22 10:47	201	334	371	131	0.1	816	0	400	86
5/3/22 11:17	207	324	323	108	0.15	816	0	400	84
5/3/22 11:48	200	327	401	89	0.15	816	0	400	84
5/3/22 12:18	210	290	451	159	0.11	816	0	400	86
5/3/22 12:48	209	317	473	103	0.13	813	0	400	86
5/3/22 13:18	202	308	316	158	0.12	816	0	400	86
5/3/22 13:49	202	312	463	91	0.19	816	0	400	84
5/3/22 14:19	205	302	285	145	0.15	816	0	400	84
5/3/22 14:49	210	305	476	87	0.19	817	0	400	86
5/3/22 15:19	201	312	275	103	0.17	816	0	400	84
5/3/22 15:49	208	281	19	6	0.01	456	0	400	86
5/3/22 16:20	208	296	317	150	0.12	816	0	400	86
5/3/22 16:50	199	240	355	122	0.16	808	0	400	84
5/3/22 17:20	206	280	416	100	0.13	816	0	400	86
5/3/22 17:50	208	270	286	143	0.18	816	0	400	90
5/3/22 18:20	199	275	460	112	0.1	815	0	400	88
5/3/22 18:51	206	266	480	87	0.18	814	0	400	88
5/3/22 19:21	204	273	468	143	0.13	815	0	400	88
5/3/22 19:51	207	274	265	116	0.14	815	0	400	88
5/3/22 20:21	208	269	428	146	0.1	815	0	400	90
5/3/22 20:52	208	268	452	131	0.14	812	0	400	88
5/3/22 21:22	206	266	398	143	0.1	815	0	400	88
5/3/22 21:52	204	263	265	152	0.18	815	0	400	88
5/3/22 22:22	195	269	0	0	0	0	0	400	88
5/3/22 22:52	201	284	0	1	0	0	0	400	88
5/3/22 23:23	197	265	0	1	0	0	0	400	88
5/3/22 23:53	197	263	0	0	0	0	0	400	88
5/4/22 0:23	196	251	0	0	0	1	0	400	88
5/4/22 0:53	207	249	1	0	0	0	0	400	88

Discussion

Recommendations/Future Works

Although we have successfully run the whole system, there is still a lot of room for improvement in this project. We still have a lot of things to do to improve this project.
At this stage, we use the valve's opening time to control the amount of fertilizer used, but in fact, this is not accurate enough for this project. In the next step of this project, we can enable load sensors to accurately calculate the amount of fertilizer. And we will also consider watering plants in different regions and using different fertilizers for different kinds of plants. We have successfully installed these two parts into our system, and we will test and put them into application in the future.

And we will consider not connecting computers to obtain data but uploading data to the cloud through the network. Now we connect the microprocessor to the computer all the time to obtain data and power the microprocessor. In the future, we will consider adding a networking function to this microprocessor, so that it can directly upload data to the cloud, which is convenient for users to receive data with other devices, possibly implementing a LoWaRAN connection. We will consider changing the power supply of the microprocessor to solar energy and using battery power so that when the system is really put into use, it will solve the problem that it is too far away from the power supply and cannot be connected.

Ethics and Professional Responsibilities

To the best of our knowledge, we believe this project was created to be scaled and implemented as ethically as possible, and that the production of it was entirely safe. This project, as it is built to protect the environment and produce healthy food, is ethical in its nature.

Safety

In order to work with many of the chemicals involved in fertilizer, we recommend storing all fertilizers in a dry, cool environment, and regularly checking stores. We also recommend using gloves to mix the solutions into water aqueously to avoid any chemical damage to people.

Accomplishments and Failures

We did manage to build the entire system, which is operational. It can take many of the data points it is supposed to, and functions how it should without leaking. It can keep plants alive without anyone checking on it for days.

However, some of our failures included not being able to get the load sensors to take data in the way we needed them to. The pH Sensor broke halfway through the project, and we were unable to get a replacement one in the time allotted. The project as a whole could have been a lot neater/professional given more time had been spent on it.

Manufacturability and Cost

The overall cost of this project was around $500 dollars. The sensors were acquired from Grove, and Amazon. The solenoid and motor were from Amazon, and the rest of the materials were from Home Depot. The acquisition of the parts was not too difficult. In the future a lot of our budget would be saved on not having to buy grow lights, a pond liner and soil as that would already be in a field.

Table 6: Budget Estimate

Item	Quantity	Cost/unit	Total Item Cost	Link
Tanks 7.5 Gallons (6x)	1	63	63	https://www.walmart.com/ip/St
Plastic Tubing	1	8	8	https://www.amazon.com/Lear
valve	6	7.49	44.94	https://www.amazon.com/dp/B
pump	1	25	25	https://www.amazon.com/gp/p
pump	1	1.27	1.27+5.26	https://www.aliexpress.com/ite
load sensor 10X	2	11	22	https://www.amazon.com/Weig
load sensor 4X	4	1.38	5.52+11.19	https://www.aliexpress.com/ite
TDS sensor	1	14.3	14.3	https://wiki.seeedstudio.com/G
TDS sensor	1	7.66+3.6		https://www.aliexpress.com/ite
ph sensor	1	44.99	44.99	https://www.seeedstudio.com/
ph sensor	1	18.16		https://www.aliexpress.com/ite
color sensor	1	11.9	11.9	https://www.seeedstudio.com/
CO2 sensor grove				https://www.seeedstudio.com/
CO2 sensor	1	56	56	https://www.mouser.com/Prod
CO2 sensor	1	26.9	26.9+4.64	https://www.aliexpress.com/ite
Resistive moisture sensor				
Moisture Sensor	1			
Conductivity Sensor	1	5.95	5.95	https://www.mouser.com/Prod
Conductivity Sensor	1	0.43	0.43+1.65	https://www.aliexpress.com/ite
microcontroller	1			
Drip Tubing	1	6.54	6.54	https://www.homedepot.com/p
T Drip connectors	10	1.96		https://www.homedepot.com/p
Drip barb connectors	6	1.96		https://www.homedepot.com/p
Silicon Caulking	1	7.58	7.58	https://www.homedepot.com/p
Plumbing Tape	1	1.97	1.97	https://www.homedepot.com/p
Gardening Soil	7	7.97	56	https://www.homedepot.com/p
Lumber	4	10	40	
Seeds	1	5	5	
Wire				
Nitrogen 5lbs bag	1		14.95	https://www.amazon.com/Urea
Phosphorus 5lbs bag	1		16	https://www.amazon.com/Tripl
Potassuium 32 fl oz	1		23	https://www.amazon.com/0-0-
Solar Panels 12	7	4.59		https://www.aliexpress.com/ite
Li 18650 Batteries	1	11		https://www.aliexpress.com/ite
Seed Starting Trays	1		5.98	https://www.amazon.com/Larg
Pond Liner	1	15	11	Amazon
Soil	6	12	16	Home Depo

Scalability

This project was designed to be highly scalable. In the future if we wanted it to be applicable, we would just need to add more tubing to cover the entire field, and longer wires for the sensors to be able to take data deep in the field. Additionally, we might scale the number of sensors to be relative to the number of acres in the field and average them to make sure that we will have adequate coverage.

Conclusion

In sum we have designed a machine with a mechanical fluid dispersion system, and electrical/computer sensing and control system that modulates and takes in environmental factors of a field. Each portion of the design plays a key role in the overall function of the machine and are closely intertwined. Through this mechanism we have received preliminary data that indicates that the machine functions as designed and is capable of getting the data that we are looking for. Based on the tables we can see that our plants live in above average conditions and based on the picture they seem to be growing well. This is substantial because in the future, with some more refinement, we could take real time information on growing conditions, optimize plant nutrient dispersion, and increase our knowledge of plants. Given a real-world application, and scaled model, we could see this being implemented worldwide to continue to survey and understand agriculture at its core. The next course of action in our project is to make it more sustainable and real world applicable through solar and LoRaWAN. Lastly, it creates a lower impact system on the environment, heavily impacting local river ecosystems and nitrogen run off.

References/Footnotes/Bibliography

1) Kim, Seungdo, and Bruce E. Dale. "Effects of Nitrogen Fertilizer Application on Greenhouse Gas Emissions and Economics of Corn Production." Environmental Science & Technology, vol. 42, no. 16, 2008, pp. 6028–6033., doi:10.1021/es800630d.
2) Muratoglu, Abdullah. "Grey Water Footprint of Agricultural Production: An Assessment Based on Nitrogen Surplus and HIGH-RESOLUTION LEACHING Runoff Fractions in Turkey." Science of The Total Environment, vol. 742, 2020, p. 140553., doi:10.1016/j.scitotenv.2020.140553.
3) "Greenhouse Hydroponics." Controlled Environment Agriculture (NEMALI Lab), 3 May 2018, www.purdue.edu/hla/sites/cea/greenhouse-hydroponics/. Accessed 8 May 2022.
4) Brueck, Holger, and Joachim Lammel. "Impact of Fertilizer N Application on the Grey Water Footprint of Winter Wheat in a NW-EUROPEAN Temperate Climate." Water, vol. 8, no. 8, 2016, p. 356., doi:10.3390/w8080356.
5) "The Effects of Fertilizer Runoff." The Effects of Fertilizer Runoff | Connection Between Agriculture & Drinking Water Contamination, Multipure, 10 Oct. 2021, https://www.multipure.com/purely-social/science/effects-fertilizer-runoff.
6) Fischels, Josie. "At Least 600 Tons of Dead Fish Have Washed up along Tampa Bay's Shore." Environment, NPR, 13 July 2021, https://www.npr.org/2021/07/13/1015312707/a-summer-red-tide-has-left-hundreds-of-tons-of-dead-fish-along-tampa-bays-shore.
7) "History of Agriculture." Wikipedia, Wikimedia Foundation, 4 Oct. 2021, https://en.wikipedia.org/wiki/History_of_agriculture.
8) "The Birth of Agriculture ." Innovature, ASTA, 2019, https://innovature.com/timeline.
9) United States. Environmental Protection Agency. Fertilizer Regulations. Washington, D.C.: GPO, 2001. Web. 4 Mar. 2021.
10) Koon, John. "Key Factors to Consider When Choosing a Microcontroller." Www.microcontrollertips.com, 15 Apr. 2019, www.microcontrollertips.com/key-factors-consider-choosing-microcontroller/.
11) Tameson. "How a Solenoid Valve Works & a Buying Guide." Tameson, Aug. 8AD, tameson.com/solenoid-valve-types.html.
12) Depo. "A Guide to Types of Valves." MegaDepot.com, 6 Mar. 2015, megadepot.com/resource/a-guide-to-types-of-valves.
13) Beningo, Jacob. "10 Steps to Selecting a Microcontroller - Embedded Blog - Arm Community Blogs - Arm Community." Community.arm.com, 12 Jan. 2014, community.arm.com/arm-community-blogs/b/embedded-blog/posts/10-steps-to-selecting-a-microcontroller. Accessed 8 May 2022.

.

Appendices

Arduino Code:

```
void moistureSensor();
void cmoiSensor();
void lightSensor();
void sunlightSensor();
void vocSensor();
void tdsSensor();
void phSensor();
void switchoff(int relay);
void switchon(int relay);


#include "Si115X.h"
#include <Arduino.h>
#include "sensirion_common.h"
#include "sgp30.h"
#include "HX711.h"

int moisensorPin = A1;   //moisture sensor
int moisensorValue = 0;

int cmoisensorPin = A2;  //capacitive moisture sensor
int cmoisensorValue = 0;

int lightsensorPin = A3;  //light sensor
int lightsensorValue = 0;

int phsensorPin = A0;  //pH Sensor
int Offset = 41.02740741;
int LED = 13;


Si115X si1151;  //sunlight sensor

int tdssensorPin = A4;  //TDS sensor
float VREF = 5.0;      // analog reference voltage(Volt) of the ADC
float averageVoltage = 0;
float tdsValue = 0;
float temperature = 22;
```

```
unsigned long time;

const int motor_relay = 2;
const int valve3_relay = 3;
const int valve4_relay = 4;
const int valve5_relay = 5;
const int valve2_relay = 6;
const int valve1_relay = 7;
const int valve6_relay = 8;     // valve 1-4 are valves to mix tank,  valve 5 is one to the left
and valve 6 to right

int tank1_time = 0;
int tank1_status = 0;

int tank2_time = 0;
int tank2_status = 0;

int tank3_time = 0;
int tank3_status = 0;

int tank4_time = 0;
int tank4_status = 0; //0 is off, 1 is on, 2 is done for today

int motor_time = 0;
int motor_status = 0; //0 is off, 1 is on

void setup() {
  uint8_t conf[4];
  Wire.begin();
  s16 err;
  u16 scaled_ethanol_signal, scaled_h2_signal;
  Serial.begin(115200);

  if (!si1151.Begin()){
      Serial.println("Si1151 is not ready!");
  }
  else {
      Serial.println("Si1151 is ready!");
  }

  /**
  pinMode(LED, OUTPUT);
*/
```

```
  #if defined(ESP8266)
  pinMode(15, OUTPUT);
  digitalWrite(15, 1);
  Serial.println("Set wio link power!");
  delay(500);
  #endif
    /*  Init module,Reset all baseline,The initialization takes up to around 15 seconds, during
which
       all APIs measuring IAQ(Indoor air quality ) output will not change.Default value is
400(ppm) for co2,0(ppb) for tvoc*/
    while (sgp_probe() != STATUS_OK) {
       Serial.println("SGP failed");
       while (1);
    }
    /*Read H2 and Ethanol signal in the way of blocking*/
    err = sgp_measure_signals_blocking_read(&scaled_ethanol_signal,
                                  &scaled_h2_signal);
    if (err == STATUS_OK) {
       Serial.println("get ram signal!");
    } else {
       Serial.println("error reading signals");
    }
    err = sgp_iaq_init();

  pinMode(tdssensorPin,INPUT);

  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  //pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  //pinMode(8,OUTPUT);

  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  //digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
  //digitalWrite(8,LOW);
```

```
}

void loop() {


  time = millis();

  if (time % 1800000 <= 20){    //1 hour 1000*60*60=3600000   measure once an hour
    //sensor
    moistureSensor();
    cmoiSensor();
    lightSensor();
    sunlightSensor();
    tdsSensor();
    vocSensor();
    Serial.println();


  }

  if (time % 43200000 <= 20 && tank1_status == 0 && tank2_status == 0
                              && tank3_status == 0 && tank4_status == 0 ){ //1day
1000*60*60*24=86400000
    tank1_time = time;
    tank2_time = time;
    tank3_time = time;
    tank4_time = time;
    switchon(valve1_relay);
    switchon(valve2_relay);
    switchon(valve3_relay);
    switchon(valve4_relay);
    tank1_status = 1;
    tank2_status = 1;
    tank3_status = 1;
    tank4_status = 1;
  }

  if (tank1_status == 1 && time - tank1_time >= 80000 ){
    switchoff(valve1_relay);
    tank1_status = 2;
  }

  if (tank2_status == 1 && time - tank2_time >= 160000 ){
```

```
    switchoff(valve2_relay);
    tank2_status = 2;

  }

  if (tank3_status == 1 && time - tank3_time >= 80000 ){
    switchoff(valve3_relay);
    tank3_status = 2;

  }

  if (tank4_status == 1 && time - tank4_time >= 240000 ){
    switchoff(valve4_relay);
    tank4_status = 2;

  }

  if (tank1_status == 2 && tank2_status == 2 && tank3_status == 2 && tank4_status == 2
&& motor_status == 0 ){
    motor_time = time;
    switchon(motor_relay);
    Serial.println("motor on");
    motor_status = 1;
  }

  if (motor_status ==1 && time - motor_time >= 300000){
    switchoff(motor_relay);
    motor_status = 0;
    tank1_status = 0;
    tank2_status = 0;
    tank3_status = 0;
    tank4_status = 0;

  }
}

void moistureSensor(){                    //moisture sensor
  moisensorValue = analogRead(moisensorPin);
  Serial.print("Moisture = " );
  Serial.println(moisensorValue);
}

void cmoiSensor(){                        //capacitive moisture sensor
  cmoisensorValue = analogRead(cmoisensorPin)-500;
```

```
  Serial.print("capacitive moisture = ");
  Serial.println(cmoisensorValue);
}

void lightSensor(){                    //light sensor
  lightsensorValue = analogRead(lightsensorPin);
  Serial.print("light = ");
  Serial.println(lightsensorValue);
}

void phSensor(){                       //pH Sensor
  static float pHValue, voltage;
  voltage = analogRead(phsensorPin) * 5.0 / 1024;
  pHValue = -19.18518519 * voltage + Offset;
  Serial.print("Voltage:");
  Serial.print(voltage, 2);
  Serial.print("    pH value: ");
  Serial.println(pHValue+1, 2);
  digitalWrite(LED, digitalRead(LED) ^ 1);
}

void sunlightSensor(){                 //sunlight sensor
  Serial.print("IR: ");
  Serial.println(si1151.ReadHalfWord());
  Serial.print("VISIBLE: ");
  Serial.println(si1151.ReadHalfWord_VISIBLE());
  Serial.print("UV: ");
  Serial.println(si1151.ReadHalfWord_UV());
}

void vocSensor(){                      //VOC sensor
  s16 err = 0;
  u16 tvoc_ppb, co2_eq_ppm;
  err = sgp_measure_iaq_blocking_read(&tvoc_ppb, &co2_eq_ppm);
  if (err == STATUS_OK) {
    Serial.print("tVOC  Concentration:");
    Serial.print(tvoc_ppb);
    Serial.println("ppb");

    Serial.print("CO2eq Concentration:");
    Serial.print(co2_eq_ppm);
    Serial.println("ppm");
  } else {
    Serial.println("error reading IAQ values\n");
```

```
  }
}

void tdsSensor(){                              //tds sensor
  averageVoltage = analogRead(tdssensorPin) * VREF / 1024.0; // read the analog value more
stable by the median filtering algorithm, and convert to voltage value
  float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0); //temperature
compensation formula: fFinalResult(25^C) = fFinalResult(current)/(1.0+0.02*(fTP-25.0));
  float compensationVolatge = averageVoltage / compensationCoefficient; //temperature
compensation
  tdsValue = (133.42 * compensationVolatge * compensationVolatge * compensationVolatge -
255.86 * compensationVolatge * compensationVolatge + 857.39 * compensationVolatge) *
0.5; //convert voltage value to tds value
  Serial.print("TDS----Value:");
  Serial.print(tdsValue, 0);
  Serial.println("ppm");
}

void switchoff(int relay){
  digitalWrite(relay,LOW); //set the pin to low
}

void switchon(int relay){
  digitalWrite(relay,HIGH); //set the pin to high

}
```